Universidade do Estado do Rio de Janeiro
Centro de Tecnologia e Ciências
Faculdade de Engenharia

Igor Pedro Pinto dos Santos

Análise de Sentimentos Usando Redes Neurais de Convolução

Rio de Janeiro
2017

Igor Pedro Pinto dos Santos

Análise de Sentimentos Usando Redes Neurais de Convolução

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Orientadora: Prof.^{a} Dr.^{a} Nadia Nedjah
Coorientadora: Prof.^{a} Dr.^{a} Luiza de Macedo Mourelle

Rio de Janeiro
2017

CATALOGAÇÃO NA FONTE

UERJ / REDE SIRIUS / BIBLIOTECA CTC/B

S237 Santos, Igor Pedro Pinto dos.
Análise de sentimentos usando redes neurais de convolução / Igor Pedro Pinto dos Santos. – 2017.
124f.

Orientador: Nadia Nedjah.
Coorientador: Luiza de Macedo Mourelle.
Dissertação (Mestrado) – Universidade do Estado do Rio de Janeiro, Faculdade de Engenharia.

1. Engenharia Eletrônica - Teses. 2. Redes neurais (Computação) - Teses. 3. Processamento da linguagem natural (Computação) - Teses. 4. Emoções - Teses. I. Nedjah, Nadia. II. Mourelle, Luiza de Macedo. III. Universidade do Estado do Rio de Janeiro. IV. Título.

CDU 681.3

Assinatura

Data

Igor Pedro Pinto dos Santos

Análise de Sentimentos Usando Redes Neurais de Convolução

Dissertação apresentada, como requisito parcial para obtenção do título de Mestre, ao Programa de Pós-Graduação em Engenharia Eletrônica, da Universidade do Estado do Rio de Janeiro. Área de concentração: Sistemas Inteligentes e Automação.

Aprovado em: 14 de agosto de 2017.

Banca Examinadora:

Prof.[a] Dr.[a] Nadia Nedjah (Orientadora)
Faculdade de Engenharia - UERJ

Prof.[a] Dr.[a] Luiza de Macedo Mourelle (Coorientadora)
Faculdade de Engenharia - UERJ

Prof. Dr. Luiz Eduardo Soares de Oliveira
Universidade Federal do Paraná - UFPR

Prof.[a] Dr.[a] Leila Weitzel Coelho da Silva
Universidade Federal Fluminense - UFF

Rio de Janeiro
2017

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo dom da vida e por suas graças.

Agradeço à minha família por ser meu esteio. À minha mãe, Tereza, por ser amorosa e preocupada comigo. Ao meu pai, Adilson, por ser minha inspiração e exemplo pessoal de resiliência. Ao meu irmão, Iuri, por ser um amigo incrível que sempre me faz sorrir. Ao meu irmão, Ives, por sua generosidade. Ao meu primo, João Paulo, por sua grandeza e por ser o irmão mais velho que a vida me deu.

Agradeço à minha avó, Urania, que infelizmente partiu durante essa minha jornada. Obrigado por sempre me incentivar e torcer pelo meu sucesso. Obrigado por ter sido uma mãe para mim.

Agradeço às minhas professoras pelas lições aprendidas nestes dois anos de curso. Agradeço à professora Luiza de Macedo Mourelle pela sua gentileza e preocupação com os alunos. Agradeço à professora Nadia Nedjah que soube ser dura quando foi necessário, mas também soube ser extremamente compreensiva e generosa quando eu precisei.

Agradeço aos meus queridos amigos Jean, Patrícia e Paulo por me ajudarem quando mais precisei deles.

Agradeço à minha terapeuta, Ilda, por ser uma grande amiga e incentivadora. Agradeço por seus conselhos e dicas que foram extremamente valiosos para mim.

Agradeço aos colegas de mestrado Alexandre, Pedro, Luneque, Tarso, Luigi, Reinaldo e Joelmir. Obrigado pelas conversas sobre as dúvidas individuais e coletivas. Os momentos de descontração regidos por papos sobre futebol e política.

Agradeço em especial ao amor da minha vida, Daiana, que soube aguentar minhas ausências, minhas reclamações e os quilos extras que adquiri ao longo do mestrado. Também gostaria de agradece-lá por me incentivar e me dar forças a continuar na luta por este objetivo.

Agradeço à CAPES e ao PEL-UERJ pelos recursos investidos nesta pesquisa.

A educação é o nosso passaporte para o futuro, pois o amanhã pertence às pessoas que se preparam hoje.

Malcolm X

RESUMO

SANTOS, Igor Pedro Pinto dos *Análise de sentimentos usando redes neurais de convolução.* 2017. 132f. Dissertação (Mestrado em Engenharia Eletrônica) – Faculdade de Engenharia, Universidade do Estado do Rio de Janeiro, Rio de Janeiro, 2017.

As redes de neurais de convolução são conhecidas por seu ótimo desempenho em visão computacional, alcançando resultados no estado da arte. No entanto, trabalhos recentes mostraram que estas redes podem funcionar também para processamento de linguagem natural. Neste caso, a ideia básica consiste em concatenar as representações vetoriais das palavras em um único bloco e usá-lo como imagem. Contudo, apesar dos bons resultados, o problema de redes de convolução é o grande números de decisões de projeto que precisam ser tomadas. Estes modelos exigem a definição de muitos hiperparâmetros, incluindo o tipo *word embeddings* que consiste na representação vetorial dos dados, a função de ativação que introduz não-linearidade ao modelo, o tamanho do filtro que aplica convolução aos dados, o número de mapas de características que são responsáveis por identificar os atributos, o método de *pooling* usado na condensação dos dados, além da constante de regularização e a taxa de *dropout* que são responsáveis por evitar o sobreajuste da rede. Em trabalhos existentes, foram apresentadas arquiteturas de redes neurais de convolução capazes de superar o desempenho de modelos tradicionais de aprendizado de máquinas, competindo com modelos mais complexos. Todavia, não foi explorado como as diferentes possibilidades de hiperparâmetros podem afetar o desempenho deste tipo de rede. Nesta dissertação, o objetivo consiste em criar um classificador de análise de sentimentos eficiente usando redes neurais de convolução por meio da análise do impacto de seus hiperparâmetros no desempenho do modelo. O interesse por análise de sentimentos vem do advento das mídias sociais e dos avanços tecnológicos que inundam a Internet com opiniões. Os resultados encontrados foram obtidos com o uso de GPU e mostram que as diferentes configurações superam os modelos de referência em sua grande maioria com ganhos de até 18% e possuem desempenho semelhante aos modelos no estado da arte com ganhos de até 2% em alguns casos.

Palavras-chave: Rede Neural de Convolução. Análise de Sentimento. Hiperparâmetros. Processamento de Linguagem Natural.

ABSTRACT

Convolutional neural networks are known for their excellent performance in computer vision, achieving results in the state of the art. However, recent work has shown that these networks can also work for natural language processing. In this case, the basic idea is to concatenate the vector representations of words into a single block and use it as an image. However, despite the good results, the problem of convolution networks is the large numbers of design decisions that need to be made. These models require the definition of many hyperparameters, including the type of word embeddings, which consists of the vector representation of the data, the activation function that introduces nonlinearity to the model, the size of the filter that applies convolution to the data, the number of feature maps which are responsible for identifying the attributes, the pooling method used in data reduction, in addition to the regularization constant and the dropout rate, which are responsible for avoiding overfitting of the network. In existing works, convolutional neural network architectures capable of overcoming the performance of traditional machine learning models were presented and these can compete with more complex models. However, it has not been explored how the different possibilities of hyperparameters can affect the performance of this type of network. In this dissertation, the objective is to create an efficient sentiment analysis classifier using convolutional neural networks by analyzing the impact of their hyperparameters on model performance. The interest in analyzing sentiment comes from the advent of social media and the technological advances that flood the Internet with opinions. The results achieved were obtained with the use of GPU and show that the different configurations exceed the reference models in the majority of them with gains of up to 18% and have similar performance to the models in the state of the art with gains of up to 2% in some cases.

Keywords: Convolutional Neural Network. Sentiment Analysis. Hyperparameters. Natural Language Processing.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS

API	*Application Programming Interface*
AS	Análise de Sentimentos
BN	*Bi-gram Naive Bayes*
BOW	*Bag of Words*
CBOW	*Continuous Bag of Words*
CCAE	*Combinatorial Category Autoencoders with combinatorial category grammar operators*
CNN	*Convolution Neural Network*
CPU	*Central Processing Unit*
CR	Constante de Regularização
CUDA	*Compute Unified Device Architecture*
DO	Taxa de *Dropout*
DT	*Decision Tree*
fTD	*fastText* Dinâmico
fTE	*fastText* Estático
GPU	*Graphics Processing Unit*
GVD	*GloVe* Dinâmico
GVE	*GloVe* Estático
KNN	*K-Nearest Neighbors*
LDA	*Latent Dirichlet Allocation*
LSA	*Latent Semantic Analysis*
LSI	*Latent Semantic Indexing*
MLP	*MultiLayer Perceptron*
Mne	Classe Muito Negativa

MO	Mineração de Opiniões
Mpo	Classe Muito Positiva
MR	*Movie Reviews dataset*
MV-RNN	*Matrix-Vectore Recursive Neural Network*
NB	*Naive Bayes*
Neg	Classe Negativa
Neu	Classe Neutra
NLP	*Natural Language Processing*
PLSA	*Probabilistic Latent Semantic Analysis*
Pos	Classe Positiva
PSO	Particle Swarm Optimization
RAE	*Recursive Autoencoders*
RBF	*Radial Basis Function*
ReLU	*Rectified Linear Unit*
RNA	Rede Neural Artificial
RNTN	*Recursive Neural Tensor Network*
SOM	*Self Organizing Map*
SRN	*Simple Recurrent Network*
SST-1	*Stanford Sentiment Treebank dataset* de 5 classes
SST-2	*Stanford Sentiment Treebank dataset* de 2 classes
SVM	*Support Vector Machine*
Tanh	Tangente Hiperbólica
TPU	*Tensor Processing Unit*
W2VD	*Word2Vec* Dinâmico
W2VE	*Word2Vec* Estático

SUMÁRIO

INTRODUÇÃO

RÁPIDA proliferação de textos *online* faz com que seja impossível para um ser humano ter conhecimento das diferentes situações que o interessa em tempo hábil. Contudo, em alguns casos, apenas alguns aspectos do texto são relevantes como: sobre quem se fala e se a avaliação sobre quem se fala é boa ou ruim. Diante deste problema, foram desenvolvidas formas de avaliar o conteúdo escrito de forma rápida e automática. A análise de sentimentos (AS), também chamada de mineração de opiniões (MO), é o campo de estudo que analisa as opiniões de pessoas, sentimentos, avaliações, atitudes e emoções em relação a entidades como produtos, serviços, organizações, indivíduos e eventos.

O termo análise de sentimentos surgiu pela primeira vez em 2002 por Pang e Lee (PANG; LEE; VAITHYANATHAN, 2002). Trata-se de uma área multidisciplinar que abrange o processamento de linguagem natural (NLP), linguística e inteligência artificial para identificar, extrair e quantificar informações subjetivas de forma sistemática. A AS é amplamente aplicada em marketing, relações públicas, investimento, campanhas políticas e em muitas outras áreas.

A mineração de opiniões é um problema fascinante que permite traçar cenários políticos a partir de comentários em mídias sociais e prever a valorização de ativos na bolsa de valores usando notícias. Todavia, a MO também é extremamente subjetiva, o que a faz um problema extremamente desafiador. Grandes dificuldades também decorrem da constante evolução e transformação da linguagem humana e de aspectos inerentes da linguagem escrita como: anáforas e sarcasmo.

Apesar de todas as peculiaridades presentes na escrita humana, a análise de sentimentos é essencialmente um problema de classificação. O objetivo final de qualquer análise de sentimentos é classificar uma informação como positiva, neutra e negativa, ou em qualquer variação destas. Tendo isto em vista, a principal abordagem ao problema é feita usando aprendizado de máquinas. Modelos como *Naive Bayes* e *Support Vector Machine* foram usados ao longo dos anos para criar sistemas de classificação de sentimentos. Além destes, mo-

delos mais complexos oriundos do aprendizado profundo têm sido usados recentemente na tentativa de melhorar a acurácia de classificação entregue pelos modelos mais simples.

As redes neurais de convolução (CNN) utilizam camadas com filtros de convolução que são aplicadas aos dados (LECUN; BENGIO, 1998). Originalmente, foram criadas para trabalhar com visão computacional, mas as CNNs têm se mostrado efetivas e obtiveram excelentes resultados em tarefas de NLP. Em (KIM, 2014), o autor apresentou uma arquitetura de CNN que pode ser usada para classificar sentenças de diferentes conjuntos de dados. Os resultados mostraram ser melhores que os entregues por modelos mais bem estabelecidos e se comparam aos resultados de modelos estado da arte do aprendizado profundo. A grande vantagem das CNN sobre os modelos mais profundos é que seu tempo de treinamento é menor, apesar de ainda ser muito alto quando comparado a algoritmos simples. Já o problema com as redes neurais de convolução é a grande possibilidade configurações que uma rede destas pode assumir.

O objetivo desta dissertação é dar continuidade aos experimentos feitos por Kim em seu trabalho. De forma que se possa avaliar o desempenho de uma rede neural de convolução, sobre dados de análise de sentimentos, quando sua topologia é alterada de forma sistemática. A variação da topologia se dá pela mudança de seus hiperparâmetros. Estes são parâmetros que são definidos fora do processo de treinamento. Vão desde o número de neurônios ao tipo de função de ativação.

Os experimentos foram feitos usando como referência a configuração de hiperparâmetros adotada por Kim em (KIM, 2014). A partir desta, os hiperparâmetros são explorados de modo que todos permaneçam fixos enquanto apenas uma classe é avaliada. Quando uma classe de hiperparâmetros é completamente avaliada, seu valor volta a ser o estabelecido como referência e uma outra classe passa a ser explorada. Os resultados são comparados entre si usando um teste de significância estatística, mas também são comparados com outros modelos mais tradicionais usados na análise de sentimentos.

A principal contribuição desta dissertação é a implementação e avaliação de diferentes modelos de análise de sentimentos usando redes neurais de convolução. Os modelos aqui testados mostram ter melhor desempenho que os usados rotineiramente e são competitivos com os modelos no estado da arte. Com os experimentos feitos nesta dissertação, também é possível traçar um indicativo de quais configurações podem melhorar o desempenho das redes neurais de convolução aplicadas a dados de análise de sentimentos.

O restante desta dissertação está organizado em seis capítulos, cujos conteúdos são resumidamente descritos a seguir. Inicialmente, o Capítulo 1 apresenta uma introdução teórica do problema de análise de sentimentos, onde são expostas suas aplicações, os tipos de análise, as diferentes abordagens e os seus desafios.

No Capítulo 2 são apresentados, de maneira sucinta, os trabalhos relacionados à análise de sentimentos. São apresentadas diferentes abordagens do problema e as soluções propostas.

O Capítulo 3 apresenta a fundamentação teórica relacionada às redes neurais artificiais. São apresentados os conceitos de neurônio artificial e aprendizado profundo. Além disso, o capítulo também define como funcionam as redes de convolução e como elas podem ser aplicadas na classificação de textos.

O capítulo 4 define os vetores de semânticos de baixa dimensionalidade que são usados em modernos sistemas de aprendizado de máquinas e têm papel fundamental nas redes neurais de convolução. Ao longo do capítulo são mostradas sua motivação e uso, além dos diferentes tipos existentes.

O Capítulo 5 descreve as características e considerações técnicas adotadas na implementação dos experimentos feitos ao longo do trabalho. São expostos aspectos relevantes aos *softwares* utilizados, dados de treinamento, modelos de referência, metodologia dos experimentos e hiperparâmetros.

Por fim, o Capítulo 6 apresenta os principais resultados e as possíveis análises que podem ser feitas a partir destes. A dissertação termina com as conclusões obtidas através dos resultados e as direções para futuros trabalhos.

Capítulo 1

ANÁLISE DE SENTIMENTOS

ANÁLISE de sentimentos (AS) é o campo de estudo que analisa as opiniões de pessoas, sentimentos, avaliações, atitudes e emoções em relação à entidades tais como produtos, serviços, organizações, indivíduos, eventos entre outros. Representa um espaço de problemas muito grande e possui diferentes nomes: análise de sentimentos, mineração de opiniões, mineração de sentimentos, análise de subjetividade, análise de emoções, análise de críticas, etc. Apesar de todas as denominações representarem o mesmo problema, elas são levemente diferentes em sua essência. O termo Análise de Sentimento surgiu pela primeira vez em (NASUKAWA; YI, 2003), enquanto Mineração de Opiniões ocorreu em (DAVE; LAWRENCE; PENNOCK, 2003). Contudo, o estudo e interesse por este problema apareceram primeiro em (PANG; LEE; VAITHYANATHAN, 2002).

Embora o NLP tenha uma longa história, pouca pesquisa foi feita sobre opiniões e sentimentos das pessoas antes dos anos 2000. Contudo, desde então, o campo tornou-se uma área de pesquisa muito ativa. Há algumas razões para isso. Primeiramente, há um amplo arranjo de aplicações, em vários domínios. A indústria da análise de sentimentos floresceu devido à proliferação de aplicações comerciais. Isso fornece uma forte motivação para a pesquisa. Em segundo lugar, o campo oferece problemas de pesquisa desafiadores, que não tinham sido estudados antes. Em terceiro lugar, pela primeira vez na história humana, há um enorme volume de dados opinativos nas redes sociais. Sem esses dados, muita pesquisa não teria sido possível. Não é surpreendentemente que o crescimento da AS coincida com os das redes sociais. Na verdade, a AS está agora no centro da pesquisa de mídias sociais. Assim, a pesquisa de mineração de opiniões não só tem um impacto importante no NLP, mas também tem um profundo efeito nas ciências de gestão, política, de economia e sociais, pois todas estas são afetadas pelas opiniões das pessoas .

1.1 Aplicação de Análise de Sentimentos

As opiniões desempenham papel fundamental em quase todas as atividades humanas porque são indicadores de comportamento. Sempre que é preciso tomar uma decisão, deseja-se conhecer as opiniões dos outros. No mundo real, as empresas e organizações sempre querem encontrar opiniões dos consumidores ou do público sobre seus produtos e serviços. Os consumidores individuais também querem saber as opiniões dos usuários existentes de um produto antes de comprá-lo, e opiniões dos outros sobre os candidatos políticos antes de decidir sobre seu voto. No passado, quando uma pessoa precisava de opiniões, perguntava a amigos e familiares. Quando uma organização ou uma empresa necessitava de opiniões públicas ou de consumidores, realizava pesquisas de opinião. Adquirir opiniões do público e do consumidor tem sido por muito tempo um negócio próprio para o *marketing*, as relações públicas e as companhias de gerenciamento de campanha política.

