Pós-Graduação em Ciência da Computação

DÉBORA DA CONCEIÇÃO ARAÚJO

AVALIAÇÃO DE COMITÊS COM CLASSIFICADORES TRADICIONAIS E PROFUNDOS PARA ANÁLISE DE SENTIMENTOS

Universidade Federal de Pernambuco
posgraduacao@cin.ufpe.br
http://cin.ufpe.br/~posgraduacao

Recife
2019

DÉBORA DA CONCEIÇÃO ARAÚJO

AVALIAÇÃO DE COMITÊS COM CLASSIFICADORES TRADICIONAIS E PROFUNDOS PARA ANÁLISE DE SENTIMENTOS

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

Área de Concentração: Aprendizado de Máquina

Orientador: Prof. Dr. Leandro Maciel Almeida

Recife

2019

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

A663a Araújo, Débora da Conceição
Avaliação de comitês com classificadores tradicionais e profundos para análise de sentimentos / Débora da Conceição Araújo. – 2019.
113 f.: il., fig., tab.

Orientador: Leandro Maciel Almeida.
Dissertação (Mestrado) – Universidade Federal de Pernambuco. CIn, Ciência da Computação, Recife, 2019.
Inclui referências.

1. Inteligência computacional. 2. Aprendizagem de máquina. I. Almeida, Leandro Maciel (orientador). II. Título.

006.3 CDD (23. ed.) UFPE- MEI 2019-046

Débora da Conceição Araújo

Avaliação de Comitês com Classificadores Tradicionais e Profundos para Análise de Sentimentos

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

Aprovado em: 20/02/2019.

BANCA EXAMINADORA

Prof. Dr. Paulo Salgado Gomes de Mattos Neto
Centro de Informática /UFPE

Prof. Dr. João Fausto Lorenzato de Oliveira
Escola Politécnica de Pernambuco /UPE

Prof. Dr. Leandro Maciel Almeida
Centro de Informática /UFPE
(**Orientador**)

aos meus pais e irmã.

AGRADECIMENTOS

Esta dissertação é fruto de horas de dedicação e trabalho, e só se tornou realidade graças as pessoas que me ajudaram a tornar esta caminhada mais leve. Agradeço aos meus pais, Edinaldo e Vanda, e à minha irmã, Dara, por todo apoio em todos os momentos, principalmente durante esses dois anos longe de casa. Agradeço às minhas primas, tias, tios e todos os familiares que se fizeram presentes, torceram e me ajudaram durante este tempo do mestrado. Aos amigos da graduação que continuaram na caminhada da pós e que foram tão importantes durante a etapa final desta dissertação, Máverick e Eraylson. Aos amigos do mestrado Raimundo e Paulo. Às amigas que Recife me trouxe, Amanda e Milena, obrigada pelos conselhos de sempre, meninas. Aos amigos de Garanhuns e Correntes que sempre estão torcendo por mim, Jadielma, Marciele, Jamilly, Aleandro, Vangéssica, Paulo Vinícius, Dorghislanny, Mayra, Jaciara, Jaqueline e Izabele. Agradeço a Iago por todo apoio, paciência e confiança, sou grata a você e à sua família. Agradeço ao meu orientador, Leandro Almeida, pelos direcionamentos que foram essenciais para a construção deste trabalho, obrigada pela confiança que depositou em mim, professor. Sou grata a Deus e a Virgem Maria por tantas bençãos e pessoas maravilhosas que estão presentes em minha vida.

"We accepted education as the means to rise above the limitations that a prejudiced society endeavored to place upon us"(GRANVILLE, 1989)

RESUMO

Análise de Sentimentos é o problema que explora documentos escritos em linguagem natural visando classificá-los em polaridades de sentimentos (classes) pré-estabelecidas. Os algortimos *Naive Bayes* e *Suport Vector Machine* estão frequentemente associados a este tipo de tarefa, porém estes classificadores apontam para um problema iminente quando se trata da análise de sentimentos em um universo não-binário de classes. Classificadores de aprendizado profundo aparecem, cada vez mais, na literatura como alternativa aos modelos tradicionais de aprendizado de máquina, apresentando bons desempenhos. Diante disto, esta pesquisa apresenta uma avaliação de desempenho entre métodos de aprendizado de máquina tradicional, métodos de aprendizado profundo e comitês de classificadores que combinam as duas abordagens. Os comitês construídos fazem uso de modelos de aprendizado profundo com um menor número de épocas de treinamento, a intenção foi desenvolver modelos com menor tempo de execução sem perder em acurácia, devido ao conhecimento dos demais modelos combinados. Para avaliar o desempenho das diferentes abordagens, foram utilizadas cinco bases de dados com múltiplas classes: *Stanford Sentiment Treebank*, IMDb *Review*, Yelp 2013, Yelp 2014 e Yelp 2015. O desempenho dos modelos foram avaliados através de um conjunto de métricas e técnicas estatísticas. Com base nos resultados obtidos, foi possível concluir que os algoritmos de aprendizado profundo e os comitês alcançaram desempenhos médios estatisticamente superiores em relação aos algoritmos de aprendizado de máquina tradicional. Apesar do maior desempenho, vale salientar que os comitês e os modelos de aprendizado profundo possuem tempo de treinamento superior em relação aos algoritmos tradicionais.

Palavras-chaves: Análise de Sentimentos. Aprendizado de Máquina. Aprendizado Profundo. Comitê de Classificadores.

ABSTRACT

Sentiment Analysis is the problem that explores documents written in natural language aiming to classify them into pre-established polarities of feelings (classes). The Naive Bayes and Suport Vector Machine algorithms are often associated with this type of task, but these classifiers point to an imminent problem when it comes to the sentiment analysis in a non-binary universe of classes. In the light of this, this research presents an alternative to the traditional methods used for the sentiment analysis, by means of deep learning cassifiers and ensembles of classifiers that mix algorithms of learning of traditional machine and deep learning, considering that these types of approaches have been showing good results in several problems of the literature. To address this problem, this research presents an alternative to the traditional methods used for the sentiment analysis by means of deep learning cassifiers and ensembles of classifiers that mix traditional machine learning algorithms and deep learning, since these types of approaches come demonstrating good results in several problems of the literature. We compared the performances of the single classifiers, traditional machine learning and deep learning, and classifier combinations, in order to observe if there is a statistical difference between the acuracies reached and the relation between the performance of the model and its execution time. To evaluate the performance of the different approaches, 5 databases were used: Stanford Sentiment Treebank, IMDb Review, Yelp Challange Dataset 2013, 2014 e 2015. The performance of the models were evaluated through a set of metrics and statistical techniques. Based on the results obtained, it is possible to infer that the deep learning algorithms and ensemble classifiers achieved statistically superior average performances in relation to the algorithms of traditional machine learning. Despite the higher performance, it is worth noting that ensembles and deep learning classifiers have a computational cost higher than the cost of traditional algorithms.

Keywords: Ensembles Classifiers. Deep Learning. Machine Learning. Sentiment Analysis.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

AM	Aprendizado de Máquina
API	*Application Programming Interface*
AS	Análise de Sentimentos
CLSTM	*Cached Long Short-Term Memory*
CNN	*Convolutional Neural Network*
FN	Falso Negativo
FP	Falso Positivo
KNN	*K Nearest Neighbor*
LSTM	*Long short-term Memory*
MCS	*Multiple Classifiers Systems*
NB	*Naive Bayes*
NLTK	*Natural Language Toolkit*
PLN	Processamento de Linguagem Natural
RNN	*Recurrent Neural Network*
SGD	*Stochastic Gradient Descent*
SSTb	*Stanford Sentiment Treebank*
SVM	*Support Vector Machine*
TFIDF	*Term Frequency-Inverse Document Frequency*
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo

SUMÁRIO

1 INTRODUÇÃO

1.1 MOTIVAÇÃO

Com o constante crescimento das mídias e redes sociais, as buscas por opiniões deixaram de se restringir à familiares e amigos e passaram a ser realizadas em sites e comunidades com comentários especializados na Internet. Já em 2008, mais de 80% dos usuários realizavam buscas na Web antes de adquirir um produto ou serviço (PANG; LEE, 2008). Em 2014, o *e-commerce Amazon* já acumalava um conjunto de dados com mais de 34 milhões de comentários sobre mais de 2 milhões de produtos (MCAULEY; LESKOVEC, 2013; PORIA; CAMBRIA; GELBUKH, 2016). Esses comentários possuem potencial para ajudar clientes à realizar suas escolhas, bem como podem auxiliar empresas na melhoria de seus processos.

Ao considerar o volume de dados acumalado, percebe-se que analisar tal conteúdo não é uma tarefa trivial. Desse modo, diversos estudos apontam para a necessidade da construção de métodos automáticos que interpretem as informações presentes nos textos (CARVALHO, 2018; CHEN et al., 2018).

Para interpretação automática de texto tem-se o Processamento de Linguagem Natural (PLN), uma área da computação que compreende o desenvolvimento de métodos inspirados no processo complexo pelo qual os humanos analisam e interpretam a linguagem (ARNOLD; TILTON, 2015). Através das técnicas de PLN, os textos descritos em linguagem natural (português, inglês etc.) passam a ser interpretáveis por um computador. Para a tarefa específica de ler e classificar sentimentos e opiniões expressas em textos, há um campo específico de PLN, conhecido por Análise de Sentimentos (AS) (AIRES et al., 2018).

Também chamada de mineração de opiniões, a análise de sentimentos tem por objetivo analisar comentários, opiniões, sentimentos e atitudes das pessoas com relação a produtos, serviços, lugares, figuras públicas, organizações ou demais aspectos (PANG; LEE, 2008; LIU, 2012; CAMBRIA, 2016). As técnicas de AS possuem aplicações na política, no cinema, comércio varejista e muitos outros segmentos da sociedade, como em Carvalho (2018) onde são realizadas investigações sobre a parcialidade de notícias políticas que se referiam às eleições brasileiras de 2014 por meio de uma classificação binária que considerou comentários positivos e negativos. Em Poria et al. (2017) foram analisadas recomendações de produtos, como perfumes e filmes, considerando as polaridades de sentimentos: *raiva*, *felicidade* ou *neutra*. É possível analisar mais aplicações de AS nos trabalhos de Tang, Qin e Liu (2015b) e Zhang, Wang e Liu (2018), que também consideram mais polaridades de sentimentos, como: muito negativa, negativa, neutra, positiva e muito positiva.

Para classificar os textos em AS utilizam-se, principalmente, dois tipos de abordagens: dicionários léxicos e métodos de Aprendizado de Máquina (AM) (HUTTO; GILBERT, 2014). As aplicações que usam dicionários léxicos têm como entrada adjetivos, nomes, verbos e

expressões idiomáticas que ajudam a identificar uma polaridade de sentimentos, em geral binária, positiva ou negativa (TABOADA et al., 2011). No caso de abordagens com algoritmos de aprendizado de máquina, métodos supervisionados são amplamente utilizados, em que o intuito é identificar padrões através de um conjunto de dados de treinamento.

Trabalhos na literatura demonstram que as técnicas de aprendizado de máquina, geralmente, possuem desempenho superior aos dicionários léxicos (MEDHAT; HASSAN; KORASHY, 2014). Um segmento crescente do aprendizado de máquina é o aprendizado profundo, do inglês *deep learning*. Bons desempenhos de algoritmos de aprendizado profundo para a tarefa de análise de sentimentos são apresentados em Zhai e Zhang (2016), porém classificadores de AM tradicional ainda possuem maior adesão da comunidade científica para este tipo de tarefa (MEDHAT; HASSAN; KORASHY, 2014). Desse cenário emerge a primeira questão de pesquisa: algoritmos de aprendizado profundo possuem desempenho superior aos algoritmos tradicionais de AM para a tarefa de AS?

Além dos classificadores únicos, do ingês *single classifiers*, Sistemas de Múltiplos Classificadores (*Multiple Classifiers Systems* (MCS)) são amplamente utilizados na literatura visando a melhora de desempenho dos modelos (KUNCHEVA, 2004). Também chamados de comitês, os MCS partem da premissa de que a combinação de várias redes, treinadas separadamente, pode aumentar significativamente a capacidade de generalização do sistema (LIMA, 2017). Esse tipo de modelo ganhou destaque devido ao seu bom desempenho, principalmente quando são considerados problemas complexos (ALMEIDA, 2011).

Segundo Roli, Giacinto e Vernazza (2001), para melhor representar e classificar os dados (textos), os MCS precisam conter diversidade por meio de métodos como: diferentes conjuntos de dados para treinar os modelos; um mesmo modelo de classificação, porém com diferentes configurações/parâmetro ou; diferentes tipos de classificadores. Considerando o contexto deste trabalho, a diversidade dos comitês construídos é apresentada por meio de diferentes tipos de classificadores, sendo combinados modelos de AM tradicional aos modelos de Aprendizado Profundo.

Nesse sentido, o presente trabalho busca corroborar com as pesquisas de AM tradicional, aprendizado profundo e Sistemas de Múltiplos Classificadores para a tarefa de AS, por meio de experimentos realizados em cinco bases de dados que visam responder as seguintes questões de pesquisa: (1) algoritmos de aprendizado profundo possuem desempenho superior aos algoritmos tradicionais de AM ao considerar o contexto das bases utilizadas? (2) em quais situações a combinação de classificadores se mostra uma melhor solução em relação a um classificador único? (3) os melhores comitês, para estas bases, combinam métodos de AM tradicional e Aprendizado Profundo ou advém da combinação de modelos do mesmo tipo? E (4) a combinação se mostrou uma solução interessante ao se observar tempo de execução e desempenho dos modelos?

1.2 OBJETIVOS

1.2.1 Objetivo Geral

Esta dissertação de mestrado tem como objetivo principal:

- Comparar a relação entre desempenho e custo computacional entre algoritmos de aprendizado de máquina tradicional, algoritmos de aprendizado profundo e sistemas de múltiplos classificadores para o problema de análise de sentimentos.

1.2.2 Objetivos Específicos

Para atingir o objetivo geral, foram definidos os seguintes objetivos específicos:

- Realizar levantamento sobre análise de sentimentos, aprendizado profundo e combinação de classificadores;
- Reunir bases de dados com múltiplas classes para a tarefa de análise de sentimentos;
- Comparar os desempenhos obtidos entre os algoritmos de aprendizado de máquina tradicional e aprendizado profundo;
- Comparar os desempenhos obtidos entre as diferentes abordagens de combinação e os classificadores únicos;
- Investigar em quais circunstâncias os diferentes métodos apresentam melhores desempenhos, sejam comitês ou classificadores únicos;
- Analisar quais as melhores abordagens de combinação de classificadores;

1.3 ESTRUTURA DA DISSERTAÇÃO

Esta dissertação encontra-se organizada em 6 capítulos, além das Referências que nortearam o trabalho:

1. **Introdução:** apresenta, de forma introdutória, o contexto de análise de sentimentos, bem como a motivação e os objetivos desta dissertação;
2. **Fundamentação Teórica:** detalha os conceitos e técnicas fundamentais para o embasamento teórico deste trabalho;
3. **Método Proposto:** apresenta as etapas que compõem as arquiteturas propostas;
4. **Metodologia dos Experimentos:** descreve as bases de dados utilizadas, destacando suas características e comportamentos, detalha, ainda, as etapas que compõem o tratamento dos dados e as configurações dos classificadores;

5. **Experimentos e Resultados:** são descritos os experimentos realizados e os resultados obtidos são discutidos;

6. **Conclusões e Trabalhos Futuros:** apresenta um resumo deste trabalho, destacando as contribuições, limitações e propostas para trabalhos futuros;

2 FUNDAMENTAÇÃO TEÓRICA

Este capítulo apresenta uma revisão da literatura, com conceitos e técnicas de análise de sentimentos, aprendizado de máquina, aprendizado profundo e combinação de classificadores, além do pré-processamento dos textos, métricas de avaliação e testes estatísticos utilizados.

2.1 ANÁLISE DE SENTIMENTOS

Emoção e opinião possuem diferentes significados de acordo com a área do conhecimento em que estão sendo estudados. Sociologia, filosofia, ciência política, psicologia e, mais recentemente, a computação são algumas das ciências que analisam tais termos por diferentes perspectivas (SENA, 2007; FRANZ; HILLMAN, 1990; SILVA, 2016).

No âmbito da psicologia e filosofia, pesquisadores apontam que a base para conceitualizar o sentimento é o estudo dos filósofos Aristóteles, Platão, Spinoza, Descartes, Kant e Hume (FRANZ; HILLMAN, 1990; CECI; ALVAREZ; GONÇALVES, 2017). De acordo com Sena (2007), para a ciência política, "a opinião corresponde sempre a um juízo formulado a respeito de qualquer fato". Ao analisar o ponto de vista da sociologia, Tarde (1991) define o termo opinião como "um agrupamento momentâneo e mais ou menos lógico de julgamentos". Para a ciência da computação o objetivo é analisar automaticamente fragmentos textuais que expressem emoção, opinião ou atitudes, meio do campo de Análise de Sentimentos (PANG; LEE, 2008). Para tal, a computação mescla conhecimentos de diferentes áreas, como "mineração de dados, aprendizado de máquina, linguística, processamento de linguagem natural e análise textual"(SILVA, 2016).

A necessidade de analisar textos de forma automática surgiu junto ao crescente uso da internet, pois com as redes sociais a quantidade de dados produzidos pelos usuários se tornou gigantesca. Somente na rede social Twitter são cerca de 500 milhões de tweets postados por dia (SILVA; COLETTA; HRUSCHKA, 2016). Em Kiss (2013) são apresentados dados da rede social Facebook que, em outubro de 2012, atingiu a marca de 1 bilhão de usuários mensais ativos e mais de 550 milhões de usuários logados diariamente. São postagens e comentários sobre os mais variados temas, que torna praticamente impossível para um humano minerar manualmente o que lhe interessa.

Os primeiros trabalhos que apontavam para tarefas envolvendo análise de sentimentos apenas realizavam uma classificação binária entre as polaridades positiva ou negativa dos sentimentos (TURNEY, 2002; PANG; LEE; VAITHYANATHAN, 2002). Hoje em dia, os sistemas podem ser construídos de forma mais ampla, para significar o tratamento computacional de qualquer opinião, sentimento e subjetividade em um texto.

As pesquisas em AS se dividem, basicamente, em três níveis de granularidade, a saber:

- **Granularidade de documento:** o objetivo deste tipo de tarefa é classificar o contexto geral do documento, presume-se que apenas uma entidade está sendo avaliada, ou seja, o documento trata de um produto ou serviço específico e o método deve retornar uma avaliação a seu respeito (SILVA, 2016; PANG; LEE; VAITHYANATHAN, 2002; TURNEY, 2002).

- **Granularidade de sentença:** "Nesta tarefa, o texto é subdividido em sentenças e análise é feita sobre tais unidades textuais com o objetivo de efinir se as mesmas expressam individualmente um sentimento positivo, negativo ou neutro"(SILVA, 2016).

- **Granularidade de entidade e de aspecto:** neste tipo de classificação o que se leva em conta não é o documento ou suas sentenças, mas as entidades presentes no texto (SILVA, 2016). Na frase "essa banda é ótima, mas o som está péssimo", tem-se duas entidades que são avaliadas de forma distinta, em que a "banda"é avaliada positivamente enquanto o "som"é avaliado negativamente.

Esta dissertação se concentra na granularidade de documentos, ao analisar comentários relacionados a filmes, produtos e serviços. As bases de dados utilizadas estão detalhadas na seção 4.1.

2.2 PRÉ-PROCESSAMENTO DE TEXTO

A fase de pré-processamento dos dados compõe diversos objetivos, tais como tratar atributos irrelevantes e alterar a estrutura dos dados, preparando-os para a etapa de extração de conhecimento (BATISTA et al., 2003). Nesta seção estão descritas as técnicas de pré-processamento utilizadas durante os experimentos desta dissertação.

2.2.1 *StopWords, Stemming* e *N-gram*

Quando se trata de aprendizado de máquina, o pré-processamento dos dados é um procedimento chave para o desempenho dos métodos de classificação (CHEN et al., 2018). Dentre as técnicas adotadas neste trabalho estão a remoção de *stopwords*, aplicação de *stemming* e n-gram, em que:

- *Stopwords* são palavras que não contribuem para o significado mais profundo da frase. Assim, a remoção de *stopwords* atinge, principalmente, palavras que representam artigos, conjunções e preposições (LO; HE; OUNIS, 2005). Na frase, *"essa banda é ótima, mas o som está péssimo"*, seriam removidas as palavras *"essa"*, *"mas"* e *"o"*. Restando apenas o essencial para o contexto, ou seja, *"banda é ótima"* e *"som está péssimo"*.

- *Stemming* é um método de redução de palavras ao seu radical (PLISSON; LAVRAC; MLADENIć, 2004). Esse método pode beneficiar a classificação do documento tanto

por reduzir o vocabulário de palavras quanto por se concentrar no sentido de um conjunto de palavras, em vez de analisar o significado de cada palavra de forma individual. Assim, palavras como *"alegre"*, *"alegria"*, *"alegrar"* e *"alegremente"*, serão agrupadas e consideradas como "alegr"no vocabulário de palavras.

- n-gram é uma técnica que consiste no agrupamento das palavras de um documento em sequências de tamanho n (AL-SHALABI; OBEIDAT, 2008). Em um exemplo considerando $n = 2$, a frase *"essa banda é ótima, mas o som está péssimo"*, seria quebrada em *"essa banda"*, *"banda é"*, *"é ótima"*, *"ótima mas"*, *"mas o"*, *"o som"* e *"som péssimo"*. Essa técnica pode ser interessante para contextos em que há uma palavra com polaridade negativa modificando o sentido de uma outra palavra, originalmente, com polaridade positiva, ex.: *"não gostei"*, a palavra *"gostei"* teve sua polaridade modificada pelo advérbio de negação *"não"*.

2.2.2 *Oversampling e undersampling*

Segundo Chawla, Japkowicz e Kotcz (2004), o desbalanceamento dos dados é um dos principais problemas do aprendizado de máquina. Esse problema consiste em uma base que possui muitos exemplos de determinada classe e uma quantidade de exemplos não representativa de uma outra classe. Ao lidar com esse tipo de dado, o algoritmo encontra dificuldades para extrair padrões da classe minoritária, o que acarreta em um classificador com pouca capacidade de generalização.

As técnicas de *oversampling* e *undersampling* buscam minimizar os impactos do desbalanceamento das bases por meio de duas mudanças nos dados de treinamento (LIU; WU; ZHOU, 2009), que são: *oversampling* - repete dados da instância minoritária nos dados de treinamento; *undersampling* - remove dados da instância majoritária nos dados de treinamento.

2.2.3 Vetorização dos textos

Algoritmos de aprendizado de máquina não trabalham com textos em linguagem natural, portanto diversas técnicas visam preparar (vetorizar) dados originalmente em textos, para a etapa de extração de conhecimento e classificação em Aprendizado de Máquina (AM) (SEBASTIANI, 2002). *Bag of Words*, *Word2Vec*, *Term Frequency-Inverse Document Frequency* (TFIDF) e *Keras Tokenizer* são alguns métodos apontados na literatura para a tarefa de vetorização de textos (ZHU et al., 2016; ZHANG; JIN; ZHOU, 2010; GUPTA et al., 2018). Neste trabalho o método TFIDF foi escolhido por ser amplamente utilizado para vetorização de textos em aprendizado de máquina, sendo apontado em diversos trabalhos, tais como Sebastiani (2002), Ferreira (2018), Chiong et al. (2018) e Gupta et al. (2018).

Para identificar o grau de importância das palavras dentro de um conjunto de documentos o TFIDF se utiliza do *Term Frequency* (TF), que consiste na quantidade de vezes

em que uma palavra aparece em um universo de documentos; e do Inverse Document Frequency (IDF), que é a quantidade de documentos em que uma palavra aparece dividido pelo total de documentos existentes. A medida TFIDF surge da multiplicação entre o TF e o IDF de cada palavra, como detalham as Equação 2.1, 2.2 e 2.3

$$TF = \frac{soma\_termo}{total\_termos} \tag{2.1}$$

$$IDF = 1 + log_e \frac{total\_doc}{docs\_termo} \tag{2.2}$$

$$TFIDF = TF * IDF \tag{2.3}$$

Em que, *soma_termo* representa a soma das vezes em que um termo aparece no documento, *total_termos* é o total de termos presentes no documento, *total_doc* é o total de documentos utilizados e *docs_termo* representa o total de documentos que apresentam determinado termo.

No caso de experimentos com classificadores de aprendizado profundo, foi utilizado para a vetorização dos textos o método *Keras Tokenizer*, apontado em estudos recentes que envolvem *deep learning* (LOZHNIKOV; DERCZYNSKI; MAZZARA, 2018; SAN, 2018; GUPTA et al., 2018). O *Keras Tokenizer* permite vetorizar um *corpus* de texto, transformando cada texto em uma sequência de inteiros (cada inteiro sendo o índice de uma palavra em um dicionário), com base na contagem de *tokens* (palavras). A medida é baseada no TFIDF. Por padrão, toda a pontuação é removida, transformando os textos em sequências de palavras separadas por espaços. Essas seqüências são, então, divididas em listas de *tokens* para serem indexados. A medida é calculada por meio da função *Tokenizer* do *Keras*.

2.3 MODELOS DE APRENDIZADO DE MÁQUINA

Algoritmos de aprendizado de máquina, identificam padrões afim de inferir conhecimento a partir de um conjunto de dados de treinamento, o aprendizado é testado através de outro conjunto de dados no qual o algoritmo deve classificar corretamente os objetos (MICHIE et al., 1994). Um classificador de aprendizado de máquina pode inferir conhecimento de três formas principais (CHAPELLE; SCHÖLKOPF; ZIEN, 2006; GUERREIRO, 2017):

- **Aprendizado supervisionado:** consiste no treinamento do modelo por meio de um conjunto de dados previamente classificado;
- **Aprendizado não supervisionado:** nesse tipo de abordagem o conjunto de treinamento não está previamente classificado. O que existe são dados distribuídos, que devem ser classificados de acordo com a densidade do conjunto apresentado. Uma configuração comum consiste no agrupamento de dados com base em suas similaridades criando grupos, os chamados *clusters*;

- **Aprendizado semi-supervisionado:** utiliza-se dados rotulados e não-rotulados, com o objetivo de classificar os dados não rotulados. A classificação é efetuada com base na aprendizagem feita por meio do conjunto rotulado. Este tipo de algoritmo faz uso de poucos dados rotulados, a pequena quantidade de dados rotulados em meio ao conjunto de dados não rotulados pode melhorar a precisão de classificação do algoritmo.

Nesta dissertação os experimentos foram realizados com dois algoritmos supervisionados, o *Support Vector Machine* e o *Naive Bayes* (NB), que estão detalhados no decorrer desta seção.

2.3.1 *Support Vector Machine*

O *Support Vector Machine* (SVM) é baseado na teoria do aprendizado estatístico (VAPNIK, 2013), foi inicialmente pensado para problemas lineares, mas também consegue lidar com problemas não-lineares. O algoritmo objetiva encontrar um hiperplano que divida as classes em um plano cartesiano. Este hiperplano é obtido a partir do treinamento dos documentos para, posteriormente, classificar os dados de teste. A Figura 1 exemplifica a separação linear e não-linear realizada por um SVM.

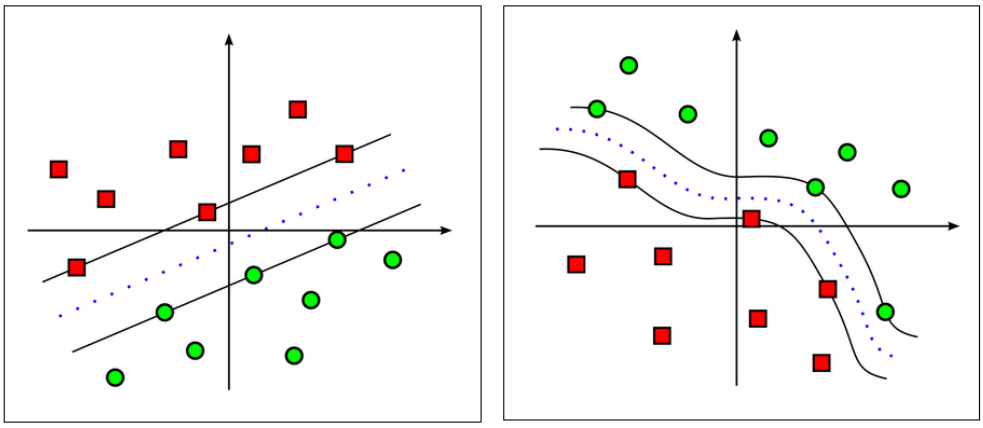

Figura 1 – Separação linear e não-linear com o SVM. Figura adaptada de Ruiz-Gonzalez et al. (2014)

Um hiperplano é considerado ótimo ao separar os vetores das classes sem erro e com distância máxima para com os vetores mais próximos. Para lidar com dados não-linearmente separáveis são utilizadas as chamadas funções kernels, tais como: Polynomial, RBF e Sigmoid. Essas permitem o mapeamento do conjunto não-linear para um espaço dimensional maior, denominado de espaço de características, tornando possível a separação linear (FERREIRA, 2018).

2.3.2 *Naive Bayes*

O NB é um algoritmo baseado no Teorema de Bayes. É conhecido como um classificador simples, mas que apresenta resultados satisfatórios em diversos problemas do mundo real (RISH, 2001; FERREIRA, 2018). O NB calcula a probabilidade de um dado elemento pertencer a uma determinada classe por meio da Eq. 2.4.

$$P(C|x_i) = \frac{P(x_i|c)P(c)}{P(x_i)} \tag{2.4}$$

Em que $P(C|x_i)$ é a probabilidade posterior de um elemento pertencer a uma dada classe; $P(x_i)$ é a probabilidade de cada atributo; $P(x_i|c)$ é a probabilidade *a priori* de um elemento pertencer a uma classe e; P(c) é a probabilidade original da classe (FERREIRA, 2018).

2.4 MODELOS DE APRENDIZADO PROFUNDO

Representation learning é um conjunto de métodos que permitem que uma máquina seja alimentada com dados brutos e descubra automaticamente as representações necessárias para previsão ou classificação dos dados. Os métodos de aprendizado profundo são métodos de *representarion learning* com múltiplos níveis de representação, obtidos pela composição de módulos não lineares, que transformam a representação original, começando com a entrada bruta, em uma representação em um nível um pouco mais abstrato (LECUN; BENGIO; HINTON, 2015). Com a composição de tais transformações, funções muito complexas podem ser aprendidas. Para tarefas de classificação, camadas mais altas de representação amplificam aspectos da entrada que são importantes para a discriminação e suprimem variações irrelevantes (LECUN; BENGIO; HINTON, 2015). Estão detalhadas nesta seção as redes de aprendizado profundo *Long short-term Memory* (LSTM) e *Convolutional Neural Network* (CNN), que foram escolhidas devido ao seu vasto uso, bem como de suas variações, para os diversos problemas da computação, incluindo AS.

2.4.1 *Long short-term Memory*

A rede LSTM é uma variante da *Recurrent Neural Network* (RNN). RNNs são redes cujos neurônios enviam sinais de feedback uns aos outros (GROSSBERG, 2013), isso permite que elas exibam um comportamento temporal dinâmico. Ao contrário das redes *feedforward*, as RNNs podem usar seu estado interno (memória) para processar as sequências de entradas. Todas as RNNs têm a forma de uma cadeia de módulos repetitivos, em RNNs padrão o módulo de repetição, normalmente, tem uma estrutura simples, o que diferencia as LSTMs que possuem uma estrutura mais complicada. Ao invés de ter uma única camada de rede neural, há quatro camadas interagindo de uma maneira especial

na LSTM, além disso, possui dois estados: *hidden state* e *cell state* (ZHANG; WANG; LIU, 2018).

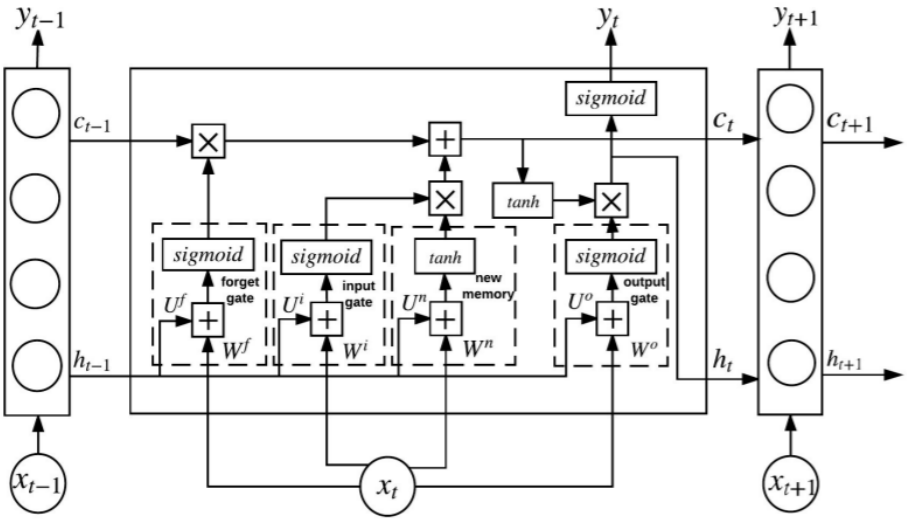

Figura 2 – LSTM (ZHANG; WANG; LIU, 2018)

A Figura 2 detalha o funcionamento de uma LSTM, em que na etapa de tempo t a rede decide quais informações deve manter na memória. Esta decisão é tomada por meio da função sigmóide, na camada chamada de *forget gate* (f_t), que gera um número entre [0 1], em que 1 significa manter completamente a informação atualmente armazenada e 0 significa esquecer completamente essa informação (ZHANG; WANG; LIU, 2018).