Com o crescimento exponencial das mídias sociais (comentários, fóruns, blogs, microblogs, comentários e postagens em redes sociais) na Internet, pessoas e empresas estão cada vez mais usando o conteúdo disponível nestes meios para tomar decisões. Atualmente, se alguém quiser comprar um produto de consumo, já não se limita a pedir a opinião de seus amigos e familiares porque há muitos comentários de usuários e discussões em fóruns públicos na Internet sobre o produto. Para uma organização, pode não ser mais necessário realizar pesquisas de opinião para reunir informações de potenciais clientes, pois há abundância destas disponíveis na Internet. Todavia, encontrar e monitorar sites de opinião e filtrar as informações relevantes contidas nestes é uma tarefa extremamente difícil por causa da proliferação diária de conteúdo *online*. Cada site normalmente contém um enorme volume de textos opinativos que nem sempre encontra-se estruturado e pronto para ser usado. O leitor humano terá dificuldade em identificar locais relevantes, extrair e resumir as opiniões contidas nesses locais. Portanto são necessários sistemas inteligentes capazes de automatizar o processo de análise de sentimentos.

Nos últimos anos, testemunhou-se que postagens opinativas em mídias sociais têm ajudado a remodelar negócios e influenciar os sentimentos e emoções públicas que têm impactado profundamente os sistemas sociais e políticos existentes. Exemplos de mobilização que foram alcançados através da comunicação em mídias sociais são a Primavera Árabe em 2011 e as manifestações de 2013 no Brasil. Com isso, é possível perceber a importância de coletar e analisar opiniões presentes na Internet. Contudo, textos opinativos não existem

apenas no mundo virtual. Há muitos documentos que são coletados por empresas por meio de pesquisas e *feedback* de clientes. Esses dados também precisam ser analisados de forma rápida para que as corporações possam tomar decisões estratégicas para o futuro do negócio.

Devido a essas aplicações, as atividades industriais têm se desenvolvido nos últimos anos. As aplicações de análise de sentimentos se espalharam para quase todos os domínios possíveis, desde produtos de consumo, serviços, assistência médica e finanças a eventos sociais e eleições. Grandes corporações como Google™, Facebook™, Microsoft™ e Amazon™ estão construindo suas próprias soluções internas para tratar este problema cada vez mais presente. Até mesmo pequenas empresas estão percebendo o valor da informação de seus atuais e potenciais clientes e estão fazendo investimentos nesta direção. As diversas aplicações práticas e o crescente interesse da indústria têm sido fortes motivações para o aumento da pesquisa em mineração de opiniões.

Muitas publicações orientadas a uma aplicação específica foram produzidas. Por exemplo, em (LIU et al., 2007), um modelo de sentimento foi proposto para prever o desempenho de vendas. Em (MCGLOHON; GLANCE; REITER, 2010), *reviews* foram usados para classificar produtos e comerciantes. Em (HONG; SKIENA, 2010), os autores estudaram as relações entre as tendências de apostas na NFL (liga de futebol americano) e opiniões públicas em blogs e Twitter™. Em (O'CONNOR et al., 2010), o sentimento extraído do Twitter™ foi vinculado a pesquisas de opinião pública. Em (TUMASJAN et al., 2010), análise de sentimentos no Twitter™ foi aplicada para prever os resultados de eleições. Em (ZHOU; CHEN; WANG, 2010), os autores estudaram pontos de vista políticos. Em (YANO; SMITH, 2010), foi relatado um método para prever volumes de comentários de blogs políticos. Em (ASUR; HUBERMAN, 2010), (JOSHI et al., 2010) e (SADIKOV; PARAMESWARAN; VENETIS, 2009), dados oriundos do Twitter™, críticas de filmes e blogs foram usados para prever receitas de bilheteria para alguns filmes. Em (MILLER et al., 2011), investigou-se o fluxo de sentimentos em redes sociais. Em (MOHAMMAD; YANG, 2013), sentimentos em e-mails foram usados para descobrir como pessoas de diferentes sexos se distinguem em aspectos emocionais. Em (BOLLEN; MAO; ZENG, 2010), *tweets* foram usados para prever o mercado de ações. Em (BAR-HAIM et al., 2011) e (FELDMAN et al., 2011), investidores especialistas em microblogs foram identificados e análise de sentimento de ações foi realizada. Em (ZHANG; SKIENA, 2010), sentimentos de blogs e notícias foram usados para estudar estratégias de negociação de ativos financeiros. Em (SAKUNKOO; SAKUNKOO,

2009), foram estudadas influências sociais nas revisões de livros *online*. Em (GROH; HAUFFA, 2011), a análise de sentimentos foi usada para caracterizar relações sociais. Um sistema abrangente de análise de sentimentos e alguns estudos de caso também foram relatados em (CASTELLANOS et al., 2011).

1.2 Tipos de Análise de Sentimentos

As aplicações reais são apenas parte da razão pela qual a análise do Sentimentos é um problema de pesquisa popular. Um outro motivo é que se trata de um problema altamente desafiador como tópico de pesquisa e abrange muitos novos subproblemas. Além disso, houve pouca pesquisa antes do ano 2000 na área de Processamento de Linguagem Natural. A razão disto é que antes deste ano havia pouco texto opinativo produzido por pessoas disponível em formato digital. Desde então, o campo tem crescido rapidamente tornando-se uma das áreas de pesquisa mais ativas do NLP. Este também é amplamente usado em extração de informação, mineração de dados e mineração *Web*. Na verdade, está presente em diversas áreas da ciência (ARCHAK; GHOSE; IPEIROTIS, 2007), (CHEN; XIE, 2008), (DAS; CHEN, 2007), (DELLAROCAS; ZHANG; AWAD, 2007) e (HU; PAVLOU; ZHANG, 2006).

Em geral, a Análise do Sentimento tem sido subdividida principalmente em três níveis: análise a nível de documento, sentença e aspecto.

1.2.1 Nível de Documento

A tarefa neste nível é classificar se o sentimento global que emana do texto de um documento expressa uma opinião positiva ou negativa (PANG; LEE; VAITHYANATHAN, 2002), (TURNEY, 2002). Por exemplo, dado o *review* de um produto, o sistema determina se a avaliação expressa uma opinião geral positiva ou negativa sobre o produto. Esta tarefa é comumente conhecida como classificação de sentimentos a nível de documento. Este nível de análise pressupõe que cada documento expresse opiniões sobre uma única entidade (por exemplo, um único produto ou única pessoa). Desse modo, não se aplica a documentos que avaliam ou comparam múltiplas entidades.

1.2.2 Nível de Sentença

A tarefa neste nível analisa sentenças e determina se cada uma expressa uma opinião positiva, negativa ou neutra. Esta última geralmente significa ausência de opinião. Esse ní-

vel de análise está intimamente relacionado com a classificação de subjetividade (WIEBE; BRUCE; O'HARA, 1999), que distingue frases, chamadas sentenças objetivas que expressam informações factuais de sentenças, chamadas sentenças subjetivas que expressam opiniões e opiniões subjetivas. Entretanto, observa-se que a subjetividade não é equivalente ao sentimento, já que muitas sentenças objetivas podem implicar opiniões, como por exemplo: O celular foi comprado há dois dias, mas já apresenta problemas.

1.2.3 Nível de Aspecto

Tanto a análise a nível de documento como a nível de sentença não descobrem o que exatamente as pessoas gostaram e não gostaram. O nível de aspecto executa análise com esse tipo de granularidade. O nível de aspecto foi anteriormente chamado de nível de recurso (mineração de opinião baseada em características e sumarização) (HU; LIU, 2004). Ao invés de examinar estruturas de linguagem (documentos, parágrafos, sentenças ou orações), a análise a nível de aspecto foca diretamente na opinião. Baseia-se na ideia de que uma opinião consiste em um sentimento (positivo ou negativo) e um alvo da opinião. Uma avaliação tem o seu uso limitado caso o seu alvo não seja identificado. Saber quem é o alvo da análise é de grande importância pois ajuda a entender melhor o problema de análise de sentimentos. Por exemplo, embora a frase "O restaurante tem um ótimo serviço, contudo há pequenos problemas com limpeza.", claramente tenha um tom positivo, não se pode dizer que esta frase é inteiramente positiva. De fato, a frase positiva sobre o restaurante (enfatizada), mas negativo sobre sua limpeza (não enfatizada). Em muitas aplicações, os alvos de opinião são descritos por entidades e/ou seus diferentes aspectos. Desse modo, o objetivo deste nível de análise é descobrir sentimentos sobre entidades e/ou seus aspectos. Por exemplo, a frase "A qualidade da chamada do celular é boa, mas a duração da bateria é curta", avalia dois aspectos: a qualidade da chamada e a duração da bateria do celular (entidade). O sentimento em relação a qualidade da chamada do celular é positivo, mas o sentimento sobre sua bateria é negativo. A qualidade da chamada e a duração da bateria do aparelho são as metas da opinião. Com base nesse nível de análise, pode ser produzido um resumo estruturado de opiniões sobre entidades e seus aspectos, que transforma um texto não-estruturado em dados estruturados e pode ser usado para todos os tipos de análises qualitativas e quantitativas. Tanto a análise a nível de documento quanto a classificação a nível de sentença já são problemas altamente desafiadores. A análise a nível de aspecto é ainda mais difícil por ser

formada por alguns subproblemas como etiquetagem de partes da fala e reconhecimento de entidades. Além disso, este tipo de análise faz uso de dados estruturados como árvores de análise sintática que é um dado caro e de difícil acesso.

1.3 Abordagens Existentes

A abordagem padrão para a representação de texto (SALTON; MCGILL, 1983) tem sido o método do saco de palavras (BOW, *bag-of-words*). De acordo com o modelo BOW, o documento é representado como um vetor de palavras no espaço Euclidiano, onde cada palavra é independente dos outras. Este saco de palavras individuais é comumente chamado de uma coleção de unigramas. O BOW é fácil de ser entendido e permite alcançar resultado de alto desempenho.

As duas abordagens principais de análise de sentimentos são abordagem não-supervisionada que é baseada em vocabulário e abordagem supervisionada que é baseada em aprendizagem máquina, ambos dependem do *bag-of-words*. No método supervisionado de aprendizagem de máquina, os classificadores usam unigramas ou suas combinações (N-gramas) como entradas do classificador. No método baseado em vocabulário, os unigramas que são encontrados no léxico recebem uma pontuação de polaridade, a pontuação total de polaridade do texto é então calculada como soma das polaridades dos unigramas.

1.3.1 Abordagem por Aprendizado de Máquinas

A abordagem por aprendizagem de máquina para a classificação de texto é feita por meio de treinamento supervisionado. O algoritmo analisa os dados que foram marcados anteriormente como positivo, negativo ou neutro; extrai as características (*features*) que modelam as diferenças entre as classes, e infere uma função de classificação, que pode ser usada para avaliar novos exemplos antes não vistos. De forma simplificada, a tarefa de classificação de texto pode ser descrita da seguinte forma: dado um conjunto de dados de rotulados $T_{treino} = \{(t_1, l_1), ..., (t_k, l_k)\}$, onde cada texto t_i pertence a um conjunto de dados T e o rótulo l_i é uma classe pré-definida dentro do grupo de classes L, o objetivo é construir um algoritmo de aprendizagem que receberá como entrada o conjunto de treinamento T_{treino} e irá gerar um modelo capaz de classificar com precisão textos não catalogados previamente.

Os algoritmos de aprendizagem mais populares para a classificação de texto são *Support Vector Machine* (SVMs) (CORTES; VAPNIK, 1995) (VAPNIK, 1995), *Naive Bayes* (NARAYANAN;

ARORA; BHATIA, 2013); Árvore de Decisão (MITCHELL, 1997). Em (BARBOSA; FENG, 2010), encontraram-se os melhores resultados para SVMs, enquanto no trabalho de (PAK; PAROUBEK, 2010), *Naive Bayes* desempenhou melhor a tarefa de classificação. Já (DUMAIS et al., 1998) obtiveram um classificador de Árvore de Decisão capaz de executar análises quase tão bem como um classificador SVM.

1.3.2 Abordagem Baseada em Vocabulário

A abordagem baseada em vocabulário calcula o sentimento de um determinado texto a partir da polaridade das palavras ou frases nesse texto (TURNEY, 2002). Para este método um léxico (dicionário) de palavras com polaridades anexadas faz-se necessário. Exemplos de léxicos existentes incluem: *SentiWordNet* (ESULI; SEBASTIANI, 2006), *LoughranMcDonald Lexicon* (MOHAMMAD; KIRITCHENKO; ZHU, 2013), *General Inquirer Lexicon3* (STONE; HUNT, 1963). O valor de sentimento (*score*) pode ser calculado como a média das polaridades transmitidas por cada uma das palavras no texto.

1.4 Desafios da Análise de Sentimentos

Os indicadores mais importantes de sentimentos são palavras de sentimento, também chamadas de palavras de opinião. Estas são palavras que são comumente usadas para expressar sentimentos positivos ou negativos. Por exemplo, bom, excelente e surpreendente são palavras de sentimento positivo; e mau, pobre e horroroso são palavras de sentimento negativo. Além de palavras individuais, também existem frases e expressões idiomáticas. Por exemplo: "me custou os olhos da cara". As palavras e as frases de sentimentos são uma ferramenta importante na análise de sentimentos por razões óbvias. Uma lista de tais palavras e frases é chamada um léxico do sentimento, léxico da opinião ou vocabulário de opinião. Todavia, este léxico não é o suficiente devido à complexidade do problema. Alguns dos desafios relacionados à análise de sentimentos são:

- Reconhecimento de Entidades: é o problema de descobrir sobre o que a pessoa realmente está falando. Exemplo: São os 300 de Esparta. Refere-se a um grupo de gregos ou a um filme?

- Resolução de Anáfora: é o problema de descobrir a o que um pronome ou frase se refere. Exemplo: Vimos o filme e fomos jantar, foi horrível.

- Sarcasmo: é o problema de definir qual é o sentimento subliminar à sentença.
- Gírias e Neologismos: é a situação em que novas palavras criadas especificamente para um contexto, exclusivas de uma localidade ou grupo.

Estes são alguns dos problemas a serem enfrentados pela pesquisa da análise de sentimentos. Há vários outros e alguns deles são de difícil interpretação até para seres humanos por serem de origem subjetiva.

1.5 Considerações Finais

A análise de sentimentos, como o Processamento de Linguagem Natural como um todo, têm ganho um crescimento muito grande impulsionados pelo advento das mídias sociais e pelos avanços tecnológicos. Tendo em vista as inúmeras aplicações, empresas, grandes ou pequenas, estão investindo muito para encontrar soluções que possam atender suas demandas. Saber o que as pessoas sentem e como se sentem em relação a um produto tornou-se um ativo de valor inestimável para as companhias.

As diferentes abordagens e tipos de análise de sentimentos mostram que não há solução definitiva para este problema e dão ideia de sua complexidade. Além disso, há também problemas em aberto, como o sarcasmo, que é de difícil interpretação para esses sistemas de avaliação.

Capítulo 2

TRABALHOS RELACIONADOS

ESTE capítulo apresenta alguns recentes trabalhos relacionados ao problema de análise de sentimentos em textos. Os trabalhos estão separados em três Seções de acordo com as possíveis abordagens ao problema. As Seções 2.1 e 2.2 apresentam soluções usando *machine learning* e *deep learning* respectivamente. Já a Seção 2.3 contém trabalhos que usam métodos baseados em vocabulário (*lexicon based*). Por fim, a Seção 2.4 apresenta algumas considerações finais.

2.1 Abordagem por Aprendizado de Máquinas

Os métodos de classificação de texto usando Aprendizagem de Máquinas (*Machine Learning*) podem ser divididos em métodos de aprendizado supervisionado e não supervisionado (MEDHAT; HASSAN; KORASHY, 2014). Os supervisionados fazem uso de um grande número de dados já classificados. Enquanto os não supervisionados são usados quando é difícil encontrar dados pré-classificados.

2.1.1 Aprendizado Supervisionado

Em (KANG; YOO; HAN, 2012), é usado um classificador *Naive Bayes* otimizado para resolver o problema de tendência de maior acurácia na classificação de amostras positivas (cerca de 10% maior que em amostras negativas). Isto cria o problema de diminuição da precisão média quando a acurácia para as duas classes é contabilizada de maneira conjunta. O trabalho mostrou que usando este algoritmo juntamente com um banco de dados de análises de restaurantes foi possível diminuir a diferença de precisão entre as classes quando comparado ao *Naive Bayes* tradicional e ao *Support Vector Machine* (SVM). As medidas de *recall* e *precision* também melhoraram.

Para evitar a hipótese de independência dos atributos (*features*) que existe em um classificador *Naive Bayes*, (ORTIGOSA-HERNáNDEZ et al., 2012) usaram uma Rede Bayesiana, que evita a consideração de independência, porém é mais cara computacionalmente, como algoritmo de classificação em um problema no qual a postura do autor é caracterizada por 3 variáveis alvo diferentes. Os autores propuseram o uso de uma Rede Bayesiana multidimensional. A rede uniu diferentes variáveis alvo em uma mesma tarefa de classificação com o intuito de explorar potenciais relações entre elas. O trabalho ainda expandiu a plataforma de classificação multidimensional para um sistema de aprendizado semi supervisionado com o propósito de utilizar o grande número de dados não classificados previamente que estavam disponíveis. Como resultado, eles mostraram que o modelo semi supervisionado superou em desempenho os algoritmos utilizados mais comumente em problemas de análise de sentimentos.

Em (CHEN; TSENG, 2011), são usados dois algoritmos multiclasse baseados em SVM: Um-contra-Todos e *Single-Machine Multi SVM* para categorizar comentários. Eles propuseram um método para avaliar a qualidade da informação em produtos analisados considerando como um problema de classificação. Usaram também uma estrutura de qualidade da informação para encontrar o conjunto de atributos orientados à informação. Eles trabalharam em comentários de câmeras digitais e aparelhos de MP3. Os resultados mostraram que o método pode classificar corretamente críticas em termos de sua qualidade e que significativamente supera métodos avançados.

SVMs foram usados em (LI; LI, 2013) como um classificador de polaridade de sentimento. Ao contrário do problema de classificação binária simples, eles argumentaram que a subjetividade de opinião e a credibilidade do expressor também devem ser levadas em consideração. Os autores propuseram uma estrutura que fornece um resumo numérico de opiniões sobre plataformas de micro-blogs. Identificaram e extraíram os tópicos mencionados das opiniões associadas aos pedidos dos usuários e, em seguida, classificaram-as usando SVM. *Tweets* foram usados em seu experimento. Com isso, descobriram que a consideração de credibilidade do usuário e subjetividade de opinião são essenciais para agregar comentários de microblog. Os autores também provaram que seu mecanismo pode efetivamente descobrir inteligência de mercado *market intelligence* para apoiar a tomada de decisão através da criação de um sistema de monitoramento assim rastreando opiniões externas sobre os diferentes aspectos de um negócio em tempo real.

Em (MORAES; VALIATI; NETO, 2013), é apresentada uma comparação empírica entre SVM e redes neurais artificiais (RNAs) sobre análise de sentimentos a nível de documento. Eles fizeram esse trabalho porque SVM é utilizado amplamente e com sucesso em análise de sentimentos, enquanto RNAs têm atraído pouca atenção como uma abordagem para a mineração de opiniões. Discutiram os requisitos, os modelos resultantes e em que contextos ambas abordagens alcançam melhores níveis de precisão de classificação. Também adotaram um contexto de avaliação padrão com métodos supervisionados populares para seleção de atributos e ponderação em um modelo *bag of words* tradicional. Os experimentos indicaram que a Rede Neural Artificial produziu resultados superiores ao SVM, exceto para alguns contextos de dados desbalanceados. Os autores usaram bancos de dados de referência sobre filmes, GPS, câmera e avaliações de livros do *amazon.com*. Como resultado, provaram que os experimentos sobre comentários de filmes com RNAs superaram SVM por uma diferença significativa estatisticamente. Eles confirmaram algumas limitações potenciais de ambos modelos, que foram raramente discutidos na literatura de análise de sentimentos, como o custo computacional do SVM no momento em execução e da RNA no momento do treinamento. O trabalho também provou que o uso de *Information Gain* (um método de seleção de atributos computacionalmente barato) pode reduzir o esforço computacional das RNAs e dos SVMs sem afetar significativamente a precisão da classificação resultante.

SVM e RNAs também foram usados para a classificação das relações pessoais em textos biográficos como apresentado por Van de Camp e Van den Bosch em (CAMP; BOSCH, 2012). Eles marcaram as relações entre duas pessoas (sendo uma delas o tema de uma biografia, a outra sendo mencionada nesta biografia) como positivo, neutro ou desconhecido. O estudo de caso foi baseado em históricas informações biográficas descrevendo pessoas em um contexto particular, região e período de tempo. O resultado mostrou que os classificadores foram capazes de rotular corretamente essas relações acima do valor esperado. Os autores descobriram que um conjunto de treinamento contendo relações, envolvendo várias pessoas, produz resultados mais desejáveis que um conjunto que se concentra em uma entidade específica. Provaram também que SVM e uma RNA de uma camada atingem os valores mais altos de precisão.

2.1.2 Aprendizado semi e não Supervisionado

O conceito de fraco e semi-supervisão é usado em muitos aplicações. Em (HE; ZHOU, 2011), foi proposta uma estratégia que trabalha fornecendo supervisão "fraca"no nível de atributos, ao invés do de amostras. Eles obtiveram um classificador inicial por incorporação de informação prévia extraída a partir de um léxico de sentimento existente em um modelo de aprendizagem de sentimento. Os autores se referem a informações prévias como atributos classificados e os usam diretamente para restringir as previsões do modelo em amostras não classificadas usando critérios de expectativa generalizada. Em seu trabalho, foi possível identificar as palavras de polaridade de domínio específico que clarificam a ideia de que a polaridade de uma palavra pode ser diferente de acordo com o contexto. O trabalho foi feito com análises de filmes e os dados multi domínio do IMDB e *amazon.com* respectivamente. O trabalho mostrou que esta abordagem obteve melhor desempenho que outros métodos de classificação de sentimento fracamente supervisionados e é aplicável a qualquer tarefa de classificação de texto onde algum conhecimento prévio relevante está disponível.

A abordagem não supervisionada foi usada também em (XIANGHUA et al., 2013) para detectar automaticamente os aspectos discutidos em redes sociais chinesas e também os sentimentos expressos em diferentes aspectos. Foi usado um modelo LDA (*Latent Dirichlet Allocation*) para descobrir temas globais multi aspecto de comentários sociais, então os pesquisadores extraíram o tema local e sentimento associado com base em um contexto de janela deslizante sobre o texto. Eles trabalharam em comentários sociais que foram extraídos de um conjunto de dados do blog (2000-SINA) e um vocabulário (300-SINA HowNet). Mostraram ainda que a sua abordagem obteve bons resultados na separação dos temas e melhorou a precisão da análise de sentimentos. O modelo também ajudou a descobrir diversos aspectos nos tópicos e sentimento associado. Já em (KO; SEO, 2000), foi proposto um método que divide os documentos em frases e classificam cada frase usando as listas de palavras-chave de cada categoria e uma medida de similaridade de sentença.

2.1.3 Aprendizado Híbrido

Em (LANE; CLARKE; HENDER, 2012), é apresentado uma abordagem por aprendizado de máquinas para resolver o problema de localização de textos que carregam uma tendência positiva ou negativa dentro de mídias sociais. O desbalanceamento na distribuição de amostras positivas e negativas, alteração dos documentos ao longo do tempo, treinamento eficaz e

procedimentos de avaliação para os modelos são os desafios que foram enfrentados para alcançar os resultados. O trabalho usou três conjuntos de dados gerados por uma empresa de análise de mídia social. Os documentos foram classificados de duas formas: detecção da presença de favorabilidade e avaliação negativa vs. positiva. Foram usados cinco diferentes tipos de atributos para criar os conjuntos de dados a partir do texto. Os autores testaram muitos classificadores e descobriram que o melhor é composto por SVM, KNN (*K-Nearest Neighbors*), NB, BN (*Bi-gram Naive Bayes*), DT (*Decision Tree*) e Classificador baseado em regras. O trabalho ainda mostrou que o equilíbrio da distribuição entre as classes de dados de treinamento pode ser benéfica na melhoria do desempenho, mas NB pode ser afetado negativamente .

As abordagens supervisionada e não-supervisionada podem ser combinadas. Isso foi feito em (MARTÍN-VALDIVIA et al., 2013). Eles propuseram a utilização de classificadores híbridos, a fim de desenvolver um sistema de classificação sentimento. O trabalho usou um corpus espanhol de comentários de filmes juntamente com um *corpus* paralelo traduzido para o inglês. Primeiro, eles geraram dois modelos independentes usando esses dois corpus, em seguida, aplicaram algoritmos de aprendizado de máquinas (SVM, NB e outros). Segundo, eles integraram o corpus de sentimento SentiWordNet ao corpus em inglês gerando assim um novo modelo não supervisionado utilizando uma abordagem de orientação semântica. Por fim, combinaram os três sistemas utilizando um classificador híbrido. Os resultados superaram os resultados do uso de um único corpus e mostraram que a abordagem adotada pode ser considerada uma boa estratégia para a classificação de sentimento quando corpus paralelos estão disponíveis.

Algoritmos baseados em aprendizado de máquinas foram usadas em (WALKER et al., 2012) para classificar o posicionamento. Este é definido como uma opinião global de uma pessoa em relação a um objeto, ideia ou postura. Posicionamento é semelhante a ponto de vista ou perspectiva e pode ser visto como a identificação do partido com o qual o autor da frase está ligado. Neste trabalho, foi classificado o posicionamento que as pessoas têm e aplicaram isso a debates políticos. Eles utilizaram 104 debates de duplo posicionamento oriundos do *convinceme.net* para 14 temas diferentes e tentaram identificar a opinião dos envolvidos. O objetivo principal foi determinar a contribuição potencial de características de diálogo na classificação de debates. O trabalho usou SVM, NB e um classificador baseado em regras para fins de classificação e conseguiram precisar a escolha do lado do debate.

2.2 Abordagem por Aprendizado Profundo

Em (ZHARMAGAMBETOV; PAK, 2015), são usados conjuntos de árvores de decisão para fazer a análise de sentimentos em comentários de filmes. Contudo, para a extração de atributos do texto, eles usaram métodos de aprendizagem profunda. A metodologia escolhida foi o *Word2Vec* que permite a captura de características semânticas das palavras. Os vetores obtidos pelo processo de extração de atributos foram clusterizados por *k-means*. Cada cluster, de um total de 2000, teve uma média de 5 palavras. Estas foram selecionadas por sua proximidade no espaço vetorial. Os clusters foram usados como entradas para o classificador. No trabalho, foi usado um banco de dados de referência com comentários de filmes. O resultado do experimento mostrou que a abordagem utilizando a extração de atributos por *deep learning* foi significativamente melhor que a utilizando *bag-of-words* com 5000 entradas.

Redes neurais profundas foram utilizadas em (HU et al., 2015) para a análise de sentimentos em *reviews* de produtos eletrônicos, filmes e hotéis. Os autores criaram uma estrutura de classificação que usa 3 métodos diferentes para extração de atributos: baseados em frequência, contexto e etiquetagem de partes de falas. Cada método alimenta uma sub rede neural que reduz a dimensionalidade do espaço de atributos. As saídas destas sub redes alimentam a rede principal que é responsável pela análise de sentimentos. A estrutura formada pelas sub redes e a rede principal é denominada rede neural profunda hierárquica. O trabalho comparou o desempenho da estrutura proposta pelos autores com métodos de aprendizado de máquinas (NB e SVM). A análise verificou que a acurácia do modelo de redes neurais profundas foi superior a NB e SVM para um número de amostras grande no conjunto de treinamento (acima de 200000). Todavia, a precisão do modelo proposto diminuiu para casos em que a dimensionalidade das entradas é grande e o número de amostras para treinamento é reduzido. O trabalho ainda mostrou que independente do contexto dos comentários a abordagem por aprendizado profundo apresenta precisão crescente com o número de amostras no treinamento. Os bancos de dados utilizados foram *amazon.com*, IMDB e *TripAdvisor*.

Análise de sentimentos a nível de sentença foi estudado em (LI et al., 2014). Para o trabalho, os autores tiveram que criar manualmente o banco de dados de sentenças a partir de uma rede social de filmes devido a falta de dados de referência no idioma chinês. Ao contrário de outros bancos de dados que possuem um sentimento global para cada sentença, este possui classificações de sentimento para cada oração dentro de cada sentença e para cada

palavra dentro de cada oração. Deste modo, as palavras alimentam as entradas de redes de alimentação direta. Estas, quando agrupadas, formam uma rede neural recursiva que tem como estágio final um classificador *soft*. O trabalho analisou a diferença de desempenho deste sistema, batizado de *Recursive Neural Deep Model* (RNDM), com métodos de aprendizado de máquinas (NB, SVM, e Regressão Logística). O resultado foi de um leve melhor desempenho do RNDM quando comparado às outras técnicas. Todavia, quando o mesmo experimento foi feito usando sentenças com conjunções adversativas, o resultado usando RNDM foi muito melhor aos outros algoritmos de referência.

Em (OUYANG et al., 2015), os autores propuseram um sistema chamadado de *Word2Vec* + CNN (Rede Neural de Convolução) para realizar a classificação de sentimentos em comentários de filmes. As palavras presentes nos *reviews* foram transformadas em vetores através do método de extração de atributos *Word2Vec*. Estes foram concatenados de modo a formar uma matriz correspondente a sentença ou documento analisado. A matriz foi usada como entrada de uma Rede Neural de Convolução de 7 camadas. O experimento feito no trabalho comparou o desempenho do modelo proposto a outras abordagens de *machine learning* (NB, SVM e NB com bigramas) e *deep learning* (Média de Vetores, Rede Neural Recursiva e MV-Rede Neural Recursiva). O resultado mostrou que para uma problema de classificação de sentimento em 5 diferentes classes (positivo, levemente positivo, neutro, levemente negativo e negativo) o sistema proposto obteve resultados melhores que todos os outros 6 algoritmos.

Análise de sentimentos a nível de sentença usando com redes de convolução também foi estudado por Kim em (KIM, 2014). Contudo, ao invés de usar redes com múltiplas camadas, o autor escolheu apenas redes com uma camada convolução. Durante o trabalho, foram avaliadas como a escolha de alguns hiperparâmetros, relativos à arquitetura da rede, afeta seu desempenho. O resultado final mostrou que o modelo teve melhor performance quando comparado a algoritmos mais simples como NB e SVM e também a mais complexos como (KALCHBRENNER; GREFENSTETTE; BLUNSOM, 2014) e (SOCHER et al., 2013).

Já em (ZHANG; WALLACE, 2015), os autores do trabalho se concentraram em um estudo mais aprofundado a respeito da influência dos hiperparâmetros no desempenho da rede criada por (KIM, 2014) quando usadas para classificar sentenças em diferentes *datasets*. Foram avaliados o tipo de entrada, tamanho dos filtros de convolução, quantidade de filtros de convolução, tipo de função de ativação, tipo de *pooling* e regularização.

2.3 Abordagem Baseada em Vocabulário

Métodos baseados em Vocabulário utilizam um léxico de palavras e frases de sentimento para determinar a polaridade de documentos (MEDHAT; HASSAN; KORASHY, 2014).

Em (QIU et al., 2010), é usada uma abordagem baseada em dicionário para identificar o sentimento de sentenças no contexto de publicidade. No trabalho, foi proposta uma estratégia de publicidade para melhorar a relevância dos anúncios e a experiência do usuário. Os autores trabalharam com dados oriundos de fóruns da web. Os resultados demostraram que o modelo proposto teve bom desempenho na extração de palavras-chave de publicidade e seleção de anúncios. Já em (HATZIVASSILOGLOU; MCKEOWN, 1997), foi usado uma lista inicial de adjetivos de opinião juntamente com um conjunto de restrições linguísticas para identificar mais palavras de opinião e suas orientações (MEDHAT; HASSAN; KORASHY, 2014).

2.4 Considerações Finais

Os trabalhos de aprendizado de máquinas se subdividiram em aprendizado supervisionado, não supervisionado e híbrido. Os métodos de aprendizado de máquinas dependem de algoritmos simples e que demandam de baixo custo computacional. Contudo, há exceções como a Rede Bayesiana apresentada por (ORTIGOSA-HERNáNDEZ et al., 2012). Os resultados encontrados pelos autores dos trabalhos foram em sua grande maioria muito satisfatórios, porém é possível observar que todos realizam alterações nos algoritmos canônicos a fim de obter algum tipo de informação extra sobre o texto.

Os trabalhos que usam métodos de Aprendizado Profundo fazem parte de uma recente área de pesquisa. O *deep learning* se mostrou uma ferramenta poderosa para análise de sentimentos. Em todos os trabalhos, os métodos de aprendizado de máquinas foram suplantados pelos modelos de *deep learning*. Um dos motivos para tal superioridade encontra-se no modo de extração de atributos que leva em consideração o contexto das palavras. Contudo, a grande desvantagem é a necessidade de muito mais dados para o treinamento que outros algoritmos mais simples.

Foram também citados dois trabalhos de Métodos Baseados em Vocabulário. Apesar de apresentar bons resultados em algumas aplicações, o método tem baixa escalabilidade porque necessita de um grande léxico de sentimento. Todavia, ainda há espaço para seu uso como abordagem complementar às outras citadas anteriormente.

Capítulo 3

REDES NEURAIS DE CONVOLUÇÃO

OS estudos que tratam problemas de análise de sentimentos usam diferentes abordagens para avaliar o sentimento adjacente a um texto. Há, todavia, dois métodos principais nos quais a maioria dos trabalhos se concentra. Métodos baseados em vocabulário e métodos de aprendizado de máquinas conforme descrito no capítulo 2. Contudo, devido aos avanços na parte de hardware e o crescimento do aprendizado profundo, as redes neurais artificiais (RNA) têm se mostrado um alternativa viável e mais poderosa para classificação de sentimentos.

As redes neurais artificiais (RNA), são algoritmos computacionais sofisticados usados para modelar relações complexas, geralmente não lineares, entre os dados de entrada e saída. Por serem capazes de descobrir tais relações, as RNAs podem tratar de problemas complexos como análise de sentimentos.

3.1 Redes Neurais Artificiais

RNAs são modelos computacionais capazes de realizar reconhecimento de padrões. São apresentadas como sistemas de neurônios interconectados que podem operar alguma forma de processamento computacional sobre os valores de entradas.

Elas têm como inspiração o cérebro humano onde o conhecimento é adquirido a partir de estímulos externos e mediante um processo de aprendizagem. As forças de conexão entre neurônios, chamadas de pesos sinápticos, são utilizadas para armazenar o conhecimento adquirido durante o processo de treinamento da rede.

3.1.1 Neurônio Artificial

Os elementos computacionais, denominados neurônios artificiais, são modelos simplificados dos neurônios biológicos. É uma espécie de neurônio mais simples e que engloba as principais características de uma rede neural biológica, ou seja, paralelismo e alta conectividade. Foi proposto inicialmente por McCulloch-Pitts (MCCULLOCH; PITTS, 1988). É o modelo mais utilizado nas arquiteturas de RNAs (HAYKIN, 2001).

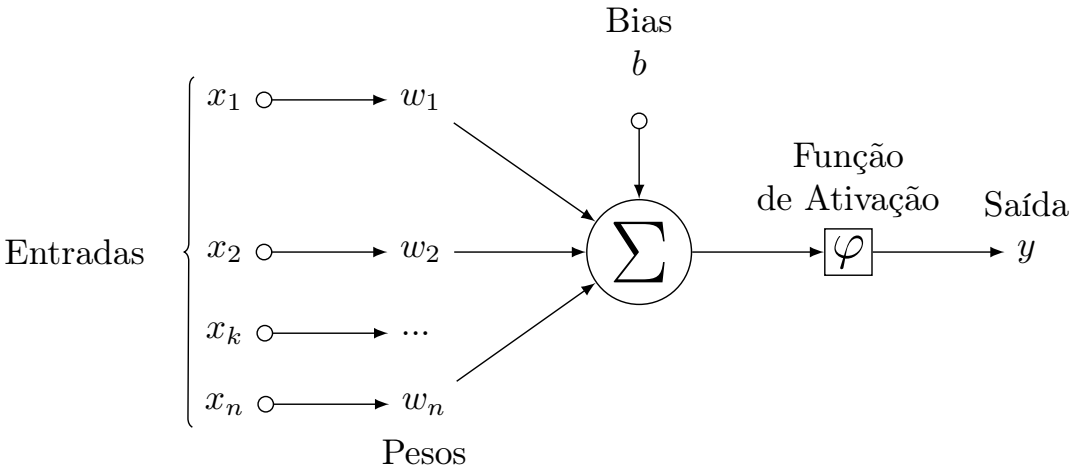


Figura 1: Modelo de Neurônio Artificial

Para um determinado neurônio artificial como ilustrado na Figura 1, seja n o número de entradas com sinais x_1 a x_n, pesos sinápticos w_1 a w_n, *bias* b e φ a função de ativação. A saída é dada pela Equação 1:

$$y = \varphi\left(\sum_{i=1}^{n}(w_j x_j + b)\right). \tag{1}$$

Esta saída é análoga ao axônio de um neurônio biológico, e seu valor se propaga para a entrada da próxima camada num processo semelhante à sinapse.