Após a LSTM decidir quais novas informações devem ser armazenadas no *cell state*, deve decidir quais valores serão atualizados, essa etapa ocorre no *input gate*. Em seguida, uma função *tanh* cria um vetor de novos valores candidatos, que serão adicionados ao *cell state*. A LSTM combina esses dois vetores para criar uma atualização para o estado atual (ZHANG; WANG; LIU, 2018).

Para atualizar o antigo estado em um novo *cell state*, a LSTM utiliza a camada *forget gate* que pode controlar o gradiente que passa por ele e permite exclusões de "memórias"(ZHANG; WANG; LIU, 2018).

Por fim, a LSTM decide o *output* (saída), que é baseada no *cell state*. Primeiro a camada sigmóide é executada e decide quais partes do *cell state* irão para o *output gate* (potão de saída). Então, a LSTM coloca o estado da célula através da função *tanh* e multiplica pela saída da função sigmóide do *output gate*, de modo que sejam produzidos apenas os valores desejados (ZHANG; WANG; LIU, 2018).

2.4.2 *Convolutional Neural Network*

A CNN é um tipo especial de rede neural *feedforward* que possui design inspirado no córtex visual humano. O córtex visual contém muitas células responsáveis pela detecção da luz, chamadas de campos receptivos. Essas células atuam como filtros locais sobre

o espaço de entrada. A CNN consiste em múltiplas camadas convolucionais, cada uma desempenha a função que é processada pelas células no córtex visual (ZHANG; WANG; LIU, 2018). A Figura 3 detalha a arquitetura de uma CNN.

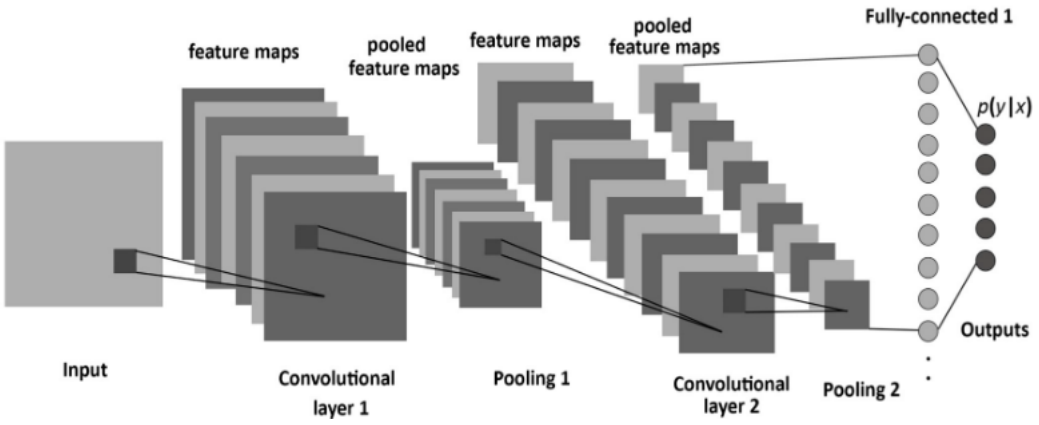

Figura 3 – CNN (ALBELWI; MAHMOOD, 2017)

A camada convolucional é composta por um conjunto de *kernels* ou filtros que visam extrair características (*features*) da entrada (*input*). Cada *kernel* é usado para calcular um *feature map*. As unidades dos *feature maps* só podem se conectar a uma pequena região da entrada, chamada de campo receptivo. Os modelos de CNN têm uma estrutura padrão que consiste em camadas convolucionais alternadas a camadas de *pooling* (frequentemente cada camada de *pooling* é colocada depois de uma camada convolucional) (ALBELWI; MAHMOOD, 2017).

Geralmente, um novo *feature map* é gerado ao deslizar um filtro sobre a entrada, seguido por uma função de ativação não linear para introduzir a não linearidade ao modelo. Todas as unidades compartilham os mesmos pesos (filtros) entre cada *feature map* (ALBELWI; MAHMOOD, 2017). A vantagem de compartilhar pesos é o número reduzido de parâmetros e a capacidade de detectar o mesmo recurso, independentemente de sua localização nas entradas (ALBELWI; MAHMOOD, 2017; CHEN et al., 2015).

2.4.3 Configurações das redes profundas

Neste trabalho foi utilizada, para os experimentos com múltiplas classes, a função de ativação softmax em lugar da tangente ($tanh$), exemplificada na Figura 2, tendo em vista que o softmax responde melhor a problemas envolvendo múltiplas classes (DUAN et al., 2003).

$$f = softmax(W_m h_l + b) \tag{2.5}$$

Na Equação 2.5 tem-se W_m como a matriz de pesos, h_l como a última saída da camada oculta e b como o valor do bias da função (CHEN et al., 2018). Os experimentos para múltiplas classes realizados neste trabalho utilizam, ainda, o *Stochastic Gradient Descent*

(SGD) como otimizador. O SGD, em vez de calcular exatamente o gradiente, calcula cada iteração com base em um único exemplo escolhido aleatoriamente (BOTTOU, 2010).

$$w_{t+1} = w_t - \gamma_t \nabla_w Q(z_t, w_t) \tag{2.6}$$

A Equação 2.6 descreve o SGD, em que cada iteração atualiza os pesos w e γ é o ganho escolhido. Cada iteração estima o gradiente com base em um único exemplo aleatório z_t. O processo estocástico $(w_t, t = 1, ...)$ depende dos exemplos aleatoriamente escolhidos em cada iteração (BOTTOU, 2010).

Para os experimentos binários, Sigmóide foi a função de ativação utilizada, detalhada na Equação 2.7, em que x representa a soma ponderada do vetor de entrada.

$$\sigma(x) = \frac{1}{1 + e^{-x}} \tag{2.7}$$

Para avaliar e melhorar o desempenho do modelo, tem-se a função objetivo, que visa, primordialmente, minimizar o erro. Neste trabalho *Cross-entropy error* foi a função de perda (objetivo) utilizada, detalhada na Equação 2.8.

$$loss = -\sum_{i \in D} \sum_{j \in C} \hat{y}_j^i log y_j^i + \lambda ||\theta||^2 \tag{2.8}$$

Em que D representa os dados de treinamento, C representa as classes/polaridades, $\hat{y}$ é a classe atual, y é a classe predita e $\lambda||\theta||^2$ representam a função de regularização e o vetor de parâmetros (CHEN et al., 2018).

2.5 SISTEMAS DE MÚLTIPLOS CLASSIFICADORES

Os Sistemas de Múltiplos Classificadores tem como foco a criação de métodos que unem/combinam diferentes abordagens de aprendizado de máquina, partindo da premissa de que a combinação de diferentes resultados provoca um desempenho superior aos resultados de classificadores únicos (FILHO, 2014).

Em MCS, vários classificadores são treinados no mesmo conjunto de treinamento ou em partições do mesmo conjunto de treinamento. A saída final do sistema é dada por uma combinação das saídas dos classificadores individuais (PINHEIRO; CAVALCANTI; TSANG, 2019). Há três motivações principais que apontam para a alternativa de combinação de classificadores: motivação estatística, representacional e computacional (FILHO, 2014; DIETTERICH; BAKIRI, 1994).

- **Motivação Estatística**: considerando que o projetista precisa escolher entre n classificadores como solução para determinado problema, é possível que o classificador com pior desempenho seja escolhido para a tarefa. Combinar alguns classificadores é uma alternativa para evitar o pior caso.

- **Motivação Representacional**: em aplicações de aprendizado de máquina, a hipótese verdadeira pode não estar representada no espaço de hipóteses disponível. Neste caso, a combinação de diversas hipóteses cria uma nova hipótese, expandindo-se o espaço de hipóteses representáveis. Logo, torna-se possível atingir um desempenho superior ao da melhor solução existente no espaço de hipóteses original (FILHO, 2014).

- **Motivação Computacional**: os diversos algoritmos de AM podem ficar presos em um ótimo local, que é a melhor solução do problema dentre as soluções vizinhas, mas, ainda assim, não é a melhor solução (HERNÁNDEZ-LOBATO; HOFFMAN; GHAHRAMANI, 2014). A combinação dos resultados de vários algoritmos é capaz de se aproximar da hipótese verdadeira de forma mais eficiente do que com um único classificador (FILHO, 2014).

Segundo Roli, Giacinto e Vernazza (2001) a construção de um *Multiple Classifiers Systems* (MCS) envolve duas etapas principais: construção do comitê e processo de combinação. A etapa de **construção do comitê** consiste em criar o conjunto de modelos, também conhecido como *pool* de classificadores, considerando que exista diversidade no conjunto. Para garantir a diversidade existem diversas abordagens, tais como:

- Diferentes conjuntos de dados de treinamentos, em que, geralmente, os subconjuntos são formados por meio de técnicas de *resampling* como *bootstrapping* (DIETTERICH, 2000) ou *bagging* (BRYLL; GUTIERREZ-OSUNA; QUEK, 2003), na maioria das vezes com reposição;

- Diferentes parâmetros de treinamento - considerando um mesmo classificador, porém com diferentes configurações. Ex.: redes neurais, utiliza-se diferentes pesos iniciais, número de camadas, funções de ativação, algoritmos de treinamento e demais parâmetros;

- Diferentes tipos de classificadores - realizando combinações entre modelos de diferentes características, por exemplo, um *multilayer perceptron*, um SVM e um NB.

A etapa de **processo de combinação** consiste em integrar os resultados dos diferentes modelos em uma classificação final.

O processo de combinação de classificadores ocorre por variados métodos, considerando a média dos resultados ou pelo voto majoritário. Neste trabalho foi escolhida uma média simples dos resultados dos classificadores. A escolha por média em detrimento ao voto majoritário, amplamente utilizado, ocorreu devido ao número de classificadores presentes no comitê em relação a quantidade de classes/polaridades. Em uma combinação com

dois ou três classificadores e cinco polaridades há grande chance da escolha pelo voto majoritário ser aleatória. A Eq. 2.9 detalha a combinação por média ponderada

$$y_f = \frac{1}{n}\sum_{i=1}^{n} w(i).y_i \tag{2.9}$$

Em que y_f é o resultado final da combinação; y_i é a classificação do modelo i e; $w(i)$ é o peso atribuído a classificação y_i (SILVA, 2017). Considerando que nem sempre as médias apresentam valores inteiros, para obter a classificação, o valor de y_f é arredondado para a extremidade mais próxima, seja superior ou infeior. Outros métodos poderiam ser adotados realizar a combinação dos classificadores, como a mediana, porém a média foi escolhida neste trabalho por seu vasto uso (FILHO, 2014), além de ter se mostrado um método interessante ao decorrer dos experimentos.

2.6 MÉTRICAS DE AVALIAÇÃO

A abordagem proposta concentra-se em métodos que buscam inferir os sentimentos e opiniões presentes em textos. Isto implica em uma tarefa de classificação que, em áreas como PLN, é comumente avaliada por meio das medidas *precision*, *recall* e *f-score* (FERRI; HERNÁNDEZ-ORALLO; MODROIU, 2009; FERREIRA, 2018), que variam entre o intervalo [0 1]. As equações 2.10, 2.11 e 2.12 detalham o cálculo das métricas.

$$precision = \frac{VP}{VP + FP} \tag{2.10}$$

$$recall = \frac{VP}{VP + FN} \tag{2.11}$$

$$f - score = 2 * \frac{precision * recall}{precision + recall} \tag{2.12}$$

Em que Verdadeiro Positivo (VP) é o número de elementos positivos classificados como positivos; Falso Positivo (FP) é o número de elementos falsos classificados como positivos e; Falso Negativo (FN) representa o número de elementos falsos classificados como falsos.

Mesmo *precision*, *recall* e *f-score* sendo amplamente utilizadas para este tipo de problema, Ferri, Hernández-Orallo e Modroiu (2009) apontam a acurácia como a medida mais comum para avaliar um classificador. É definida como o grau de previsões corretas de um modelo (ou, inversamente, a porcentagem de erros de classificação incorreta).

$$accuracy = \frac{VP + VN}{VP + FP + VN + FN} \tag{2.13}$$

A Equação 2.13 detalha o cálculo da acurácia, sendo Verdadeiro Negativo (VN) o total de elementos positivos classificados como negativos. A acurácia também retorna um valor entre [0 1], sendo 0 o pior caso e 1 o melhor caso.

2.7 ANÁLISES ESTATÍSTICAS

Para decidir qual teste estatístico deve ser utilizado para analisar se há diferença entre as amostras, deve ser realizado um teste de normalidade. Caso os dados obedeçam a uma distribuição normal, um teste paramétrico deve ser utilizado, caso os dados não obedeçam a uma distribuição normal, uma alternativa não paramétrica será utilizada.

2.7.1 Teste de *Shapiro-Wilk*

O teste de *Shapiro-Wilk* é utilizado para verificar a normalidade dos dados, antes de se definir o teste de hipótese que deve ser utilizado para verificar diferença entre amostras (GHASEMI; ZAHEDIASL, 2012). O teste pode ser dividido em 3 partes, a saber:

1. Elaboração das hipóteses:
   H_0 : *a amostra provém de uma população normal*
   H_1 : *a amostra não provém de uma população normal*

2. Definir o nível de significância do teste:
   $\alpha = 0.05$, ou seja, o teste terá um nível de confiança de 95%.

3. Cálculo do $p - value$ e tomada de decisão:
   Rejeita H_0 se *p-value* $< \alpha$;
   Não é possível rejeitar H_0 se *p-value* $> \alpha$.

Caso a amostra rejeite H_0, um teste não paramétrico deverá ser utilizado para comparação das amostras, neste estudo o teste de *Wilcoxon*. Caso não seja possível rejeitar H_0 um teste paramétrico deverá ser utilizado, neste estudo o teste *t-Student*.

2.7.2 Análises de Distribuições

As análises de distribuições tem como objetivo verificar se as amostras distintas apresentam semelhanças (SILVA, 2017). Caso a amostra provenha de uma distribuição normal, é utilizado o teste *t-Student*, que consiste nos seguintes passos (SILVA, 2017):

1. Elaboração das hipóteses:
   $H_0 : \mu_1 = \mu_2$: *as amostras possuem a mesma variância*
   $H_1 : \mu_1 \neq \mu_2$: *as amostras não possuem a mesma variância*

2. Definir o nível de significância, ou seja, o valor de α. Neste trabalho considerou-se $\alpha = 0.05$, em um nível de confiança de 95%.

3. Cálculo do t e tomada de decisão:
   Rejeita H_0 se $t_{calculado} \neq t_\alpha$, em que t_α é o valor crítico da distribuição *t-Student*;

No caso de amostras que não obedecem a uma distribuição normal, o teste de *Wilcoxon* deve ser realizado, seguindo os seguintes passos:

1. Elaboração das hipóteses:
   $H_0 : \triangle = 0$: *a mediana da diferença entre as amostras é nula*
   $H_1 : \triangle \neq 0$: *a mediana da diferença entre as amostras não é nula*

2. Definir nível de significância, $\alpha = 0.05$, dessa forma tem-se um nível de 95% de confiança;

3. Cálculo da estatística w e tomada de decisão:
   Rejeitar a H_0 caso o $w_{calculado} < w_{\alpha 1}$, ou $w_{calculado} > w_{\alpha 2}$ onde $w_{\alpha 1}$ e $w_{\alpha 2}$ são os valores críticos da distribuição *Wilcoxon*.

2.8 TRABALHOS RELACIONADOS

A tarefa de Análise de Sentimentos (AS) pode estar inserida nos mais variados segmentos da sociedade e, portanto, têm motivado a condução de muitas pesquisas. Nesta dissertação é realizada uma comparação entre os classificadores de aprendizado de máquina tradicional e de aprendizado profundo mais utilizados para análise de sentimentos.

Alguns trabalhos na literatura fazem uso apenas de dicionários léxicos para a tarefa de AS, como em Taboada et al. (2011) que apresentam uma abordagem baseada em léxico para extrair sentimentos a partir de opiniões sobre livros, hoteis, filmes e outros. O método proposto obteve acurácia entre 70% e 80%, a depender do domínio analisado. Comparações com modelos de aprendizado de máquina não foram realizadas.

Em Dey et al. (2016) são utilizados algoritmos de aprendizado de máquina tradicional, como o NB e o *K Nearest Neighbor* (KNN), para facilitar a rápida descoberta de opiniões envolvendo resenhas de filmes e análises de hotéis. Os dados encontram-se subdividos em duas polaridades de sentimentos (positiva ou negativa). A métrica *precision* foi utilizada para avaliar o desempenho dos classificadores e o melhor desempenho obtido em seus experimentos foi de 84.84%, por meio do classificador Naive Bayes. No trabalho não foram comparados os desempenhos de métodos de aprendizado profundo ou combinações de classificadores, também não foram consideradas outras polaridades de sentimentos, além de positiva ou negativa.

Em Xu et al. (2016) é apresentada uma variação da rede de aprendizado profundo LSTM, denominada de *Cached Long Short-Term Memory* (CLSTM). A CLSTM captura a informação semântica geral em textos longos ao introduzir um mecanismo de cache, que divide a memória em vários grupos com diferentes taxas de esquecimento e, assim, permite que a rede mantenha as informações de sentimentos dentro de uma unidade recorrente. A CLSTM teve seu desempenho comparado a demais algoritmos de aprendizado profundo,

não considerando algoritmos tradicionais que poderiam obter desempenho similar com um menor tempo de execução.

Em Akhtar et al. (2017) é construído um comitê de classificadores, utilizando os classificadores *Maximum Entropy*, SVM e *Conditional Random Field*, e um modelo de otimização, o *Particle Swarm Optimization* (PSO). São realizados experimentos em duas bases de dados, sendo a primeira com opiniões acerca de um restaurante e a segunda contendo opiniões de *laptops*. A classificação ocorre nas polaridades de sentimentos positiva, negativa ou neutra e o comitê se mostrou uma alternativa interessante tendo em vista seus resultados competitivos. No trabalho não foram consideradas abordagens de aprendizado profundo.

Neste trabalho, além da comparação de desempenho entre os classificadores únicos e os comitês que combinam algoritmos de AM tradicional e aprendizado profundo, tem-se um ajuste nas redes de aprendizado profundo que fazem parte das combinações de classificadores: as redes possuem seu número de épocas de treinamento reduzido. Espera-se com isso que haja uma redução no tempo de treinamento dos comitês. Todos os modelos são testados em experimentos binários e com múltiplas classes (5 polaridades de sentimentos), a fim de analisar o comportamento dos modelos de acordo com os dois cenários.

2.9 CONSIDERAÇÕES DO CAPÍTULO

Este capítulo abordou os principais conceitos de análise de sentimentos, considerando seu estudo através não só da computação, mas também através das demais áreas do conhecimento. Foram explicitadas as técnicas de pré-processamento de texto utilizadas nesta dissertação como remoção de *stopwords*, aplicação de *stemming* e n-gram, além das técnicas de *oversampling* e *undersampling* que visam lidar com o problema de dados desbalanceados. As técnicas de vetorização dos textos, TFIDF e Tokenizer, também foram apresentadas, considerando suas particularidades.

Os principais conceitos envolvendo algoritmos de aprendizado de máquina e aprendizado profundo e sistemas de múltiplos classificadores também foram considerados neste capítulo, bem como os algoritmos utilizados nos experimentos realizados.

Foram apresentadas, ainda, as métricas *precision*, *recall*, *f-score* e acurácia, utilizadas para avaliação do desempenho dos classifcadores, bem como os testes estatísticos utilizados para averiguar a diferença entre os desempenhos obtidos. Por fim, tem-se a descrição de alguns trabalhos relacionados ao tema desta dissertação.

3 MÉTODO PROPOSTO

Este capítulo apresenta uma sistematização acerca do problema de pesquisa, detalha a arquitetura do método proposto, além dos demais modelos utilizados com fins de comparação. Por fim, encontram-se as considerações do capítulo.

3.1 ANÁLISE DO PROBLEMA

A análise de sentimentos visa a classificação de textos de forma automática. Para tal, são encontrados diversos métodos na literatura, como o uso de dicionários léxicos e métodos de aprendizado de máquina tradicional e de aprendizado profundo. Os métodos de AM tradicional e aprendizado profundo possuem destaque em termos de desempenho para a classificação de sentimentos.

Trabalhos na literatura também apresentam comitês de classificadores que alcançam resultado superior em relação aos classificadores únicos ao combinar métodos de AM tradicional ou métodos de aprendizado profundo, não contando com a combinação das duas técnicas (AM tradicional ou profunda) em um mesmo comitê. Tendo em vista os resultados dos classificadores únicos e dos comitês que consideram uma das técnicas, este trabalho propõe a construção de comitês de classificadores que combinam modelos de AM tradicional e profunda, visando uma melhora de desempenho em relação aos métodos já testados.

Como o tempo de execução é um problema tanto ao lidar com comitês (KUNCHEVA, 2002) quanto com aprendizado profundo (SUNDERMEYER; SCHLÜTER; NEY, 2012), o método proposto atua neste problema ao desenvolver combinações em que é utilizado um menor número de épocas de treinamento para os algoritmos de aprendizado profundo. Espera-se com isso minimizar o tempo de execução dos comitês sem que se perca no quesito desempenho, devido aos conhecimento das combinações. A comparação entre os comitês construídos e os modelos únicos também são pontos abordados nesta pesquisa, a fim de analisar as vantagens e desvantagens de cada método.

3.2 ARQUITETURA

A arquitetura de construção dos comitês se divide em 4 etapas principais. A primeira etapa consiste no pré-processamento dos dados, com a aplicação das técnicas de *stopwords, stemming* e *n-gram*, nessa etapa os textos também são divididos em três conjuntos: treinamento (80%), validação (10%) e teste (10%). O percentual utilizado para a divisão dos dados foi estabelecido ao considerar trabalhos apresentados na literatura que fazem uso das bases de dados selecionadas nesta dissertação (LE; MIKOLOV, 2014; TANG; QIN; LIU, 2015a; TANG; QIN; LIU, 2015b; CHEN et al., 2016; YANG et al., 2016).

A segunda etapa envolve o processo de vetorização dos textos, tendo em vista que os classificadores não conseguem atuar com texto em linguagem natural (SEBASTIANI, 2002), essa é uma etapa fundamental para que os modelos possam extrair conhecimento dos dados. Ao considerar os experimentos apresentados na literatura (ver 2.2.3, pág. 20), optou-se por vetorizar os textos de duas formas: algoritmos de aprendizado de máquina tradicional teriam como entrada os vetores do TFIDF, pois essa técnica está associada a bons resultados com AM tradicional; para os algoritmos de aprendizado profundo tem-se como entrada os vetores advindos do *Keras Tokenizer*, pois esta técnica é muito utilizada quando se trata deste tipo de rede. A Figura 4 detalha esse processo.

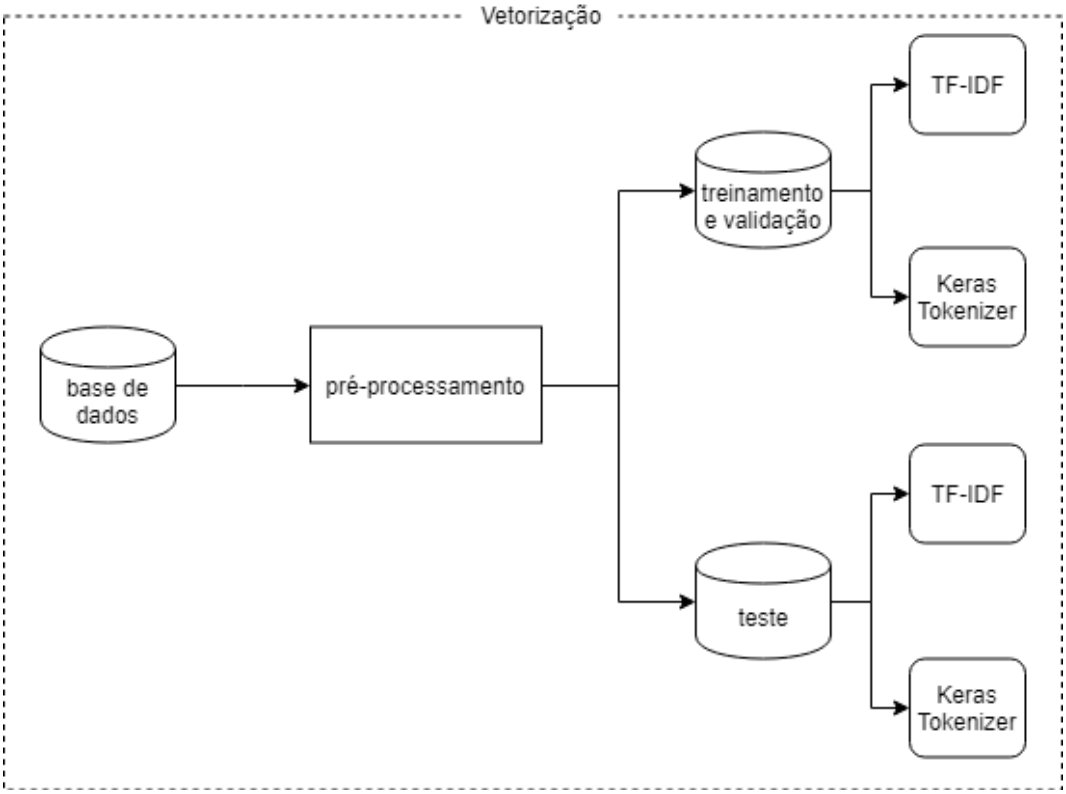

Figura 4 – Pré-processamento dos dados

O primeiro ponto apresentado na Figura 4 está ligado ao pré-processamento dos textos que consiste na remoção dos *stopwords* (palavras que não agregam ao contexto), na aplicação da técnica de *stemming* (redução das palavras ao seu radical) e na aplicação da técnica de n-gram (agrupa n palavras), com $n = 2$. Essas técnicas são utilizadas a fim de deixar os dados mais representativos. O segundo ponto equivale à divisão dos dados nos conjuntos de treinamento (80%), validação (10%) e teste (10%), para que se possa treinar e avaliar os classificadores. A partir da divisão dos dados, o último ponto refere-se a vetorização dos textos, por meio do TFIDF e do *Keras Tokenizer*, para que classificadores de AM tradicional possam ter os vetores construídos com o TFIDF como entrada, enquanto que os vetores construídos com o *Keras Tokenizer* servem de entradas para as redes profundas.

A próxima etapa da arquitetura proposta consiste na contrução do chamado *Pool* de classificadores, que consiste na seleção dos classificadores que serão combinados (FILHO, 2014). O *pool* inicial indica a quantidade de classificadores que farão parte do conjunto.

O método de geração de comitês, *Bagging* (BREIMAN, 1996), é utilizado para treinar os modelos presentes no *pool* de classificadores. De acordo com Filho (2014), o *Bagging* é baseado na geração de amostras, a partir do conjunto de dados de treinamento. As amostras mantém o tamanho do conjunto original, porém com pequenas modificações. O método *Bagging* foi escolhido para este trabalho por ser ideal para trabalhar com redes neurais (FILHO, 2014), e boa parte dos modelos utilizados nesta dissertação são redes de aprendizado profundo. Muitas aplicações deste método também são descritas utilizando algoritmos de AM tradicional, como o SVM em Mordelet e Vert (2014). Ao finalizar a criação do comitê, pode-se utilizar diversos métodos para classificação final, segundo Filho (2014) os mais comuns são a média ou a escolha da classe com maior frequência (voto majoritário). A Figura 5 detalha o esquema para construção dos modelos, em que os dados de treinamento são vetorizados em um primeiro momento para, posteriormente, seguirem de entrada ao *pool* de classificadores. Os classificadores são combinados na etapa de *pool*, formando o novo comitê, que tem seu desempenho avaliado através dos dados de teste.

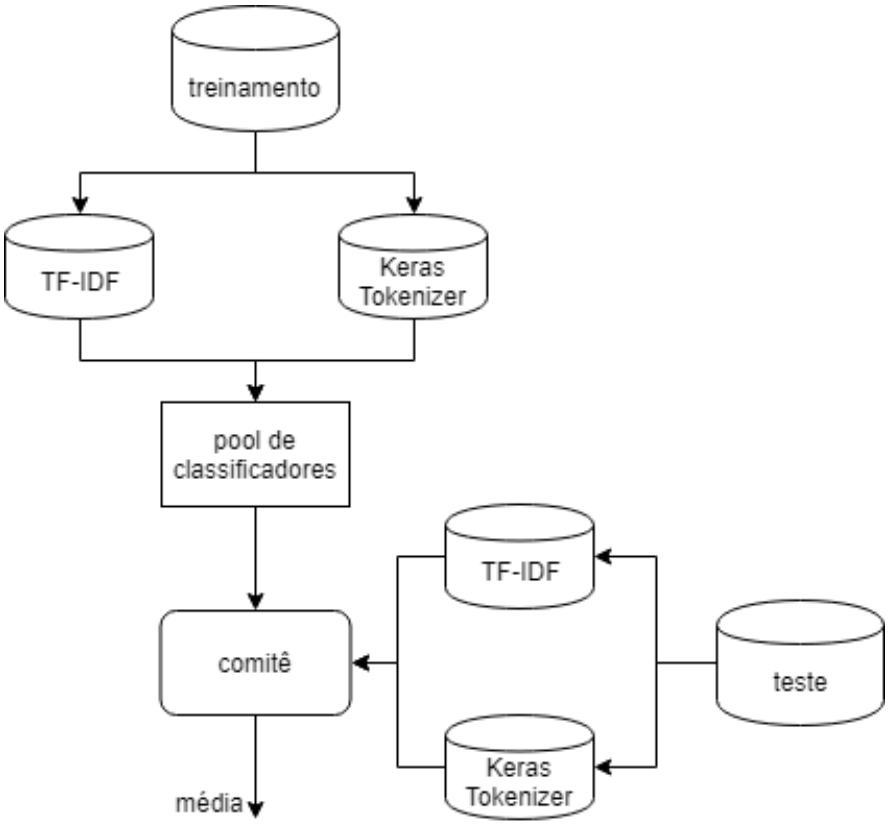

Figura 5 – Combinação de classificadores

O resultado do comitê ocorre por meio da média, apontada por Filho (2014) como um dos métodos mais utilizados para a classificação final de comitês. Essa escolha da

média considera que os exemplos de algumas classes podem ser muito similares. Ex.: uma combinação realizada por meio de três classificadores, em que o primeiro classifica o comentário como *muito negativo* (1), o segundo classifica como *negativo* (2) e o terceiro classifica como *neutro* (3), a decisão por voto majoritário seria tomada de forma arbitrária. Ao considerar a média, a classificação final seria para o comentário *negativo* (2), o que se mostra uma decisão interessante por adotar uma solução moderada, em detrimento de escolher entre extremos.

Os critérios utilizados para gerar diversidade e selecionar os classificadores que serão combinados, foram: 1 comitê formado apenas por classificadores de aprendizado de máquina tradicional; 1 comitê formado apenas por classificadores de aprendizado profundo; 2 comitês que combinam abordagens de AM tradicional e aprendizagem profunda. Esses critérios foram definidos ao considerar a questão de pesquisa que busca analisar se os melhores comitês combinam métodos de AM tradicional e aprendizado profundo ou advém de modelos do mesmo tipo. Assim, serão comparados os desempenhos tanto dos quatro comitês gerados quanto dos classificadores únicos.

3.3 COMPARAÇÃO DE CLASSIFICADORES

Foram realizadas comparações entre os desempenhos dos classificadores únicos e dos comitês, além disso a variável tempo de execução também foi analisada para entender como a diminuição do número de épocas para os classificadores dos comitês influenciou no tempo e desempenho dos modelos. A Figura 6 traz um esquema de como foi realizada a comparação do desempenho entre os diferentes modelos, por meio da acurácia.

Como detalha a Figura 6, são treinados, a partir dos mesmos dados, n modelos de classificadores únicos e i modelos de comitês de classificadores. A acurácia final dos modelos é utilizada como métrica para comparação dos desempenhos.

Para viabilizar as análises estatíscas e comparação entre os desempenhos, foi utilizado o método *k-fold crossvalidation* (BROWNE, 2000), com k igual a 10. Através desse método, são realizadas k repetições de cada classificador, de modo a se atingir uma amostra de tamanho y com os valores de acurácia dos modelos. A partir dessas amostras tornou-se possível analisar se há diferença estatística entre os desempenhos dos diferentes classificadores.

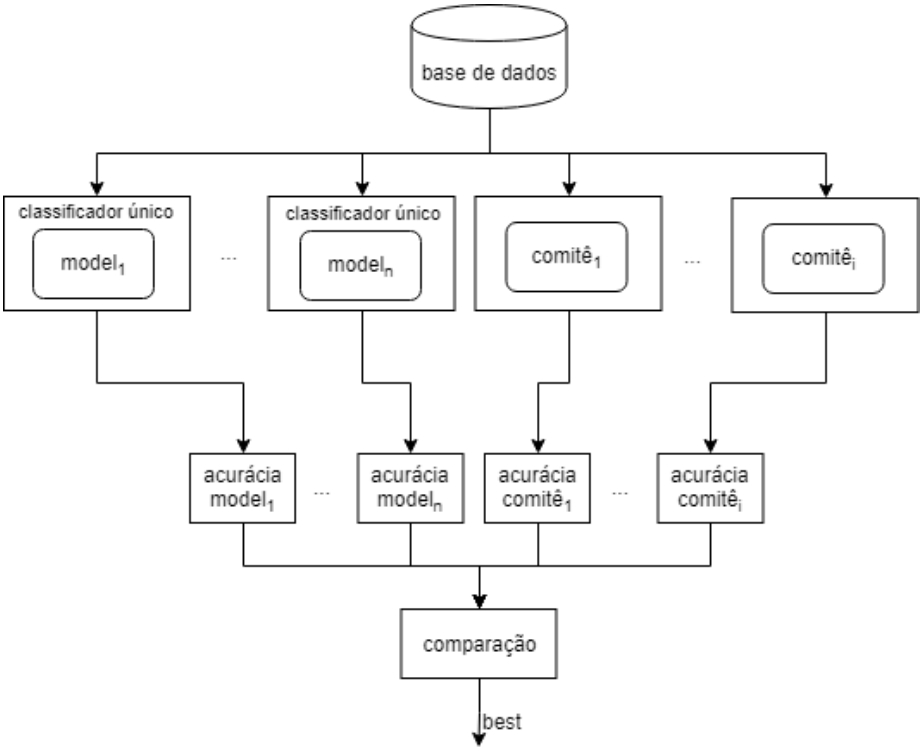

Figura 6 – Comparação de desempenhos dos classificadores

3.4 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo foi abordada a arquitetura do método proposto. A diversidade dos modelos está na combinação de classificadores de diferentes características, por meio de modelos de AM tradicional e modelos de aprendizado profundo. Um diferencial importante presente nas combinações realizadas está na diminuição do número de épocas de treinamento dos algoritmos de aprendizado profundo, por visar a diminuição do tempo de execução das combinações, sem perder desempenho, tendo em vista que o conhecimento dos demais classificadores será combinado. Os critérios que estabeleceram o TFIDF como técnica de vetorização para a entrada dos algoritmos de aprendizado de máquina tradicional e o *Keras Tokenizer* como técnica para vetorização para as entradas dos algoritmos de aprendizado profundo também foram abordados. A Comparação entre os classificadores únicos e as combinações é detalhada na seção 3.3, finalizando o capítulo.