3.1.2 Multilayer Perceptron

Uma *Multilayer Perceptron* (MLP) é um modelo de rede neural artificial *feedforward*, onde as conexões entre os neurônios não formam um ciclo que mapeia conjuntos de dados de entrada para um conjunto de saídas apropriadas. Uma MLP consiste em várias camadas de nós em um grafo direcionado, com cada camada totalmente conectada à próxima. Exceto

para os nós de entrada, cada nó é um neurônio (ou unidade de processamento) com uma função de ativação, geralmente não-linear. A MLP foi a rede neural mais utilizada em problemas de aprendizagem de máquinas durante muito tempo. Sua popularidade foi oriunda do seu baixo custo computacional e sua capacidade de distinguir dados que não são linearmente separáveis, algo que seus antecessores não conseguiam fazer. A Figura 2 mostra um exemplo de MLP.

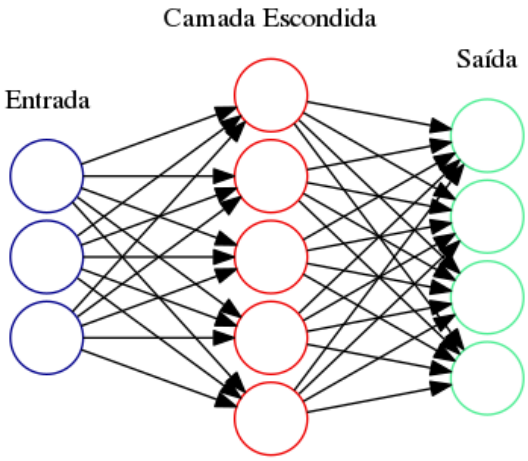


Figura 2: Multilayer Perceptron

As MLPs foram uma solução popular de aprendizado de máquinas nos anos 80, encontrando aplicações em diversos campos, como reconhecimento de fala, reconhecimento de imagens e tradução automática (WASSERMAN; SCHWARTZ, 1988), mas na década de 1990 enfrentaram uma forte concorrência dos SVMs por serem algoritmos tão eficientes quanto e muito mais simples. Mais recentemente, há um crescente interesse renovado em redes neurais devido aos casos de sucesso obtidos pelo aprendizado profundo.

3.2 Aprendizado Profundo

Aprendizado Profundo é um ramo do aprendizado de máquinas baseado em um conjunto de algoritmos que tentam modelar um alto nível de abstração nos dados, utilizando múltiplas camadas de processamento para implementar transformações não lineares. Uma das áreas promissoras de uso do *Deep Learning* é na substituição de características escolhidos manualmente por algoritmos que identificam as principais características automaticamente (NIELSEN, 2015).

Não há uma divisão clara entre o *Deep Learning* e o *Shallow Learning* (Aprendizado Raso), mas em geral especialistas concordam que uma rede com mais de 2 camadas escondidas pode ser considerada profunda (NIELSEN, 2015).

3.2.1 Conceitos Básicos

A ideia básica do Aprendizado Profundo é considerar que os dados estão organizados em múltiplas camadas que correspondem a diferentes níveis de abstração ou composição (NIELSEN, 2015). Um exemplo que ilustra esse conceito pode ser visualizado na Figura 3 que mostra uma rede neural de reconhecimento facial. O projeto da rede é apenas ilustrativo, mas exemplifica como uma composição complexa de características, tal como o rosto humano pode se dividida em pequenas partes simples assim como boca, olhos e nariz.

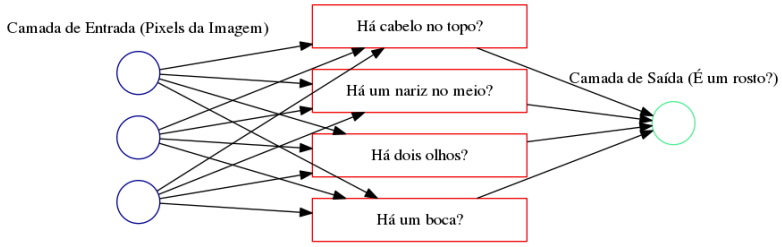


Figura 3: Sistema de Reconhecimento Facial

Como a ideia básica do aprendizado profundo é a composição de níveis, a rede neural da Figura 3 pode ser dividida em sub-redes. Estas são responsáveis pela identificação de características mais simples. Essas características simples podem ser decompostos em outros mais simples ainda. Por exemplo, os olhos podem ser divididos em íris, pálpebras e sobrancelhas conforme ilustrado na Figura 4.

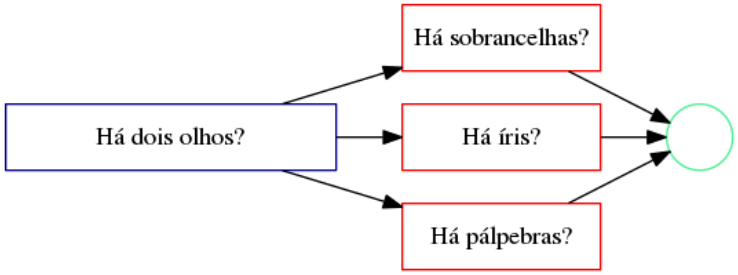


Figura 4: Conceito de Abstração

A sequência de composições de características simples em outros mais complexos é feita pela composição de transformações não lineares representadas pelas diversas camadas escondidas de uma rede neural profunda, ilustrada na Figura 5.

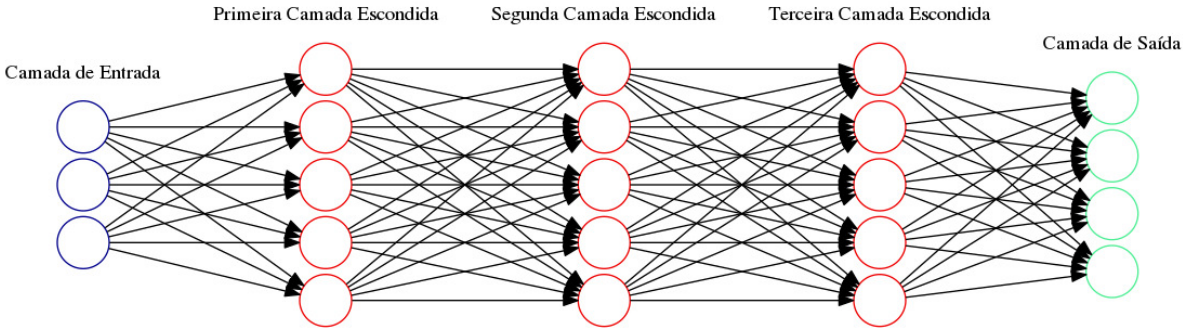


Figura 5: Rede Neural Profunda

3.2.2 Redes de Convolução

As redes neurais de convolução (*convolution neural networks*) formam uma classe de redes muita utilizada no Aprendizado Profundo. Suas maiores aplicações são em processamento de imagens digitais devido à sua capacidade intrínseca de utilizar a geometria das imagens (NIELSEN, 2015). As redes de convolução fazem uso de três ideias básicas: campos de recepção local, pesos compartilhados e agrupamento.

3.2.2.1 Campos de Recepção Local

Nas redes totalmente conectadas, como na Figura 5, as entradas são retratadas como uma linha vertical de neurônios. Em uma rede de convolução, consideram-se as entradas como um quadrado de n por n neurônios, por exemplo, cujos valores correspondem aos n por n *pixels* de intensidades da imagem de entrada (NIELSEN, 2015) conforme a Figura 6.

Os *pixels* de entrada são conectados a uma camada de neurônios ocultos. Todavia, não há conexão entre todos os *pixels* de entrada e todos os neurônios da camada escondida. Em vez disso, apenas regiões pequenas fazem conexão.

Para ser mais exato, cada neurônio na camada escondida é primeiro ligado a uma pequena região dos neurônios de entrada, por exemplo, uma região de txt, correspondendo a t^2 *pixels* de entrada. Assim, para um neurônio escondido em particular, esta conexão pode ser ilustrada conforme a Figura 7, onde $t = 5$.

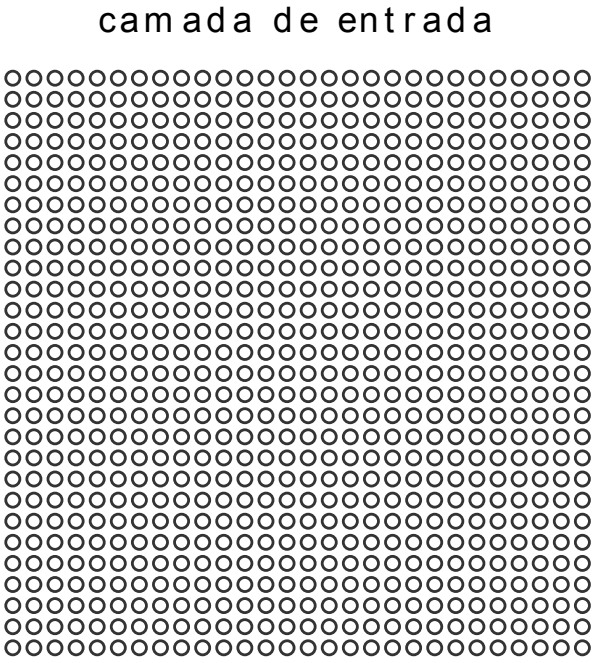


Figura 6: Neurônios de Entrada

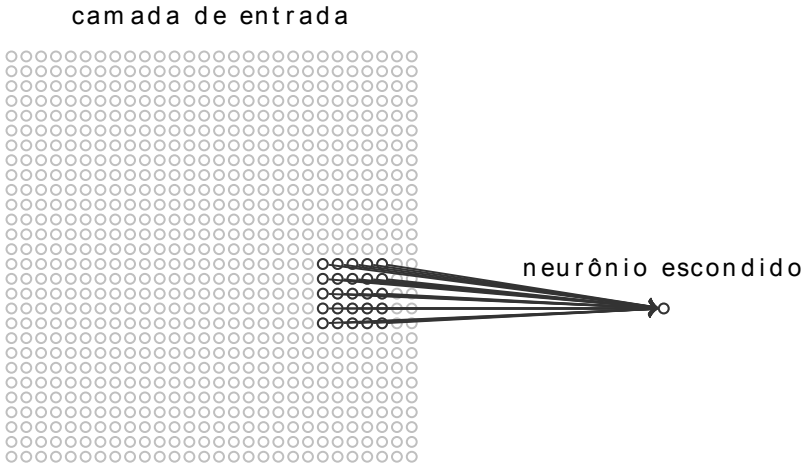
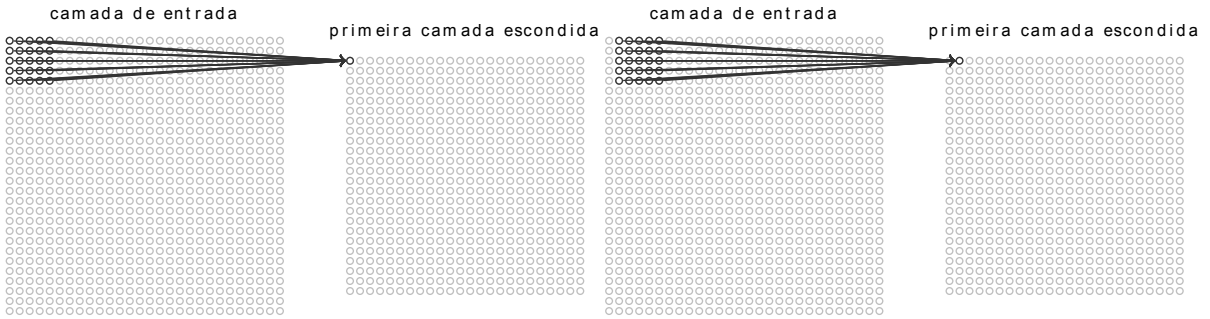


Figura 7: Neurônios da Camada Escondida

Essa região na imagem de entrada é chamada de campo de recepção local para o neurônio escondido é definido por um filtro de convolução de tamanho igual. É uma pequena janela sobre os *pixels* de entrada. Cada conexão aprende um peso. Portanto, o neurônio escondido aprende um *bias* geral. Simultaneamente, o filtro desliza sobre a imagem de entrada inteira. Para cada um, há um neurônio oculto diferente na primeira camada escondida. As Figuras 8 (a) e 8 (b) ilustram esse processo para o primeiro e o segundo neurônios da primeira linha da matriz de neurônios escondidos, respectivamente.



(a) Primeiro neurônio da camada escondida (b) Segundo neurônio da camada escondida

Figura 8: Neurônios da camada escondida

As camadas escondidas são criadas pelo deslizamento dos filtros sobre a camada de entrada. Ao final do processo, o tamanho da primeira camada escondida será de $(n-t+1)x(n-t+1)$.

3.2.3 Pesos Compartilhados

Em uma rede de convolução, todos os neurônios de uma mesma camada escondida compartilham os pesos e *bias*. Isto é, há um compartilhamento de pesos entre os diferentes campos de recepção local.

Como todos os neurônios da camada escondida compartilham o mesmo peso, isso significa que todos eles detectam exatamente a mesma característica, porém em diferentes localizações da imagem de entrada. Por esta razão, o mapeamento da camada de entrada para as camadas escondidas é chamado de mapa de características (NIELSEN, 2015). Uma grande vantagem do compartilhamento de pesos e *biases* é que ele reduz significativamente o número de parâmetros referentes a uma rede convolução.

Como cada camada escondida é capaz de detectar apenas um tipo de característica, é necessário usar várias camadas escondidas, conforme ilustrado na Figura 9 para fazer o reconhecimento de imagem. No exemplo da Figura 9, foram gerados 3 mapas de características de tamanho $(n-t+1)x(n-t+1)$ com um filtro de tamanho txt. Esta rede, ilustrada na Figura 9, será capaz de identificar estas características ao longo da imagem de entrada. A Figura 10 mostra alguns mapas de características da primeira camada escondida. As regiões mais escuras indicam os locais da imagem de entrada onde foram encontradas alguma característica detectável pelo mapa.

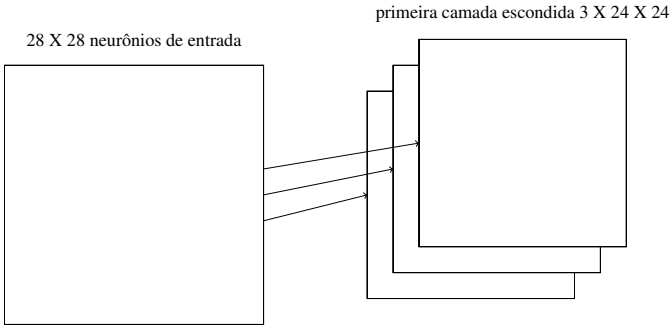


Figura 9: Conexão com as Camadas Escondidas

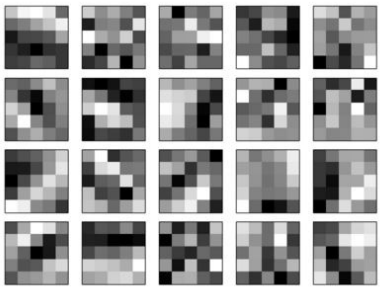


Figura 10: Exemplo de mapas de características

3.2.3.1 Agrupamento

Além das camadas de convolução (escondidas) descritas, as redes neurais de convolução também contém camadas de agrupamento (*Pooling*). Estas são normalmente utilizadas imediatamente após as camadas de convolução. As camadas de agrupamento permitem simplificar a informação de saída das camadas escondidas (NIELSEN, 2015).

Uma camada de agrupamento considera cada saída do mapa de características a partir da camada escondida e prepara um mapa de características condensado. Por exemplo, cada unidade da camada de agrupamento pode resumir uma região de mxm neurônios da camada escondida, onde $m < (n - t + 1)$. Um procedimento comum de agrupamento é o *pooling*-máximo (*max-pooling*) (NIELSEN, 2015), onde uma unidade de *pooling* emite simplesmente a ativação máxima na região de mxm de entrada. Todos os mapas de características são submetidos ao processo de *pooling* sendo que o tamanho final do mapa passa a ser $\frac{(n-t+1)}{m}$. A Figura 11 ilustra este procedimento para $m = 2$.

As camadas de agrupamento permitem uma melhor localização da região da imagem que foi ativada na camada escondida. Cada camada escondida tem uma camada de agrupamento respectiva, conforme ilustrado na Figura 12, onde cada mapa de $24x24$ origina uma camada de $12x12$.

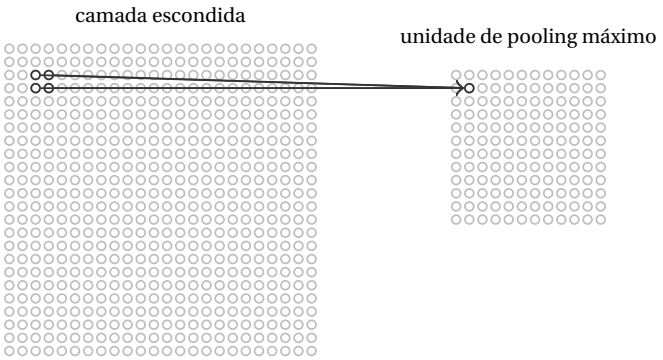


Figura 11: Camada de Pooling

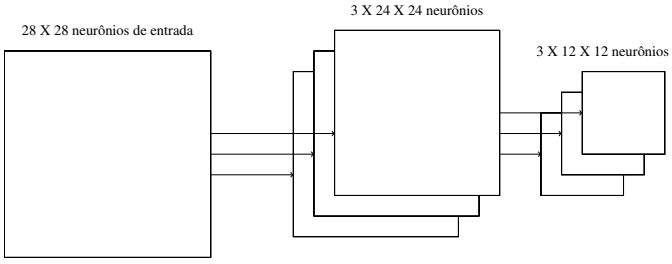


Figura 12: Conexão entre as Camadas

Por exemplo, a topologia de uma rede neural de convolução de reconhecimento de dígitos manuscritos é representado na Figura 13. São adicionadas 10 unidades de saída (dígitos de 0 a 9) e cada neurônio das camadas de agrupamento é conectado a cada uma das saídas.

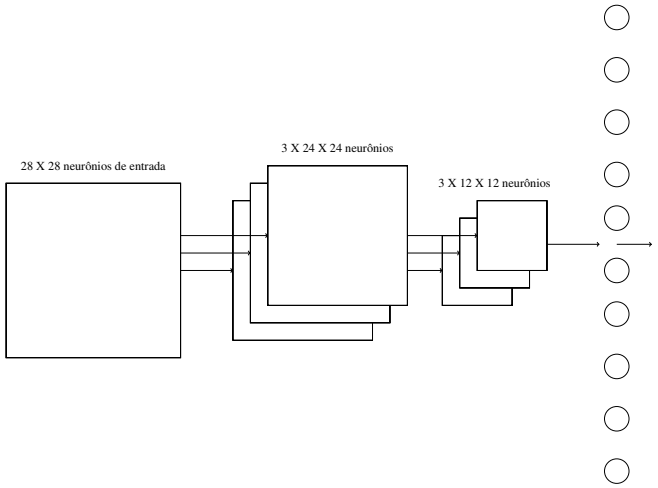


Figura 13: Topologia Final da Rede

3.3 Redes de Convolução em Classificação de Texto

A motivação deste trabalho se dá pelos recentes e promissores resultados encontrados por alguns autores (KIM, 2014) e (ZHANG; WALLACE, 2015) no uso de redes de convolução em áreas de classificação de textos como mineração de opinião, categorização de tópicos e filtro *spam*.