4 METODOLOGIA DOS EXPERIMENTOS

Neste capítulo estão detalhadas as bases de dados utilizadas para os experimentos desta dissertação. A sequência em que são adotadas as técnicas de pré-processamento dos dados também são contempladas no capítulo, além da descrição dos parâmetros utilizados em cada modelo algorítmico.

4.1 BASES DE DADOS

Para realização dos experimentos desta dissertação foram selecionadas cinco bases de dados para análise de sentimentos. Foram realizados experimentos considerando o problema binário (positivo vs negativo) e com múltiplas classes. As bases selecionadas foram: IMDb *Review*, *Stanford Sentiment Treebank*, Yelp 2013, Yelp 2014 e Yelp 2015. Para realizar os experimentos binários, as bases de dados passaram por um processo de redistribuição de classes, como é explicado ao decorrer desta seção. O processo de redistribuição de classes também foi realizado nas bases com mais de 5 (cinco) classes, devido a baixa quantidade de exemplos de determinadas polaridades de sentimentos.

4.1.1 IMDb *Review*

O IMDb Reviews, apresentado por Diao et al. (2014), é uma base de dados composta por 135.669 comentários de filmes. Esses comentários estão organizados em 10 classes (polaridades de sentimentos), definidas de acordo com a quantidade de estrelas presentes em cada comentário. O comentário classificado com uma estrela é considerado extremamente negativo enquanto o comentário classificado com dez estrelas é considerado extremamente positivo. A tabela 1 detalha a quantidade de exemplos disponíveis em cada uma das 10 classes presentes na base.

Tabela 1 – Quantidade de exemplos por classe - IMDb Review

classe	1	2	3	4	5	6	7	8	9	10
exemplos	4.394	2.997	4.144	6.244	9.654	15.200	23.127	33.848	17.385	18.665

É possível perceber um forte desbalanceamento da base ao observar o número de exemplos disponíveis por classe. A classe 2, com o menor número de exemplos, conta com menos de 3 mil sentenças, enquanto a classe 8, com o maior número de exemplos, conta com mais de 33 mil. Como tentativa de minimizar o problema de desbalanceamento da base, os experimentos desta pesquisa foram realizadas em duas etapas, em que o primeiro considera 5 classes, sendo elas: muito negativa, negativa, neutra, positiva e muito positiva. A tabela 2 descreve como foi realizada a aglutinação dos dados em 5 classes.

Tabela 2 – Reorganização em 5 classes - IMDb Review

classe original	1 e 2	3 e 4	5 e 6	7 e 8	9 e 10
nova classe	muito negativa	negativa	neutra	positiva	muito positiva

Restaram para os experimentos com 5 *labels*: 7.391 exemplos da classe *muito negativa*, 10.388 exemplos da classe *negativa*, 24.854 exemplos da classe *neutra*, 56.975 exemplos da classe *positiva* e 36.050 exemplos da classe *muito negativa*. O desbalanceamento dos dados continua sendo um problema eminente, assim, foram aplicadas as téncias de *oversampling* e *undersampling*, apresentadas na seção 2.2.2 nos dados de treinamento.

A segunda etapa de experimentos considera dados binários, a aglutinação da base em duas classes teve como critério: $label < 5 = negativo$ e $label > 6 = positivo$. As *labels* 5 e 6, apontadas como *neutra*, foram excluídas durante os experimentos binários.

Para os experimentos com dados binários restaram: 42.633 exemplos para a classe negativa e 93.025 exemplos para a classe positiva. Mais uma vez o desbalanceamento dos dados se mostrou um problema eminente e foram aplicadas as ténicas de *oversampling* e *undersampling* nos dados de treinamento.

4.1.2 *Stanford Sentiment Treebank*

A base de dados *Stanford Sentiment Treebank* (SSTb) é composta por um total de 239.232 sentenças, subdivididas em 11 polaridades de sentimentos que estão no intervalo entre 0 e 10. A polaridade 0 (zero) representa uma avaliação extremamente negativa enquanto a polaridade 10 (dez) significa uma avaliação extremamente positiva. A base do SSTb foi rotulado por 3 julgadores humanos (SOCHER et al., 2013) e os respectivos valores correspondentes a cada uma das 11 classes presentes na base estão detalhados na tabela 3.

Tabela 3 – Quantidade de exemplos por classe - SSTb

classe	**quantidade de exemplos**	**classe**	**quantidade de exemplos**
0	605	6	36.624
1	4.613	7	24.611
2	15.323	8	20.263
3	21.779	9	5.942
4	31.391	10	585
5	77.496		

Devido a grande quantidade de classes presentes na SSTb, muitos trabalhos presentes na literatura minimizam a quantidade de classes da base de dados para realizar expe-

rimentos (LE; MIKOLOV, 2014; SOCHER et al., 2013; LEI; YANG; YANG, 2018). É comum subdividir esta base em três polaridades, sendo positiva, negativa e neutra (SOCHER et al., 2013; LEI; YANG; YANG, 2018). Neste trabalho, assim como o IMDb Review, o SSTb será subdivido de duas formas, a primeira forma considerando 5 classes e a segunda forma considerando a binarização da base. A tabela 4 detalha a aglutinação em 5 classes do SSTb.

Tabela 4 – Reorganização em 5 classes - SStb

classe	0, 1 e 2	3 e 4	5 e 6	7 e 8	9 e 10
novas classes	muito negativa	negativa	neutra	positiva	muito positiva
quantidade de exemplos	20.541	53.170	114.120	44.874	6.527

No caso dos experimentos binários, foi adotado o mesmo limiar do IMDb Review, sendo $label < 5 = negativa$ e $label > 6 = positiva$. As classes 5 e 6 foram desconsideradas, por representarem a polaridade *neutra*. Para os experimentos binários restaram: 74.111 exemplos negativos e 51.401 exemplos positivos. As técnicas de *oversampling* e *undersampling* foram aplicadas no conjunto de treinamento a fim de minimizar o problema de desbalanceamento da base.

4.1.3 Yelp 2013, 2014 e 2015

O *Yelp Dataset Challenge* possui 3 bases de dados utilizadas nos experimentos deste trabalho, sendo: Yelp 2013; Yelp 2014 e; Yelp 2015. O conjunto de dados do Yelp é um subconjunto dos negócios da empresa, avaliações e dados de usuário que são utilizados para fins pessoais, educacionais e acadêmicos (YELP..., 2013; YELP..., 2014; YELP..., 2015). As bases do Yelp Challenge estão, originalmente, divididas em 5 polaridades de sentimentos, portanto não houve aglutinação de classes para os experimentos com 5 polaridades de sentimentos. A tabela 4.1.3 apresenta a quantidade de instâncias por classe referente a cada base do Yelp.

Classes (Yelp 2013)

classe	instâncias
1	28.584
2	29.635
3	47.597
4	109.813
5	119.389

Classes (Yelp 2014)

classe	instâncias
1	110.772
2	102.737
3	163.761
4	342.143
5	406.044

Classes (Yelp 2015)

classe	instâncias
1	159.812
2	140.608
3	222.719
4	466.599
5	579.527

No caso dos experimentos binários, foram considerados: $label < 3 = negativa$ e $label > 3 = positiva$. A classe *neutra* (3) foi excluída dos experimentos. Restando para a base do Yelp 2013 58.219 exemplos negativos e 229.202 exemplos positivos. A binarização para o Yelp 2014 ficou com 213.509 exemplos negativos e 748.187 exemplos negativos. Por fim, a base do Yelp 2015 binarizada ficou com 300.420 exemplos negativos e 1.046.216 exemplos positivos.

Como as bases apresentam grande desbalanceamento tanto em sua divisão original quanto binarizada, as técnicas de *oversampling* e *undersampling* foram utilizadas nos dados de treinamento, visando minimizar tal problema.

4.1.4 Etapas do Pré-processamento dos dados

Cada base de dados passa por um processo que agrupa quatro etapas. A primeira etapa consiste no pré-processamento dos dados, a partir da remoção dos stopwords e aplicação das técnicas de stemming e n-gram, apresentadas na seção 2.2. Para remoção das *stopwords* e aplicação do *stemming*, foi utilizado o conjunto de bibliotecas para Processamento de Linguagem Natural (PLN), *Natural Language Toolkit* (NLTK), por meio de dois métodos: NLTK stopwords[2] e NLTK RSLPStemmer[3]. Para a aplicação da técnica de n-gram, estabeleceu-se um valor de $n = 2$, assim palavras com conotação positiva, caso precedidas por palavras negativas, como o advérbio "não", podem passar por uma mudança de polaridade, seguindo o sentido da frase.

Como neste trabalho foram realizados dois tipos de experimentos em cada base de dados, o primeiro considerando múltiplas classes e o segundo considerando dados binários, a segunda etapa consiste no processo de separação das classes de acordo com o experimento a ser realizado. A Figura 7 apresenta a sequência das duas etapas.

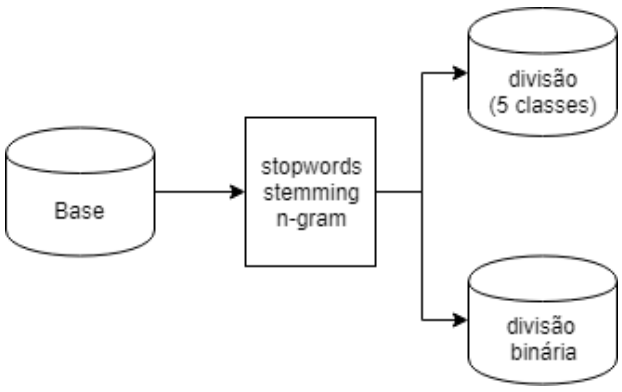

Figura 7 – Pré-processamento e divisão das bases (múltiplas classes e binária)

Para os experimentos com múltiplas classes, as bases de dados do Yelp Challenge Dataset não sofreram alterações, tendo em vista que possuem 5 classes, agrupadas neste

[2] https://pythonspot.com/nltk-stop-words/
[3] https://www.nltk.org/api/nltk.stem.html

trabalho como: *muito negativa*, *negativa*, *neutra*, *positiva* e *muito positiva*. Por outro lado, as bases de dados IMDb e SSTb possuem de 1-10 e 0-10 classes, respectivamente. Para respeitar os experimentos com 5 *labels* estabelecidos nesta pesquisa, foi considerado o agrupamento apresentado na seção 4.1 (pág. 37).

No caso dos experimentos binários, os dados referentes às classes *neutra* foram removidos dos testes e estabeleceu-se o seguinte padrão para as bases do Yelp: $label < 3 = negativa$ e $label > 3 = postiva$. De forma similar, a binarização das bases IMDb e SSTb ocorreu considerando: $label < 5 = negativa$ e $label > 5 = postiva$.

A terceira etapa aborda a divisão das bases em dois conjuntos de dados, o primeiro é o conjunto de treinamento utilizado para treinar e validar os modelos, e o segundo é o conjunto de teste, utilizado na avaliação de desempenho dos algoritmos. O percentual utilizado para a divisão do conjunto de dados foi estabelecido em: 80% da base para treinamento, 10% para validação e os 10% restantes para teste. Ainda nesta etapa, foram aplicadas as técnicas de *oversampling* e *undersampling*, a fim de minimizar o problema de desbalanceamento dos dados (ver seção 4.1, pág. 37).

Finalizando o processo de tratamento dos textos, na quarta etapa tem-se o processo de vetorização dos dados, para que os modelos computacionais possam interpretá-los. Foram utilizadas as técnicas TF-IDF, para os modelos de AM tradicional, e *Keras Tokenizer*, para os modelos de aprendizado profundo, detalhados na seção 2.2.3.

4.2 CONFIGURAÇÕES DOS MODELOS

Após definição e tratamentos dos dados, faz-se necessário o ajuste dos modelos que serão treinados. Para realização dos experimentos com os modelos de aprendizado de máquina tradicional, foi utilizada a biblioteca *scikit-learn*[4], que proporciona a implementação de diversos algoritmos de aprendizado de máquina por meio da linguagem de programação Python.

Para construção do algoritmo Naive Bayes utilizou-se o método do *scikit-learn*, MultinomialNB. O classificador Naive Bayes multinomial é adequado para classificação com características discretas (por exemplo, contagens de palavras para classificação de texto). A distribuição multinomial requer contagens de recursos inteiros ou fracionais, como o TF-IDF. Os parâmetros utilizados nos experimentos com NB foram:

- alpha=0 - Parâmetro de suavização, permite "suavizar"os dados considerando combinações de classe, é útil ao tentar desenvolver um modelo de classificação usando um conjunto de treinamento pequeno que pode não constituir uma amostra suficientemente representativa da população (1 para suavizar);
- fit_prior=True - aprender as probabilidades anteriores da classe ou não. Se falso, um uniforme anterior será usado;

[4] https://scikit-learn.org/

- class_prior=None - probabilidades anteriores das classes. Se especificado, os antecedentes não são ajustados de acordo com os dados.

Para a construção do modelo SVM, foi utilizado o método do *scikit-learn*, SGDClassifier. Este estimador implementa modelos regularizados com aprendizagem por meio do *Stochastic Gradient Descent* (SGD). O gradiente da perda é estimado em cada amostra e o modelo é atualizado ao longo do caminho com um cronograma de força decrescente (também conhecido como taxa de aprendizado). Os parâmetros utilizados nos experimentos com o SVM seguiram o padrão aprensendado pelo SGDClassifier:

- alpha=0.0001 - Constante que multiplica o termo de regularização. Também utilizada para calcular a taxa de aprendizagem quando definida como 'ótima';
- learning_rate='optimal' - taxa de aprendizado, 1.0 ($alpha*(t_t0)$) onde $t0$ é escolhido por uma heurística proposta por Leon Bottou.

Para a construção dos modelos de aprendizado profundo foi utilizada a Interface de Programação de Aplicações, do inglês *Application Programming Interface* (API), Keras. O Keras é uma API de redes neurais de alto nível, escrita em Python e capaz de rodar em cima de diversas bibliotecas para aprendizado de máquina. TensorFlow foi a biblioteca de aprendizado de máquina utilizada nestes experimentos, pela sua arquitetura flexível que permite a computação dos dados em uma ou mais CPUs ou GPUs em um desktop, servidor ou dispositivo móvel sem reescrever o código. O TensorFlow foi desenvolvido originalmente por pesquisadores e engenheiros que trabalham na equipe do Google Brain na organização *Google's Machine Intelligence Research*, com o objetivo de realizar aprendizado de máquina e pesquisa em redes neurais profundas (GOOGLE, 2018). A tabela 5 detalha os parâmetros utilizados nos experimentos com a *Long short-term Memory* (LSTM).

Em que:

- input_dim: representa a dimensão das *features*, neste caso o tamanho do vocabulário que deve ser lido pela rede;
- output_dim: é o tamanho do espaço vetorial no qual as palavras serão incorporadas. Ele define o tamanho dos vetores de saída dessa camada para cada palavra;
- input_lengh: comprimento das sequências de entrada;
- hidden_nodes: é o número de neurônios da LSTM. Um número maior, pode deixar a rede mais poderosa. No entanto, o número de parâmetros a aprender também aumenta, isso significa que também aumenta o tempo de treinamento da rede;

Tabela 5 – Parâmetros da LSTM

Parâmetro	**Valor**
input_dim	10000
output_dim	32
input_lenght	100
hidden_nodes	100
units	1 / 5
activation	sigmoid / softmax
loss	binary_crossentropy / categorical_crossentropy
optimizer	adam / sgd
epochs	5 / 3
batch_size	64

- units: dimensionalidade do espaço de saída. Nos experimentos binários utiliza-se um 1, enquanto nos experimentos com múltiplas classes, utiliza-se a quantidade de *labels*;
- activation: funções de ativação que introduz um componente não linear na rede, fazendo com que elas possam aprender mais do que relações lineares entre as variáveis dependentes e independentes;
- loss: função objetivo que visa, primordialmente, minimizar o erro do modelo. A *binary_crossentropy* foi utilizada nos experimentos binários, enquanto a *categorical_crossentropy* foi utilizada nos experimentos com múltiplas classes;
- optimizer: o otimizador é um dos argumentos necessários para compilar um modelo Keras. O *adam* foi utilizado para os experimentos binários, enquanto o *sgd* foi utilizado para os experimentos com múltiplas classes.
- epochs: número de épocas para treinar o modelo. Uma época é uma iteração sobre todos os dados x (*features*) e y (*labels*) fornecidos. Os classificadores únicos foram treinados com 5 épocas, enquanto os comitês foram treinados com 3 épocas;
- batch_size: número de amostras por atualização do gradiente.

Os parâmetros utilizados nos experimentos com a *Convolutional Neural Network* (CNN) estão detalhados na tabela 6, em que:

- filters: é a dimensionalidade do espaço de saída, ou seja, o número de filtros de saída na convolução;
- kernel_size: Um inteiro ou tupla/lista de um único inteiro, especificando o comprimento da janela de convolução 1D;

- padding: pode ser *causal*, que resulta em convoluções causais (dilatadas), onde a saída t não depende da entrada $[t + 1 :]$; *valid*, significando "sem preenchimento"ou; *same*, que resulta em preenchimento da entrada de forma que a saída tenha o mesmo comprimento que a entrada original;

- pool_size: tamanho das janelas do *max pooling* (camada de agrupamento de neurônios que combinam as saídas usando o valor máximo de cada *cluster* de neurônios na camada anterior.

.

Tabela 6 – Parâmetros da CNN

Parâmetro	**Valor**
input_dim	10000
output_dim	32
input_lenght	100
filters	32
kernel_size	3 / 5
padding	same
pool_size	2
activation	sigmoid / softmax
hidden_nodes	250
units	1 / 5
loss	binary_crossentropy / categorical_crossentropy
optimizer	adam / sgd
epochs	5 / 3
batch_size	64

Os demais parâmetros da CNN correspondem as mesmas configurações apresentadas na LSTM. Os parâmetros apresentados nas tabelas 5 e 6 também foram selecionados para a construção dos comitês, tendo como principal diferencial a quantidade de épocas das redes profundas. Enquanto os classificadores únicos foram treinados com 5 épocas, as combinações foram treinadas com 3 épocas. Foram realizadas 4 combinações de classificadores, sendo elas:

1. comitê 1: SVM-NB;

2. comitê 2: LSTM-CNN;

3. comitê 3: LSTM-CNN-SVM;

4. comitê 4: LSTM-CNN-NB.

O comitê 1 combina apenas classificadores de aprendizado de máquina tradicional; o comitê 2 combina apenas classificadores de aprendizado profundo e; os comitês 3 e 4 combinam algoritmos de aprendizado de máquina tradicional e aprendizado profundo. A partir dos experimentos realizados com os classificadores únicos e as combinações aqui apresentadas, será possível responder as questões de pesquisas levantadas no início desta dissertação.

4.3 CONSIDERAÇÕES DO CAPÍTULO

Neste capítulo foram descritas as bases de dados utilizadas durante os experimentos, bem como é detalhado o processo de redistribuição de classes adotado para a realização de testes com múltiplas classes e com dados binários. As técnicas e etapas para tratamento dos dados também foram descritas. Por fim, estão detalhadas as configurações/parâmetros utilizados em cada classificador.

5 EXPERIMENTOS E RESULTADOS

Para responder as questões de pesquisa, este capítulo apresenta os resultados obtidos por oito classificadores, sendo quatro classificadores únicos, um comitê apenas com algortimos de aprendizado de máquina tradicional, um comitê apenas com algoritmos de aprendizado profundo e dois comitês que combinam algoritmos de AM tradicional e aprendizado profundo.

Os comitês de classificadores que combinam algoritmos de aprendizado de máquina tradicional e aprendizado profundo tiveram seu número de épocas minimizados, enquanto os classificadores únicos contam com 5 épocas em seus experimentos, as combinações são realizadas com 3 épocas. Através dessa medida, espera-se diminuir o tempo de execução dos comitês.

O classificador que alcançar maior acurácia e o que atingir o menor tempo de execução em relação a cada base de dados, terão suas performances comparadas aos demais classificadores por meio de testes estatísticos. Assim, foi possível analisar o modelo mais adequado à cada situação, bem como seus pontos positivos e negativos. O teste de normalidade de *Shapiro-Wilk* foi utilizado neste trabalho para analisar estatisticamente a natureza da distribuição dos dados, com um nível de confiança de 95%, ou seja, α=0.05. Para analisar se há diferença entre as amostras, são utilizados os testes de *t-Student* e de *Wilcoxon*, a depender da distribuição dos dados (ver seção 2.7.2, pág. 29).

Os testes realizados através do 10-*fold crossvalidation*, foram executados em diferentes máquinas. Para se obter um resultado aproximado do tempo de execução de cada classificador, todos os testes foram realizados com uma execução (1-*fold*) em um notebook com processador Intel Core i7-7500U com 16GB de memória RAM e 2.70GHz.

5.1 ANÁLISES DO IMDB REVIEW

Esta seção está dividida em duas subseções, em que a primeira apresenta os resultados dos classificadores e análises estatísticas referentes aos experimentos com múltiplas classes do IMDb Review e; a segunda subseção apresenta os resultados e análises estatísticas referentes aos experimentos binários que foram realizados.

5.1.1 Experimentos com Múltiplas Classes - IMDb Review

O IMDb Review possui, originalmente, um total de 10 classes, este trabalho considera experimentos divididos em 5 polaridades de sentimentos ao distribuir as 10 classes iniciais em: muito negativa, negativa, neutra, positiva e muito positiva. Foram realizados um total de 8 experimentos com diferentes modelos, a fim de analisar os melhores classificadores em termos de acurácia e tempo de execução. A Tabela 7 detalha a acurácia obtida por cada

classificador em relação a base de dados. Além disso, apresenta informações referentes ao desvio padrão (sd) obtido por cada modelo, bem como seu tempo de execução em segundos (s).

Tabela 7 – Performance média dos classificadores para o IMDb *Review multclass*

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	40.24	0.0004	**4**
SVM	38.82	0.03	42
LSTM	25.44	0.001	4226
CNN	**43.99**	0.006	196
SVM-NB	39.87	0.004	47
LSTM-CNN	35.55	0.02	2654
LSTM-CNN-SVM	39.83	0.005	2701
LSTM-CNN-NB	39.40	0.002	2660

O classificador CNN atingiu o maior valor em termos de acurácia para essa base, enquanto que o NB obteve o menor tempo de execução. Assim CNN e NB são os dois classificadores que terão suas performances comparadas as dos demais.

Antes de definir qual teste de hipótese entre amostras deve ser utilizado, é necessário conhecer a natureza da distribuição dos dados por meio de um teste de normalidade, principalmente se tratando de uma amostra pequena, com $n = 10$. Os resultados do teste de normalidade de *Shapiro-Wilk* são apresentados na Tabela 8.

Tabela 8 – Teste de normalidade dos classificadores para o IMDb *Review multclass*

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.4203	não	não é possível rejeitar H_0
SVM	32.844e-05	sim	rejeita H_0
LSTM	0.01339	sim	rejeita H_0
CNN	0.4624	não	não é possível rejeitar H_0
SVM-NB	0.01004	sim	rejeita H_0
LSTM-CNN	0.7846	não	não é possível rejeitar H_0
LSTM-CNN-SVM	0.8691	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.8009	não	não é possível rejeitar H_0

Dentre os classificadores únicos, não é possível rejeitar a hipótese nula, de que os dados provém de uma distribuição normal para as acurácias do NB e da rede de aprendizado profundo, CNN. No caso das acurácias referentes à LSTM e ao SVM a hipótese nula é rejeitada, portanto os dados não provém de uma distribuição normal. No caso dos comitês, apenas os dados do SVM-NB rejeitam H_0. Não é possível rejeitar a hipótese nula, de que os dados provém de uma distribuição normal para LSTM-CNN, LSTM-CNN-SVM e LSTM-CNN-NB.

Conhecendo a distruibuição dos dados, tem-se como próximo passo verificar se há diferença estatística entre os desempenhos dos classificadores. Como a rede CNN obteve maior acurácia, o seu desempenho foi comparado aos desempenhos dos demais modelos. Foram escolhidos dois testes que podem verificar a diferença entre as acurácias dos classificadores, o teste paramétrico *t-Student* para as análises com os dados que não rejeitaram H_0 e o teste de *Wilcoxon*, alternativa não paramétrica, para os dados que rejeitaram H_0 (ver 2.7.2, pág. 29). A Tabela 9 apresenta a comparação entre o desempenho da CNN e dos demais classificadores.

Tabela 9 – Comparação da performance entre a CNN e demais classificadores para o IMDb *Review multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
CNNxNB	*t-Student*	1.107e-13	sim	rejeita H_0
CNNxSVM	*Wilcoxon*	3.188e-07	sim	rejeita H_0
CNNxLSTM	*Wilcoxon*	2.2e-16	sim	rejeita H_0
CNNxSVM-NB	*Wilcoxon*	1.083e-05	sim	rejeita H_0
CNNxLSTM-CNN	*t-Student*	2.2e-16	sim	rejeita H_0
CNNxLSTM-CNN-SVM	*t-Student*	2.2e-16	sim	rejeita H_0
CNNxLSTM-CNN-NB	*t-Student*	2.2e-16	sim	rejeita H_0

Como é possível analisar por meio da Tabela 9, a rede CNN possui desempenho estatisticamente superior em relação a todos os modelos. Dentre os classificadores de aprendizado profundo, esta técnica tem também o menor tempo de execução, apesar de esse tempo de execução ainda ser superior ao dos métodos tradicionais. Para melhor analisar a diferença entre as acurácias dos classificadores, as Figuras 8 e 9 trazem por meio de *box-plots* a comparação da distribuição das acuráricas entre a CNN e demais classificadores.

Por meio dos *box-plots* se observa que as acurácias da CNN estão distruibuídas bem acima das acurácias dos demais classificadores, corroborando com as análises estatísticas que apontaram diferença entre as médias das amostras.

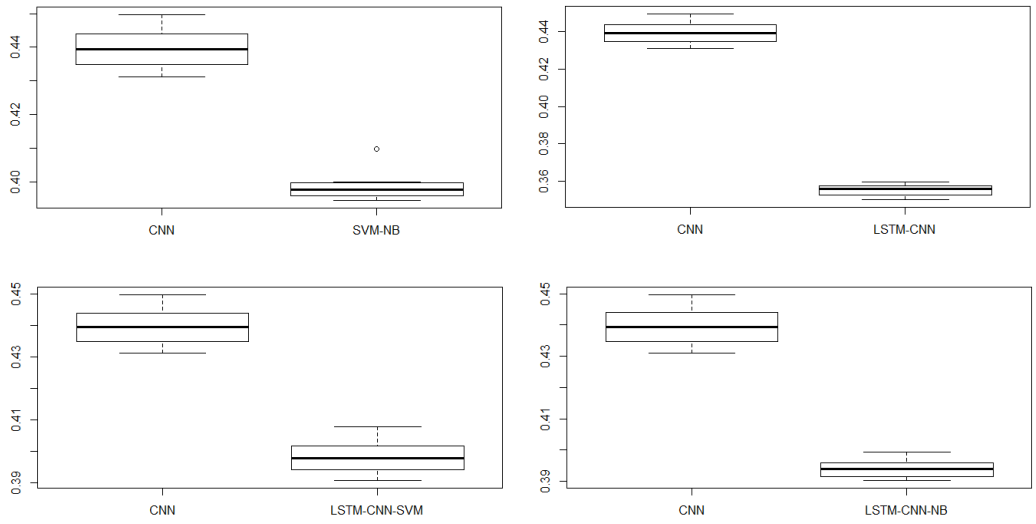

Figura 8 – Comparação da distribuição das acurácias entre CNN e os comitês para o IMDb *Review multclass*

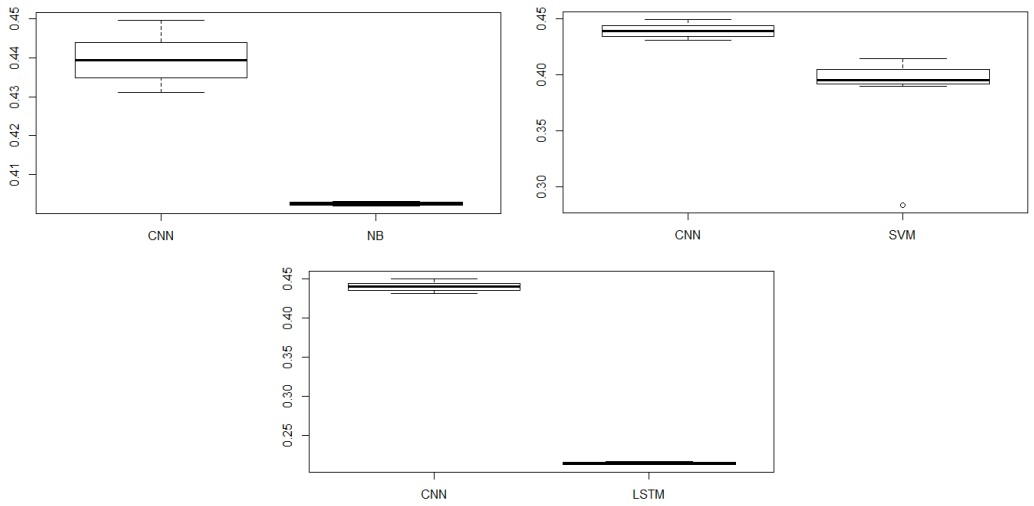

Figura 9 – Comparação da distribuição das acurácias entre CNN e demais classificadores únicos para o IMDb *Review multclass*

Como o Naive Bayes foi o classificador que apresentou o menor tempo de execução, seu desempenho também foi analisado em relação aos demais classificadores, como forma de averiguar se este modelo possui desempenho médio igual ou superior aos demais classificadores. A Tabela 10 apresenta a comparação entre os desempenhos do NB com os demais classificadores.

Como é possível analisar, o NB possui desempenho estatisticamente diferente em relação a todos os modelos, exceto em relação ao SVM, em que não foi possível rejeitar H_0. Assim, em termos de tempo de execução os classificadores tradicionais se mostraram boas

Tabela 10 – Comparação da performance entre NB e demais classificadores para o IMDb *Review multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*Wilcoxon*	0.1431	não	não é possível rejeitar H_0
NBxLSTM	*Wilcoxon*	0.0001817	sim	rejeita H_0
NBxCNN	*t-Student*	1.107e-13	sim	rejeita H_0
NBxSVM-NB	*Wilcoxon*	0.001505	sim	rejeita H_0
NBxLSTM-CNN	*t-Student*	1.033e-12	sim	rejeita H_0
NBxLSTM-CNN-SVM	*t-Student*	0.03816	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	3.668e-06	sim	rejeita H_0

alternativas, sendo NB melhor opção em termos de tempo e igualmente bom ao SVM em termos de performance. No entanto, a rede de aprendizado profundo, CNN, aponta para um desempenho melhor, com acurácia estatisticamente superior. As figuras 10 e 11 apresentam os *box-plots* que comparam visualmente o desempenho do NB aos demais modelos.

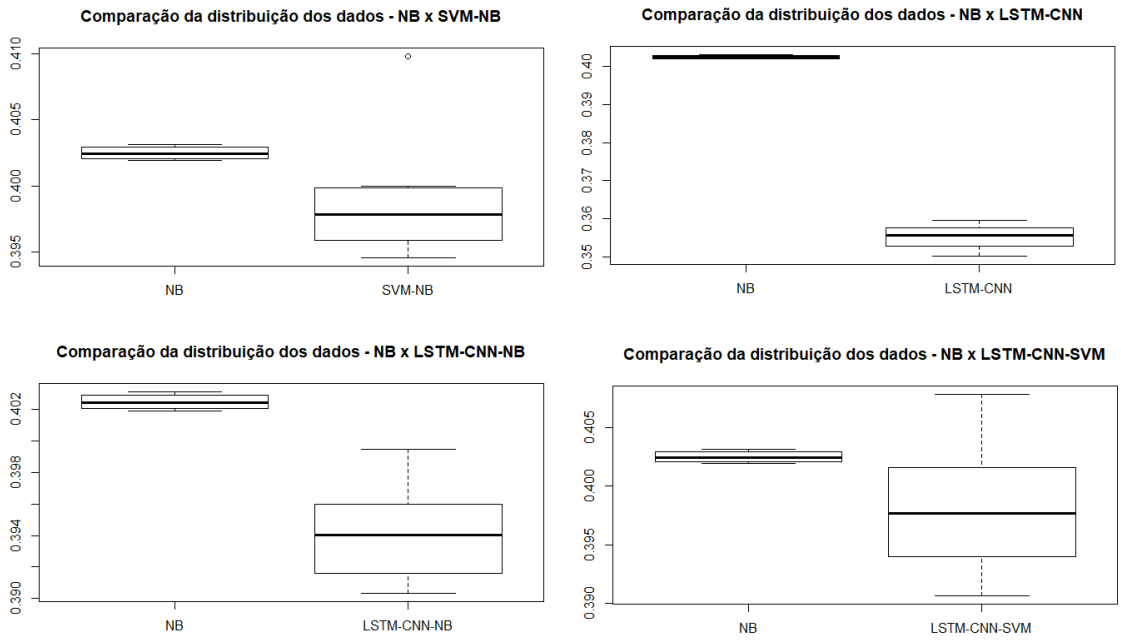

Figura 10 – Comparação da distribuição das acurácias entre NB e os comitês para o IMDb *Review multclass*

Através dos *box-plots* é possível observar tanto a variação dos dados de cada classificador, quanto comparar a intersecção ou disparidade entre as amostras de dois classificadores. O NB se mostra um bom classificador, considerando seu desempenho em relação algoritmo de AM tradicional SVM, atingindo também um desempenho superior ao *ensemble* SVM-NB, porém atingiu acurácia muito inferior em relação aos classificadores de

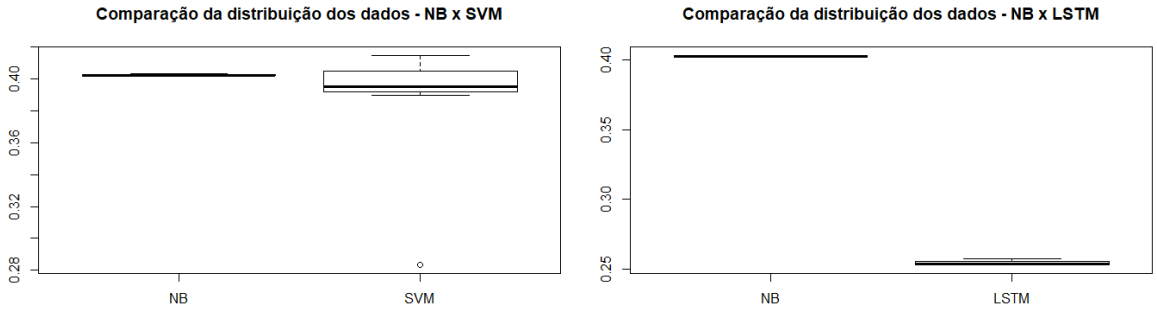

Figura 11 – Comparação da distribuição das acurácias entre NB e demais classificadores únicos para o IMDb *Review multclass*

aprendizado profundo, bem como em relção aos comitês que continham algoritmos de aprendizado profundo.

Por fim, foram analisados os desempenhos da CNN e do NB em relação a cada polaridade de sentimentos da base, de forma individual, por meio das métricas *precision*, *recall* e *f-score* (ver seção 2.6, pág. 28), como mostra a Tabela 11.

Tabela 11 – Desempenho da CNN e do NB de acordo com cada polaridade de sentimento do IMDb Review *multclass*

	CNN/NB					
	muito negativa	**negativa**	**neutra**	**positiva**	**muito positiva**	**total**
precision	0.36 / 0.39	0.40 / 0.00	0.40 / 0.38	0.48 / 0.49	0.55 / 0.40	0.44 / 0.35
recall	0.79 / 0.68	0.02 / 0.00	0.32 / 0.12	0.49 / 0.34	0.59 / 0.71	0.44 / 0.41
f-score	0.71 / 0.49	0.04 0.00	0.28 / 0.18	0.48 0.40	0.57 / 0.67	0.39 / 0.34

Por meio da Tabela 11 é possível perceber que a rede CNN tem acertividade superior ao NB em relação a todas as polaridades de sentimentos. Observou-se ainda que o NB pode oscilar muito em termos de desempenho para uma classe em específico, como para a classe *negativa*. A rede de aprendizado profundo, CNN, por sua vez, alcançou resultados significativamente superiores em relação a todas as polaridades individualmente, além de ser o algoritmo com menor tempo de execução dentre os que envolveram aprendizado profundo no experimento abordado, podendo ser considerada a melhor opção para esta base de dados. Trabalhos na literatura apontam para uma acurácia de 49.2% para a base do IMDb *Review* (LIU; LAPATA, 2018), em Angelidis e Lapata (2018) uma acurácia de 63.97% é apresentada, mas são consideradas apenas três polaridade de sentimentos. Tanto em Liu e Lapata (2018) quanto em Angelidis e Lapata (2018) não são apresentados valores de variância, desvio padrão ou testes estatísticos que permitam uma maior análise

dos resultados.

5.1.2 Experimentos Binários - IMDb Review

Para estes experimentos a base de dados do IMDb Review foi binarizada, restando apenas as polaridades de sentimentos positiva e negativa. A Tabela 12 detalha a acurácia, desvio padrão e tempo de execução obtidos por cada classificador em relação a base de dados binarizada, para que se verifique os melhores modelos.

Tabela 12 – Performance dos classificadores para o IMDb *Review* binarizado

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	89.84	0.003	**0.1**
SVM	93.40	0.006	0.7
LSTM	90.44	0.001	712
CNN	91.42	0.001	96
SVM-NB	89.46	0.002	1
LSTM-CNN	90.42	0.003	486
LSTM-CNN-SVM	**94.52**	0.003	487
LSTM-CNN-NB	94.44	0.003	487

O comitê LSTM-CNN-SVM atingiu o maior valor em termos de acurácia para essa base, enquanto que o NB obteve o menor tempo de execução. Por meio da Figura 12 é possível comparar a distribuição entre as acurácias dos dois classificadores.

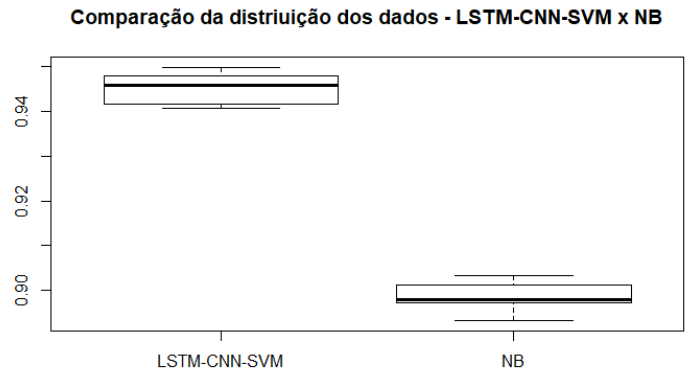

Figura 12 – Comparação da distribuição das acurácias entre LSTM-CNN-SVM x NB para o IMDb *Review* binarizado

A LSTM-CNN-SVM apresenta uma distribuição das acurácias em um quadro consideravalmente superior ao NB. Através do teste de *Shapiro-Wilk*, na Tabela 13, é possível analisar estatisticamente a distribuição dos dados.

Tabela 13 – Teste de normalidade dos classificadores para o IMDb *Review* binarizado

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.73123	não	não é possível rejeitar H_0
SVM	0.8428	não	não é possível rejeitar H_0
LSTM	0.013399	sim	rejeita H_0
CNN	0.1799	não	não é possível rejeitar H_0
SVM-NB	0.03976	sim	rejeita H_0
LSTM-CNN	0.7587	não	não é possível rejeitar H_0
LSTM-CNN-SVM	0.1533	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.5866	não	não é possível rejeitar H_0

Dentre os classificadores únicos, não é possível rejeitar a hipótese nula, de que os dados provém de uma distribuição normal, para as acurácias do NB, SVM e CNN. No caso das acurácias referentes à LSTM, a hipótese nula é rejeitada, portanto os dados não provém de uma distribuição normal. No caso dos comitês, apenas os dados do SVM-NB rejeitam H_0. Não é possível rejeitar a hipótese nula, de que os dados provém de uma distribuição normal, para LSTM-CNN, LSTM-CNN-SVM e LSTM-CNN-NB.

Como o comitê LSTM-CNN-SVM obteve maior acurácia, o seu desempenho foi comparado aos desempenhos dos demais classificadores. O teste paramétrico *t-Student* foi utilizado para as análises com os dados que não rejeitaram H_0 e o teste de *Wilcoxon*, alternativa não paramétrica, foi utilizado para os dados que rejeitaram H_0. A Tabela 14 apresenta a comparação entre o desempenho do comitê LSTM-CNN-SVM e dos demais classificadores.

Tabela 14 – LSTM-CNN-SVM vs. demais classificadores (IMDb *Review* binarizado)

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
LSTM-CNN-SVMxNB	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNN-SVMxSVM	*t-Student*	0.0002662	sim	rejeita H_0
LSTM-CNN-SVMxLSTM	*Wilcoxon*	0.0001776	sim	rejeita H_0
LSTM-CNN-SVMxCNN	*t-Student*	5.83e-13	sim	rejeita H_0
LSTM-CNN-SVM x SVM-NB	*Wilcoxon*	0.0001786	sim	rejeita H_0
LSTM-CNN-SVM x LSTM-CNN	*t-Student*	0.7827	não	não é possível rejeitar H_0
LSTM-CNN-SVM x LSTM-CNN-NB	*t-Student*	0.5905	não	não é possível rejeitar H_0

Como é possível analisar por meio da Tabela 14, a combinação de classificadores

LSTM-CNN-SVM possui desempenho estatisticamente superior em relação a todas os classificadores únicos. Com relação ao desempenho dos comitês, o LSTM-CNN-SVM possui desempenho superior apenas em relação ao SVM-NB, no entanto, não foi possível rejeitar H_0 no que se refere as demais combinações. Assim, os comitês que contemplam algoritmos de aprendizado profundo se mostraram como os modelos de melhor desempenho para este problema.

As figuras 13 e 14 trazem por meio de *box-plots* a comparação da distribuição das acurácias entre a LSTM-CNN-SVM e os demais modelos.

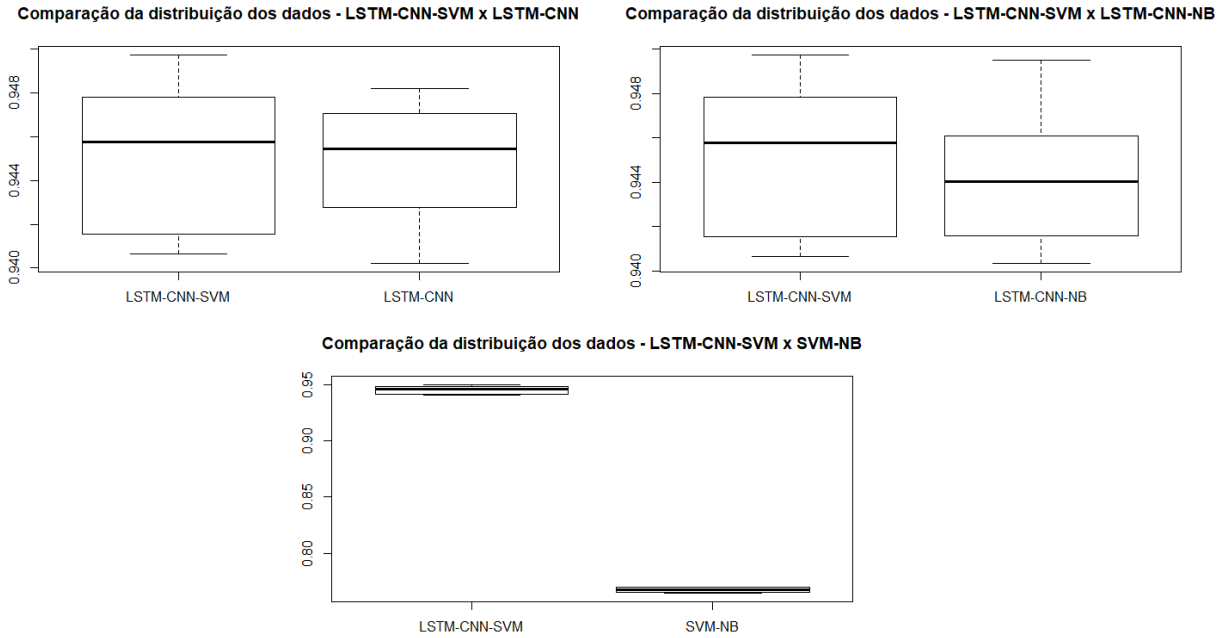

Figura 13 – Comparação da distribuição das acurácias entre LSTM-CNN-SVM e os demais comitês para o IMDb *Review* binarizado

Se torna ainda mais perceptível por meio da Figura 13 que há interseção entre os desempenhos das combinações de classificadores com algoritmos de aprendizado profundo. Assim como no IMDb Review *multclass*, o Naive Bayes também foi o classificador que apresentou o menor tempo de execução para a base binarizada. O desempenho do NB foi, então, analisado em relação aos demais classificadores, como forma de averiguar em quais situações o modelo com menor tempo de execução também apresenta bons desempenhos em acurácia.

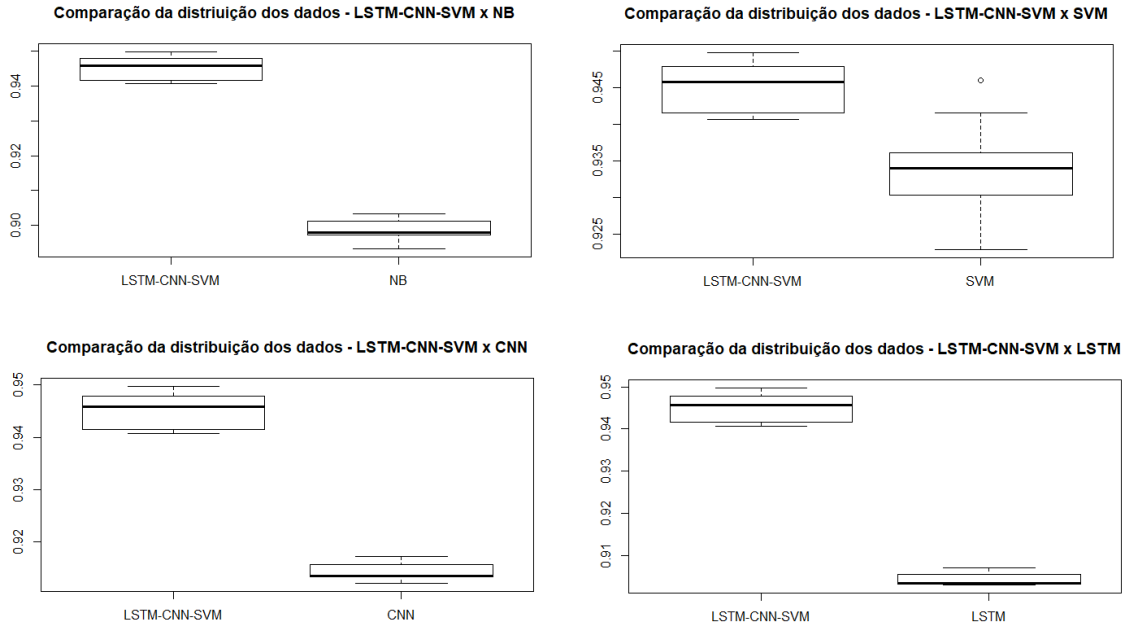

Figura 14 – Comparação da distribuição das acurácias entre LSTM-CNN-SVM e os classificadores únicos para o IMDb *Review* binarizado

A Tabela 15 apresenta a comparação entre os desempenhos do NB com os demais classificadores.

Tabela 15 – Comparação da performance entre NB e demais classificadores para o IMDb *Review* binarizado

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*t-Student*	6.211e-12	sim	rejeita H_0
NBxLSTM	*Wilcoxon*	0.0005801	sim	rejeita H_0
NBxCNN	*t-Student*	5.054e-11	sim	rejeita H_0
NBxSVM-NB	*Wilcoxon*	1.083e-055	sim	rejeita H_0
NBxLSTM-CNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-SVM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	2.2e-16	sim	rejeita H_0

Como é possível analisar, o NB possui desempenho estatisticamente diferente em relação a todos os classificadores e, apesar de obter o menor tempo de execução, também figura entre os classificadores com menor acurácia, não se mostrando um classificador tão interessante neste problema.

As figuras 15 e 16 apresentam *box-plots* que comparam visualmente o desempenho do NB aos demais modelos. É possível observar que o NB figura na parte mais baixa dos gráficos em, praticamente, todas as comparações, excetuando-se a comparação com o SVM-NB.

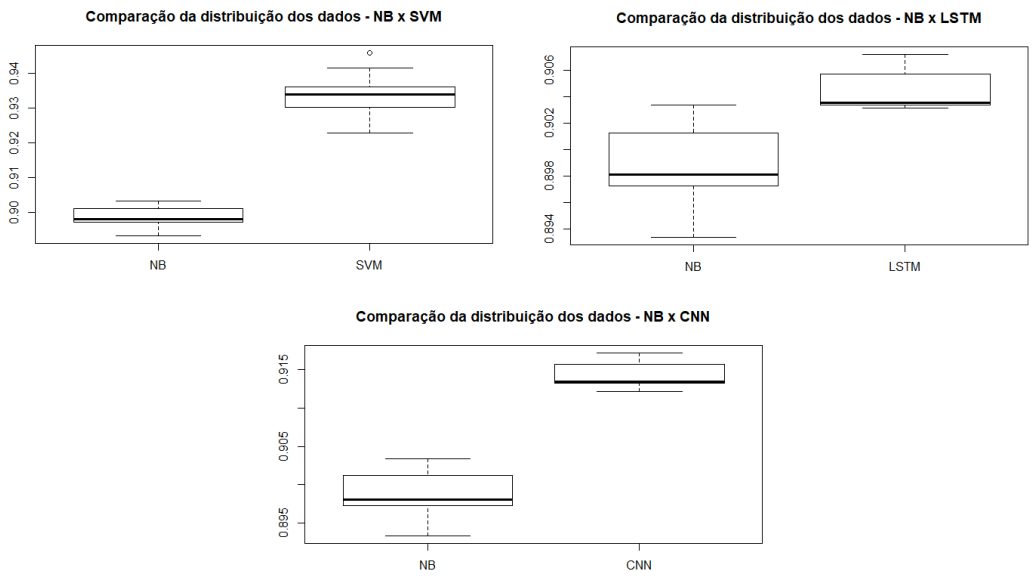

Figura 15 – Comparação da distribuição das acurácias entre NB e os comitês para o IMDb *Review* binarizado

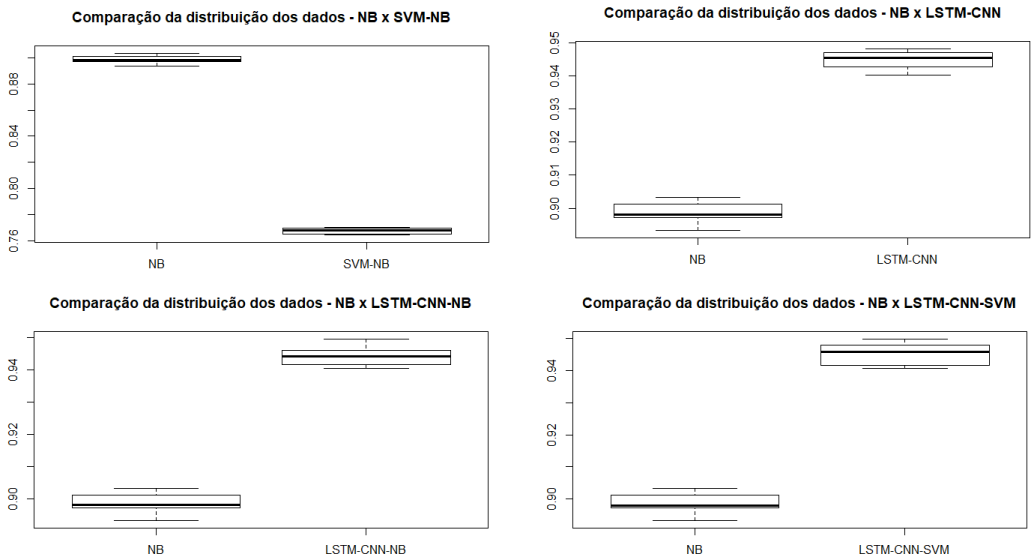

Figura 16 – Comparação da distribuição das acurácias entre NB os classificadores únicos para o IMDb *Review* binarizado

Foram analisados, por fim, os desempenhos da LSTM-CNN-SVM e do NB em relação a sua acertividade para cada polaridade de sentimentos, por meio das métricas *precision*, *recall* e *f-score*, como mostra a Tabela 16.

Tabela 16 – Desempenho da LSTM-CNN-SVM e do NB de acordo com cada polaridade de sentimento para o IMDb Review binarizado

	LSTM-CNN-SVM/NB	
	negativa	**positiva**
precision	0.93 / 0.88	0.97 / 0.94
recall	0.97 / 0.95	0.95 / 0.87
f-score	0.95 / 0.90	0.95 / 0.90

Através da Tabela 16 pode-se observar que a combinação de classificadores (LSTM-CNN-SVM) não obteve em nenhuma métrica resultado inferior à 93%, já o NB obteve como menor métrica um *recall* de 87%. Assim, para a base do IMDb Review binarizada, os comitês de classificadores que continham modelos de aprendizado profundo alcançaram os melhores desempenhos, com acurácia estatisticamente maior em relação a todos os outros classificadores. No que se refere ao tempo de execução, essas combinações alcançam tempo maior em relação aos algoritmos tradicionais, porém em relação as redes profundas, devido ao menor número de épocas de treinamento, as combinações atingem um menor tempo de execução.

5.2 *STANFORD SENTIMENT TREEBANK*

Esta seção está dividida em duas subseções, em que a primeira apresenta os resultados e análises estatísticas referentes aos experimentos com múltiplas classes para o *Stanford Sentiment Treebank* (SSTb) e a segunda, apresenta os resultados e análises estatísticas referentes aos experimentos binários que foram realizados.

5.2.1 Experimentos com Múltiplas Classes - SSTb

O SSTb está dividido, originalmente, em 11 classes (ver 4.1.2, pág. 39), redistruibuídas neste trabalho em 5, a saber: muito negativa, negativa, neutra, positiva e muito positiva. A Tabela 17 apresenta a acurácia, desvio padrão (sd) e tempo de execução de cada um dos 8 classificadores em relação ao SSTb, a fim de analisar os melhores classificadores em termos de desempenho e custo computacional.

Tabela 17 – Performance dos classificadores para o SSTb *multclass*

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	41.48	0.001	**2**
SVM	40.67	0.002	2,5
LSTM	43.13	0.005	2740
CNN	41.88	0.003	234
SVM-NB	39.62	0.002	3
LSTM-CNN	40.86	0.003	1785
LSTM-CNN-SVM	**49.28**	0.001	1788
LSTM-CNN-NB	40.19	0.0001	1787

O classificador LSTM-CNN-SVM atingiu o maior valor em termos de acurácia para o SSTb *multclass*, enquanto que o NB obteve o menor tempo de execução. Assim, os dois classificadores que terão sua performance comparada frente aos demais são LSTM-CNN-SVM e NB. A Tabela 18, apresenta os resultados do teste de *Shapiro-Wilk* em relação a cada classificador.

Tabela 18 – Teste de normalidade dos classificadores para o SSTb *multclass*

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.7703	não	não é possível rejeitar H_0
SVM	0.3973	não	não é possível rejeitar H_0
LSTM	0.02857	sim	rejeita H_0
CNN	0.009923	sim	rejeita H_0
SVM-NB	0.2518	não	não é possível rejeitar H_0
LSTM-CNN	0.9552	não	não é possível rejeitar H_0
LSTM-CNN-SVM	0.02288	sim	rejeita H_0
LSTM-CNN-NB	0.08787	não	não é possível rejeitar H_0

Os classificadores LSTM, CNN e LSTM-CNN-SVM rejeitaram H_0, o que significa que não obedecem a uma distribuição normal. No caso dos demais classificadores, não foi possível rejeitar a hipótese nula. Considerando que o combinação LSTM-CNN-SVM obteve melhor desempenho, sua performance será comparada estatisticamente em relação as performances dos demais classificadores por meio do teste de *Wilcoxon*, tendo em vista que o comitê a ser comparado não obedece a uma distribuição normal. Os resultados do teste estão detalhados na Tabela 19.

Tabela 19 – Comparação da performance entre o LSTM-CNN-SVM e demais classificadores para o SSTb *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
LSTM-CNN-SVMxNB	*Wilcoxon*	0.0001736	sim	rejeita H_0
LSTM-CNN-SVMxSVM	*Wilcoxon*	0.0001736	sim	rejeita H_0
LSTM-CNN-SVMxLSTM	*Wilcoxon*	0.0001736	sim	rejeita H_0
LSTM-CNN-SVMxCNN	*Wilcoxon*	0.0001602	sim	rejeita H_0
LSTM-CNN-SVM x SVM-NB	*Wilcoxon*	0.0001736	sim	rejeita H_0
LSTM-CNN-SVM x LSTM-CNN	*Wilcoxon*	0.0001736	sim	rejeita H_0
LSTM-CNN-SVM x LSTM-CNN-NB	*Wilcoxon*	0.000164	sim	rejeita H_0

Como mostra a Tabela 19, o desempenho apresentado pela combinação LSTM-CNN-SVM é superior ao desempenho de todos os outros modelos. As figuras 17 e 18 trazem uma comparação da distribuição das amostras que permite uma melhor visualização da diferença entre as acurácias de cada classificador em relação ao comitê com melhor resultado.

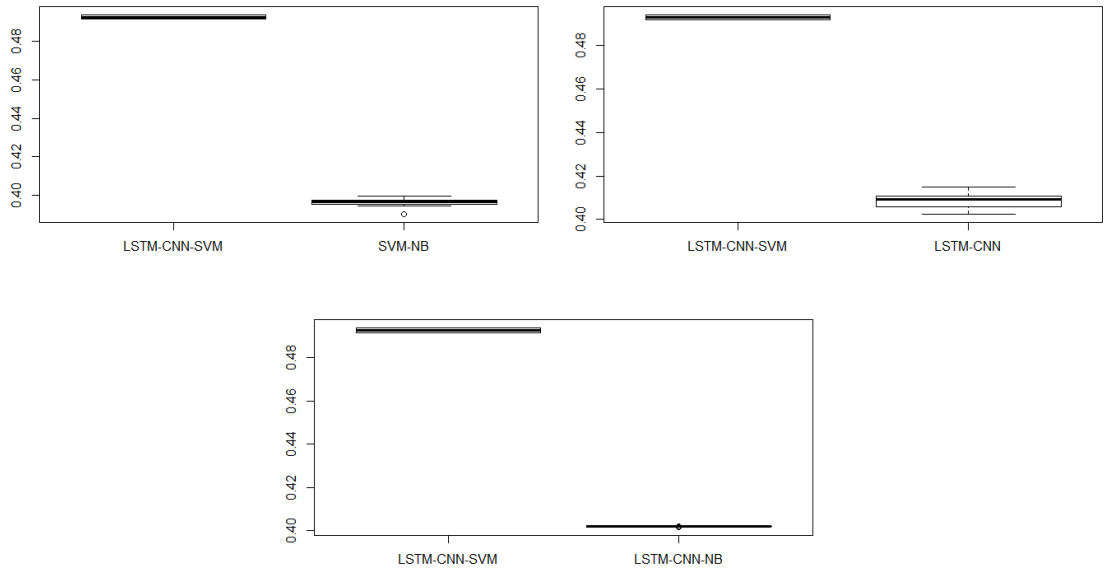

Figura 17 – Comparação da distribuição das acurácias entre LSTM-CNN-SVM e os comitês para o SSTb *multclass*

Por meio dos *box-plots* é possível visualizar que a combinação de classificadores LSTM-

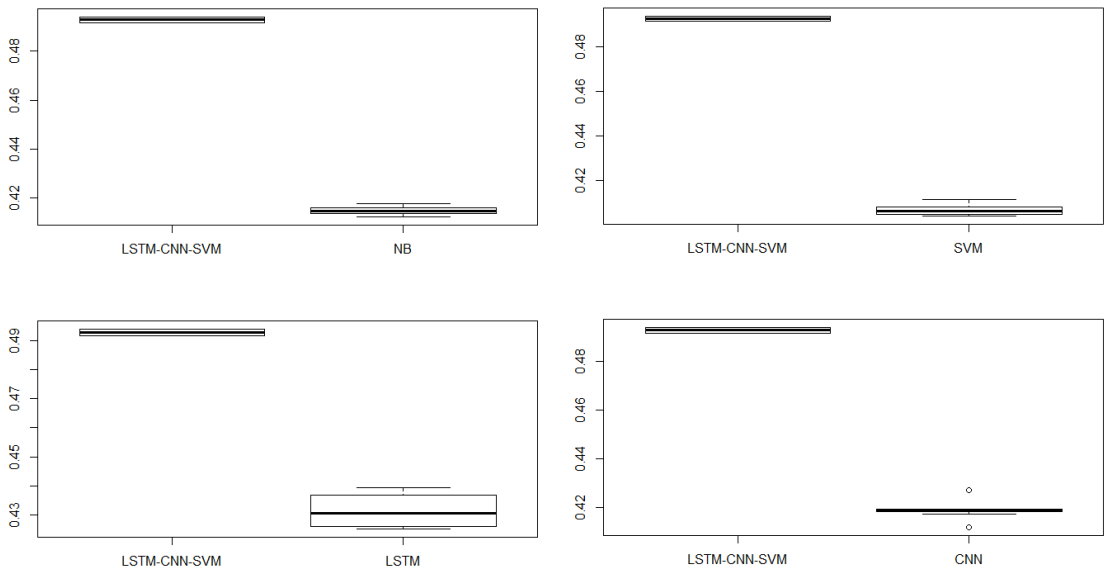

Figura 18 – Comparação da distribuição das acurácias entre LSTM-CNN-SVM e classificadores únicos para o SSTb *multclass*

CNN-SVM possui desempenho muito acima dos demais classificadores. Considerando que o NB foi o classificador com menor tempo de execução, sua acurácia foi comparada a acurácia dos demais classificadores, a fim de analisar o quão o NB pode ser considerado uma opção interessante. A Tabela 20 detalha os resultados desta comparação.

Tabela 20 – Comparação da performance entre NB e demais classificadores para o SSTb *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*t-Student*	1.558e-07	sim	rejeita H_0
NBxLSTM	*Wilcoxon*	1.083e-05	sim	rejeita H_0
NBxCNN	*Wilcoxon*	0.0043758	sim	rejeita H_0
NBxSVM-NB	*t-Student*	3.965e-12	sim	rejeita H_0
NBxLSTM-CNN	*t-Student*	0.00025862	sim	rejeita H_0
NBxLSTM-CNN-SVM	*Wilcoxon*	0.0001736	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	1.822e-09	sim	rejeita H_0

O NB foi considerado estatisticamente diferente de todos os modelos. Através dos *boxplots* das figuras 19 e 20 é possível analisar em relação a quais modelos o NB pode ser considerado superior.

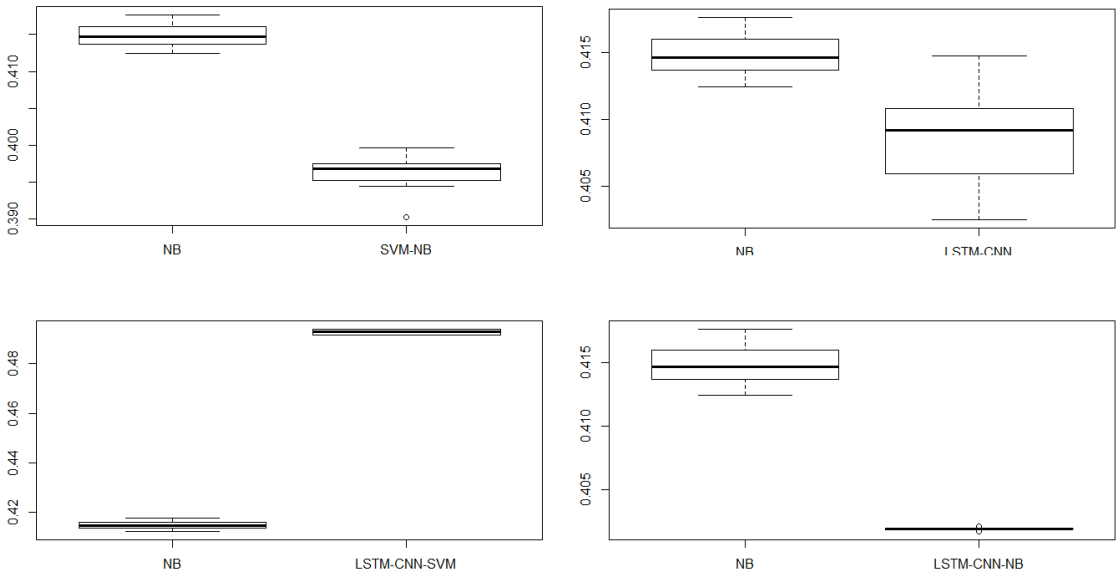

Figura 19 – Comparação da distribuição das acurácias entre NB e os comitês para o SSTb *multclass*

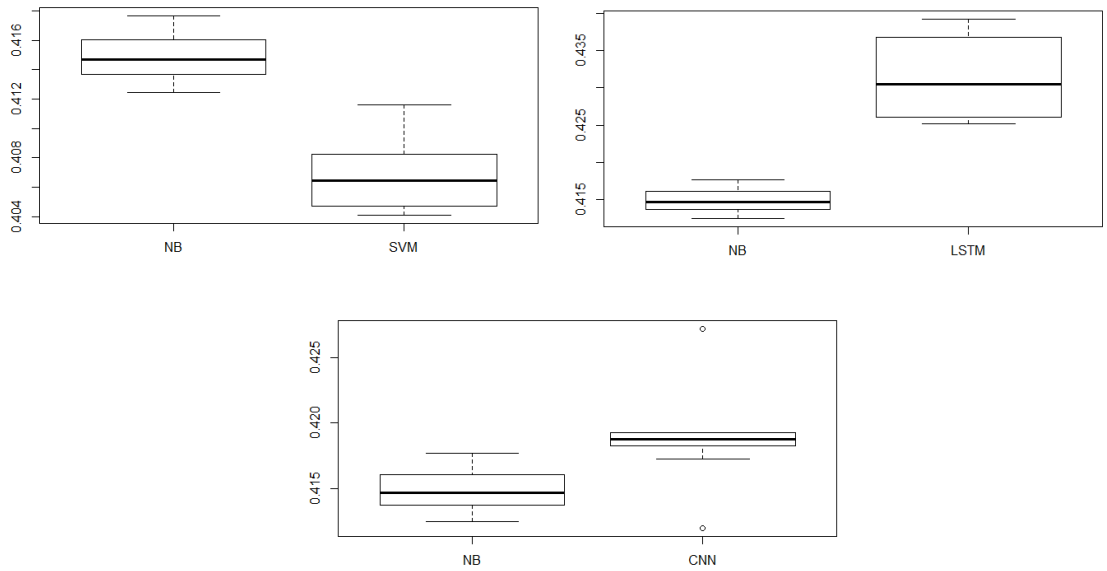

Figura 20 – Comparação da distribuição das acurácias entre NB e os classificadores únicos para o SSTb *multclass*

Por meio das figuras 19 e 20 percebe-se que o NB possui desempenho superior em relação as combinações SVM-NB, LSTM-CNN e LSTM-CNN-NB. No caso dos classificadores únicos, o NB apresenta desempenho superior apenas em relação ao SVM. Sendo o NB inferior as redes LSTM, CNN e ao comitê LSTM-CNN-SVM, que apresenta o melhor

desempenho para a base do SSTb *multclass*.