Ao invés de *pixels* da imagem, a entrada para a maioria das tarefas de NLP são frases ou documentos representados como uma matriz. Cada linha da matriz corresponde a um símbolo, tipicamente uma palavra, mas poderia ser um caractere. Isto é, cada linha é um vetor, representando uma palavra. Normalmente, esses vetores são representações de baixa dimensionalidade (*word embeddings*) como word2vec (MIKOLOV et al., 2013) ou *GloVe* (PENNINGTON; SOCHER; MANNING, 2014), mas também podem ser representações de alta dimensionalidade (vetores *one-hot*). Uma frase de cinco palavras usando uma representação de 100 dimensões se torna uma matriz de $5x100$ e será usada como entrada da rede. A única diferença para o caso com imagens é o tamanho dos filtros. Em textos, estes terão a mesma largura dos vetores de entradas, mas diferentes alturas. A Figura 14 mostra o uso de uma arquitetura de rede de convolução aplicada a um texto como a usado por Kim (KIM, 2014).

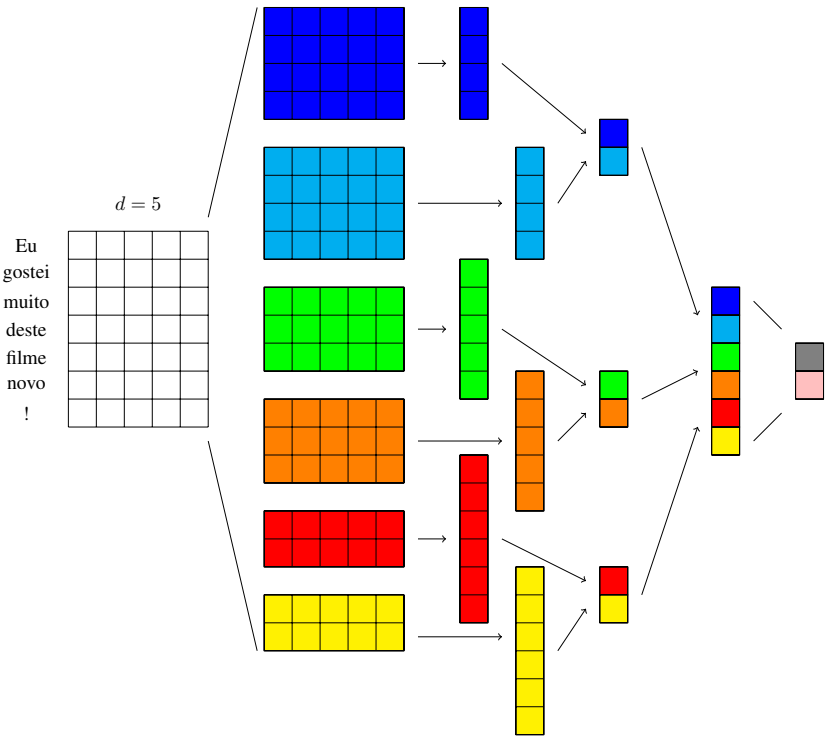


Figura 14: Arquitetura de rede de convolução para classificação binária de sentenças

Na Figura 14 por exemplo, há uma sentença de 6 palavras e 1 sinal de pontuação: "*Eu gostei muito deste filme novo!*"representados por vetores de 5 dimensões. Usa-se então, por exemplo, 3 filtros de altura 4, 3 e 2 para gerar 6 mapas de características. Estes têm tamanhos $4x1$, $5x1$ e $6x1$ como pode ser observado. Os mapas sofrem o processo de *pooling* e são concatenados formando um único vetor de tamanho $6x1$. Este, por final, é submetido a um classificador simples como MLP para obtenção de duas classes.

3.4 Considerações Finais

Uma rede neurais artificial são um algoritmo de aprendizado de máquinas inspirado no cerébro humano. Seu bom desempenho em tarefas de reconhecimento de fala, visão computacional e tradução automática fizeram de seu modelo mais famoso, MLP, uma poderosa ferramenta no tratamento de tais problemas. Contudo, devido a limitações de hardware dos anos 1990, as MLPs perderam espaço para outros algoritmos igualmente capazes e com baixo custo computacional.

Os avanços tecnológicos do século XXI permitiram a evolução de hardware necessária para o advento do aprendizado profundo. Novas redes neurais artificiais surgiram e com elas novos resultados estado da arte para problemas complexos. As redes neurais de convolução são atualmente o modelo padrão em problemas de visão computacional pois alcançam um nível de desempenho superior a qualquer outro modelo. Além disso, trabalhos recentes indicaram que seu uso em classificação de texto também é notável.

Capítulo 4

ESPAÇOS VETORIAIS SEMÂNTICOS DE BAIXA DIMENSIONALIDADE

UMA das tendências mais fortes no processamento de linguagem natural (NLP), no momento, é o uso de *word embeddings*, que são vetores cujas similaridades relativas se correlacionam com as similaridades semânticas. Tais vetores são usados tanto de maneira independente para calcular semelhanças entre termos, quanto como base de representação para tarefas de NLP tais como classificação de texto, agrupamento de documentos, etiquetagem de partes de falas, reconhecimento de entidades, análise de sentimentos, entre outras.

Word embeddings é o termo dominante para designação destes vetores, sem dúvida devido à popularidade atual dos métodos provenientes da comunidade de aprendizado profundo. Na linguística computacional, muitas vezes se usa o termo modelo semântico distributivo uma vez que a teoria semântica subjacente é chamada de semântica distributiva. Há também muitos outros termos alternativos em uso. Desde os mais genéricos como representação distribuída, aos mais específicos como espaço vetorial semântico ou simplesmente espaço de palavras. Para a consistência, será usado *word embeddings* ao longo deste trabalho.

De maneira sucinta, *embeddings* podem ser definidos como uma forma de representação de dados esparsos e de alta dimensionalidade, como é o caso de textos, em dimensões reduzidas que sejam capazes de codificar relações semânticas presentes nos dados. As primeiras tentativas de encontrá-los para quantificar a semelhança semântica entre palavras foram obtidas por meio de recursos feitos à mão. Os diferenciais semânticos introduzidos em (OSGOOD, 1952) na década de 1960 são um bom exemplo, e representações similares também foram usadas nos primeiros trabalhos sobre conexões e inteligência artificial na década de 1980.

Métodos para o uso de características contextuais geradas automaticamente foram desenvolvidos quase que simultaneamente em 1990 em várias áreas de pesquisa diferentes. Um dos modelos iniciais mais influentes foi a análise/indexação semântica latente (LSA x LSI), desenvolvida no contexto da recuperação da informação. Aproximadamente ao mesmo tempo, vários modelos diferentes foram desenvolvidos em pesquisas sobre redes neurais artificiais que usavam representações contextuais. Os mais conhecidos são, provavelmente, os Mapas Auto-Organizadores (SOM) (KOHONEN; SCHROEDER; HUANG, 2001) e as Redes Recorrentes Simples (SRN) (ELMAN, 1991), dos quais o último é o precursor dos modelos neuro-linguísticos de hoje .

Desenvolvimentos posteriores são basicamente apenas refinamentos desses modelos iniciais. Os modelos de tópicos ((WALLACH, 2006)) são refinamentos do LSA e incluem métodos como LSA probabilístico (PLSA) (HOFMANN, 1999) e *Latent Dirichlet Allocation* (LDA) (BLEI; NG; JORDAN, 2003). Os modelos neuro-linguísticos baseiam-se na mesma aplicação de redes neurais como SRN e incluem arquiteturas como redes neurais de convolução e *auto-encoders* (VINCENT et al., 2010).

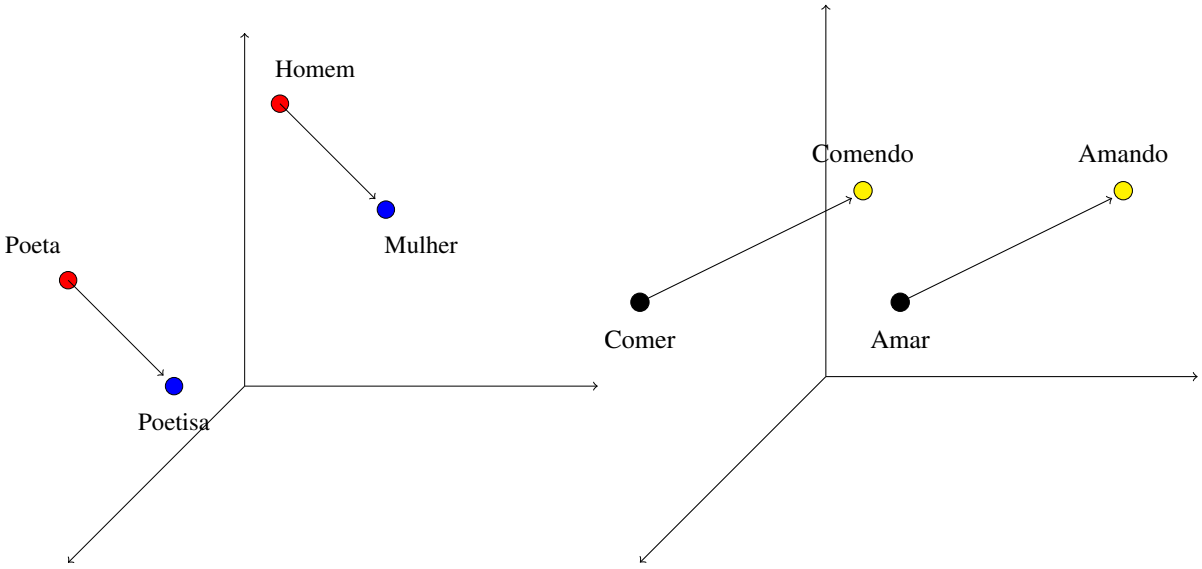
4.1 A Motivação para os Word Embeddings

Os sistemas de processamento de imagem e áudio funcionam com conjuntos de dados ricos e de alta dimensionalidade codificados como vetores das intensidades de *pixels* para dados de imagem ou, por exemplo, coeficientes de densidade espectral de potência para dados de áudio. Para tarefas como reconhecimento de objeto ou fala, todas as informações necessárias para a execução ser bem-sucedida são codificadas nos dados. No entanto, os sistemas de processamento de linguagens naturais tradicionalmente tratam palavras como símbolos atômicos discretos e, portanto, "hotel"pode ser representado por um identificador e "motel"por um outro identificador. Essas codificações são arbitrárias e não fornecem informações úteis ao sistema quanto às relações que podem existir entre os símbolos individuais. Isso significa que o modelo pode aproveitar muito pouco do que aprendeu sobre "hotéis"quando está processando dados sobre "motéis"de modo que ambos são locais, com quartos, banheiros, camas e outras particularidades.

Representar palavras como identificações únicas e discretas também leva à dispersão de dados. O algoritmo *bag-of-words* (UIJLINGS; SMEULDERS, 2011) cria vetores muito esparsos onde a dimensionalidade dos vetores é igual ao tamanho do vocabulário. Já os *word*

embeddings são representações que podem superar alguns desses obstáculos porque criam vetores com dimensões menores e mais densas que são capazes de preservar as similaridades semânticas existentes entre as palavras.

Uma característica interessante dos *word embeddings* é que por serem representações numéricas de similaridades contextuais entre palavras, estes podem ser manipulados aritmeticamente como qualquer outro vetor. As Figuras 15 (a) e 15 (b) ilustram estas relações para formas verbais e gênero.



(a) Relação entre gêneros

(b) Relação entre formas verbais

Figura 15: *Word Embeddings*

4.2 Tipos de *Word Embeddings*

Encontrar *word embeddings* é encontrar representações de baixa dimensionalidade para as palavras. Geralmente a redução de dimensionalidade é alcançada por meio de algoritmos de aprendizado não-supervisionado. *Word Embeddings* são de fato uma grande vitória do aprendizado não-supervisionado na indústria devido às suas diversas aplicações.

A principal diferença entre os vários modelos existentes é o tipo de informação contextual que estes usam. Os modelos do tipo LSA usam documentos como contexto. Os modelos neuro-linguísticos usam as palavras como contextos, o que é indiscutivelmente mais natural a partir de uma perspectiva linguística e cognitiva. Essas diferentes representações contextuais capturam diferentes tipos de semelhança semântica. Os modelos com contexto

baseados em documentos capturam relação semântica, tais como aquela entre carro e estrada enquanto os modelos baseados em palavras capturam similaridade semântica, tais como aquela entre carro e ônibus. Outra diferença significativa é a forma como os vetores são obtidos: *GloVe* e LSA usam fatoração de matrizes já que são modelos baseados em contagem enquanto *Word2Vec* e *fastText* usam redes neurais.

Os modelos baseados em contagem calculam as estatísticas de quantas vezes algumas palavras ocorrem junto de suas palavras vizinhas em um grande *corpus* de texto e, em seguida, mapeiam essas estatísticas de contagem para um vetor pequeno e denso para cada palavra. Modelos preditivos tentam prever diretamente uma palavra a partir de seus vizinhos.

4.2.1 Análise Semântica Latente

A Análise Semântica Latente (LSA) é uma técnica usada para criar representações vetoriais de documentos. O contexto dos *word embeddings* é formado por documentos e não palavras. A representação vetorial de um documento oferece uma maneira fácil de comparar textos por sua similaridade calculando o *cosseno* entre os vetores. Além disso, estes vetores fornecem um forma de representar documentos como entradas para algoritmos de aprendizado de máquinas e assim classificar documentos para determinar a qual grupo estes pertencem. Contudo, a classificação implica que alguns documentos já estejam rotulados. Para, tentar identificar agrupamentos naturais entre os documentos, é possível aplicar clusterização (PAPADIMITRIOU et al., 1998).

O primeiro passo para se obter vetores por LSA é aplicar o algoritmo chamado *tf-idf* (*term-frequency inverse-document-frequency*) (RAMOS, 1999). A essência do algoritmo é que cada posição no vetor corresponde a uma palavra diferente e um documento é representado contando o número de vezes que cada palavra aparece. Além disso, cada uma das contagens de palavras é normalizada pela frequência da palavra correspondente na coleção de documentos. Deste modo, termos menos frequentes têm mais peso.

Os vetores obtidos por *tf-idf* já são suficiente para realizar análises de aprendizado de máquinas. Contudo, eles geralmente são longos e esparsos, pois estes têm um componente para cada palavra no vocabulário. Para evitar esse problema, o LSA leva o *tf-idf* um passo adiante aplicando um algoritmo de redução de dimensionalidade, geralmente SVD (*Singular Value Decomposition*) (STEWART, 1993). Esta redução de dimensionalidade torna os vetores

mais gerenciáveis para operações como agrupamento ou classificação. Além disso, os vetores obtidos tendem a acentuar as características latentes de cada documento. Tornando assim, documentos semelhantes mais semelhantes, e documentos distintos, mais distintos.

4.2.2 Word2Vec

Google^{TM} usou o algoritmo com um *corpus* de 100 bilhões de palavras não catalogadas. O modelo original está disponível na internet (DAV, 2016).

Word2Vec é um modelo preditivo particularmente eficiente computacionalmente para aprender *embeddings* de palavras a partir de texto não-estruturado (MIKOLOV et al., 2013). Este é indiscutivelmente o modelo de *word embeddings* mais popular atualmente. Como este algoritmo forma um bloco de construção fundamental de modelos de aprendizado profundo para NLP, geralmente também é considerado um algoritmo de aprendizado profundo. No entanto, tecnicamente, o *Word2Vec* não faz parte do aprendizado profundo, pois sua arquitetura não é profunda nem usa não-linearidade.

4.2.2.1 *CBOW*

O *CBOW* (*Continous bag-of-words*) é um tipo de arquitetura rede do *Word2Vec*, onde são usadas as palavras do contexto (palavras vizinhas) para descobrir a palavra central. A Figura 16 mostra o *CBOW* com contexto de 2 palavras.

Os vetores *CBOW* são obtidos maximizando-se a função de custo J dada pela Equação 2:

$$J = \frac{1}{V}\sum_{i=1}^{V} log\left(P(w_i|w_{i-m},...,w_{i-1},w_{i+1},...,w_{i+m})\right). \tag{2}$$

O objetivo é maximizar a probabilidade $P(w_i|w_{i-m},...,w_{i-1},w_{i+1},...,w_{i+m})$ de se observar a palavra w_i dado um contexto. Esta é dada pela Equação 3:

$$P(w_{i+j}|w_i) = \frac{exp(w_i \cdot w_{i+j})}{\sum_{k\epsilon V} exp(w_i \cdot w_k)}, \tag{3}$$

onde V é o tamanho do vocabulário.

4.2.2.2 *Skip-Gram*

O *Skip-Gram* é um outro tipo de arquitetura rede do *Word2Vec*. Ao contrário do *CBOW*, ele usa a palavra central para descobrir as palavras do contexto. A Figura 17 mostra o *Skip-Gram*

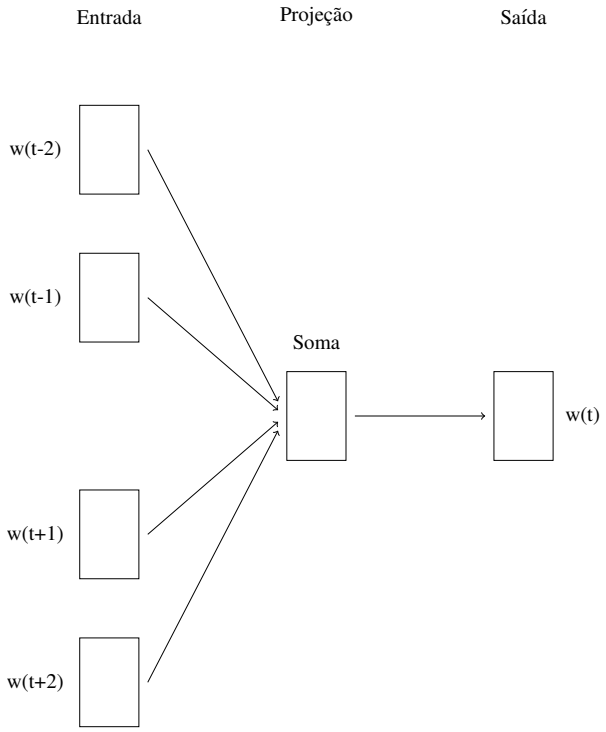


Figura 16: Arquitetura da rede utilizando o modelo *CBOW*

com contexto de 2 palavras. Os vetores *Skip-Gram* são obtidos maximizando-se a função de custo J, semelhante ao *CBOW*, dada pela Equação 4:

$$J = \frac{1}{V}\sum_{i=1}^{V}\sum_{-m\leq j\leq m, j\neq 0} logP(w_{i+j}|w_i), \tag{4}$$

onde V é o tamanho do vocabulário.