Os desempenhos da combinação LSTM-CNN-SVM e do NB também foram comparados considerando a acertividade do classificador em cada polaridade de sentimentos individualmente, como mostra a Tabela 21.

Tabela 21 – Desempenho do comitê LSTM-CNN-SVM e do NB de acordo com cada polaridade de sentimento do SSTb *multclass*

	LSTM-CNN-SVM/NB					
	muito negativa	**negativa**	**neutra**	**positiva**	**muito positiva**	**total**
precision	0.51 / 0.54	0.33 / 0.00	0.34 / 0.41	0.53 / 0.51	0.58 / 0.00	0.47 / 0.38
recall	0.67 / 0.06	0.23 / 0.00	0.28 / 0.97	0.53 / 0.15	0.66 / 0.00	0.50 / 0.42
f-score	0.58 / 0.11	0.26 0.00	0.31 / 0.57	0.53 0.24	0.62 / 0.00	0.48 / 0.29

Como é possível observar por meio da Tabela 21, o NB oscila muito em termos de acertividade das polaridades de sentimentos, enquanto possui um *recall* de 97% para a classe *neutra*, possui também um *recall* de 0.00 para as classes *negativa* e *muito positiva*. Esse comportamento é apresentado nas outras métricas também, chegando a uma precisão e *f-score* de 0.00 em alguns momentos. Portanto, mesmo sendo um classificador rápido, não é aconselhável seu uso nesse problema. Sendo o comitê a melhor solução, pois consegue manter um nível de acertividade mais variado em relação a todas as polaridades de sentimentos. Trabalhos recentes da literatura apontam para uma acurácia de 47.0% para a base SSTb (HAN; BAI; LIU, 2018), no entanto não são utilizados testes estatísticos que permitam uma análise mais detalhada dos resultados.

5.2.2 Experimentos Binários - SSTb

Para os experimentos na base do SSTb de forma binária, os dados foram separados nas polaridades de sentimentos positiva e negativa, como detalha a seção 4.1.2. Na Tabela 22 estão acurácia, desvio padrão e tempo de execução de cada classificador, os melhores classificadores no que refere ao desempenho e tempo de execução serão analisados estatisticamente em relação aos demais.

Tabela 22 – Performance dos classificadores para o SSTb binarizado

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	89.04	0.01	**0.25**
SVM	86.15	0.01	1.5
LSTM	90.04	0.003	1277
CNN	90.43	0.0008	171
SVM-NB	89.46	0.002	3
LSTM-CNN	90.42	0.003	870
LSTM-CNN-SVM	90.23	0.0005	872
LSTM-CNN-NB	**90.62**	0.002	871

Em termos absolutos, o classificador LSTM-CNN-NB atingiu a maior acurácia para essa base de dados e, mais uma vez, o NB aparece como o classificador mais rápido. Assim, o comitê LSTM-CNN-NB e o classificador único NB terão suas acurácias comparadas as dos demais, para que se possa discutir a relação entre desempenho do classificador e custo computacional. A Tabela 23, detalha os resultados do teste de normalidade

Tabela 23 – Teste de normalidade dos classificadores para o SSTb binarizado

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.6567	não	não é possível rejeitar H_0
SVM	0.6725	não	não é possível rejeitar H_0
LSTM	0.2119	não	não é possível rejeitar H_0
CNN	0.1572	não	não é possível rejeitar H_0
SVM-NB	0.8766	não	não é possível rejeitar H_0
LSTM-CNN	0.01927	sim	rejeita H_0
LSTM-CNN-SVM	0.26973	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.2518	não	não é possível rejeitar H_0

De acordo com o teste de *Shapiro-Wilk*, apenas a combinação LSTM-CNN não rejeita H_0, quanto aos demais classificadores, a hipótese nula, de que os dados obedecem a uma distribuição normal, é rejeitada. Conhecendo a distribuição dos dados, estabeleu-se que o teste a ser utilizado para verificar a diferença entre as amostras será o teste de *Wilcoxon*. Os resultados acerca da diferença entre o LSTM-CNN-NB (classificador com maior acurácia) em relação aos demais classificadores estão na Tabela 24.

Tabela 24 – LSTM-CNN-NB vs. demais classificadores (SSTb binarizado)

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
LSTM-CNN-NBxNB	*Wilcoxon*	0.003886	sim	rejeita H_0
LSTM-CNN-NBxSVM	*Wilcoxon*	1.083e-05	sim	rejeita H_0
LSTM-CNN-NBxLSTM	*Wilcoxon*	0.0003248	sim	rejeita H_0
LSTM-CNN-NBxCNN	*Wilcoxon*	0.01398	sim	rejeita H_0
LSTM-CNN-NB x SVM-NB	*Wilcoxon*	1.083e-05	sim	rejeita H_0
LSTM-CNN-NB x LSTM-CNN	*Wilcoxon*	0.2404	não	não é possível rejeitar H_0
LSTM-CNN-NB x LSTM-CNN-SVM	*Wilcoxon*	0.002796	sim	rejeita H_0

Não foi possível rejeitar a hipótese nula, de que as amostras são diferentes, para as acurácias dos comitês LSTM-CNN-NB e LSTM-CNN. Quanto aos demais classificadores, o teste de *Wilcoxon* aponta diferença estatística. As figuras 21 e 22 comparam, por meio de *box-plots* a distribuição dos dados.

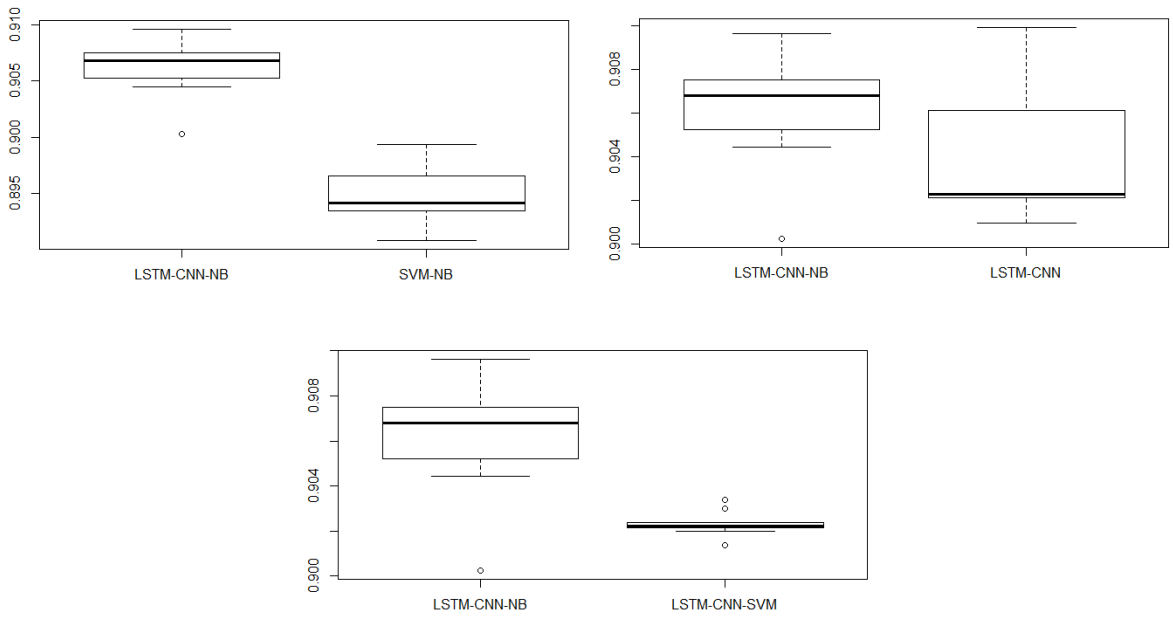

Figura 21 – Comparação da distribuição das acurácias entre LSTM-CNN-NB e os demais comitês para o SSTb binarizado

Por meio dos *box-plots* pode-se observar que a distribuição das acurácias da combinação LSTM-CNN-NB se mantêm sempre acima das acurácias dos demais classificadores, corroborando com os testes estatísticos. Também foi realizada a comparação do desem-

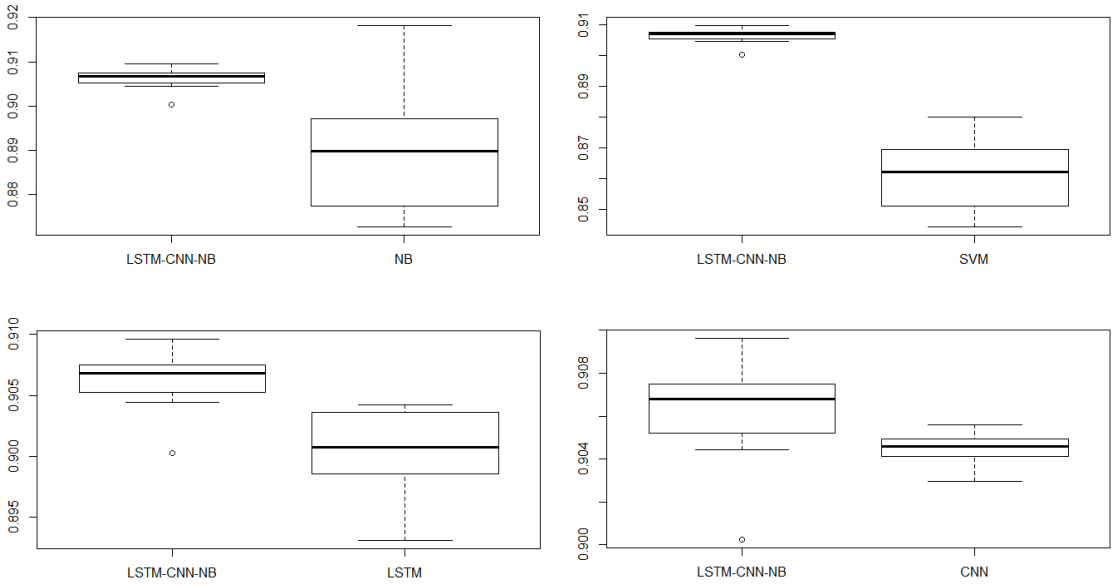

Figura 22 – Comparação da distribuição das acurácias entre LSTM-CNN-NB e os classificadores únicos para o SSTb binarizado

penho do NB (classificador mais rápido) em relação aos demais classificadores. A Tabela 25 detalha os resultados do teste.

Tabela 25 – Comparação da performance entre NB e demais classificadores para o SSTb binarizado

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*t-Student*	0.0001339	sim	rejeita H_0
NBxLSTM	*Wilcoxon*	0.03546	sim	rejeita H_0
NBxCNN	*Wilcoxon*	0.02569	sim	rejeita H_0
NBxSVM-NB	*Wilcoxon*	0.24755	não	não é possível rejeitar H_0
NBxLSTM-CNN	*Wilcoxon*	0.01118	sim	rejeita H_0
NBxLSTM-CNN-SVM	*Wilcoxon*	0.02558	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	0.006493	sim	rejeita H_0

A amostra de acurácias do NB é estatisticamente diferente das amostras de todos os classificadores, excetuando-se o SVM-NB, em que não foi possível rejeitar a hipótese nula. Através das figuras 23 e 24 é possível analisar, por meio de *box-plots*, em relação a quais classificadores o NB se torna uma opção melhor.

Observando a comparação da distribuição dos dados nas Figuras 23 e 24, o NB se mostra superior apenas ao classificador SVM, sendo estatisticamente compatível com o SVM-NB e inferior aos demais classificadores.

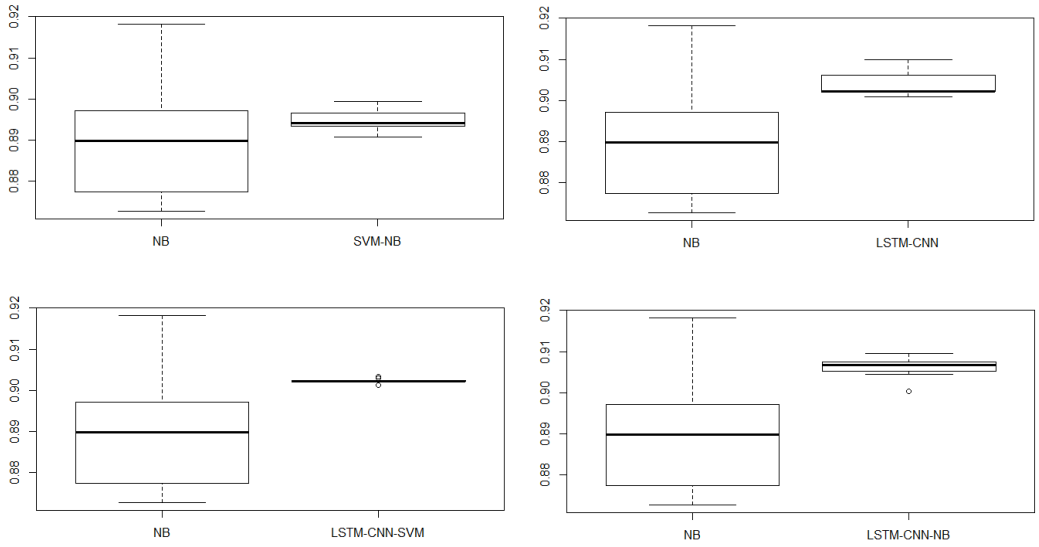

Figura 23 – Comparação da distribuição das acurácias entre NB e os comitês para o SSTb binarizado

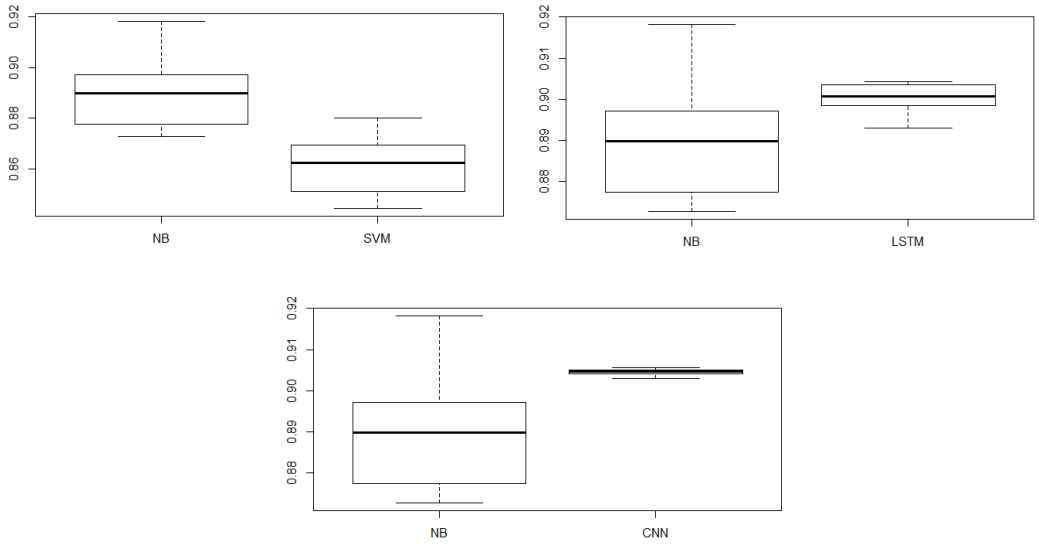

Figura 24 – Comparação da distribuição das acurácias entre NB os classificadores únicos para o SSTb binarizado

Para analisar a acertivadade da combinação LSTM-CNN-NB (maior acurácia) e do NB (menor tempo de execução) em cada polaridade de sentimentos individualmente, tem-se as métricas *precision*, *recall* e *f-score* distribuídas na Tabela 26.

Tabela 26 – Desempenho da LSTM-CNN-NB e do NB de acordo com cada polaridade de sentimentos para o SSTb binarizado

	LSTM-CNN-SVM/NB	
	negativa	**positiva**
precision	0.89 / 0.90	0.92 / 0.89
recall	0.95 / 0.92	0.83 / 0.86
f-score	0.92 / 0.91	0.98 / 0.87

Os dois classificadores demonstraram desempenho acima de 80% em todas as métricas e, mesmo o comitê alcançando um desempenho estatisticamente maior, o NB pode ser uma boa opção para esse problema, devido ao tempo de execução do classificador. Uma acurácia de 87.2% é alcançada por (HAN; BAI; LIU, 2018) para a base do SSTb com uma divisão binária de classes, porém não são realizados testes estatísticos que analisem o modelo de forma detalhada.

5.3 ANÁLISES DO YELP CHALLENGE 2013

Esta seção está dividida em duas subseções, a primeira contendo os experimentos e análises do Yelp 2013 com múltiplas classes e a segunda contendo os experimentos e análises do Yelp 2013 binarizado.

5.3.1 Análises do Yelp Challenge 2013 com múltiplas classes

Foram realizados um total de 8 experimentos com diferentes modelos de classificadores, a fim de analisar os melhores classificadores em termos de acurácia e tempo de execução. A Tabela 27 detalha as informações de cada classificador acerca de acurácia, desvio padrão e tempo de execução.

Tabela 27 – Performance dos classificadores para o Yelp Challenge 2013 *multclass*

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	40.72	0.001	**1.5**
SVM	39.58	0.003	13.5
LSTM	23.94	0.0003	5142
CNN	**57.82**	0.002	1060
SVM-NB	39.28	0.0008	16
LSTM-CNN	36.28	0.003	3722
LSTM-CNN-SVM	34.46	0.004	3735
LSTM-CNN-NB	34.35	0.0002	3723

Os melhores classificadores foram: CNN, em temos de acurácia, e NB, em termos de tempo de execução. Para analisar se há diferença entre esses classificadores e os demais por

meio de testes de hipóteses estatísticos, é necessário conhecer a natureza da distribuição dos dados. Um teste de normalidade é a forma mais segura para se decidir por rejeitar ou não a hipótese de que os dados obedecem a uma distribuição normal. A Tabela 28 detalha o resultado do teste de *Shapiro-Wilk* para as amostras dos classificadores do Yelp 2013 com múltiplas classes.

Tabela 28 – Teste de normalidade dos classificadores para o Yelp 2013 *multclass*

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.9206	não	não é possível rejeitar H_0
SVM	0.3433	não	não é possível rejeitar H_0
LSTM	0.003484	sim	rejeita H_0
CNN	00.7567	não	não é possível rejeitar H_0
SVM-NB	0.3574	não	não é possível rejeitar H_0
LSTM-CNN	0.0004494	sim	rejeita H_0
LSTM-CNN-SVM	0.01823	sim	rejeita H_0
LSTM-CNN-NB	0.002128	sim	rejeita H_0

Não é possível rejeitar a hipótese nula de que os classificadores obedecem a uma distribuição normal para os modelos NB, SVM, CNN e SVM-NB, portanto, nos testes de hipóteses para comparar a diferença em relação a estes classificadores será utilizado o teste paramétrico de *t-Student*. Quanto ao classificador LSTM e aos comitês LSTM-CNN, LSTM-CNN-SVM e LSTM-CNN-NB a hipótese nula foi rejeitada e estes não obedecem a uma distribuição normal, assim, para estes classificadores será utilizada a alternativa não paramétrica, teste de *Wilcoxon*. A Tabela 29 compara estatisticamente o desempenho da rede CNN (maior acurácia) em relação aos demais classificadores.

Tabela 29 – Comparação da performance entre a CNN e demais classificadores para o Yelp 2013 *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
CNNxNB	*t-Student*	2.2e-16	sim	rejeita H_0
CNNxSVM	*t-Student*	2.2e-16	sim	rejeita H_0
CNNxLSTM	*Wilcoxon*	0.0001776	sim	rejeita H_0
CNNxSVM-NB	*t-Student*	2.2e-16	sim	rejeita H_0
CNNxLSTM-CNN	*Wilcoxon*	0.0001776	sim	rejeita H_0
CNNxLSTM-CNN-SVM	*Wilcoxon*	0.0001727	sim	rejeita H_0
CNNxLSTM-CNN-NB	*Wilcoxon*	1.083e-05	sim	rejeita H_0

De acordo com os testes estatísticos da Tabela 29, a rede CNN possui desempenho diferente em relação a todos os outros modelos de classificadores. Por meio dos *box-plots* nas figuras 25 e 26 é possível comparar visualmente a diferença na distribuição dos dados.

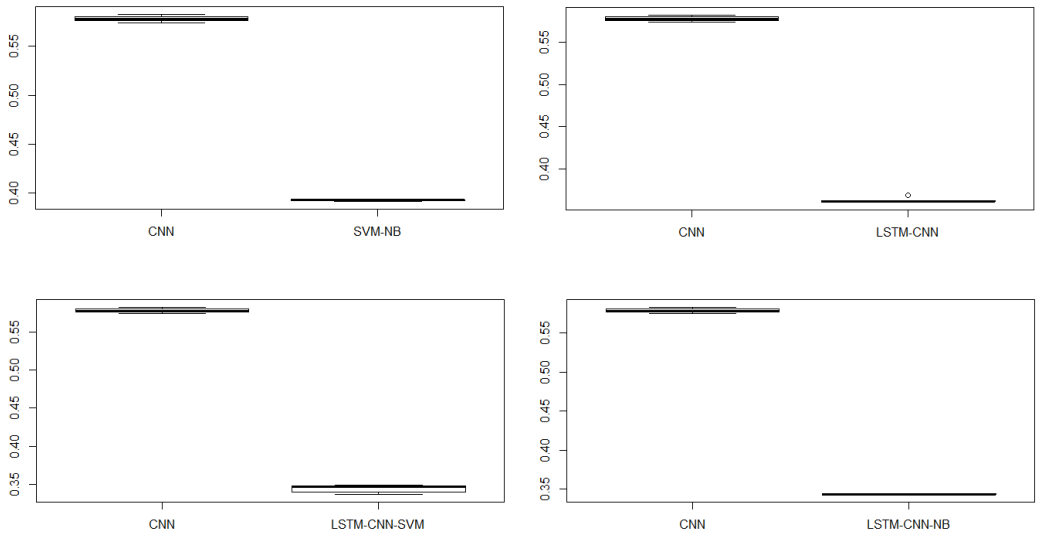

Figura 25 – Comparação da distribuição das acurácias entre CNN e os comitês para o Yelp 2013 *multclass*

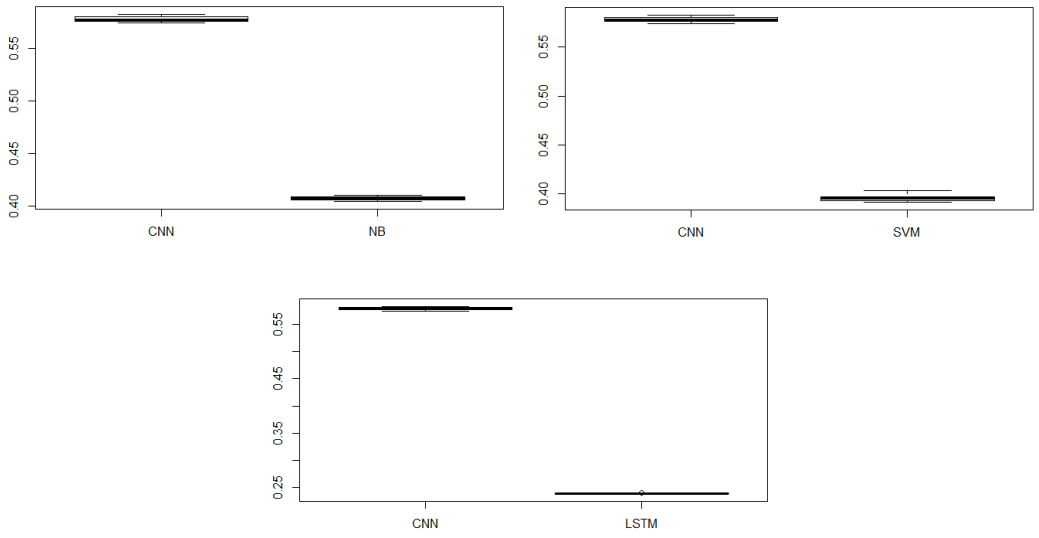

Figura 26 – Comparação da distribuição das acurácias entre CNN demais classificadores únicos para o Yelp 2013 *multclass*

A rede CNN aparece nos *box-plots* bem acima dos demais classificadores, alcançando uma acurácia com mais de 20% de diferença em alguns casos.

Como o classificador NB obteve o menor tempo de execução para esta base, seu desempenho foi comparado aos demais classificadores, a fim de analisar em quais situações o NB pode ser a melhor escolha. A Tabela 30 traz os resultados dos testes estatísticos que comparam o desempenho do NB aos demais modelos.

Tabela 30 – Comparação da performance entre NB e demais classificadores para o Yelp 2013 *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*t-Student*	0.01580618	sim	rejeita H_0
NBxLSTM	*Wilcoxon*	0.0001776	sim	rejeita H_0
NBxCNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxSVM-NB	*t-Student*	2.155e-11	sim	rejeita H_0
NBxLSTM-CNN	*Wilcoxon*	0.0001776	sim	rejeita H_0
NBxLSTM-CNN-SVM	*Wilcoxon*	1.083e-05	sim	rejeita H_0
NBxLSTM-CNN-NB	*Wilcoxon*	0.0001727	sim	rejeita H_0

O NB possui desempenho diferente em relação a todos os outros classificadores, por meio dos *box-plots*, nas figuras 27 e 28, é possível analisar em quais situações o NB possui desempenho superior

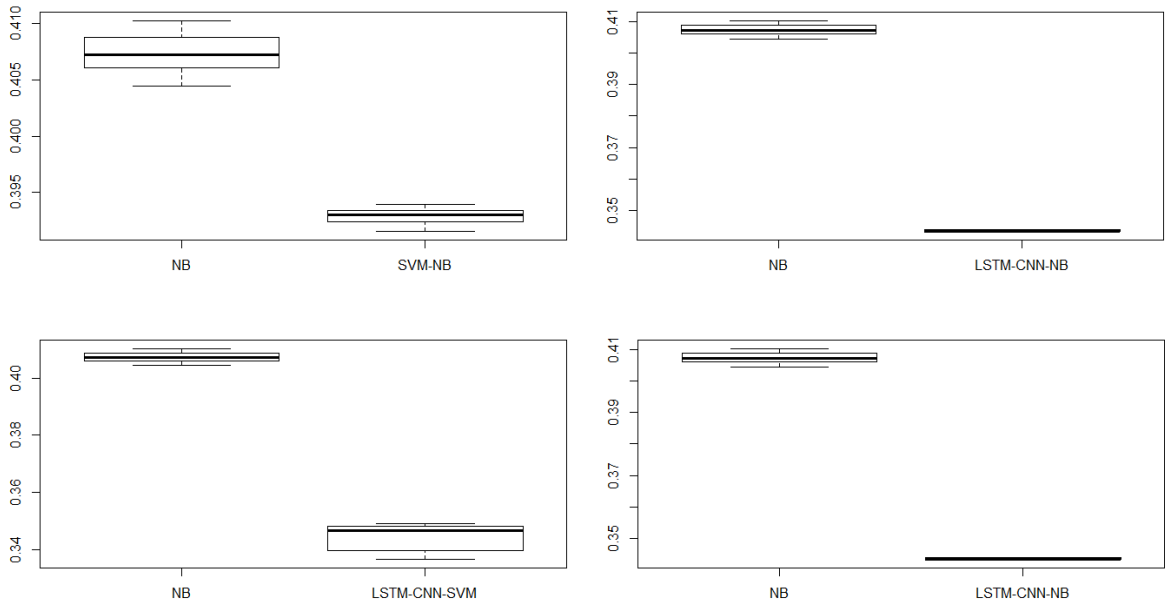

Figura 27 – Comparação da distribuição das acurácias entre NB e os comitês para o Yelp 2013 *multclass*

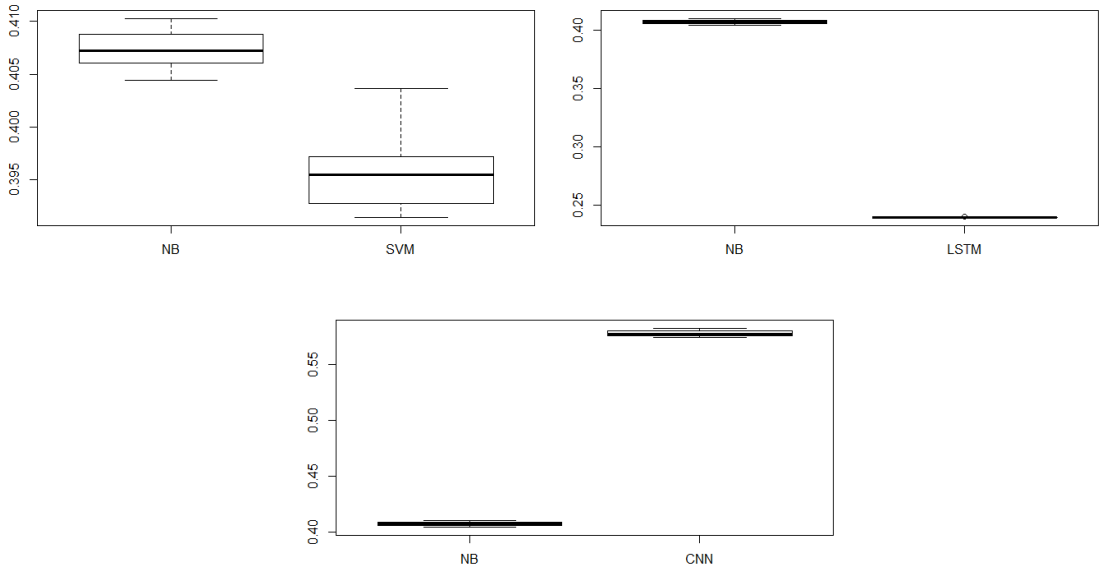

Figura 28 – Comparação da distribuição das acurácias entre NB demais classificadores únicos para o Yelp 2013 *multclass*

Como se pode observar, com exceção da rede CNN, que obteve os melhores desempenhos, o algoritmo NB se mostra estatisticamente superior a todos os outros modelos testados. Mesmo o NB se tornando uma boa opção em relação aos demais classificadores, ele ainda possui um desempenho cerca de 15% inferior a rede CNN. Assim, a rede de aprendizado profundo ainda se mostra mais robusta para este tipo de problema.

Por meio da Tabela 31 pode-se observar o desempenho da CNN e do NB em relação a cada polaridade de sentimentos, de acordo com as métricas *precision*, *recall* e *f-score*.

Tabela 31 – Desempenho da CNN e do NB de acordo com cada polaridade de sentimento do Yelp 2013 *multclass*

	CNN/NB					
	muito negativa	**negativa**	**neutra**	**positiva**	**muito positiva**	**total**
precision	0.67 / 0.51	0.53 / 0.36	0.56 / 0.37	0.48 / 0.35	0.60 / 0.41	0.57 / 0.40
recall	0.78 / 0.51	0.50 / 0.19	0.33 / 0.21	0.42 / 0.37	0.82 / 0.69	0.57 / 0.41
f-score	0.72 / 0.51	0.51 0.24	0.42 / 0.26	0.45 0.36	0.70 / 0.52	0.56 / 0.38

Considerando cada polaridade individualmente, a rede CNN também se mostra superior, atingindo maior acerto em todas as classes. A menor métrica atingida pelo NB foi um *recall* de 19% para a classe *muito negativa* e seu maior desempenho foi um *recall* de 69% para a classe *muito positiva*. No caso da CNN o seu menor desempenho foi um

recall de 33% referente a classe *neutra* e seu melhor desempenho foi uma *recall* de 82% para a classe *muito positiva*. Na literatura, tem-se pesquisas que apontam uma acurácia de 62.1% (GONG et al., 2018) para esta base, mas não são utilizados testes estatísticos que permitam maior análise deste resultado.

5.3.2 Análises do Yelp 2013 binarizado

Nesta seção são analisados estatisticamente os desempenhos dos classificadores com maior acurácia e menor tempo de execução em relação a base de dados Yelp 2013 binarizada (ver seção 4.1.3, pág. 41). A Tabela 32 detalha os valores de acurácia, desvio padrão e tempo de execução de cada classificador testado.