O objetivo é maximizar a probabilidade $P(w_{i+j}|w_i)$ de se observar a palavra w_{i+j} dada uma palavra central. Esta também é dada pela Equação 3.

4.2.2.3 Visualização

A Figura 18 mostra o resultado de uma simulação feita em Python usando o módulo Gensim (ŘEHŮŘEK; SOJKA, 2010) que permite o uso de *Word2Vec*, e um *corpus* formado pelos livros de *Game of Thrones* (MARTIN, 1997). Foram obtidos *embeddings* num espaço vetorial de 300 dimensões e depois reduzidos para 2 dimensões usando a técnica TSNE (MAATEN; HINTON, 2008). O TSNE foi escolhido como forma de redução de dimensionalidade pois este consegue preservar as distâncias relativas entre os vetores. Esta qualidade é importante pois permite observar as relações que são obtidas quando se usa *Word2Vec*. O mesmo não acontece quando se usa PCA (JOLLIFFE, 1986) para fazer grandes reduções de dimensionalidade.

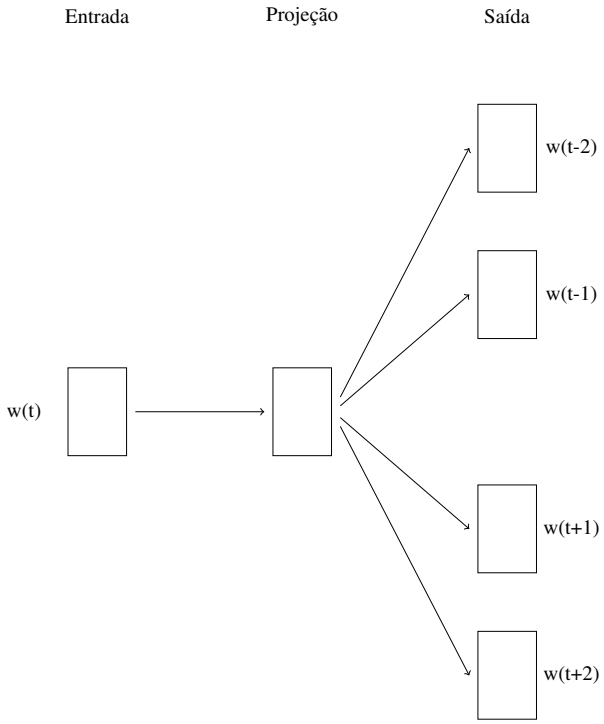


Figura 17: Arquitetura da rede utilizando o modelo *Skip-Gram*

É importante ressaltar que as componentes x e y obtidas por TSNE não têm qualquer significado. A técnica tem apenas como objetivo manter próximo vetores similares.

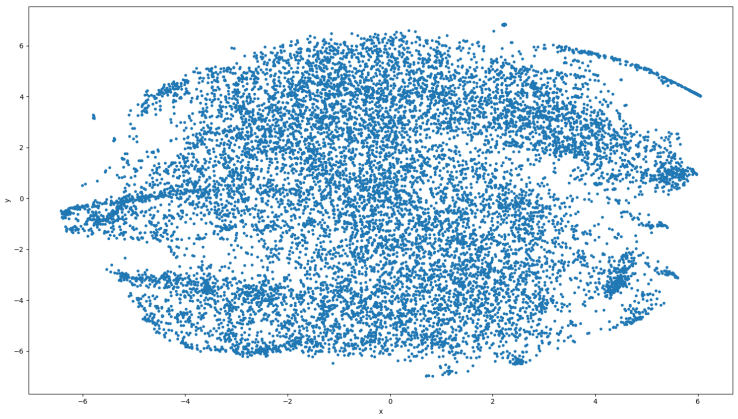
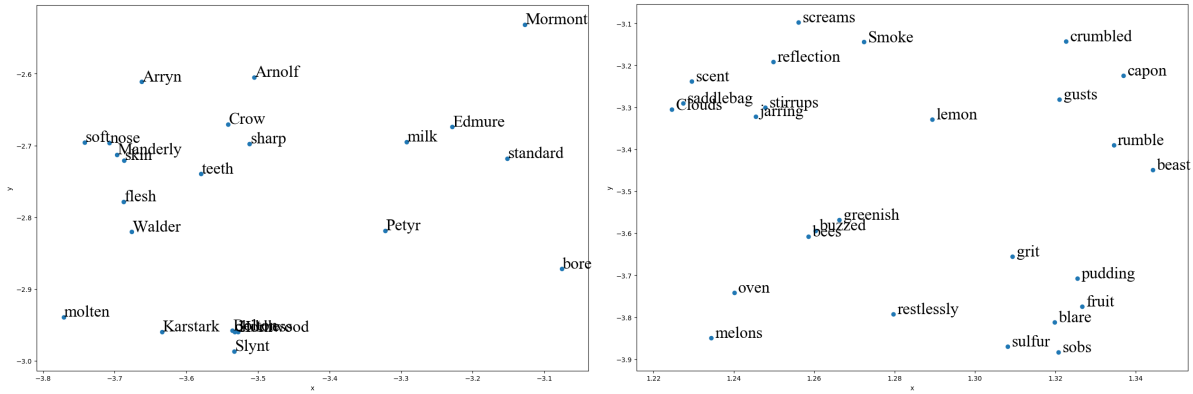


Figura 18: Word2Vec aplicado aos livros de Game of Thrones

Pela Figura 18, não é possível observar as relações entre os vetores pois há muias palavras no *corpus*. Já as Figuras 19 (a) e 19 (b) mostram algumas regiões específicas da Figura 18 com mais detalhes. É possível observar que personagens da história estão agrupados na mesma região enquanto o mesmo acontece para diferentes pratos de comida.



(a) Personagens de Game of Thrones (b) Comida de Game of Thrones

Figura 19: Clusters formados pelos livros de Game of Thrones

4.2.3 *GloVe*

Após o lançamento de *Word2Vec* em (MIKOLOV et al., 2013), houve um aumento considerável no número de artigos sobre diferentes tipos de *word embeddings*. Um dos trabalhos mais bem sucedidos na área foi (PENNINGTON; SOCHER; MANNING, 2014), que explicou por que tais algoritmos funcionam e reformula os processos de otimização obtidos por *Word2Vec* como um tipo especial de fatorização para matrizes de ocorrência conjunta de palavras.

O *GloVe* aprende os vetores de palavras examinando as ocorrências de palavras dentro de um *corpus* de texto. Antes de treinar o modelo, é preciso construir uma matriz de ocorrência conjunta X, onde o elemento X_{ij} representa a frequência com que a palavra i aparece no mesmo contexto da palavra j. O algoritmo percorre o *corpus* apenas uma vez para construir a matriz X. O modelo usa apenas a matriz de ocorrência conjunta para formular *word embeddings*. O *corpus* de texto não é mais utilizado. Uma vez encontrada a matriz X, o modelo usa a Equação 5:

$$\vec{w}_i^T \vec{w}_j + \vec{b}_i + \vec{b}_j = log(X_{ij}), \tag{5}$$

para definir os vetores w_i e w_j. O objetivo é encontrar vetores que se relacionam com a ocorrência conjunta das palavras i e j, onde b_i e b_j são *biases* associados às palavras i e j respectivamente. Os vetores são obtidos minimizando-se a função de custo J dada pela Equação 6:

$$J = \sum_{i=1}^{V} \sum_{j=1}^{V} f(X_{ij})(\vec{w}_i^T \vec{w}_j + \vec{b}_i + \vec{b}_j - log X_{ij})^2, \tag{6}$$

esta equação avalia a soma de todos os erros quadráticos tendo como base a Equação 5, além de ser ponderada por uma função positiva f, onde V é o tamanho do vocabulário de palavras presentes no *corpus* de treinamento.

A escolha de f deve ser feita de modo a evitar que pares de palavras muito comuns (pares com altos valores de X_{ij}) causem um *offset* grande no processo de otimização. Um possível caso de função que pode ser utilizada é mostrado na Equação 7:

$$f(X_{ij}) = \begin{cases} \left(\frac{X_{ij}}{x_{max}}\right)^{\alpha} & X_{ij} \leq x_{max} \\ 1 & em\ outro\ caso. \end{cases} \tag{7}$$

Quando pares de palavras extremamente comuns são encontrados ($X_{ij} \leq xmax$), a função de ponderação irá limitar sua saída para 1. Para todos os outros pares de palavras, será retornado algum peso no intervalo $[0,1]$, onde a distribuição de pesos neste intervalo é decidida por α. A Figura 20 representa a função da Equação 7 para um α igual a $\frac{3}{4}$.

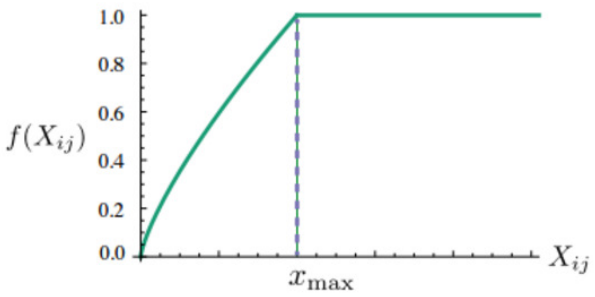


Figura 20: Função de ponderação

4.2.4 *fastText*

No final de 2016, o Facebook Research™ publicou um projeto de *word embeddings*: *fastText*, um método para aprender vetores e executar a classificação de texto. O objetivo principal dos *fastText embeddings* é levar em consideração a estrutura interna das palavras ao invés de aprender representações das palavras inteiras. Isto é notavelmente útil para linguagens morfologicamente ricas, de modo que as representações para diferentes formas morfológicas de palavras seriam aprendidas de forma independente (BOJANOWSKI et al., 2016).

O *fastText* funciona deslizando uma janela sobre o texto de entrada e aprendendo todas as palavras do contexto a partir da palavra central assim como o *Skip-Gram*. O aprendizado pode ser visto como uma série de atualizações para uma rede neural com duas camadas de pesos e três camadas de neurônios, nas quais as duas camadas externas possuem um

neurônio para cada palavra no vocabulário e a camada intermediária possui tantos neurônios quanto existem de dimensões no espaço vetorial utilizado. Esta abordagem é muito semelhante ao *Word2Vec*. Não uma coincidência, uma vez que ambos foram idealizados pelo mesmo autor: Tomas Mikolov (MIKOLOV et al., 2013) e (BOJANOWSKI et al., 2016). No entanto, ao contrário do *Word2Vec*, *fastText* também pode aprender vetores para sub-partes de palavras: os chamados n-gramas. Isso garante que, por exemplo, as palavras amor, amado e amando tenham representações vetoriais semelhantes, mesmo que elas tendam a aparecer em diferentes contextos. Este recurso melhora a aprendizagem em linguagens fortemente flexionadas (BOJANOWSKI et al., 2016). A diferença entre *fastText* e *Skip-Gram* é mostrada na Equação 8:

$$P(w_{i+j}|w_i) = \frac{exp(h_{w_i} \cdot w_{i+j})}{\sum_{k \epsilon V} exp(h_{w_i} \cdot w_k)}, \tag{8}$$

onde a função de probabilidade dada pela Equação 3 é alterada de modo que os *n-gramas* sejam contabilizados.

A representação da soma dos vetores obtidos para os *n-gramas* da palavra w_i é dada pela Equação 9:

$$h_{w_i} = \sum_{g \epsilon w_i} x_g, \tag{9}$$

onde g e x_g representam o conjunto de possíveis n-gramas e suas representações vetoriais respectivamente.

A metodologia do *fastText* considera que uma palavra é a soma de suas partes. Isso faz com que seus vetores sejam muito eficientes na representação de palavras, porém o processo de treinamento é mais demorado comparado a outros modelos.

4.2.5 Outros Modelos

Eigenwords são computados como os vetores singulares da matriz de ocorrência conjunta de palavras e seus contextos. Eles podem ser inconscientes do contexto, não dependendo do contexto da palavra, mas apenas da palavra ou sensível ao contexto, dependendo do contexto (DHILLON; FOSTER; UNGAR, 2015).

WordRank é um algoritmo de incorporação de palavras que estima os *word embeddings* através de um ranqueamento robusto. Semelhante ao *GloVe*, o treinamento do *WordRank* é realizado na matriz agregada de ocorrência conjunta de palavras de um *corpus*.

Todavia, difere do *GloVe* onde uma função de custo por regressão é empregada, o *WordRank* otimiza uma função de custo baseada em ranqueamento (JI et al., 2015).

O *VarEmbed* é um modelo que propõe melhorar os *word embeddings* através de informações morfológicas, capturando recursos de sub-palavras compartilhadas. Combina informações morfológicas e distributivas em um quadro probabilístico unificado, em que o *embedding* é uma variável latente. A informação morfológica fornece uma distribuição prévia sobre os *word embeddings*, que por sua vez condicionam uma função de verossimilhança sobre um *corpus* observado (BHATIA; GUTHRIE; EISENSTEIN, 2015).

4.3 Considerações Finais do Capítulo

Com o advento do aprendizado profundo, os *word embeddins* obtiveram grande popularide na comunidade científica. Seu propósito é transformar palavras em números, os quais os algoritmos de aprendizado profundo possam consumir e processar para formular uma compreensão da linguagem natural de forma eficiente.

Os vetores gerados pelos modelos possuem propriedades de similaridade semântica que são bastante interessante para solução de problemas complexos do NLP. Contudo, por serem parte importante do aprendizado profundo, são geralmente denominados, erroneamente, como modelos profundos. Na verdade, os *word embeddings* são obtidos por algoritmos que exploram a linearidade e não-profundidade. Outro erro comum que é cometido é pensar que há diferença qualitativa entre os modelos neuro-linguísticos e os baseados em contagem. Eles são diferentes formas matemáticas de se chegar ao mesmo modelo semântico. Vários trabalhos recentes demonstraram, tanto teórica como empiricamente, a correspondência entre esses diferentes tipos de modelos (LEVY; GOLDBERG, 2014) e (ÖSTERLUND; ÖDLING; SAHLGREN, 2015). Neste trabalho, são usados *Word2Vec*, *GloVe* e *fastText* devido a sua maior popularidade entre outros autores e a disponibilidade de encontrá-los gratuitamente na Internet.

Capítulo 5

ASPECTOS DE IMPLEMENTAÇÃO

NESTE capítulo são apresentados os aspectos e considerações técnicas adotadas na implementação de um sistema de classificação de sentenças por redes neurais de convolução. Ao longo deste capítulo, são descritas: as especificações técnicas do sistema; as características dos conjuntos de dados usados nos treinamentos das redes; as etapas de pré-processamento aplicadas aos dados de treinamento; a configuração do modelo de referência; as configurações testadas nesta dissertação; e a metodologia adotado durante os experimentos.

5.1 Especificações Técnicas

A implementação dos experimentos deste trabalho é feita usando a linguagem de programação Python (ROSSUM, 1995). Esta foi escolhida devido à sua simplicidade e grande número de módulos bem documentados que auxiliam desde o pré-processamento de dados à aplicação de modelos de aprendizado de máquinas. Apesar do Python não ser uma linguagem tão rápida quanto Julia (BALBAERT, 2015) ou Scala (ODERSKY; SPOON; VENNERS, 2008), suas operações com maior custo computacional podem ser facilmente executadas por interfaces em C (KERNIGHAN, 1988) ou Fortran (PRESS et al., 1993). Neste trabalho, os principais módulos utilizados são: *Tensorflow* (ABADI et al., 2015), *Scikit-Learn* (PEDREGOSA et al., 2011) e *Numpy* (WALT; COLBERT; VAROQUAUX, 2011). Estes são módulos de computação científica que possuem interfaces em C e que permitem a implementação tanto de redes neurais artificiais quanto de outros algoritmos com maior agilidade e com códigos mais eficientes por possuírem APIs (*Application Programming Interface*) de alto nível, as quais reduzem de maneira significativa a quantidade de código necessário para a modelagem e simulação de alguns modelos de aprendizado de máquinas.

O *Tensorflow* é usado para modelar as redes neurais de convolução testadas nesta dissertação. Este é uma biblioteca de *software* de código aberto para computação numérica usando gráficos de fluxo de dados. Os nós no gráfico representam operações matemáticas, enquanto as bordas do gráfico representam os dados sendo processados ao longo da topologia. A arquitetura flexível permite implantar computação para uma ou mais CPUs (*Central Processing Unit*) ou GPUs (*Graphics Processing Unit*) com uma única API. Esta permite um rápido e fácil interfaceamento com CUDA (*Compute Unified Device Architecture*) (NICKOLLS et al., 2008) que é uma extensão da linguagem C e que possibilita o uso de computação paralela. O *Tensorflow* foi originalmente desenvolvido para fins de pesquisa em redes neurais profundas, mas o sistema é geral o suficiente para ser aplicável em uma ampla variedade de outros domínios. A Figura 21 mostra a arquitetura de rede neural de convolução usada neste trabalha simulada no *Tensorflow*. Na figura, os dados de entrada são transformado em *word embeddings* pela camada *Embeddings*; em seguida o texto transformado é submetido a 3 filtros de convolução de diferentes tamanhos identificados na figura por *Conv1D*, *Conv1D-1* e *Conv1D-2*; os mapas obtidos através dessas operações são concatenados pelo bloco *Merge* e submetidos ao processo de *pooling* por *GlobalMaxPooling*; em seguida os dados são submetidos ao processo de *dropout* pelo bloco com o mesmo nome; e finalmente é usada uma rede MLP identificada por *FullyConnected* para definir a classificação do texto. Esta rede funciona de maneira idêntica à rede mostrada na Figura 14 do Capítulo 3. Os outros blocos não descritos, representam operações de auxílio ao processo de treinamento .

Para simular modelos de aprendizado de máquinas usados como referência neste trabalho, usa-se o *Scikit-Learn*. Este é um módulo de aprendizado de máquinas para Python criado sobre o *SciPy* (JONES et al., 2001–), que é um ecossistema Python para computação científica. Este possui uma variedade muito grande de modelos de regressão, classificação e clusterização de fácil implementação devido à sua simples API e grande documentação .

As operações matriciais feitas nos experimentos fazem uso do módulo *Numpy* que é uma biblioteca que adiciona suporte para grandes vetores e matrizes multidimensionais, juntamente com uma grande coleção de funções matemáticas de alto nível.