Tabela 32 – Performance dos classificadores para o Yelp 2013 binarizado

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	83.63	0.001	**0.04**
SVM	93.81	0.02	0.23
LSTM	96.08	0.001	2338
CNN	94.28	0.003	316
SVM-NB	79.64	0.003	2
LSTM-CNN	96.45	0.001	1592
LSTM-CNN-SVM	**96.54**	0.0008	1593
LSTM-CNN-NB	96.49	0.0007	1593

Em termos de acurácia, o comitê LSTM-CNN-SVM alcançou o maior valor em números absolutos. Quando a questão é custo computacional o NB aparece, de novo, com o menor tempo de execução. Assim, a combinação de classificadores LSTM-CNN-SVM e o NB terão seus desempenhos comparadados aos demais algoritmos, a fim de verificar estatisticamente se há diferença entre seus desempenhos e os dos demais classificadores. Para decidir entre a utilização de testes de hipóteses paramétricos ou não paramétricos, o teste de normalidade de *Shapiro-Wilk* é aplicado a fim de analisar estatisticamente a distribuição das amostras, como detalha a Tabela 33.

Tabela 33 – Teste de normalidade dos classificadores para o Yelp 2013 binarizado

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.8993	não	não é possível rejeitar H_0
SVM	0.01649	sim	rejeita H_0
LSTM	0.1996	não	não é possível rejeitar H_0
CNN	0.0004494	sim	rejeita H_0
SVM-NB	0.4572	não	não é possível rejeitar H_0
LSTM-CNN	0.836	não	não é possível rejeitar H_0
LSTM-CNN-SVM	0.07182	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.457	não	não é possível rejeitar H_0

Foi possível rejeitar a hipótese nula, de que os dados obedecem a uma distribuição normal, apenas para os classificadores SVM e CNN. Portanto, para verificar se há diferença estatística entre as amostras desses classificadores o teste não paramétrico de *Wilcoxon* será utilizado. No caso dos demais classificadores, em que não foi possível rejeitar H_0, o teste utilizado para verificar a diferença dos dados será o *t-Student*. A Tabela 34 apresenta os resultados dos testes que avaliam se há diferença entre o comitê LSTM-CNN-SVM (maior acuácia) e os demais modelos.

Tabela 34 – LSTM-CNN-SVM vs. demais classificadores (Yelp 2013 binarizado)

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
LSTM-CNN-SVMxNB	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNN-SVMxSVM	*Wilcoxon*	0.0001756	sim	rejeita H_0
LSTM-CNN-SVMxLSTM	*t-Student*	1.777e-08	sim	rejeita H_0
LSTM-CNN-SVMxCNN	*Wilcoxon*	0.0001707	sim	rejeita H_0
LSTM-CNN-SVM x SVM-NB	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNN-SVM x LSTM-CNN	*t-Student*	0.08104	não	não é possível rejeitar H_0
LSTM-CNN-SVM x LSTM-CNN-NB	*t-Student*	0.1407	não	não é possível rejeitar H_0

Como mostra a Tabela 34, o comitê LSTM-CNN-SVM possui acurácia diferente em relação a todos os classificadores únicos, além da combinação SVM-NB. Não foi comprovada diferença estatística entre as amostras do LSTM-CNN-SVM, LSTM-CNN-NB e LSTM-CNN. Por meio dos *box-plots*, nas Figuras 29 e 30, é possível comparar a distribuição das acurácias dos diferentes algoritmos.

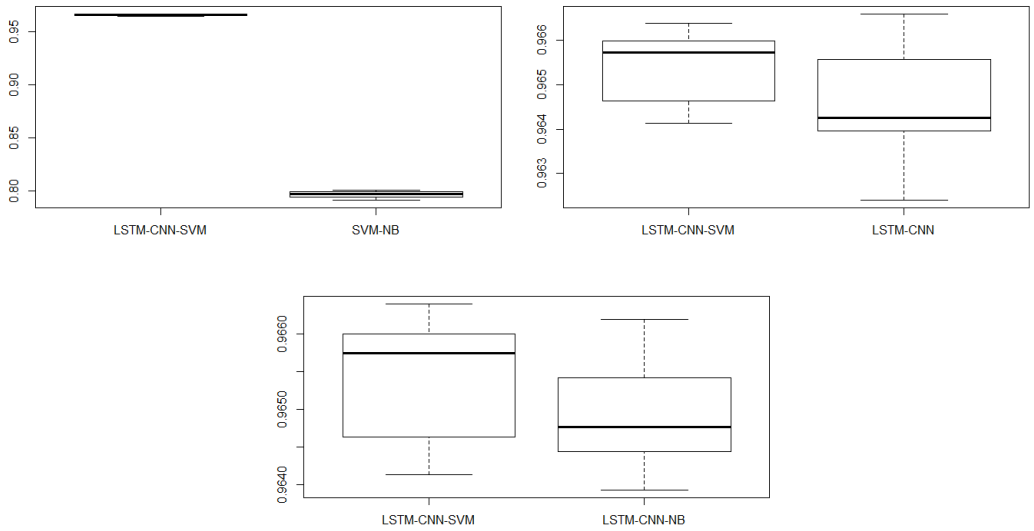

Figura 29 – Comparação da distribuição das acurácias entre LSTM-CNN-SVM e os demais comitês para o Yelp 2013 binarizado

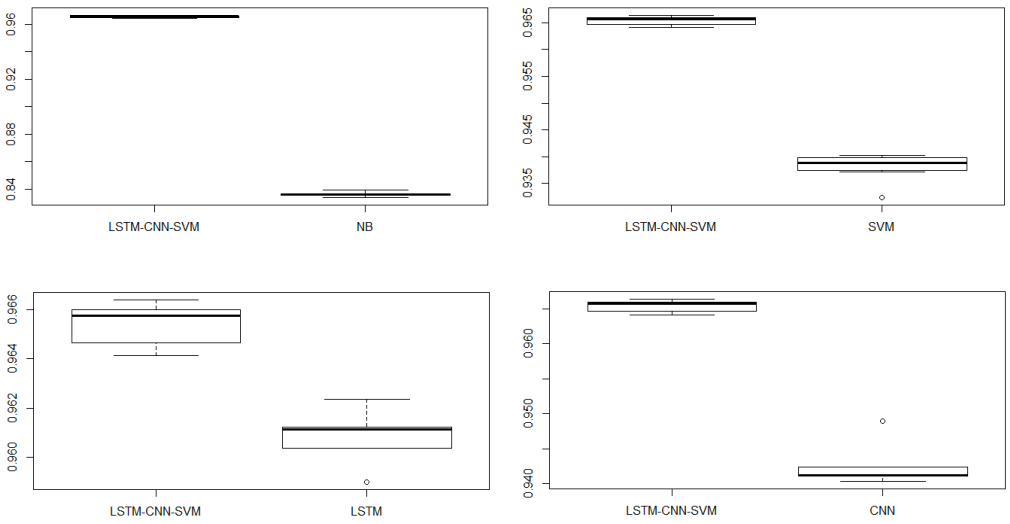

Figura 30 – Comparação da distribuição das acurácias entre LSTM-CNN-SVM e os classificadores únicos para o Yelp 2013 binarizado

É perceptível na Figura 29 que há interseção entre as acurácias dos comitês que possuem em sua composição classificadores de aprendizado profundo. Em 30 os classificadores únicos aparecem muito próximos a linha inferior dos *box-plots*, apontando superioridade nos desempenhos das combinações de classificadores.

Sabendo que o NB foi o modelo que obteve menor custo computacional, seu desempenho também foi comparado aos demais classificadores, os resultados dos teste estão na Tabela 35.

Tabela 35 – Comparação da performance entre NB e demais classificadores para o Yelp2013 binarizado

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*Wilcoxon*	1.083e-05	sim	rejeita H_0
NBxLSTM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxCNN	*Wilcoxon*	1.083e-05	sim	rejeita H_0
NBxSVM-NB	*t-Student*	8.613e-14	sim	rejeita H_0
NBxLSTM-CNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-SVM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	2.2e-16	sim	rejeita H_0

Tendo em vista que o desempenho do NB se mostrou estatisticamente diferente de todas as outras amostras, as figuras 31 e 32 trazem *box-plots* que comparam a distribuição das acurácias, a fim de analisar em quais situações o NB pode ser a melhor opção.

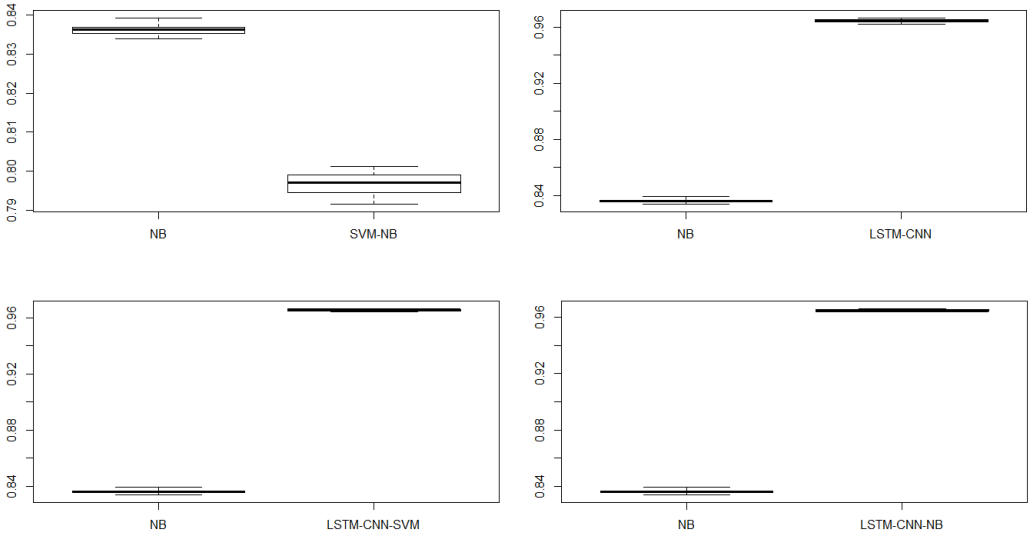

Figura 31 – Comparação da distribuição das acurácias entre NB e os comitês para o Yelp 2013 binarizado

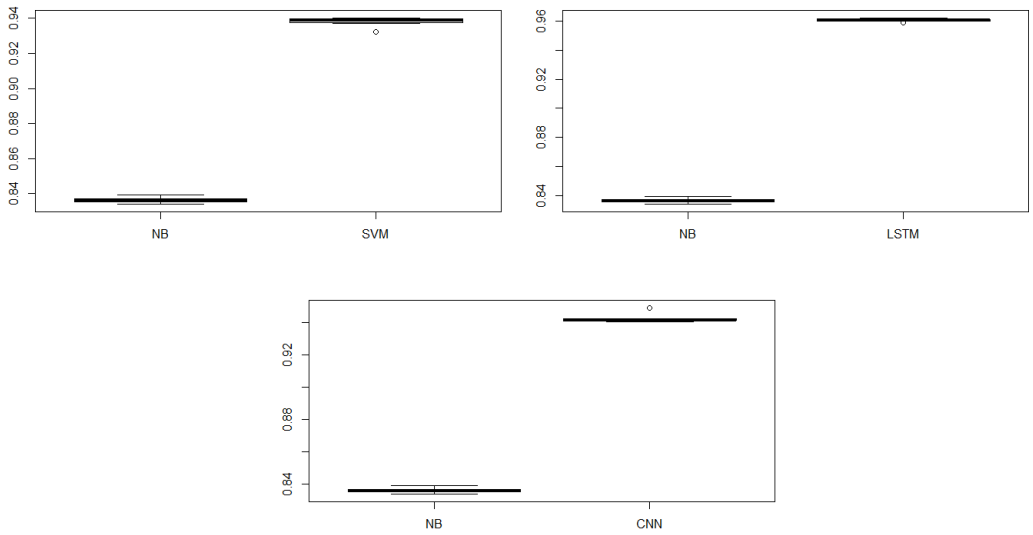

Figura 32 – Comparação da distribuição das acurácias entre NB os classificadores únicos para o Yelp 2013 binarizado

Como é possível analisar por meio dos *box-plots*, o NB possui desempenho consideravelmente inferior em relação a todos os modelos, exceto a combinação SVM-NB. Assim, em termos de desempenho os comitês que possuem algoritmos de aprendizado profundo (LSTM-CNN; LSTM-CNN-SVM e; LSTM-CNN-NB) podem ser considerados as melhores opções para este problema. A Tabela 36 conta com as métricas *precision*, *recall* e *f-score* para analisar o desempenho da combinação LSTM-CNN-SVM e do NB em relação a cada polaridade de sentimento.

Tabela 36 – Desempenho da LSTM-CNN-SVM e do NB de acordo com cada polaridade de sentimento para o Yelp 2013 binarizado

	LSTM-CNN-SVM/NB	
	negativa	**positiva**
precision	0.95 / 0.78	0.98 / 0.80
recall	0.99 / 0.80	0.94 / 0.78
f-score	0.96 / 0.79	0.96 / 0.79

O comitê LSTM-CNN-SVM atinge os maiores resultados nas duas polaridades de sentimentos consideradas individualmente, com métricas acima de 94%. Enquando a máxima do NB está relacionada à 80% de *precision* para a polaridade positiva e 80% de recall para a polaridade negativa, a combinação de classificadores chega a atingir 98% de *precision* para a classe positiva e 99% re *recall* para a classe negativa. Neste problema os comitês que fazem uso de algoritmos de aprendizado profundo se mostram como as melhores opções, perdendo dos classificadores tradicionais no quesito tempo de execução.

5.4 ANÁLISES DO YELP CHALLENGE 2014

Esta seção está dividida em duas subseções: a primeira aborda os experimentos e análises do Yelp 2014 com múltiplas classes; a segunda, os experimentos e análises do Yelp 2014 binarizado.

5.4.1 Análises do Yelp 2014 com múltiplas classes

A base de dados para análise de sentimentos do Yelp 2014 conta, originalmente, com 5 classes e foi submetida à experimentos com 8 classificadores, considerando 4 *single classifiers* e 4 comitês. O desempenho médio de cada classificador, a partir do 10 *fold cross validation* está detalhado na Tabela 37, junto ao tempo de execução e desvio padrão de cada modelo.

Tabela 37 – Performance dos classificadores para o Yelp Challenge 2014 *multclass*

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	41.21	0.001	**10**
SVM	41.68	0.002	37
LSTM	53.19	0.001	14638
CNN	57.39	0.001	791
SVM-NB	40.44	0.001	48
LSTM-CNN	**58.79**	0.0001	9258
LSTM-CNN-SVM	51.56	0.0002	9295
LSTM-CNN-NB	50.53	0.004	9268

Em números exatos, o comitê com algoritmos de aprendizado profundo, LSTM-CNN, aparece com maior desempenho. O algoritmos NB aparece com um dos menores desempenhos, porém apresenta também menor tempo de execução dentre todos os classificadores. Assim, LSTM-CNN e NB tiveram seus desempenhos comparados estatisticamente aos demais classificadores, a fim de analisar qual o melhor classificador para este problema. Para saber qual teste estatístico deve ser aplicado para verificar diferença entre as amostras, um teste de normalidade é aplicado anteriormente. A Tabela 38 traz os resultados do teste de normalidade de *Shapiro-Wilk* em relação as amostras de cada classificador.

Tabela 38 – Teste de normalidade dos classificadores para o Yelp 2014 *multclass*

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.6717	não	não é possível rejeitar H_0
SVM	0.0004232	sim	rejeita H_0
LSTM	0.08787	não	não é possível rejeitar H_0
CNN	0.8299	não	não é possível rejeitar H_0
SVM-NB	0.01339	sim	rejeita H_0
LSTM-CNN	0.5265	não	não é possível rejeitar H_0
LSTM-CNN-SVM	0.1825	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.06604	não	não é possível rejeitar H_0

Apenas as amostras dos classificadores SVM e SVM-NB rejeitaram a hipótese nula, quanto aos demais classificadores não foi possível rejeitar a hipótese de que os classificadores pertencem a uma distribuição normal. Assim, para analisar se há diferença entre os desempenhos dos classificadores foram utilizados os teste de *Wilcoxon*, para experimentos com SVM e SVM-NB e o teste *t-Student* para os experimentos com os demais classificadores. A Tabela 39 detalha os testes estatísticos que comparam a combinação LSTM-CNN aos demais classificadores.

Tabela 39 – Comparação da performance entre a LSTM-CNN e demais classificadores para o Yelp 2014 *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
LSTM-CNNxNB	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNNxSVM	*Wilcoxon*	1.083e-05	sim	rejeita H_0
LSTM-CNNxLSTM	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNNxCNN	*t-Student*	3.805e-10	sim	rejeita H_0
LSTM-CNNxSVM-NB	*Wilcoxon*	0.0001817	sim	rejeita H_0
LSTM-CNNxLSTM-CNN-SVM	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNNxLSTM-CNN-NB	*t-Student*	2.848e-13	sim	rejeita H_0

Analisando os resultados da Tabela 39, o comitê LSTM-CNN possui desempenho estatisticamente superior a todos os outros classificadores. As figuras 33 e 34 auxiliam na visualização dessa superioridade de desempenho, por meio de *box-plots* que comparam a distribuição das amostras.

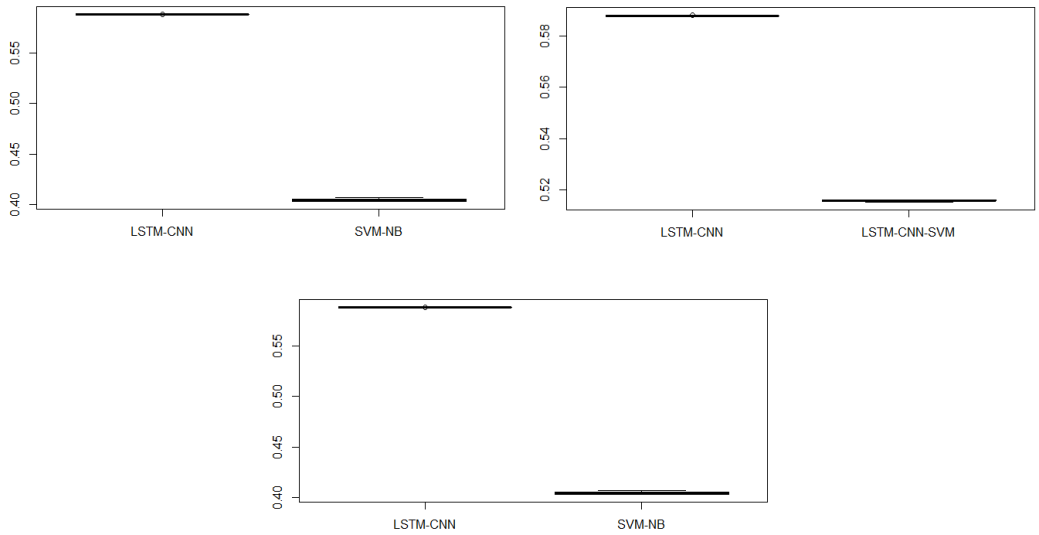

Figura 33 – Comparação da distribuição das acurácias entre LSTM-CNN e os comitês para o Yelp 2014 *multclass*

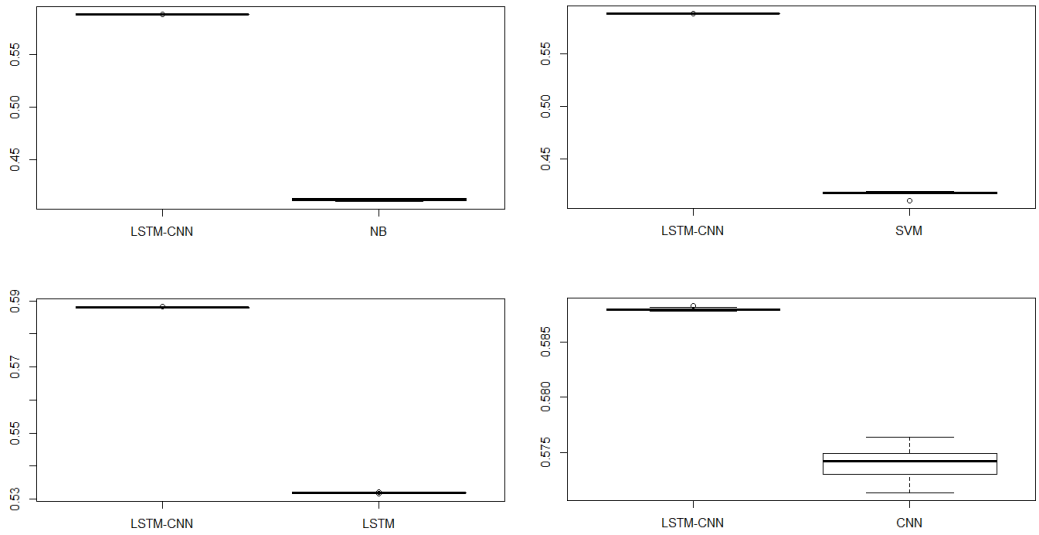

Figura 34 – Comparação da distribuição das acurácias entre LSTM-CNN demais classificadores únicos para o Yelp 2014 *multclass*

O classificador NB também teve seu desempenho comparado aos demais, a fim de analisar em quais situações ele demonstra um desempenho superior, tendo em vista que já possui o menor tempo de execução. A Tabela 40 detalha os resultados dos testes com o NB.

Tabela 40 – Comparação da performance entre NB e demais classificadores para o Yelp 2014 *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*Wilcoxon*	0.001505	sim	rejeita H_0
NBxLSTM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxCNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxSVM-NB	*Wilcoxon*	0.0001817	sim	rejeita H_0
NBxLSTM-CNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-SVM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	5.481e-15	sim	rejeita H_0

O NB obteve desempenho diferente de todos os outros classificadores. Por meio dos *box-plots* nas figuras 35 e 36 será possível analisar em relação a quais classificadores o NB é superior.

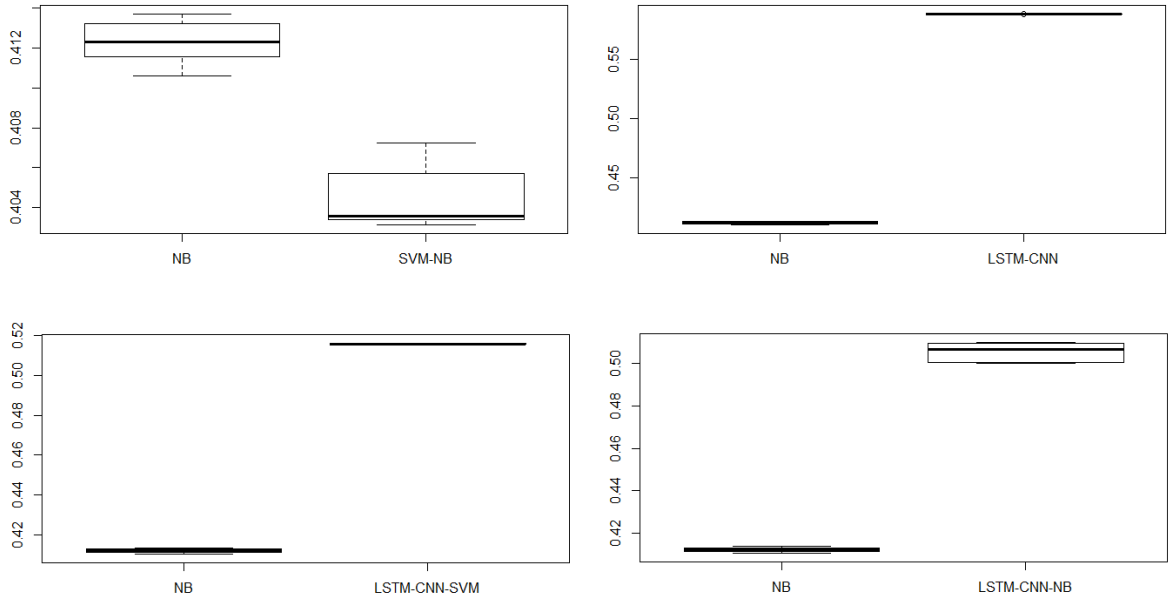

Figura 35 – Comparação da distribuição das acurácias entre NB e os comitês para o Yelp 2014 *multclass*

Percebe-se que o NB obteve desempenho superior acima da combinação SVM-NB, porém se mostra menos eficiente, do ponto de vista de desempenho, em relação a todos os outros classificadores, atingindo uma acurácia média bem abaixo dos demais.

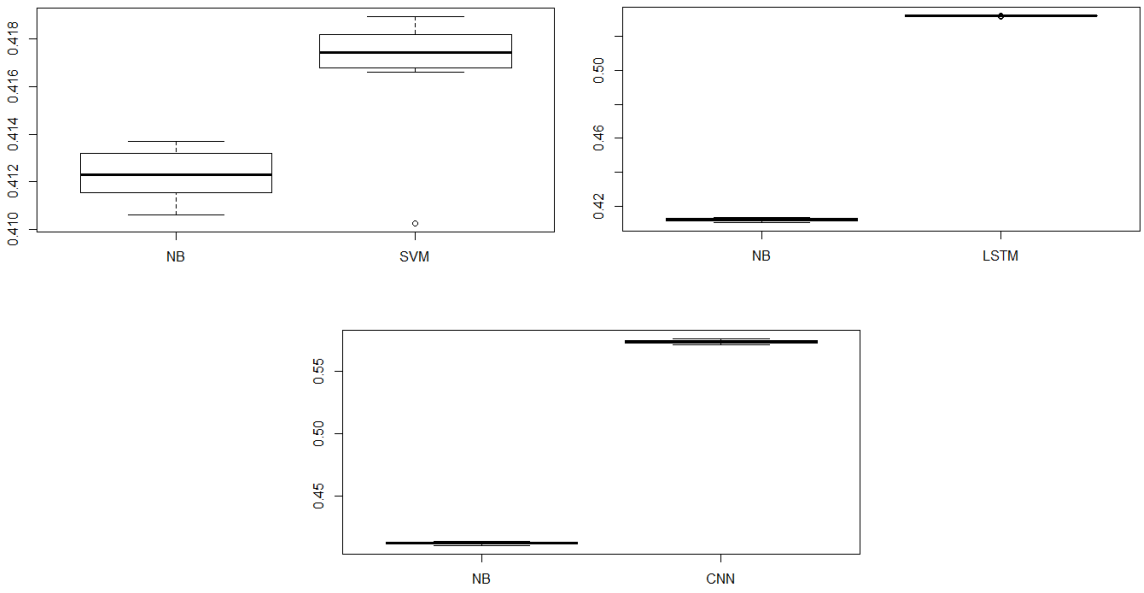

Figura 36 – Comparação da distribuição das acurácias entre NB demais classificadores únicos para o Yelp 2014 *multclass*

Para analisar o desempenho da LSTM-CNN (maior acurácia) e do NB (menor tempo de execução) em relação a cada polaridade de sentimentos individualmente, foram calculadas as métricas *precision*, *recall* e *f-score*, apresentadas na Tabela 41.

Tabela 41 – Desempenho da LSTM-CNN e do NB de acordo com cada polaridade de sentimento do Yelp 2014 *multclass*

	CNN/NB					
	muito negativa	**negativa**	**neutra**	**positiva**	**muito positiva**	**total**
precision	0.75 / 0.57	0.35 / 0.57	0.45 / 0.33	0.42 / 0.38	0.69 / 0.43	0.54 / 0.44
recall	0.53 / 0.39	0.33 / 0.00	0.51/ 0.49	0.50 / 0.28	0.63 / 0.71	0.52 / 0.41
f-score	0.62 / 0.46	0.34 0.00	0.48 / 0.41	0.46 0.32	0.66 / 0.53	0.52 / 0.37

É possível analisar que o comitê LSTM-CNN obteve melhor performance em relação a acertividade para todas as classes, individualmente. O algoritmo NB, apesar de ter obtido menor tempo de execução, possui performance falha, inclusive chegando à 0.00 acertos em determinadas métricas. A combinação de classificadores LSTM-CNN se mostrou como melhor classificador em termos de desempenho para esta base de dados e, ainda, apresentou menor tempo de execução dentre os classificadores de aprendizado profundo. Em Gong et al. (2018) é apontada uma acurácia de 63.0% para esta base, mas não são indi-

cados valores de variância, desvio padrão ou testes estatísticos que permitam analisar o resultado.

5.4.2 Análises do Yelp 2014 binarizado

O Yelp 2014 que possui, originalmente, 5 classes, foi reduzido as classes positiva e negativa para esses experimentos (ver seção 4.1.3, pág. 41). Foram realizados experimentos com 8 classificadores, a Tabela 42 apresenta as acurácias médias de cada classificador, bem como seu desvio padrão e tempo de execução.

Tabela 42 – Performance dos classificadores para o Yelp 2014 binarizado

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	89.47	0.001	**0.1**
SVM	93.65	0.001	1
LSTM	91.52	0.003	8413
CNN	95.61	0.001	1137
SVM-NB	79.32	0.002	1.5
LSTM-CNN	97.26	0.001	6003
LSTM-CNN-SVM	97.20	0.001	6004
LSTM-CNN-NB	**97.36**	0.002	6003

A combinação de classificadores LSTM-CNN-SVM obteve o maior valor de acurácia, enquanto o NB obteve menor tempo de execução. Para analisar se há diferença estatística entre esses dois classificadores e os demais foram realizados testes de nomalidade e de diferença de médias. A Tabela 43 apresenta os resultados do teste de normalidade de *Shapiro-Wilk* para cada classificador.

Tabela 43 – Teste de normalidade dos classificadores para o Yelp 2014 binarizado

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.7213	não	não é possível rejeitar H_0
SVM	0.4614	não	não é possível rejeitar H_0
LSTM	0.9644	não	não é possível rejeitar H_0
CNN	0.07286	não	não é possível rejeitar H_0
SVM-NB	0.04432	sim	rejeita H_0
LSTM-CNN	0.0469	sim	rejeita H_0
LSTM-CNN-SVM	0.3543	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.08441	não	não é possível rejeitar H_0

Apenas os comitês SVM-NB e LSTM-CNN rejeitaram a hipótese nula de que os dados obedecem a uma distribuição normal. Assim, os testes de hipóteses envolvendo SVM-NB e LSTM-CNN foram realizados a partir do teste de *Wilcoxon*. O teste *t-Student* foi utilizado

para os experimentos com os demais classificadores. A Tabela 44 apresenta os resultados das comparações entre o comitê LSTM-CNN-NB (maior valor de acurácia) e os demais classificadores.

Tabela 44 – LSTM-CNN-NB vs. demais classificadores (Yelp 2014 binarizado)

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
LSTM-CNN-NBxNB	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNN-NBxSVM	*t-Student*	5.075e-15	sim	rejeita H_0
LSTM-CNN-NBxLSTM	*t-Student*	5.075e-15	sim	rejeita H_0
LSTM-CNN-NBxCNN	*t-Student*	9.034e-12	sim	rejeita H_0
LSTM-CNN-NB x SVM-NB	*Wilcoxon*	0.0001766	sim	rejeita H_0
LSTM-CNN-NB x LSTM-CNN	*Wilcoxon*	0.3436	não	não é possível rejeitar H_0
LSTM-CNN-NB x LSTM-CNN-SVM	*t-Student*	0.06624	sim	rejeita H_0

A amostra das acurácias do LSTM-CNN-NB apresentou diferença em relação aos modelos NB, SVM, LSTM, CNN, SVM-NB e LSTM-CNN-SVM. Na comparação com o comitê LSTM-CNN não há como constatar diferença entre as amostras. Por meio dos *box-plots* das figuras 37 e 38 é possível observar que a LSTM-CNN-NB possui sua distribuição bem acima dos demais modelos, mas em relação à combinação LSTM-CNN há grande interseção.

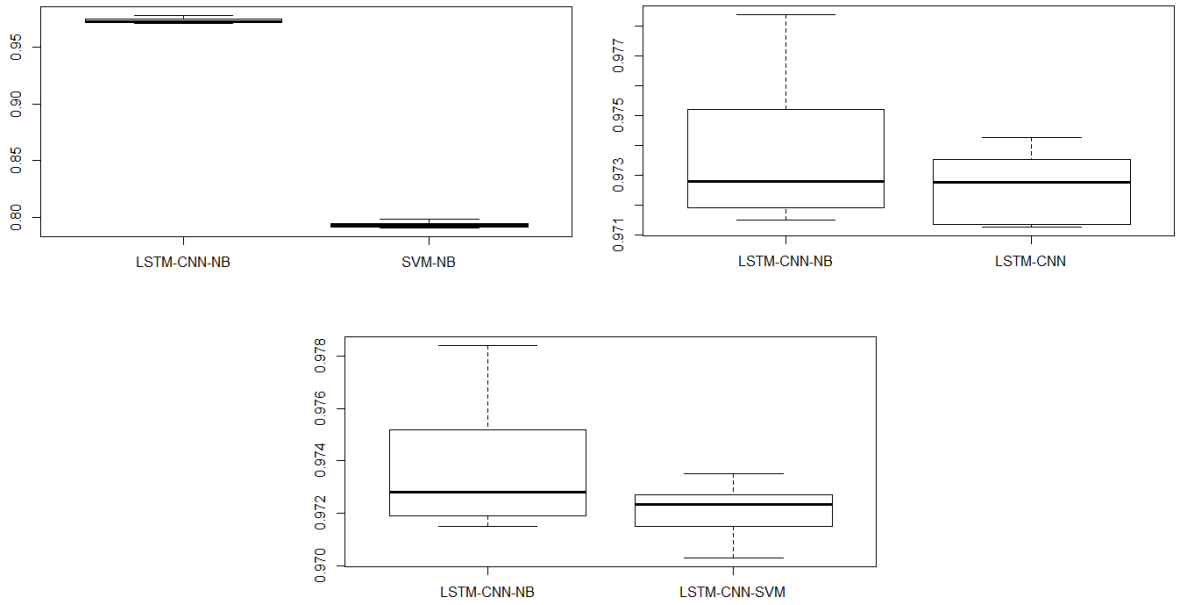

Figura 37 – Comparação da distribuição das acurácias entre LSTM-CNN-NB e os demais comitês para o Yelp 2014 binarizado

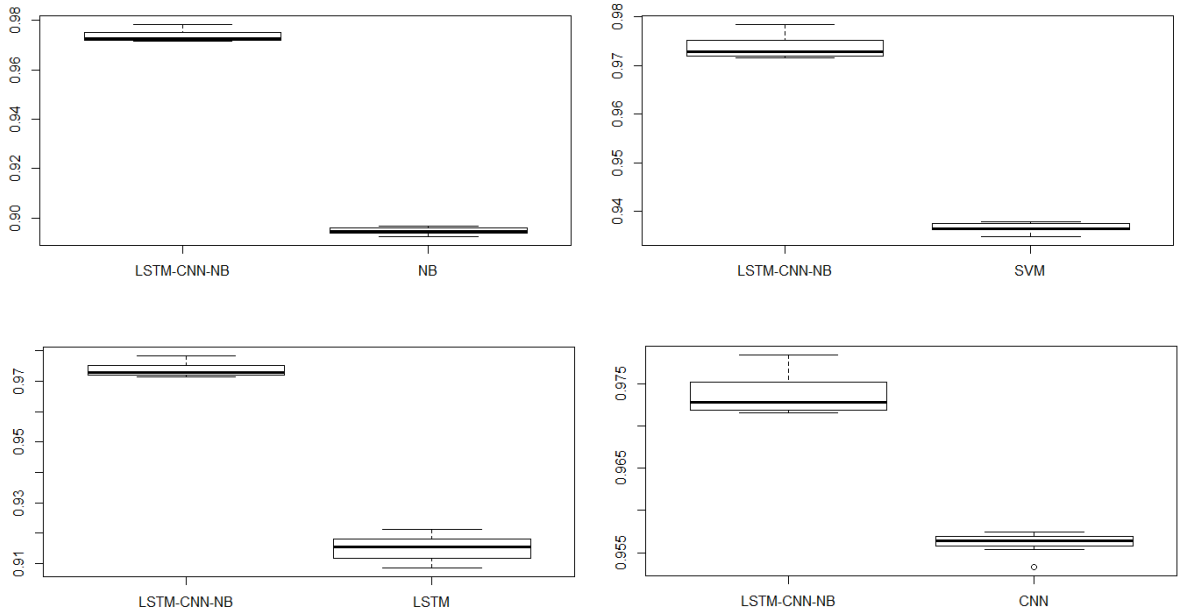

Figura 38 – Comparação da distribuição das acurácias entre LSTM-CNN-NB e os classificadores únicos para o Yelp 2014 binarizado

Como o classificador NB foi o que obteve menor tempo de execução, sua performance também foi comparada aos demais classificadores, a Tabela 45 apresenta os resultados dos testes de hipótese que analisam a diferença entre a acurácia dos classificadores.

Tabela 45 – Comparação da performance entre NB e demais classificadores para o Yelp2014 binarizado

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM	*t-Student*	4.312e-09	sim	rejeita H_0
NBxCNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxSVM-NB	*Wilcoxon*	0.0001786	sim	rejeita H_0
NBxLSTM-CNN	*Wilcoxon*	0.0001786	sim	rejeita H_0
NBxLSTM-CNN-SVM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	2.2e-16	sim	rejeita H_0

A Tabela 45 aponta que há diferença estatística entre as amostras dos NB e demais classificadores. Por meio dos *box-plots* presentes nas figuras 39 e 40 é possível analise se o NB é uma alternativa melhor, em termos de performance, em relação a algum outro modelo.

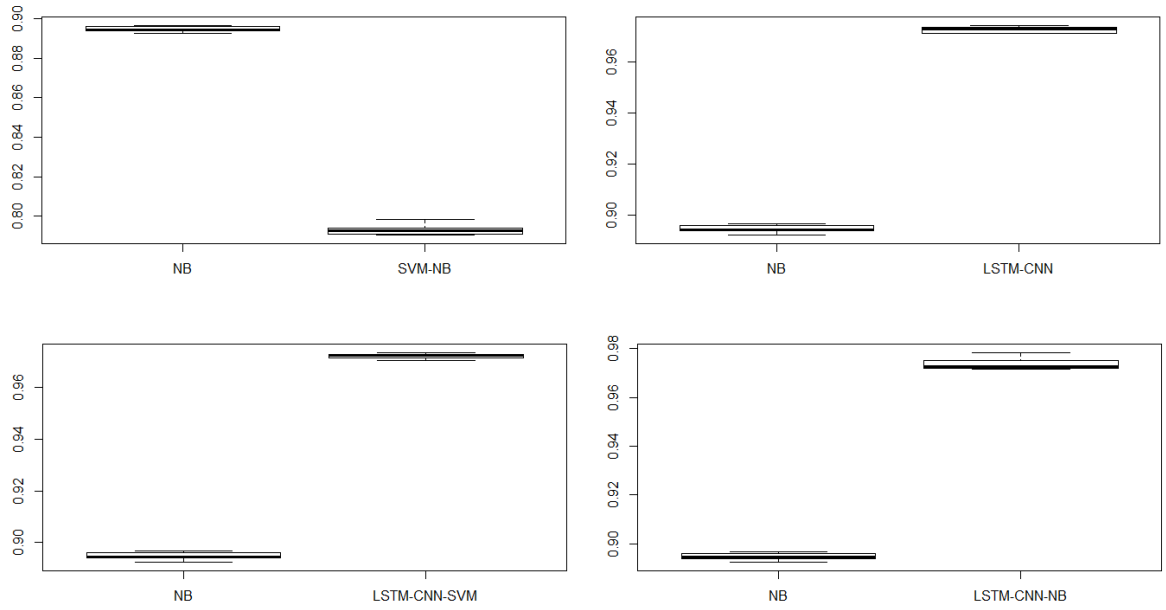

Figura 39 – Comparação da distribuição das acurácias entre NB e os comitês para o Yelp 2014 binarizado

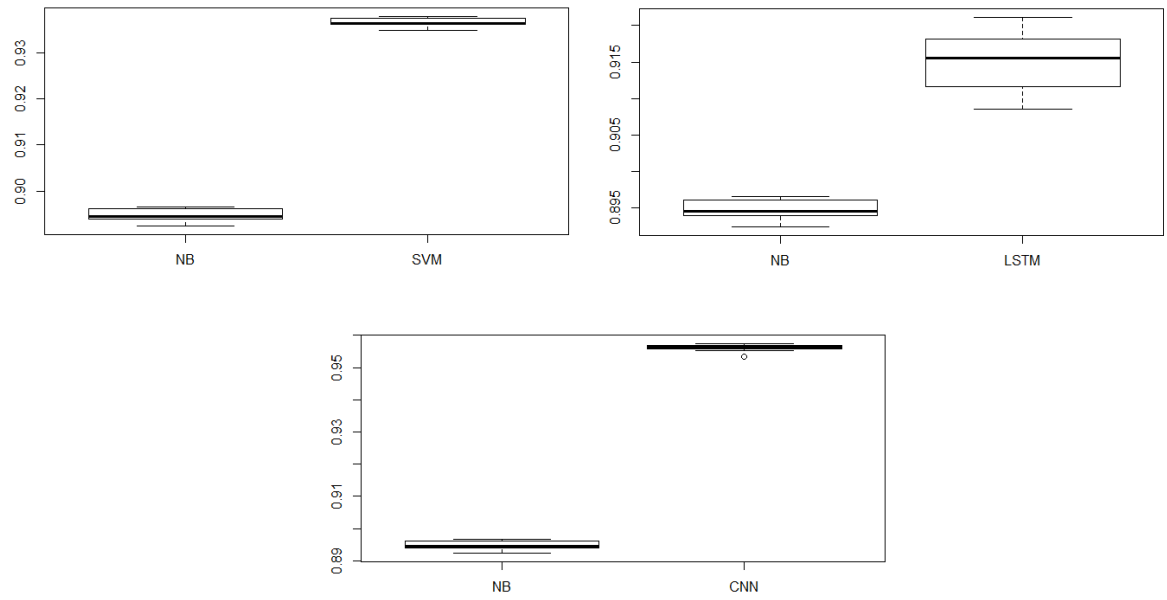

Figura 40 – Comparação da distribuição das acurácias entre NB os classificadores únicos para o Yelp 2014 binarizado

Observa-se que todos os modelos possuem desempenho superior ao NB, o que demonstra que mesmo sendo rápido, não possui desempenho tão interessante para o problema. Por fim, foram analisados os desempenhos da combinação de classificadores LSTM-CNN-NB (maior acurácia) e do NB (menor tempo de execução) de acordo com cada classe individualmente, como mostra a Tabela 46.

Tabela 46 – Desempenho da LSTM-CNN-NB e do NB de acordo com cada polaridade de sentimento para o Yelp 2014 binarizado

	LSTM-CNN-SVM/NB	
	negativa	**positiva**
precision	0.97 / 0.77	0.98 / 0.81
recall	0.98 / 0.84	0.97 / 0.73
f-score	0.97 / 0.80	0.97 / 0.79

Enquando o NB figura tanto para a polaridade positiva quanto negativa com métricas entre 70% e 80%, o comitê LSTM-CNN-NB figura entre 97% e 98% em todas as métricas, para as duas polaridades. Apesar de tempo de execução muito superior, a combinação LSTM-CNN-NB também um ganho muito superior em termos de desempenho, além de possuir tempo de execução inferior aos *single classifiers* de aprendizado profundo.

5.5 ANÁLISES DO YELP CHALLENGE 2015

Nesta seção serão analisados os experimentos que fizeram uso da base de dados Yep Challenge 2015. A primeira subseção trata dos resultados e análises da base com múltiplas classes (5) e a segunda subseção aborda os resultados e análises do Yelp 2015 binarizado.

5.5.1 Análises do Yelp 2015 com múltiplas classes

A base de dados do Yelp 2015 possui 5 classes, originalmente, que foram classificadas neste trabalho por meio de 8 classificadores. Os resultados contendo acurácia, desvio padrão e tempo de execução de cada classificador estão detalhados na Tabela 47.

Tabela 47 – Performance dos classificadores para o Yelp Challenge 2015 *multclass*

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	43.19	0.0006	**12**
SVM	44.98	0.001	53
LSTM	52.61	0.009	18376
CNN	**60.22**	0.003	1024
SVM-NB	43.03	0.004	65
LSTM-CNN	53.92	0.003	11640
LSTM-CNN-SVM	53.69	0.0006	11696
LSTM-CNN-NB	52.72	0.001	11655

A rede de aprendizado profundo, CNN, alcançou maior valor de acurácia, enquanto o classificado de aprendizado de máquina tradicional, NB, atingiu menor tempo de execução. Assim, CNN e NB terão seus desempenhos comparados aos demais classificadores por meio de testes estatísticos. O teste de *Shapiro-Wilk*, que testa se as amostras dos classificadores provém de uma distribuição normal, tem seus resultados apresentados na Tabela 48.

Tabela 48 – Teste de normalidade dos classificadores para o Yelp 2015 *multclass*

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.009145	sim	rejeita H_0
SVM	0.8011	não	não é possível rejeitar H_0
LSTM	0.1632	não	não é possível rejeitar H_0
CNN	0.2221	não	não é possível rejeitar H_0
SVM-NB	0.5565	não	não é possível rejeitar H_0
LSTM-CNN	0.03921	sim	rejeita H_0
LSTM-CNN-SVM	0.08451	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.9206	não	não é possível rejeitar H_0

O classificador único NB e a combinação LSTM-CNN rejeitaram a hipótese nula de que os dados provém de uma distribuição normal, assim o teste para comparar as amostras

destes classificadores será a alternativa não-paramétrica de *Wilcoxon*. Para os demais classificadores será utilizado o teste *t-Student*. A Tabela 49 apresenta os resultados das comparações entre a rede CNN (maior valor de acurácia) e os demais classificadores.

Tabela 49 – Comparação da performance entre a CNN e demais classificadores para o Yelp 2015 *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
CNNxNB	*Wilcoxon*	0.0001796	sim	rejeita H_0
CNNxSVM	*t-Student*	2.2e-16	sim	rejeita H_0
CNNxLSTM	*t-Student*	1.648e-14	sim	rejeita H_0
CNNxSVM-NB	*t-Student*	2.2e-16	sim	rejeita H_0
CNNxLSTM-CNN	*Wilcoxon*	0.0001756	sim	rejeita H_0
CNNxLSTM-CNN-SVM	*t-Student*	1.996e-13	sim	rejeita H_0
CNNxLSTM-CNN-NB	*t-Student*	2.2e-16	sim	rejeita H_0

A rede CNN apresentou desempenho diferente de todos os outros modelos, se mostrando estatisticamente superior para esta base. Os *box-plots* apresentados nas Figuras 41 e 42 corroboram com os testes estatísticos ao comparar visualmente a distribuição das amostras.

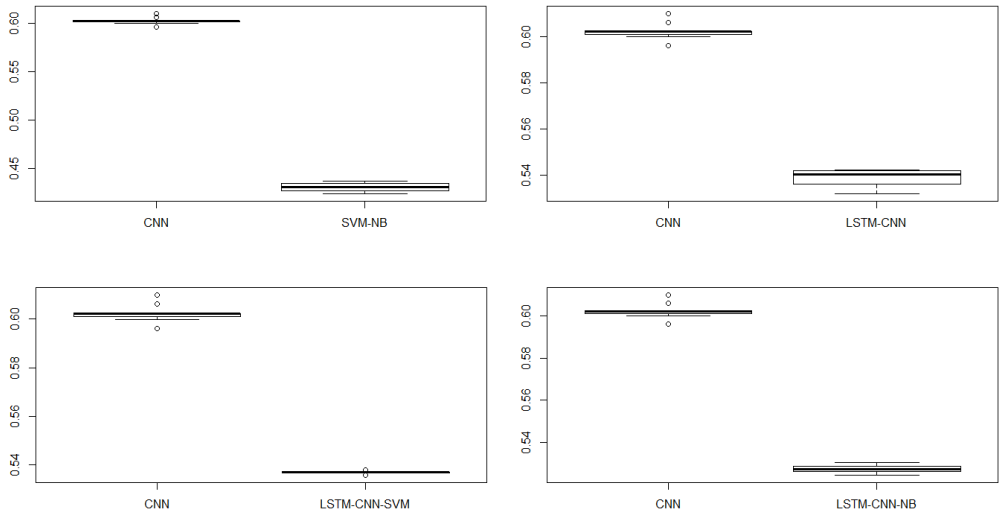

Figura 41 – Comparação da distribuição das acurácias entre LSTM-CNN e os comitês para o Yelp 2015 *multclass*

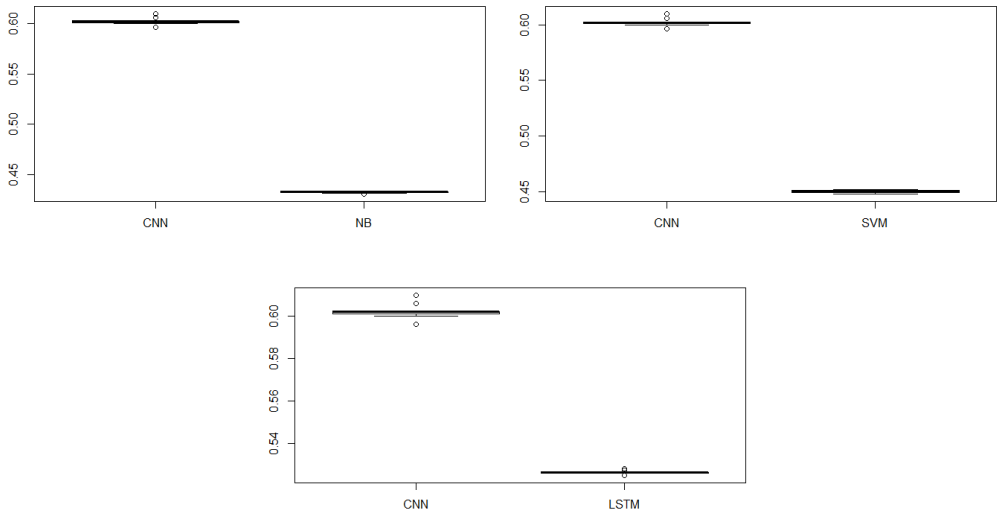

Figura 42 – Comparação da distribuição das acurácias entre LSTM-CNN demais classificadores únicos para o Yelp 2015 *multclass*

O classificador NB também foi avaliado, tendo em vista que apresentou menor tempo de execução. A Tabela 50 apresenta os resultados que comparam a performance do NB com os demais classificadores.

Tabela 50 – Comparação da performance entre NB e demais classificadores para o Yelp 2015 *multclass*

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*Wilcoxon*	1.083e-05	sim	rejeita H_0
NBxLSTM	*Wilcoxon*	0.0001707	sim	rejeita H_0
NBxCNN	*Wilcoxon*	0.0001796	sim	rejeita H_0
NBxSVM-NB	*Wilcoxon*	0.5288	não	não é possível rejeitar H_0
NBxLSTM-CNN	*Wilcoxon*	0.0001786	sim	rejeita H_0
NBxLSTM-CNN-SVM	*Wilcoxon*	0.0001817	sim	rejeita H_0
NBxLSTM-CNN-NB	*Wilcoxon*	1.083e-05	sim	rejeita H_0

De acordo com a Tabela 50 o NB possui desempenho diferente em relação a todos os modelos, exceto a combinação SVM-NB. Por meio dos *box-plots* 43 e 44 é possível analisar a comparação da distribuição das amostras dos classificadores.

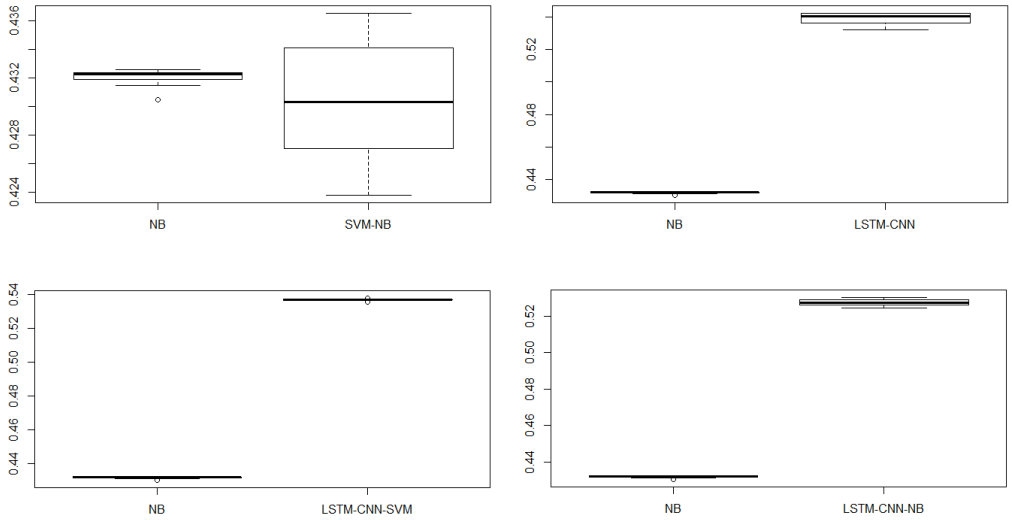

Figura 43 – Comparação da distribuição das acurácias entre NB e os comitês para o Yelp 2015 *multclass*

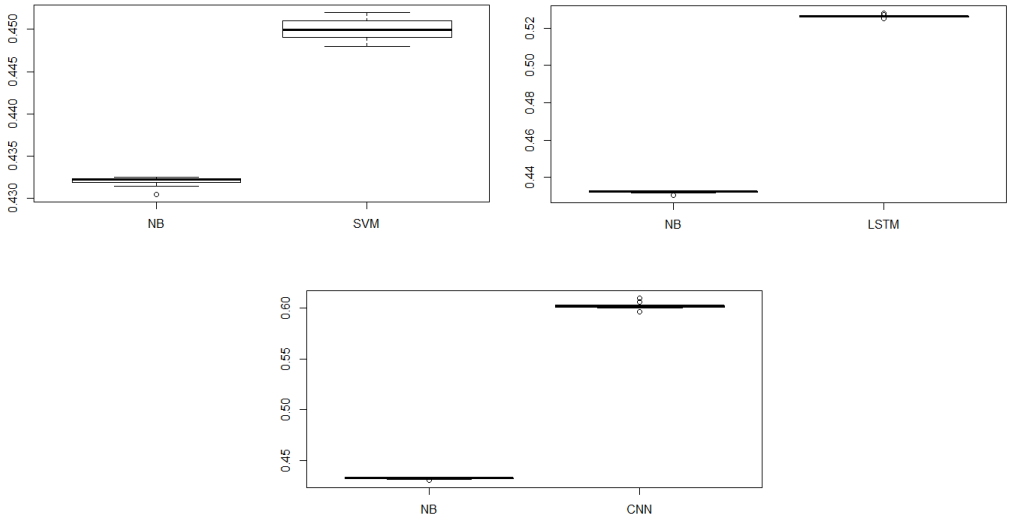

Figura 44 – Comparação da distribuição das acurácias entre NB demais classificadores únicos para o Yelp 2015 *multclass*

O NB não aparece com desempenho acima de nenhum classificador, porém pode ter desempenho igual ao comitê SVM-NB. Os desempenhos da CNN (maior acurácia) e do NB (menor tempo de execução) também foram avaliados considerando cada polaridade de sentimentos individualmente, como mostra a Tabela 51.

Tabela 51 – Desempenho da CNN e do NB de acordo com cada polaridade de sentimento do Yelp 2015 *multclass*

	CNN/NB					
	muito negativa	**negativa**	**neutra**	**positiva**	**muito positiva**	**total**
precision	0.65 / 0.57	0.43 / 0.42	0.41 / 0.00	0.58 / 0.35	0.76 / 0.43	0.60 / 0.38
recall	0.87 / 0.48	0.58 / 0.04	0.29/ 0.00	0.45 / 0.35	0.70 / 0.84	0.60 / 0.43
f-score	0.62 / 0.52	0.34 0.07	0.48 / 0.00	0.46 0.35	0.66 / 0.57	0.52 / 0.36

O NB apresenta desempenho inferior a rede de aprendizado profundo CNN no que se refere a todas as polaridades de sentimento, chegando à métricas 0.00 em algumas polaridades. Mesmo sendo um algoritmo muito rápido, o desempenho do NB não o coloca como um algoritmo interessante para este problema. A rede CNN, além de aparecer com performance superior, é o algoritmo mais rápido quando considerados comitês e algoritmos únicos de aprendizado profundo, sendo assim uma boa opção para este problema. Em Rao et al. (2018) é apresentado um classificador que atinge acurácia de 65.3% para a base do Yelp 2015, mas não são apresentados valores de desvio padrão ou testes estatísticos para melhor analisar o resultado.

5.5.2 Análises do Yelp 2015 binarizado

O Yelp 2015 possui, originalmente, 5 classes. Para estes experimentos, a base de dados foi binarizada, restando as classes positiva e negativa (ver seção 4.1.3, pág. 41). Os experimentos realizados com 8 classificadores possuem seus resultados detalhados na Tabela 52.

Tabela 52 – Performance dos classificadores para o Yelp Challenge 2015 *multclass*

classificador	**acc (%)**	**sd**	**tempo (s)**
NB	90.25	0.0006	**0.3**
SVM	93.65	0.0009	1.8
LSTM	96.35	0.0002	11606
CNN	96.02	0.0006	1640
SVM-NB	80.04	0.001	2
LSTM-CNN	97.39	0.001	7947
LSTM-CNN-SVM	97.49	0.001	7948
LSTM-CNN-NB	**97.54**	0.001	7947

Os classificadores que terão seu desempenho comparado ao dos demais modelos são: LSTM-CNN-NB, por ter atingido o maior valor de acurácia, e NB, por ter atingido o

menor tempo de execução. A Tabela 53 apresenta os resultados do teste de normalidade das amostras de todos os classificadores, a fim de identificar qual o teste de hipótese que deve ser utilizado para analisar se há diferença estatística entre os modelos.

Tabela 53 – Teste de normalidade dos classificadores para o Yelp 2015 binarizado

classificador	***p-value***	***p-value*** $< \alpha$	**resultado**
NB	0.6533	não	não é possível rejeitar H_0
SVM	0.4614	não	não é possível rejeitar H_0
LSTM	0.002128	sim	rejeita H_0
CNN	0.2951	não	não é possível rejeitar H_0
SVM-NB	0.6416	não	não é possível rejeitar H_0
LSTM-CNN	0.1768	não	não é possível rejeitar H_0
LSTM-CNN-SVM	0.2997	não	não é possível rejeitar H_0
LSTM-CNN-NB	0.1287	não	não é possível rejeitar H_0

Apenas a amostra da rede de aprendizado profundo LSTM rejeitou H_0, ou seja, não obedece a uma distribuição normal, assim nos experimentos que comparam o desempenho desta rede será utilizado o teste de *Wilcoxon*. Para os experimentos que analisam a diferença entre os demais classificadores, foi utilizado o teste *t-student*, como mostra a Tabela 54, que compara o desempenho da LSTM-CNN-NB (maior valor de acurácia) ao desempenho dos demais classificadores.

Tabela 54 – LSTM-CNN-NB vs. demais classificadores (Yelp 2015 binarizado)

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
LSTM-CNN-NBxNB	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNN-NBxSVM	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNN-NBxLSTM	*Wilcoxon*	0.0001717	sim	rejeita H_0
LSTM-CNN-NBxCNN	*t-Student*	3.01e-15	sim	rejeita H_0
LSTM-CNN-NB x SVM-NB	*t-Student*	2.2e-16	sim	rejeita H_0
LSTM-CNN-NB x LSTM-CNN	*t-Student*	0.01813	sim	rejeita H_0
LSTM-CNN-NB x LSTM-CNN-SVM	*t-Student*	0.4086	não	não é possível rejeitar H_0

De acordo com os testes estatísticos, o desempenho do comitê LSTM-CNN-NB é estatisticamente superior ao desempenho dos demais modelos. Os *box-plots*, presentes nas figuras 45 e 46 apresentam uma comparação da distribuição dos dados analisadas na Tabela 54.

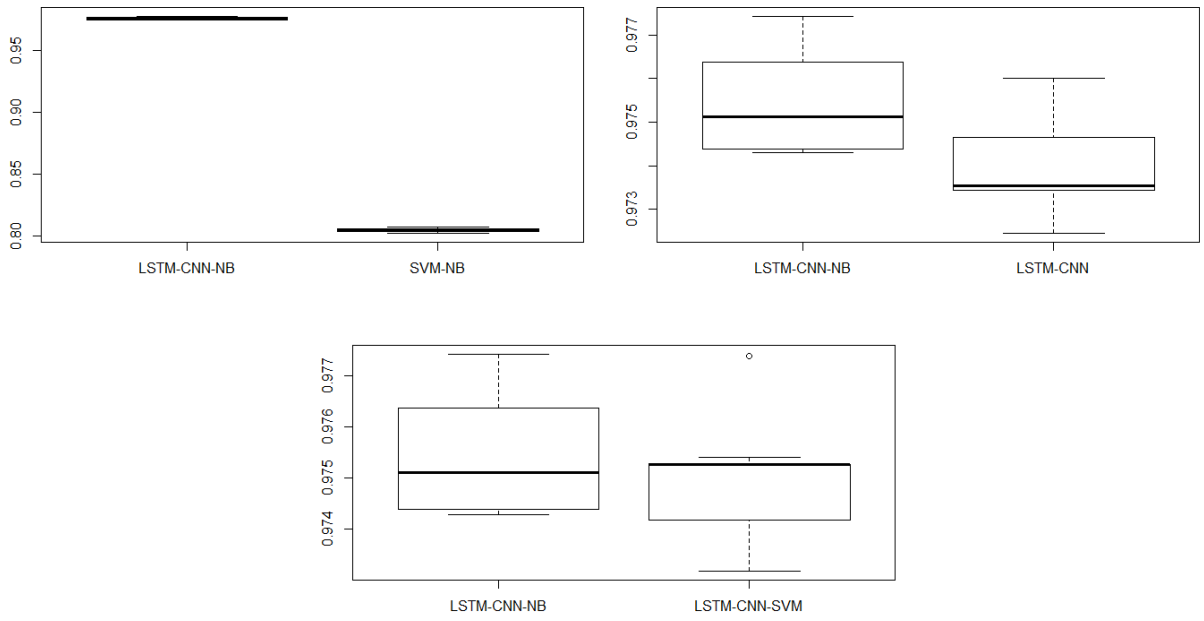

Figura 45 – Comparação da distribuição das acurácias entre LSTM-CNN-NB e os demais comitês para o Yelp 2015 binarizado

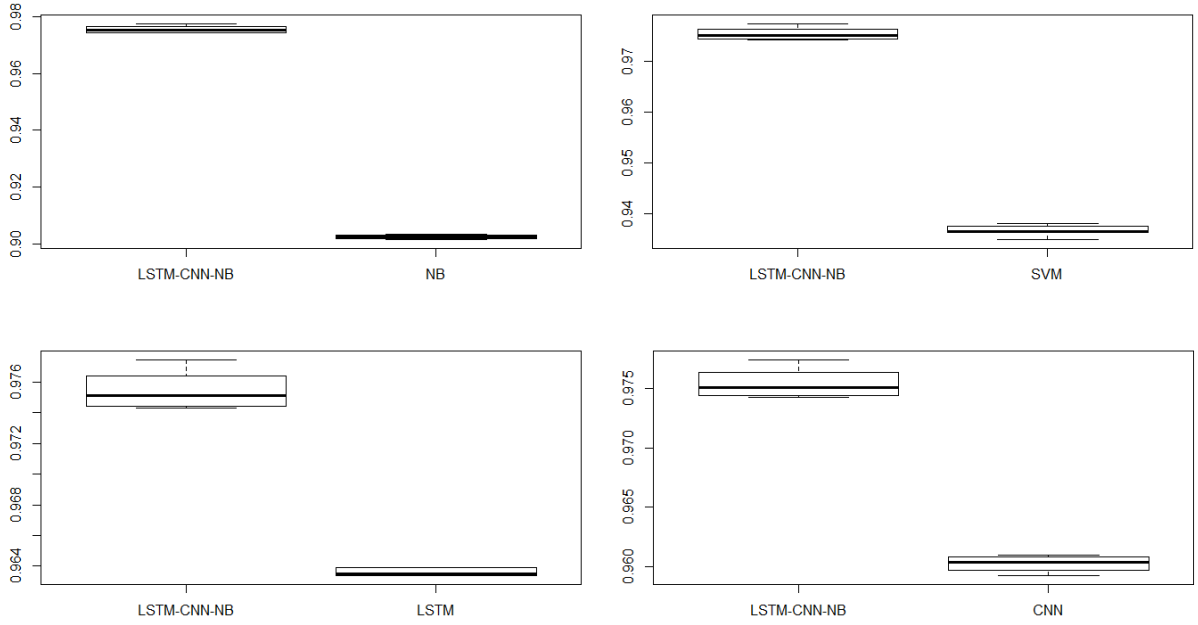

Figura 46 – Comparação da distribuição das acurácias entre LSTM-CNN-NB e os classificadores únicos para o Yelp 2015 binarizado

O modelo do NB também teve seu desempenho comparado ao desempenho dos outros modelos, tendo em vista que o NB apresentou o menor tempo de execução para classificação da base Yelp 2015 binarizada. Os resultados dos testes estatísticos estão detalhados na Tabela 55.

Tabela 55 – Comparação da performance entre NB e demais classificadores para o Yelp 2015 binarizado

classificador	**teste**	***p-value***	***p-value*** $< \alpha$	**resultado**
NBxSVM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM	*Wilcoxon*	0.0001727	sim	rejeita H_0
NBxCNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxSVM-NB	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-SVM	*t-Student*	2.2e-16	sim	rejeita H_0
NBxLSTM-CNN-NB	*t-Student*	2.2e-16	sim	rejeita H_0

O NB apresentou desempenho diferente em relação a todos os modelos de classificadores. Os *box-plots* das figuras 47 e 48 analisam se essa diferença, apontada pelos testes estatísticos, pode indicar superioridade de desempenho do NB em relação a algum classificador.

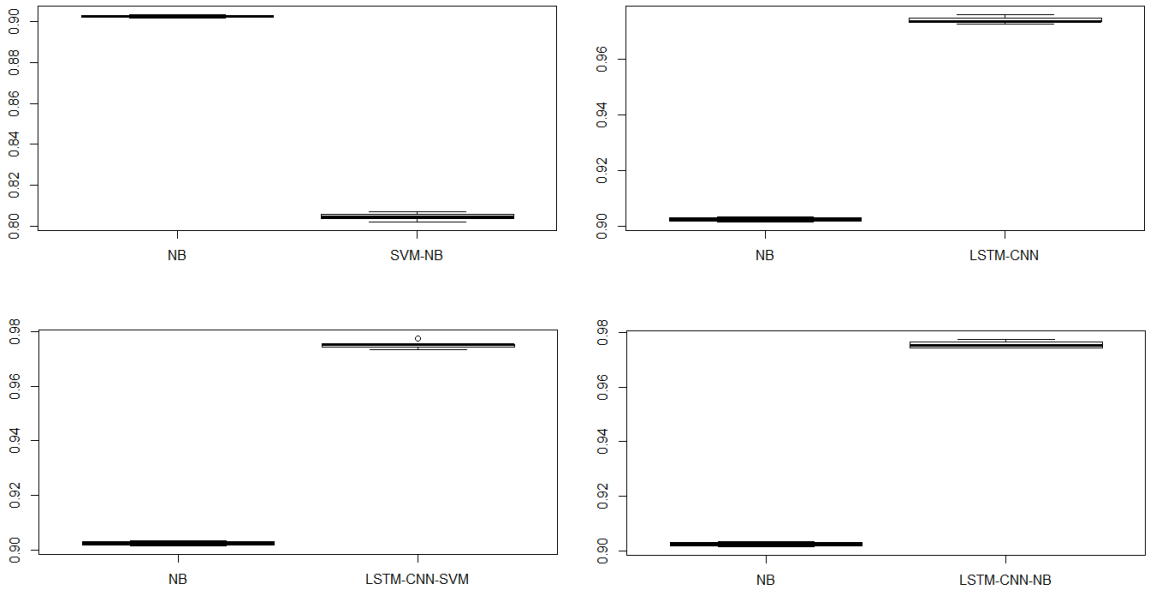

Figura 47 – Comparação da distribuição das acurácias entre NB e os comitês para o Yelp 2015 binarizado

Como é possível analisar nas figuras 47 e 48, a distribuição das amostras do NB está em um campo inferior em relação a todos os modelos. Mesmo esse classificador sendo o mais rápido, em termos de desempenho figura como o menos eficaz.