Os experimentos feitos neste trabalho fazem uso de um ambiente composto por sistema operacional *Linux*, distribuição *Ubuntu 16.04*, processador *Intel Core i7* com 2,6 *GHz*, *16GB* de memória RAM e placa de vídeo *NVIDIA GeForce GTX 1060 GPU* com 6*GB* de memória dedicada.

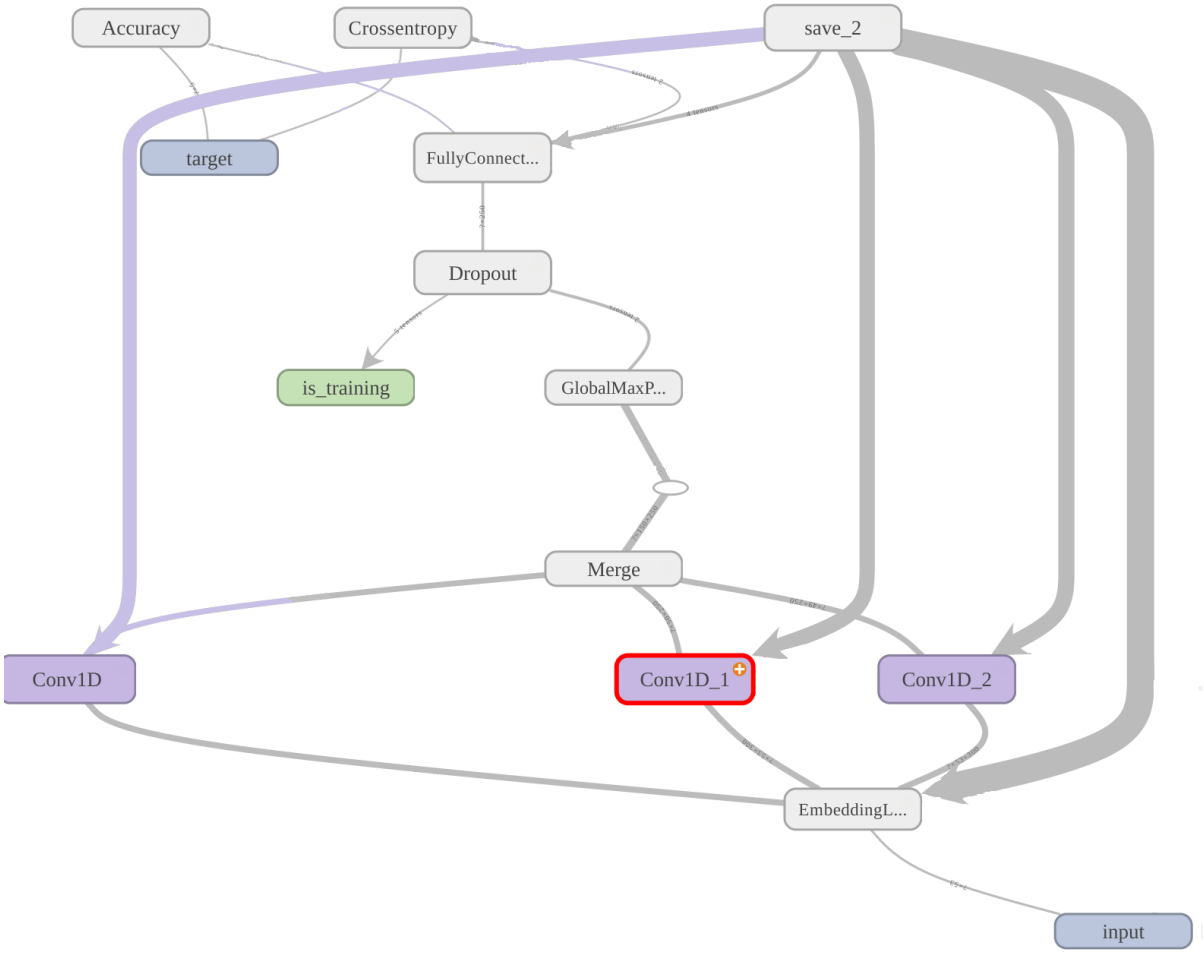


Figura 21: Visualização de Rede de Convolução pelo *Tensorflow*

5.2 Pré-processamento dos Dados de Treinamento

Os dados usados nos experimentos desta dissertação são os mesmos usados em (KIM, 2014). São dados formados por sentenças escritas no idioma inglês e são usados em diversos trabalhos que tratam de análise de sentimentos. Estes dados podem ser encontrados facilmente na Internet. Alguns de seus aspectos estão descritos na Tabela 1, onde c representa o número de classes, l representa o comprimento médio da sentença, N é o tamanho do conjunto de dados e V é o tamanho do vocabulário.

Tabela 1: Aspectos dos dados usados durante o treinamento

Conjunto de dados	c	l	N	V
MR	2	20	10662	18765
SST-1	5	18	11855	17836
SST-2	2	19	9613	16185

- *MR*: Conjunto de dados de críticas de filmes com uma sentença por crítica. As críticas tem 2 classes possíveis: positiva (Pos) e negativa (Neg) (DATA, 2002) .

- *SST-1*: *Stanford Sentiment Treebank* é uma extensão do MR com divisões de treinamento, teste e validação pré-determinadas. Possui 5 classes: muito positivo (Mpo), positivo (Pos), neutro (Neu), negativo (Neg) e muito negativo (Mne) (GLOVE..., 2014) .

- *SST-2*: Derivado do SST-1, porém sem críticas neutras e apenas com 2 classes. As críticas muito positivas se tornam positivas enquanto as muito negativas se tornam negativas (GLOVE..., 2014).

As classes presentes nestes dados estão todas bem balanceadas. Isto significa que as sentenças estão bem distribuídas pelas classes. Não há nenhuma classe com um número muito maior de sentenças em relação à outra. Deste modo, a classificação de sentimentos nestes conjuntos de dados pode ser interpretada como um problema de classificação. Caso houvesse alguma assimetria na distribuição destes dados, uma abordagem por detecção de anomalias pudesse ser mais adequada.

Os experimentos fazem uso de vetores (*word embeddings*) pré-treinados e disponíveis publicamente. Estes são usados para codificar os textos em uma representação de baixa dimensionalidade. Os *embeddings* usados nesta dissertação são: vetores *fastText*, lançados pela equipe de pesquisa do Facebook™; vetores *Word2Vec* treinados em 100 bilhões de palavras do Google News™; e *GloVe* que é treinado em um *corpus* de 840 bilhões de tokens de dados da *Web* e distribuído pela Universidade de Stanford. Cada um dos vetores presentes nestes conjuntos possui dimensão $300x1$.

O pré-processamento destes dados possui 6 etapas. Estas têm como objetivo remover ruído presente nas sentenças e codificar os textos de maneira que seja possível usá-los no treinamento de redes de convolução. As etapas descritas aqui tentam seguir o mesmo pré-processamento feito por Kim em seu trabalho. Contudo, não está claro se as etapas 3 e 4 que foram adotadas nesta dissertação foram utilizadas em (KIM, 2014).

- Etapa 1: Caracteres especiais e sinais de pontuação são removidos.

- Etapa 2: Todas as palavras são escritas em letras minúsculas.

- Etapa 3: Números são substituídos por um token especial: NUM.

- Etapa 4: As *stopwords* são removidas. Estas são palavras que não acrescentam muito significado ao texto. Geralmente são artigos, conjunções e preposições.

- Etapa 5: Substitui-se as palavras contidas nos conjuntos de dados por seus respectivos vetores pré-treinados.

- Etapa 6: As palavras, que estão presentes nos conjuntos de dados, mas não nos vetores de palavras pré-treinados, são inicializadas aleatoriamente usando uma distribuição normal com média 0 e desvio-padrão 1.

5.3 Configurações Investigadas

O objetivo desta dissertação consiste em criar um classificador de análise de sentimentos eficiente usando redes neurais de convolução por meio da análise do impacto de seus hiperparâmetros no desempenho do modelo. Em (KIM, 2014), são encontrados resultados melhores que os alcançados por modelos simples de aprendizado de máquinas como NB e SVM, e além disso, encontra resultados competitivos com modelos no estado da arte para análise de sentimentos em sentenças. Todavia, o autor obteve tais resultados sem testar diferentes configurações de hiperparâmetros. Em seu trabalho, ele justifica a ausência desta análise por falta de *hardware* capaz de treinar várias redes em tempo razoável. Nesta dissertação, algumas possibilidades de configurações de redes neurais de convolução serão investigadas tendo como referência a configuração adota em (KIM, 2014). Com isso, será possível avaliar se há alguma melhora de desempenho significativa ao se ajustar os hiperparâmetros de uma CNN.

A configuração de hiperparâmetros de referência utilizada neste trabalho está descrita na Tabela 2. Tomando esta topologia como a de referência para a rede de convolução, considera-se o efeito de diferentes decisões de arquitetura sobre o desempenho da rede. Infelizmente, mesmo fazendo uso de GPUs, é inviável testar todas as combinações possíveis de hiperparâmetros devido ao alto tempo de treinamento de cada rede. Sendo assim, para se explorar os universo de hiperparâmetros de maneira viável segue-se a seguinte metodologia: seleciona-se um hiperparâmetro dentre os 7 possíveis contidos na Tabela 2; fixa-se todos os outros 6 de acordo com os valores apresentados na Tabela 2; varia-se o hiperparâmetro selecionado dentro do universo de possibilidades presentes na Tabela 3. Cada variação de hiperparâmetro é considerada uma nova topologia de CNN e, para cada uma, o experimento

é repetido por 10 vezes seguindo um esquema de validação cruzada (*k-fold*). Cada configuração é treinada por 50 épocas e o critério de parada adotado é a Parada-Antecipada. Esta é uma forma de se evitar o sobreajuste de um algoritmo de aprendizado de máquinas. Os resultados obtidos com os treinamentos são submetidos a um teste de significância estatística com o objetivo de avaliar se há diferença significativa de desempenho dependendo da configuração adotada e se algum indício de padrão de comportamento apresentado nos diferentes conjuntos de dados.

Tabela 2: Configuração Topológica de Referência (KIM, 2014)

Hiperparâmetro	Valor
Word Embedding	Word2Vec estático
Função de Ativação	ReLU
Tamanho do Filtro	(3, 4, 5)
Mapas de Características	100
Pooling	Pooling máximo
CR	3
DO	0.5

Tabela 3: Hiperparâmetros Testados

Hiperparâmetro	Valores Testados
Word Embedding	W2VE, W2VD, GVE, GVD, fTE, fTD
Função de Ativação	ReLU, Sigmoid, Tanh, Linear
Tamanho do Filtro	F1, F2,..., F5
Mapas de Característica	50, 100,..., 250
Pooling	Pooling máximo, Pooling médio
CR	0, 3,..., 12
DO	0,0; 0,2; 0,5; 0,7; 0,9

A métrica de avaliação usada durante os experimentos é a acurácia. Esta é escolhida porque os dados estão bem balanceados. Caso os dados fossem assimétricos, outra métrica deveria ser escolhida: AUC (área sob a curva ROC) poderia ser usada nesse caso. Os valores de acurácia obtidos são submetidos a um teste de significância estatística. Devido ao baixo número de amostras, o teste estatístico usado é o Teste-T de Stutent(GOSSET, 1908). Este é o teste indicado na literatura para situações com baixo número de dados.

5.3.1 *Word Embeddings*

Word embeddings são a forma de representação dos dados usados neste trabalho. Eles mapeiam palavras a vetores de baixa dimensionalidade. Neste trabalho, são experimentados *Word2Vec*, *GloVe* e *fastText*. O uso destes vetores durante os experimentos é dividido em 2 situações: Estático, quando o vetor é usado na forma como é obtido na origem, e Dinâmico, as componentes do vetor são consideradas como pesos da rede e são ajustadas durante o treinamento, sendo o objetivo deste processo tentar adaptar os vetores aos dados treinados. Os valores de *word embeddins* testados neste trabalho são *Word2Vec* Estático (W2VE), *Word2Vec* Dinâmico (W2VD), *GloVe* Estático (GVE), *GloVe* Dinâmico (GVD), *fastText* Estático (fTE) e *fastText* Dinâmico (fTD) .

5.3.2 Função de Ativação

A função de ativação é o elemento que adiciona não-linearidades aos dados que fluem pela rede neural. Através destas, as redes podem compor abstrações sobre os dados (NIELSEN, 2015). As funções de ativação usadas são *Rectified Linear Unit* (ReLU), Sigmoid, Tangente Hiperbólica (Tanh) e Linear. Estas estão definidas nas Equações 10 - 13:

$$f(x) = max(0, x); \tag{10}$$

$$f(x) = \frac{1}{1 + e^{-x}}; \tag{11}$$

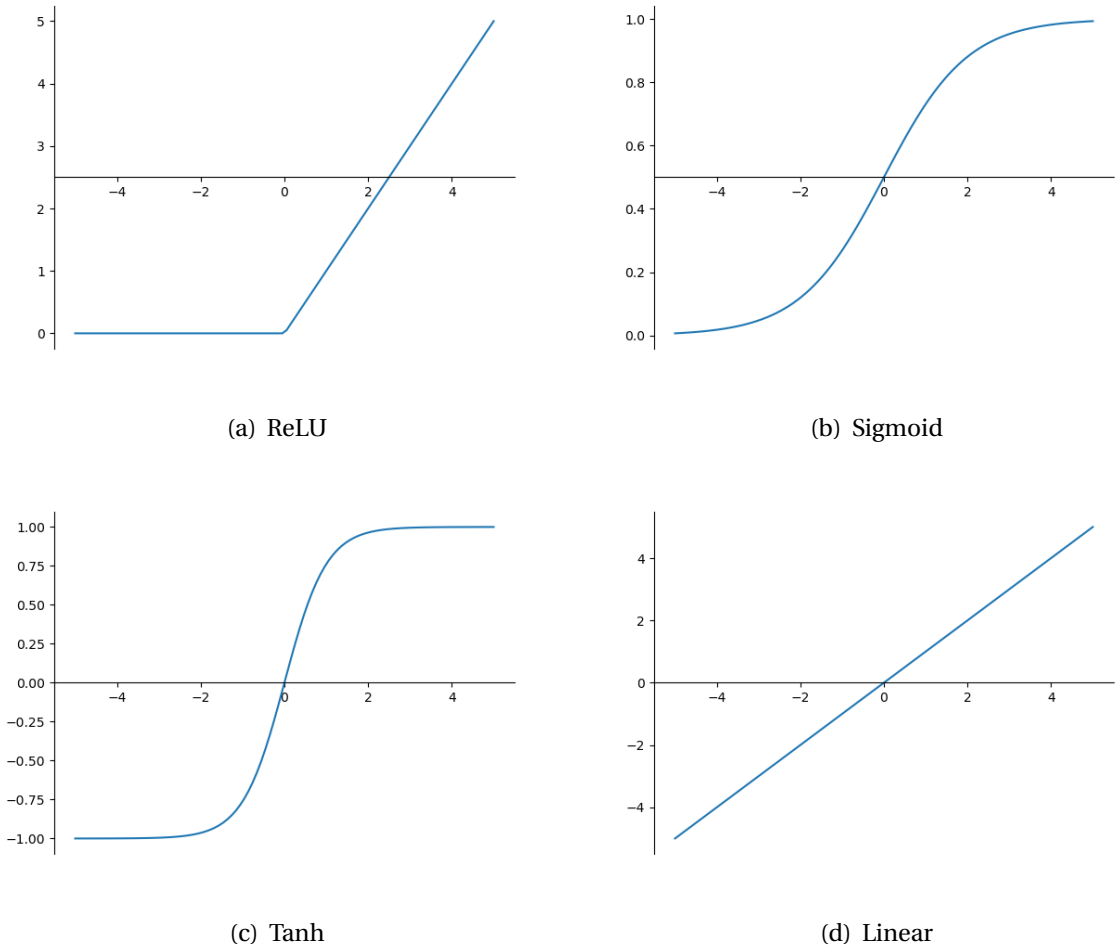
$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}}; \tag{12}$$

$$f(x) = x. \tag{13}$$

Os comportamentos das função de ativação escolhidas são ilustrados na Figura 22 . Note que as funções Sigmoid e Tanh sofrem o efeito do desaparecimento do gradiente por possuírem derivadas próximas de 0, ao contrário de outras duas funções.

5.3.3 Tamanho do Filtro

Os filtros são os elementos de uma CNN que aplicam operações de convolução sobre os dados. O tamanho dos filtros está diretamente associado ao número de pesos da rede neural. Maior o filtro, maior o número de pesos a serem treinados. Os tamanhos dos filtros podem ser interpretados como n-gramas, ou seja, um filtro de n linhas opera convolução sobre n



(a) ReLU

(b) Sigmoid

(c) Tanh

(d) Linear

Figura 22: Funções de ativação utilizadas

palavras. Os filtros são indicados por uma terna não-ordenada (*X, Y, Z*) onde cada componente representa um tamanho de filtro. O tamanho horizontal é fixo e é dado pelo tamanho dos *word embeddings*, que neste trabalho são 300 unidades. Enquanto a altura é dada por *X, Y* e *Z*. Uma terna (1, 2, 3) representa 3 filtros de tamanhos $300x1$, $300x2$ e $300x3$. Nos experimentos, são usados filtros de tamanhos (1, 2, 3), (2, 3, 4), (3, 4, 5), (4, 5, 6) e (5, 6, 7) identificados por F1, F2, F3, F4 e F5 respectivamente sendo F3 a referência usada em (KIM, 2014).

5.3.4 Mapas de Características

Os Mapas de Características são responsáveis por identificar diferentes características presentes nos dados. Cada mapa está associado a um atributo do dado de entrada. Em sistemas de visão computacional, estes identificam atributos como nariz, olhos e boca. De modo se-

melhante, estes podem identificar características como tempo verbal, gênero e plural em tarefas de *NLP*. A Figura 10 do Capítulo 3 mostra mapas de características, também chamados de mapas de atributos, para uma rede de convolução de imagens. Os experimentos desta dissertação testam a variação da quantidade de mapas de características. Maior o número de mapas, maior será o número de características identificadas, porém maior também será o tempo de treinamento. São testadas redes com 50, 100, 150, 200 e 250 mapas.

5.3.5 *Pooling*

Pooling é uma operação para simplificar a informação entregue na saída de um mapa de características. A informação pode ser transformada por meio da média dos valores de um mapa ou pelo valor máximo do mapa.