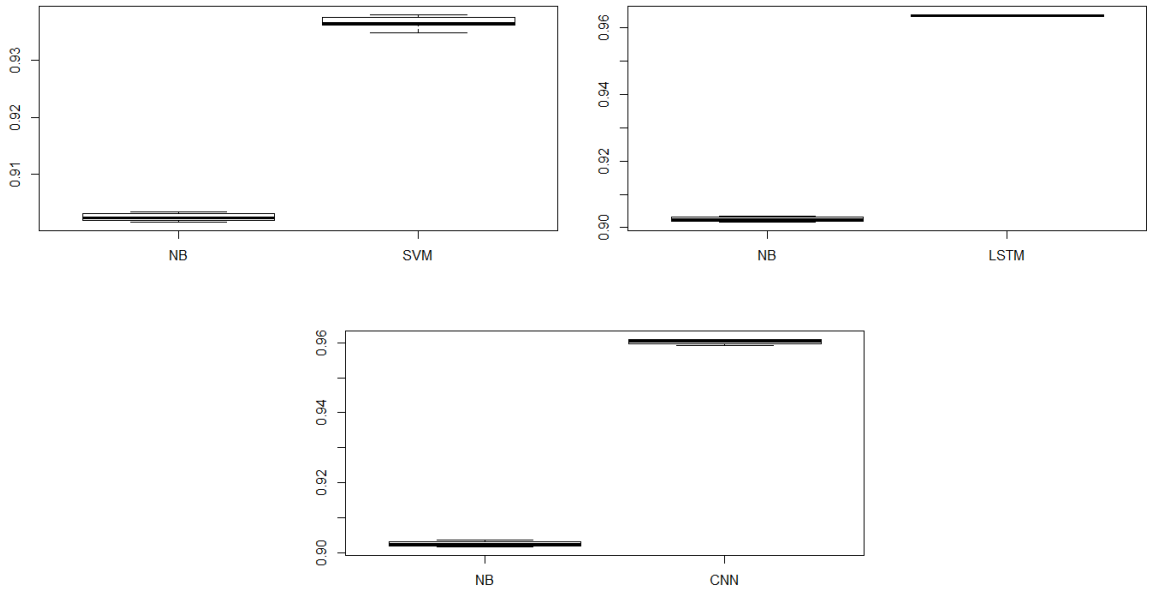

Figura 48 – Comparação da distribuição das acurácias entre NB os classificadores únicos para o Yelp 2015 binarizado

A comparação entre o desempenho da combinação LSTM-CNN-NB e do NB em relação a cada polaridade de sentimentos, individualmente, por meio das métricas *precision*, *recall* e *f-score*, é apresentada na Tabela 56.

Tabela 56 – Desempenho da LSTM-CNN-NB e do NB de acordo com cada polaridade de sentimento para o Yelp 2015 binarizado

	LSTM-CNN-SVM/NB	
	negativa	**positiva**
precision	0.97 / 0.78	0.98 / 0.82
recall	0.98 / 0.84	0.97 / 0.76
f-score	0.98 / 0.81	0.81 / 0.78

O NB se mostra um classificador inferior em relação a todas as polaridades de sentimentos, seus valores de *precision*, *recall* e *f-score* possuem acertividade entre 70% e 80%, enquanto o comitê apresenta acertividade entre 97% e 98%. Assim, mesmo com um tempo de execução muito superior ao tempo dos classificadores de AM tradicional, a combinação entre algoritmos de aprendizado de máquina e aprendizado profundo, LSTM-CNN-NB, se mostra como melhor alternativa para o problema abordado.

5.6 CONSIDERAÇÕES DO CAPÍTULO

Neste Capítulo foram detalhados os experimentos e dicussões acerca dos resultados obtidos de acordo com cada base de dados. Também são apresentados resultados recentes

presentes na literatura para tais bases. Alguns trabalhos na literatura apresentam desempenho superior a esta pesquisa para as bases IMDb *Review* (LIU; LAPATA, 2018), Yelp 2013 , Yelp 2014 (GONG et al., 2018) e Yelp 2015 (RAO et al., 2018), no entanto não são indicados valores de desvio padrão ou demais testes estatísticos que possam comprovar a diferença estatística das acurácias apontadas. No caso da base SSTb, um comitê formulado nesta pesquisa demonstra desempenho superior em relação a pesquisas recentes, como em Han, Bai e Liu (2018). Os valores médios de acurácia encontrados por cada modelo construído nesta pesquisa estão dispostos na Tabela 49, de modo a proporcionar uma análise sistêmica dos resultados.

Como é possível observar por meio da Tabela 49, no caso dos experimentos com múltiplas classes, a rede CNN obteve desempenho estatisticamente superior em três dos cinco experimentos, que foram com as bases de dados IMDb, Yelp 2013 e Yelp 2015. O comitê LSTM-CNN-SVM obteve melhor desempenho estatístico na base de dados SSTb, enquanto o comitê LSTM-CNN alcançou os melhores resultados para a base de dados Yelp 2014, também com diferença estatísta.

No caso dos experimentos binários, o destaque ficou para a combinação LSTM-CNN-NB, que atingiu os melhores resultados em três dos cinco experimentos (SSTb, Yelp 2014 e Yelp 2015). No caso das bases SSTb e Yelp 2014 o LSTM-CNN-NB atingiu superioridade estatística com relação a todos os demais modelos, exceto com relação a combinação LSTM-CNN. Ao se tratar da base Yelp 2015, o LSTM-CNN-NB atingiu desempenho estatístico superior aos demais modelos, exceto com relação ao LSTM-CNN-SVM. Nos experimentos binários para as bases IMDb e Yelp 2013, a combinação LSTM-CNN-SVM também obteve destaque ao atingir maior valor de acurácia. Ao analisar os resultados, percebe-se que o comitê LSTM-CNN-SVM apresenta desempenho estatisticamente superior a todos os modelos, exceto com relação aos modelos LSTM-CNN e LSTM-CNN-NB, tanto para a base do IMDb quanto para o Yelp 2013.

Observou-se que o algoritmo de AM tradicional, NB, superou todos os outros algoritmos em termos de menor tempo de execução, em todos os experimentos. Porém, quando o desempenho do classificador é a variável observada, os algoritmos de AM tradicional não obtém destaque, incluindo o NB. Os melhores modelos no que se refere a desempenho são: a rede de aprendizado profundo CNN e os comitês que possuem algoritmos de aprendizado profundo, sendo eles: LSTM-CNN, LSTM-CNN-SVM e LSTM-CNN-NB.

Os experimentos que envolvem combinação de classificadores com algoritmos de aprendizado profundo utilizaram redes com um menor número de épocas, em relação aos classificadores únicos. Percebeu-se que mesmo diminuindo o número de épocas, as combinações não perderam desempenho, atingindo resultados sempre acima dos classificadores de AM tradicional, da rede LSTM e resultados próximos ou acima da rede CNN. Além de diminuir consideravelmente o tempo de execução dos modelos, principalmente em relação ao tempo de execução da LSTM.

	acurácia (%)									
	IMDb	IMDb binário	SSTb	SSTb binário	Yelp 2013	Yelp 2013 binário	Yelp 2014	Yelp 2014 binário	Yelp 2015	Yelp 2015 binário
NB	40.24	89.84	41.48	89.04	40.72	83.63	41.21	89.47	43.19	90.25
SVM	38.82	93.40	40.67	86.15	39.58	93.81	41.68	93.65	44.98	93.65
LSTM	25.44	90.44	43.13	90.04	23.94	96.08	53.19	91.52	52.61	96.35
CNN	**43.99**	91.42	41.88	90.43	**57.82**	94.28	57.39	95.61	**60.22**	96.02
SVM-NB	39.87	89.46	39.62	89.46	39.28	79.64	40.44	79.32	43.03	80.04
LSTM-CNN	35.55	90.42	40.86	90.42	36.28	96.45	**58.79**	97.26	53.92	97.39
LSTM-CNN-SVM	39.83	**94.52**	**49.28**	90.23	34.46	**96.54**	51.56	97.20	53.69	97.49
LSTM-CNN-NB	39.40	94.44	40.19	**90.62**	34.35	96.49	50.53	**97.36**	52.72	**97.54**

Figura 49 – Desempenho dos modelos em relação a cada base de dados

Em seis dentre os dez experimentos realizados, os comitês que combinam algoritmos de AM tradicional e aprendizado profundo (com menor número de épocas) alcançaram os

melhores desempenhos, se mostrando uma proposta interessante para a tarefa de AS.

6 CONCLUSÃO

Nesta dissertação foi abordada a tarefa de análise de sentimentos automática através do contexto de aprendizado de máquina, aprendizado profundo e combinação de classificadores. Foram utilizadas técnicas existentes na literatura como os algoritmos de AM tradicional SVM e *Naive Bayes* e as redes de aprendizado profundo, LSTM e CNN.

Visando a melhoria do desempenho dos modelos únicos, foram construídos comitês de classificadores. Um modelo que une o conhecimento de ambos os métodos se apresentou como uma alternativa interessante para gerar modelos com maior acurácia em AS. A arquitetura proposta fez uso de redes de aprendizado profundo com um menor número de épocas de treinamento, para minimizar o tempo de execução dos comitês.

Análises com fins comparativos foram realizadas entre as combinações contruídas e os classificadores únicos para verificar a viabilidade da arquitetura. Para tal, foram utilizadas cinco bases de dados: IMDb *Review*, SSTb, Yelp 2013, Yelp 2014 e Yelp 2015. Para analisar os resultados de cada base de dados foi utilizado um conjunto de métricas de desempenho e testes estatísticos, que verificaram se há diferença com relevância estatística entre os desempenhos dos modelos.

Foram realizados dois tipos de experimentos, o primeiro com múltiplas classes (muito negativa, negativa, neutra, positiva ou muito positiva) e o segundo, binário, apenas com as polaridades de sentimentos positiva ou negativa. Entre testes binários e com múltiplas classes, foram realizados um total de dez experimentos. Em três experimentos a rede CNN alcançou os melhores resultados estatisticamente; em um experimento o melhor resultado, com relevância estatística, foi atingido por um comitê que combinou apenas redes de aprendizado profundo e; em seis experimentos os melhores desempenhos foram obtidos por modelos que combinam classificadores de AM tradicional e aprendizado profundo. A rede de aprendizado profundo CNN fez parte de todos os modelos que apresentaram os melhores desempenhos nos dez experimentos realizados, se mostrando um classificador interessante para estratégias de combinações.

O método proposto nesta dissertação, que combina classificadores de AM tradicional e aprendizado profundo, com uma menor quantidade de épocas de treinamento, apresentou bons resultados na maior parte dos experimentos, se mostrando uma alternativa viável para futuras pesquisas científicas e aplicações práticas. Para nortear o desenvolvimento desta dissertação foram elaboradas quatro questões de pesquisa:

1. Algoritmos de aprendizado profundo possuem desempenho superior aos algoritmos tradicionais de AM ao considerar o contexto das bases utilizadas?

   Com base nos resultados obtidos, observou-se que os melhores desempenhos foram atingidos por algoritmos de aprendizado profundo ou por comitês com classificadores

de aprendizado profundo, sendo estes superior em relação aos métodos tradicionais em todos os experimentos realizados.

2. Em quais situações a combinação de classificadores se mostrou uma melhor solução em relação a um classificador único?

   Os comitês se mostraram boas alternativas para a tarefa de AS, obtendo os melhores desempenhos nos experimentos binários. Nos experimentos com múltiplas classes, a rede CNN obteve os melhores resultados em três dos cinco experimentos, enquanto comitês alcançaram os melhores resultados nos demais. O desempenho superior da CNN pode estar relacionado a facilidade que esta rede possui em detectar múltiplos padrões, ao disparar o resultado de cada convolução quando um padrão especial é detectado. Assim, o método de generalização desta rede se mostrou o mais eficiente dentre os experimentos realizados com múltiplas classes.

3. Os melhores comitês, para estas bases, combinam métodos de AM tradicional e Aprendizado Profundo ou advém da combinação de modelos do mesmo tipo?

   Os experimentos realizados apontam que os melhores modelos combinam classificadores de AM tradicional e aprendizado profundo. Comitês que combinam os dois métodos atingiram resultado superior aos demais modelos em seis dos dez experimentos realizados.

4. A combinação se mostrou uma solução interessante ao se observar tempo de execução e desempenho dos modelos?

   Os comitês que combinam algoritmos de aprendizado profundo possuem um tempo de execução acima dos classificadores únicos de AM tradicional, mas também possuem desempenho superior a estes classificadores. Ao comparar comitês aos modelos de aprendizado profundo, as combinações possuem menor tempo de execução em relação a rede LSTM e tempo de execução maior em relação a rede CNN.

A partir dos experimentos e das questões de pesquisa analisadas, se torna possível destacar como contribuições desta dissertação dois pontos principais:

- Um método para geração de comitês que combina modelos de AM tradicional e aprendizado profundo com um menor número de épocas treinamento. Ao diminuir o número de épocas os modelos minimizaram seu tempo de execução, ao passo em que foi possível elevar o desempenho das combinações em relação aos classificadores únicos na maior parte dos experimentos (7/10).

- Análises de modelos de classificadores únicos e comitês para a tarefa de AS em problemas binários e com múltiplas classes. Os experimentos realizados apontam a rede profunda CNN como uma boa alternativa para os problemas com múltiplas

classes. No caso de problemas binários, comitês que combinam AM tradicional e aprendizado profundo atingiram os melhores resultados.

Esta pesquisa tem como limitação a quantidade de bases de dados utilizada, devido ao tempo de execução e análise de resultados só foram consideradas 5 bases de dados. Assim, o comportamento obtido por cada modelo nestes experimentos pode não se refletir em outras bases. Outras técnicas de Sistemas de Múltiplos Classificadores poderiam ter sido adotadas para construção dos comitês, mas também por questões de tempo de execução e análise de resultados não foram consideradas nesta pesquisa.

Como trabalhos futuros espera-se explorar a influência das técnicas de vetorização de textos para o desempenho dos diferentes tipos de classificadores. Espera-se também realizar uma análise mais detalhada da rede CNN que figura entre os melhores modelos em todos os experimentos. Além disto, mostra-se relevante trabalhar no desenvolvimento de comitês que levam em conta apenas a rede CNN, porém com diferentes configurações de parâmetros. Por fim, acredita-se que a utilização de diferentes técnicas de seleção de classificadores para a construção das combinações é um ponto que deve ser considerado em pesquisas futuras.

REFERÊNCIAS

AIRES, J. P.; PADILHA, C.; QUEVEDO, C.; MENEGUZZI, F. A deep learning approach to classify aspect-level sentiment using small datasets. In: IEEE. *Int Joint Conference on Neural Networks (IJCNN)*. RJ, Brasil: IEEE, 2018. p. 1–8. ISBN 978-1-5090-6014-6.

AKHTAR, M. S.; GUPTA, D.; EKBAL, A.; BHATTACHARYYA, P. Feature selection and ensemble construction: A two-step method for aspect based sentiment analysis. *Knowledge-Based Systems*, Elsevier, v. 125, p. 116–135, 2017.

AL-SHALABI, R.; OBEIDAT, R. Improving knn arabic text classification with n-grams based document indexing. In: *Int Conference on Informatics and Systems, Cairo, Egypt*. [S.l.: s.n.], 2008. p. 108–112.

ALBELWI, S.; MAHMOOD, A. A framework for designing the architectures of deep convolutional neural networks. *Entropy*, Multidisciplinary Digital Publishing Institute, v. 19, n. 6, p. 242, 2017.

ALMEIDA, L. M. *Construção de sistemas de múltiplos classificadores por meio de hibridização e otimização de técnicas de agrupamento e classificação de dados*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2011.

ANGELIDIS, S.; LAPATA, M. Multiple instance learning networks for fine-grained sentiment analysis. *Transactions of the Association of Computational Linguistics*, MIT Press, v. 6, p. 17–31, 2018.

ARNOLD, T.; TILTON, L. Natural language processing. In: *Humanities Data in R*. [S.l.]: Springer, 2015. p. 131–155.

BATISTA, G. E. d. A. P. et al. *Pré-processamento de dados em aprendizado de máquina supervisionado*. Tese (Doutorado) — Universidade de São Paulo, 2003.

BOTTOU, L. Large-scale machine learning with stochastic gradient descent. In: *Proceedings of COMPSTAT'2010*. [S.l.]: Springer, 2010. p. 177–186.

BREIMAN, L. Bagging predictors. *Machine learning*, Springer, v. 24, n. 2, p. 123–140, 1996.

BROWNE, M. W. Cross-validation methods. *Journal of mathematical psychology*, Elsevier, v. 44, n. 1, p. 108–132, 2000.

BRYLL, R.; GUTIERREZ-OSUNA, R.; QUEK, F. Attribute bagging: improving accuracy of classifier ensembles by using random feature subsets. *Pattern recognition*, Elsevier, v. 36, n. 6, p. 1291–1302, 2003.

CAMBRIA, E. Affective computing and sentiment analysis. *IEEE Intelligent Systems*, IEEE, v. 31, n. 2, p. 102–107, 2016.

CARVALHO, C. M. A. *Estudo comparativo de análise de sentimentos aplicado à notícias públicas*. Dissertação (Mestrado) — Universidade Federal do Maranhão, 2018.

CECI, F.; ALVAREZ, G. M.; GONÇALVES, A. L. Análise de sentimento e mineração de opinião: uma revisão bibliométrica da literatura. *Revista Spacios*, v. 38, n. 14, 2017.

CHAPELLE, O.; SCHÖLKOPF, B.; ZIEN, A. *Semi-supervised learning, ser. Adaptive computation and machine learning*. [S.l.]: Cambridge, MA: The MIT Press, 2006.

CHAWLA, N.; JAPKOWICZ, N.; KOTCZ, A. Special issue on learning from imbalanced datasets. *ACM SIGKDD Explorations Newsletter*, ACM, New York, NY, USA, v. 6, n. 1, p. 1–6, jun. 2004.

CHEN, H.; SUN, M.; TU, C.; LIN, Y.; LIU, Z. Neural sentiment classification with user and product attention. In: *Proceedings of the 2016 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2016. p. 1650–1659.

CHEN, S.; PENG, C.; CAI, L.; GUO, L. A deep neural network model for target-based sentiment analysis. In: IEEE. *2018 International Joint Conference on Neural Networks (IJCNN)*. [S.l.]: IJCNN, 2018. p. 1–7.

CHEN, T.; XU, R.; HE, Y.; WANG, X. A gloss composition and context clustering based distributed word sense representation model. *Entropy*, Multidisciplinary Digital Publishing Institute, v. 17, n. 9, p. 6007–6024, 2015.

CHEN, Y.; ZHOU, B.; ZHANG, W.; GONG, W.; SUN, G. Sentiment analysis based on deep learning and its application in screening for perinatal depression. In: IEEE. *IEEE International Conference on Data Science in Cyberspace (DSC)*. [S.l.], 2018. p. 451–456.

CHIONG, R.; FAN, Z.; HU, Z.; ADAM, M. T.; LUTZ, B.; NEUMANN, D. A sentiment analysis-based machine learning approach for financial market prediction via news disclosures. In: ACM. *Proceedings of the Genetic and Evolutionary Computation Conference Companion*. [S.l.], 2018. p. 278–279.

DEY, L.; CHAKRABORTY, S.; BISWAS, A.; BOSE, B.; TIWARI, S. Sentiment analysis of review datasets using naive bayes and k-nn classifier. *Int Journal of Information Engineering and Electronic Business*, v. 8, n. 4, p. 54–62, 2016.

DIAO, Q.; QIU, M.; WU, C.-Y.; SMOLA, A. J.; JIANG, J.; WANG, C. Jointly modeling aspects, ratings and sentiments for movie recommendation (jmars). In: ACM. *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. [S.l.], 2014. p. 193–202.

DIETTERICH, T. G. Ensemble methods in machine learning. In: SPRINGER. *International workshop on multiple classifier systems*. [S.l.], 2000. p. 1–15.

DIETTERICH, T. G.; BAKIRI, G. Solving multiclass learning problems via error-correcting output codes. *Journal of artificial intelligence research*, v. 2, p. 263–286, 1994.

DUAN, K.; KEERTHI, S. S.; CHU, W.; SHEVADE, S. K.; POO, A. N. Multi-category classification by soft-max combination of binary classifiers. In: SPRINGER. *International Workshop on Multiple Classifier Systems*. [S.l.], 2003. p. 125–134.

FERREIRA, M. A. D. *Uma Abordagem para Extração Automática de Learning Analytics Relacionados à Colaboração em Fóruns Educaionais*. [S.l.]: Dissertacão (Mestrado em Informática Aplicada) - Universidade Federal Rural de Pernambuco, 2018.

FERRI, C.; HERNÁNDEZ-ORALLO, J.; MODROIU, R. An experimental comparison of performance measures for classification. *Pattern Recognition Letters*, Elsevier, v. 30, n. 1, p. 27–38, 2009.

FILHO, L. V. S. *Uma arquitetura para combinação de classificadores otimizada por métodos de poda com aplicação em credit scoring*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2014.

FRANZ, M.-L. V.; HILLMAN, J. A tipologia de jung. *Trad.: Adail Ubirajara Sobral. São*, 1990.

GHASEMI, A.; ZAHEDIASL, S. Normality tests for statistical analysis: a guide for non-statisticians. *International journal of endocrinology and metabolism*, Kowsar Medical Institute, v. 10, n. 2, p. 486, 2012.

GONG, J.; QIU, X.; WANG, S.; HUANG, X. Information aggregation via dynamic routing for sequence encoding. *ACL - Association for Computational Linguistics*, 2018.

GOOGLE. Ai for everyone: inside tensorflow, our open-source machine learning platform. 2018. Disponível em: https://ai.google/stories/tensorflow/, Accessed: 2018-12-10. Disponível em: <https://ai.google/stories/tensorflow/>.

GRANVILLE, E. B. My life as a mathematician. *Sage: A Scholarly Journal of Black Women*, American Association for the Advancement of Science, v. 6, n. 2, 1989.

GROSSBERG, S. Recurrent neural networks. *Scholarpedia*, v. 8, n. 2, p. 1888, 2013.

GUERREIRO, L. *Aprendizado semi-supervisionado utilizando modelos de caminhada de partículas em grafos*. Dissertação (Mestrado) — Universidade Estadual Paulista (UNESP), 2017.

GUPTA, M.; BAKLIWAL, A.; AGARWAL, S.; MEHNDIRATTA, P. A comparative study of spam sms detection using machine learning classifiers. In: IEEE. *Eleventh Int Conference on Contemporary Computing (IC3)*. [S.l.], 2018. p. 1–7.

HAN, H.; BAI, X.; LIU, J. Attention-based resnet for chinese text sentiment classification. In: ATLANTIS PRESS. *2018 Int Conference on Computer Science, Electronics and Communication Engineering (CSECE)*. [S.l.], 2018.

HERNÁNDEZ-LOBATO, J. M.; HOFFMAN, M. W.; GHAHRAMANI, Z. Predictive entropy search for efficient global optimization of black-box functions. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2014. p. 918–926.

HUTTO, C. J.; GILBERT, E. Vader: A parsimonious rule-based model for sentiment analysis of social media text. In: *Int AAAI conference on weblogs and social media*. [S.l.: s.n.], 2014.

KISS, J. Facebook hits 1 billion users a month (retrieved february 2013). *The Guardian*, 2013.

KUNCHEVA, L. I. Switching between selection and fusion in combining classifiers: An experiment. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 32, n. 2, p. 146–156, 2002.

KUNCHEVA, L. I. *Combining pattern classifiers: methods and algorithms*. [S.l.]: Wiley-Interscience, Hoboken, NJs, 2004.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: *International Conference on Machine Learning*. [S.l.: s.n.], 2014. p. 1188–1196.

LECUN, Y.; BENGIO, Y.; HINTON, G. Deep learning. *nature*, Nature Publishing Group, v. 521, n. 7553, p. 436, 2015.

LEI, Z.; YANG, Y.; YANG, M. Saan: A sentiment-aware attention network for sentiment analysis. In: ACM. *Int ACM SIGIR Conference on Research & Development in Information Retrieval*. [S.l.], 2018. p. 1197–1200.

LIMA, T. P. F. d. *Sistema híbrido inteligente para geração, seleção e combinação de classificadores*. Tese (Doutorado) — Universidade Federal de Pernambuco, 2017.

LIU, B. Sentiment analysis and opinion mining. *Synthesis lectures on human language technologies*, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–167, 2012.

LIU, X.-Y.; WU, J.; ZHOU, Z.-H. Exploratory undersampling for class-imbalance learning. *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, IEEE, v. 39, n. 2, p. 539–550, 2009.

LIU, Y.; LAPATA, M. Learning structured text representations. *Transactions of the Association of Computational Linguistics*, MIT Press, v. 6, p. 63–75, 2018.

LO, R. T.-W.; HE, B.; OUNIS, I. Automatically building a stopword list for an information retrieval system. In: *Journal on Digital Information Management: Special Issue on the 5th Dutch-Belgian Information Retrieval Workshop (DIR)*. [S.l.: s.n.], 2005. v. 5, p. 17–24.

LOZHNIKOV, N.; DERCZYNSKI, L.; MAZZARA, M. Stance prediction for russian: Data and analysis. In: SPRINGER. *International Conference in Software Engineering for Defence Applications*. [S.l.]: Springer, 2018. ISBN 978-3-030-14687-0.

MCAULEY, J.; LESKOVEC, J. Hidden factors and hidden topics: understanding rating dimensions with review text. In: ACM. *Proceedings of the 7th ACM conference on Recommender systems*. [S.l.], 2013. p. 165–172.

MEDHAT, W.; HASSAN, A.; KORASHY, H. Sentiment analysis algorithms and applications: A survey. *Ain Shams Engineering Journal*, Elsevier, v. 5, n. 4, p. 1093–1113, 2014.

MICHIE, D.; SPIEGELHALTER, D. J.; TAYLOR, C. et al. Machine learning. *Neural and Statistical Classification*, Technometrics, v. 13, 1994.

MORDELET, F.; VERT, J.-P. A bagging svm to learn from positive and unlabeled examples. *Pattern Recognition Letters*, Elsevier, v. 37, p. 201–209, 2014.

PANG, B.; LEE, L. Opinion mining and sentiment analysis. *Foundations and Trends® in Information Retrieval*, Now Publishers, Inc., v. 2, n. 1–2, p. 1–135, 2008.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up?: sentiment classification using machine learning techniques. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*. [S.l.], 2002. p. 79–86.

PINHEIRO, R. H.; CAVALCANTI, G. D.; TSANG, R. Combining binary classifiers in different dichotomy spaces for text categorization. *Applied Soft Computing*, Elsevier, v. 76, p. 564–574, 2019.

PLISSON, J.; LAVRAC, N.; MLADENIć, D. A rule based approach to word lemmatization. In: *Proceedings of IS-2004*. [S.l.: s.n.], 2004. p. 83–86.

PORIA, S.; CAMBRIA, E.; GELBUKH, A. Aspect extraction for opinion mining with a deep convolutional neural network. *Knowledge-Based Systems*, Elsevier, v. 108, p. 42–49, 2016.

PORIA, S.; PENG, H.; HUSSAIN, A.; HOWARD, N.; CAMBRIA, E. Ensemble application of convolutional neural networks and multiple kernel learning for multimodal sentiment analysis. *Neurocomputing*, Elsevier, v. 261, p. 217–230, 2017.

RAO, G.; HUANG, W.; FENG, Z.; CONG, Q. Lstm with sentence representations for document-level sentiment classification. *Neurocomputing*, Elsevier, 2018.

RISH, I. An empirical study of the naive bayes classifier. In: IBM NEW YORK. *IJCAI 2001 workshop on empirical methods in artificial intelligence*. [S.l.], 2001. v. 3, n. 22, p. 41–46.

ROLI, F.; GIACINTO, G.; VERNAZZA, G. Methods for designing multiple classifier systems. In: SPRINGER. *International Workshop on Multiple Classifier Systems*. [S.l.], 2001. p. 78–87.

RUIZ-GONZALEZ, R.; GOMEZ-GIL, J.; GOMEZ-GIL, F.; MARTÍNEZ-MARTÍNEZ, V. An svm-based classifier for estimating the state of various rotating components in agro-industrial machinery with a vibration signal acquired from a single point on the machine chassis. *Sensors*, Multidisciplinary Digital Publishing Institute, v. 14, n. 11, p. 20713–20735, 2014.

SAN, A. Random decision syntax trees at semeval-2018 task 3: Lstms and sentiment scores for irony detection. In: *Proceedings of The Int Workshop on Semantic Evaluation*. [S.l.: s.n.], 2018. p. 560–564.

SEBASTIANI, F. Machine learning in automated text categorization. *ACM computing surveys (CSUR)*, ACM, v. 34, n. 1, p. 1–47, 2002.

SENA, N. M. Espaço público, opinião e democracia. In: *Estudos em Comunicação*. [S.l.]: Instituto Superior de Ciências Sociais e Políticas - Universidade Técnica de Lisboa, 2007.

SILVA, E. G. d. *Previsão de séries temporais usando sistemas de múltiplos preditores*. Dissertação (Mestrado) — Universidade Federal de Pernambuco, 2017.

SILVA, N. F. F. d. *Análise de sentimentos em textos curtos provenientes de redes sociais*. Tese (Doutorado) — Universidade de São Paulo, 2016.

SILVA, N. F. F. D.; COLETTA, L. F.; HRUSCHKA, E. R. A survey and comparative study of tweet sentiment analysis via semi-supervised learning. *ACM Computing Surveys (CSUR)*, ACM, v. 49, n. 1, p. 15, 2016.

SOCHER, R.; PERELYGIN, A.; WU, J.; CHUANG, J.; MANNING, C. D.; NG, A.; POTTS, C. Recursive deep models for semantic compositionality over a sentiment treebank. In: *Proceedings of the 2013 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2013. p. 1631–1642.

SUNDERMEYER, M.; SCHLÜTER, R.; NEY, H. Lstm neural networks for language modeling. In: *Annual conference of the international speech communication association*. [S.l.: s.n.], 2012.

TABOADA, M.; BROOKE, J.; TOFILOSKI, M.; VOLL, K.; STEDE, M. Lexicon-based methods for sentiment analysis. *Computational linguistics*, MIT Press, v. 37, n. 2, p. 267–307, 2011.

TANG, D.; QIN, B.; LIU, T. Document modeling with gated recurrent neural network for sentiment classification. In: *Proceedings of the 2015 conference on empirical methods in natural language processing*. [S.l.: s.n.], 2015. p. 1422–1432.

TANG, D.; QIN, B.; LIU, T. Learning semantic representations of users and products for document level sentiment classification. In: *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. [S.l.: s.n.], 2015. v. 1, p. 1014–1023.

TARDE, G. *A opinião e a Multidão*. [S.l.]: Europa-América, Biblioteca Universitária, 1991.

TURNEY, P. D. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 40th annual meeting on association for computational linguistics*. [S.l.], 2002. p. 417–424.

VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer science & business media, 2013.

XU, J.; CHEN, D.; QIU, X.; HUANG, X. Cached long short-term memory neural networks for document-level sentiment classification. *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing - EMNLP*, Association for Computational Linguistics, p. 1660—-1669, 2016.

YANG, Z.; YANG, D.; DYER, C.; HE, X.; SMOLA, A.; HOVY, E. Hierarchical attention networks for document classification. In: *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. [S.l.: s.n.], 2016. p. 1480–1489.

YELP dataset challenge 2013. 2013. <https://www.yelp.com/dataset/challenge>. Accessed: 10-01-2018.

YELP dataset challenge 2014. 2014. <https://www.yelp.com/dataset/challenge>. Accessed: 10-01-2018.

YELP dataset challenge 2015. 2015. <https://www.yelp.com/dataset/challenge>. Accessed: 10-01-2018.

ZHAI, S.; ZHANG, Z. M. Semisupervised autoencoder for sentiment analysis. In: *Association for the Advancement of Artificial Intelligence - AAAI*. [S.l.: s.n.], 2016. p. 1394–1400.

ZHANG, L.; WANG, S.; LIU, B. Deep learning for sentiment analysis: A survey. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, Wiley Online Library, p. e1253, 2018.

ZHANG, Y.; JIN, R.; ZHOU, Z.-H. Understanding bag-of-words model: a statistical framework. *International Journal of Machine Learning and Cybernetics*, Springer, v. 1, n. 1-4, p. 43–52, 2010.

ZHU, W.; ZHANG, W.; LI, G.-Z.; HE, C.; ZHANG, L. A study of damp-heat syndrome classification using word2vec and tf-idf. In: IEEE. *2016 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*. [S.l.], 2016. p. 1415–1420.