5.3.6 Constante de Regularização

A regularização é uma técnica utilizada na tentativa de resolver o problema de sobreajuste em modelos estatísticos. Uma forma de regularização muito utilizada é a regularização *L2-norm*. Para aplicá-la, penaliza-se a função de custo C_0 da rede neural, conforme mostrado na Equação 14:

$$C = C_0 + \lambda \sum_w w^2, \tag{14}$$

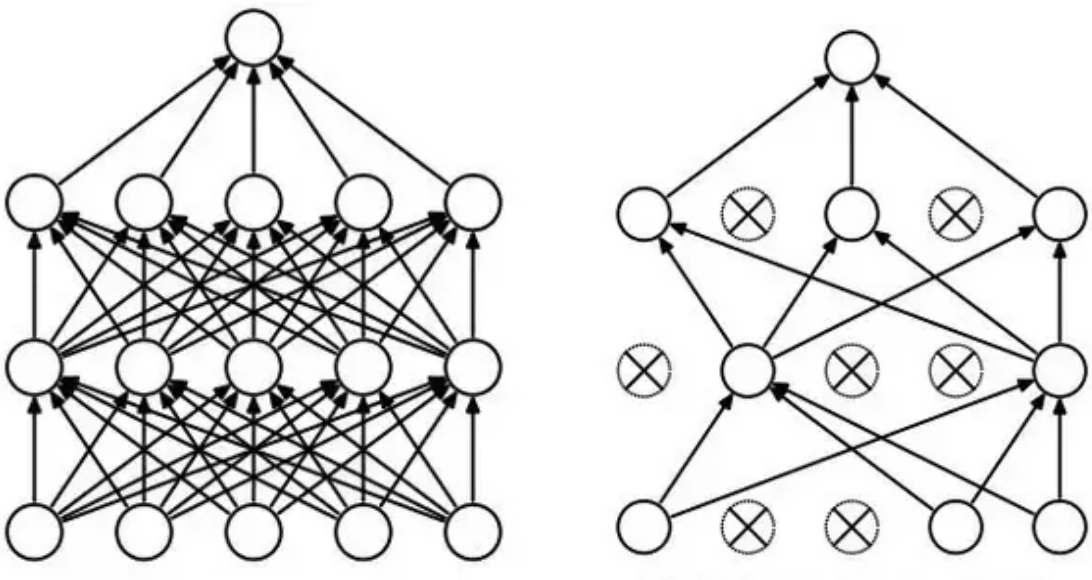
onde C é a nova função de custo, w são os os pesos da rede e λ é a constante real não-negativa que determina o grau de regularização. Esta é chamada de constante de regularização (CR). Ao aumentar o valor de λ, a rede precisa diminuir os valores w para que a nova função de custo seja minimizada adequadamente. Os valores de CR testados são: 0, 3, 6, 9 e 12.

5.3.7 Taxa de *Dropout*

O *dropout* é uma forma de regularização. Este faz com que uma rede neural artificial aprenda múltiplas representações independentes dos dados, alternando aleatoriamente os neurônios desabilitados na fase de treinamento. É importante destacar que esta técnica de regularização só é utilizada durante o treinamento.

A taxa de *Dropout* (DO) varia no intervalo $[0,1)$, sendo que quão maior a taxa, maior a quantidade neurônios que não são ajustados durante o treinamento da rede. Isso faz com

que os poucos neurônios que restam aprendam formas de suprimir a ausência dos outros. Deste modo, cria-se um sistema de redundância onde um neurônio pode responder por outro. A Figura 23 mostra a aplicação de *dropout* em uma rede neural. Os valores de DO testados são: 0,0; 0,2; 0,5; 0,7; 0,9.



(a) Antes do *Dropout* (b) Após o *Dropout*

Figura 23: Aplicação de *Dropout*

5.4 Considerações Finais do Capítulo

Este capítulo apresentou todas as considerações técnicas usadas na implementação de CNNs aplicadas à análise de sentimentos. Os dados e as etapas de pré-processamento foram enumeradas. A metologia dos experimentos foi apresentada juntamente com as possíveis configurações de hiperparâmentros que determinam a topologia da rede. No capítulo seguinte, são apresentados os resultados obtidos pelas diversas configurações topológicas de uma CNN quando treinadas usando os conjuntos de dados descritos neste capítulo.

Capítulo 6

RESULTADOS DE DESEMPENHO

NESTE capítulo são apresentados e analisados os resultados alcançados com a metodologia descrita no Capítulo 5. Na Seção 6.1, são expostos os resultados referentes às diferentes configurações de CNN apresentadas na Seção 5.3 do Capítulo 5. Na Seção 6.1, são relatados os resultados do teste estatístico na comparação de desempenho entre as arquiteturas. A Seção 6.2 mostra a comparação de desempenho entre as melhores redes obtidas na seção anterior e os modelos de trabalhos relacionados. Vale ainda ressaltar que os resultados completos obtidos nos experimentos estão contidos nos 3 anexos ao final desta dissertação.

6.1 Avaliação das Topologias Propostas

A comparação entre as diversas configurações de CNN usa a acurácia como métrica. O teste estatístico de comparação escolhido é o Teste-T de Student bilateral com 9 graus de liberdade e um valor de T igual a 2,262 (α = 95%). As redes foram treinadas usando uma taxa de aprendizado de $0,001$ e o tempo médio de treinamento foi de 30 minutos para cada rede. Este varia de acordo com o hiperparâmetro avaliado.

6.1.1 *Word Embeddings*

Os resultados apresentados nesta seção correspondem à primeira variação topológica proposta: *word embeddings*. A rede foi treinada com 3 tipos diferentes de representações vetoriais. Contudo, os experimentos foram realizados de modo que cada representação possa assumir 2 tipos de estado conforme explicado na Seção 5.3 do Capítulo 5.

A Tabela 4 apresenta os resultados do Teste-T de Student aplicado na comparação das diferentes configurações. O valores em vermelho apontam alguma diferença relevante no desempenho, ou seja, a hipótese nula não é válida.

Tabela 4: Resultados do Teste-T ($T = 2,262$ e $\alpha = 95\%$) considerando *Embedding*

Word Embedding	MR	SST-1	SST-2
W2VE - W2VD	1,798	2,496	1,894
W2VE - GVE	0,575	3,142	0,525
W2VE - GVD	3,577	2,109	2,897
W2VE - fTE	0,574	1,773	0,445
W2VE - fTD	2,502	3,551	2,502
W2VD - GVE	1,767	0,527	1,657
W2VD - GVD	0,780	0,477	0,680
W2VD - fTE	1,165	0,662	1,467
W2VD - fTD	0,331	0,105	0,354
GVE - GVD	3,111	1,032	3,111
GVE - fTE	0,796	1,339	0,789
GVE - fTD	1,684	0,758	1,684
GVD - fTE	2,640	0,222	2,450
GVD - fTD	1,334	0,706	1,244
fTE - fTD	1,245	0,848	1,375

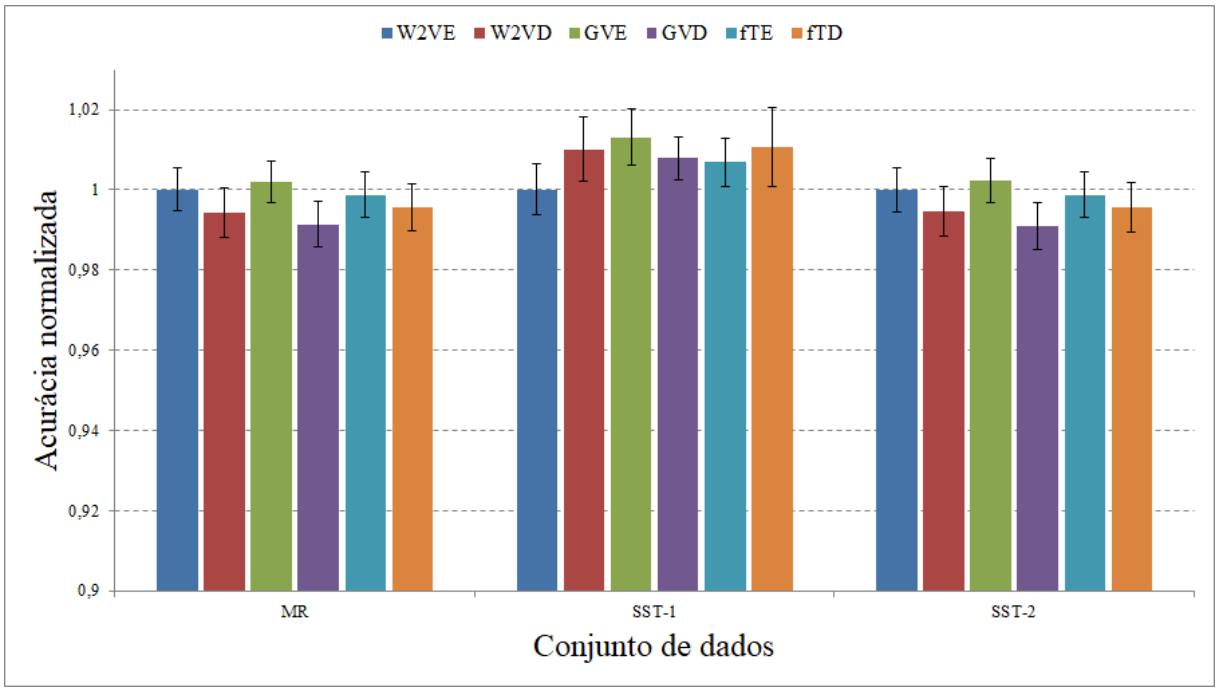


Figura 24: Impacto do tipo de *Word Embedding* na acurácia

A Figura 24 mostra os valores de acurácia normalizada em relação ao modelo de referência da Tabela 2 obtidos para todas as configurações de *word embeddings* testadas. Há uma perda de desempenho ao se usar as representações fTD (80,6% e 86,2% de acurácia) e GVD (80,2% e 84,7%) em comparação à referência de W2VE (80,8% e 86,6%) nos conjuntos MR e SST-2 respectivamente. Nestes conjuntos, o ganho de desempenho é menor que

1. Em SST-1, a comparação das representações W2VD (46,2%), GVE (47,7%) e fTD (46,6%) com a referência (45,7%) não confirma a hipótese nula. Nestes casos, há ganhos reais de desempenho ao se usar estas formas de representação em relação à referência.

Pela Tabela 4, é possível observar que a CNN se comporta de maneira semelhante em MR e SST-2, ambos são conjuntos com 2 classes de dados (positiva e negativa). Os Testes indicam que pode haver um padrão de comportamento: vetores estáticos funcionam melhor que dinâmicos para dados binários. A Tabela 4 também mostra que para o conjunto SST-1 que possui 5 classes, há uma melhora significativa de desempenho quando são usados vetores dinâmicos com exceção de *GloVe*. Para este, a versão estática funciona melhor.

6.1.2 Função de Ativação

Nesta seção, avalia-se o comportamento da CNN mediante à variação da função de ativação. Quatro tipos de funções foram escolhidas: ReLU, Sigmoid, Linear e Tanh. Estas foram escolhidas por serem as mais comumente usadas em aprendizado de máquinas.

Pela Tabela 5, é possível observar que a hipótese nula na comparação da função de ativação ReLU com Tanh e Sigmoid não se confirma para os 3 conjuntos de dados testados. ReLU que é a referência tem desempenho superior (80,8%, 45,7% e 86,6%) à Tanh (80,2%, 41,2% e 86,1%) e Sigmoid (79,9%, 26,1% e 85,9%) em MR, SST-1 e SST-2 respectivamente. O motivo talvez seja por ReLU não sofrer com derivadas nulas durante o treinamento. Quando a derivada da função de ativação se anula durante o treinamento, o aprendizado fica comprometido. As funções Tanh e Sigmoid podem sofrer deste fenômeno que é chamado de Desaparecimento do Gradiente (NIELSEN, 2015). Também é interessante observar que a função de ativação Linear se comporta de maneira similar à ReLU. Isto indica que não é preciso introduzir não-linearidades para que a CNN consiga detectar os sentimentos presentes nas sentenças.

Tabela 5: Resultados do Teste-T (T= 2,262 e α = 95%) considerando a Função de ativação

Modelos	MR	SST-1	SST-2
ReLU - Sigmoid	2,539	17,485	2,754
ReLU - Tanh	3,138	8,380	2,655
ReLU - Linear	0,296	1,416	0,167
Sigmoid - Tanh	0,499	14,066	0,674
Sigmoid - Linear	2,870	17,255	2,67
Tanh - Linear	2,270	10,073	2,232

A Figura 25 permite ver que o uso das funções Tangente Hiperbólica e Sigmoid causam uma perda de desempenho significativa. Além disso é possível concluir que o conjunto SST-1 é muito mais sensível ao tipo de função de ativação escolhido. Neste conjunto, há uma perda de aproximadamente 10% em relação à referência ao se usar Tanh e 45% com Sigmoid.

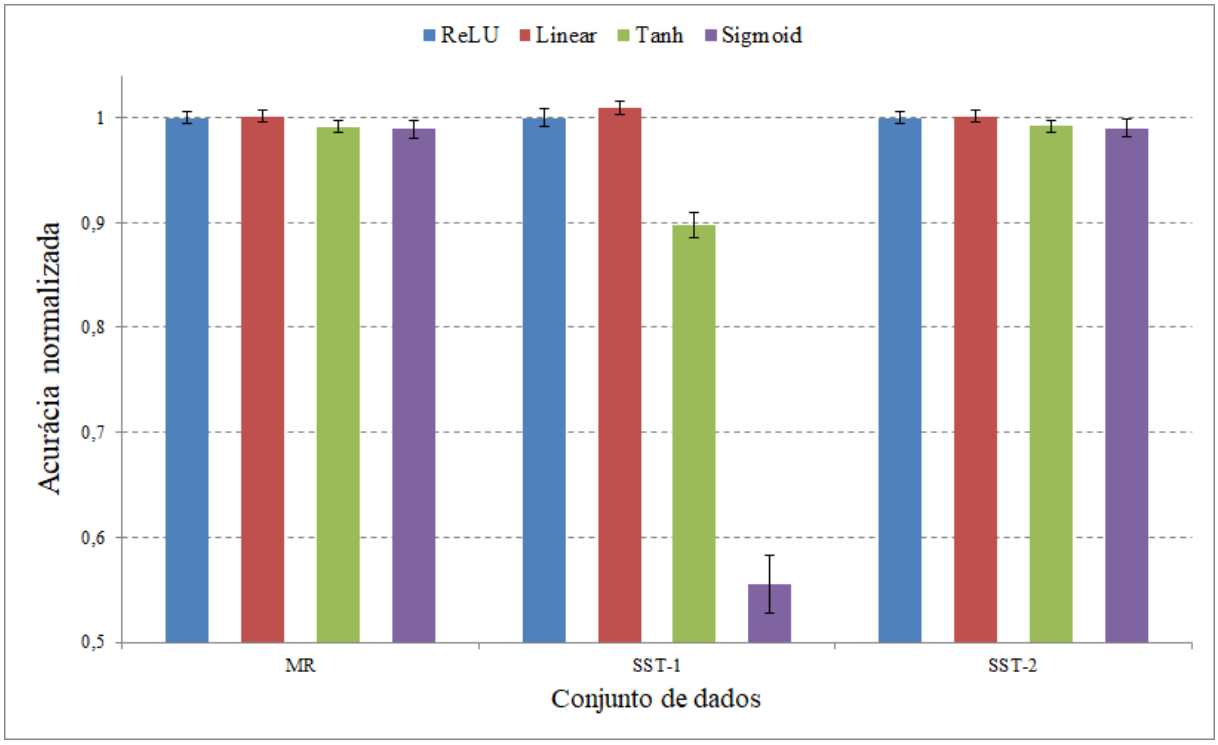
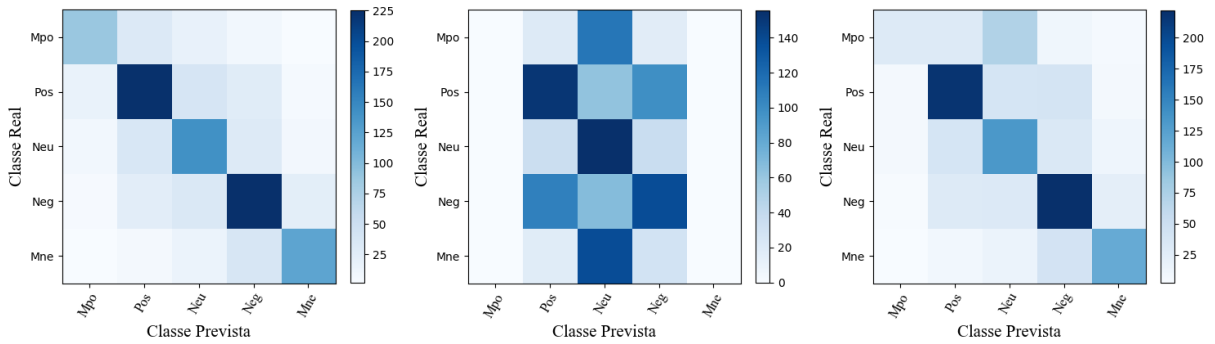


Figura 25: Impacto do tipo de Função de Ativação na acurácia



(a) ReLU e Linear (b) Sigmoid (c) Tanh

Figura 26: Matrizes de Confusão para SST-1

A Figura 26 mostra as matrizes de confusão no conjunto de dados SST-1 para as diferentes funções de ativação. É possível observar o fraco desempenho que as funções Sigmoid e Tanh alcançam quando comparadas a ReLu e Linear. A figura confirma os resultados obtidos pelo Teste-T. É interessante observar que a função Sigmoid funciona muito bem para

as classes mais centrais da matriz (Pos, Neu e Neg). Sendo mais eficiente que ReLU na identificação de sentenças neutras. Contudo, Sigmoid falha terrivelmente na identificação das classes Mpo e Mne, razão pela qual seu desempenho é muito inferior.

6.1.3 Tamanho do Filtro

O tamanho dos filtros de convolução é avaliado nesta seção. Os filtros são considerados como um conjunto de 3 filtros de diferentes dimensões.

A Tabela 6 indica que filtros menores como F1 e F2 têm desempenho pior que filtros maiores como F4 e F5. A hipótese nula não é confirmada em nenhum dos casos de comparação entre estes. O filtro F1 é a pior escolha pois proporciona uma queda no desempenho da CNN em qualquer um dos conjuntos de dados. A queda chega a 3,0% em relação à F3 em SST-1.

Tabela 6: Resultados do Teste-T (T= 2,262 e α = 95%) considerando o tamanho dos filtros

Tamanho do Filtro	MR	SST-1	SST-2
F3 - F1	3,340	3,965	3,625
F3 - F2	3,518	1,360	4,618
F3 - F4	0,121	0,399	0,311
F3 - F5	0,042	1,192	0,121
F1 - F2	2,823	3,505	3,762
F1 - F4	2,974	3,551	5,050
F1 - F5	4,324	5,385	2,574
F2 - F4	2,443	2,759	2,673
F2 - F5	2,932	2,841	2,632
F4 - F5	0,119	1,409	0,243

Apesar da Figura 27 indicar que há ganho considerável de F5 em relação à F3, a Tabela 6 indica que não há diferença de desempenho entre redes com estes filtros. Com isso, é possível concluir que é melhor se usar o filtro F3 pois este possui menos parâmetros a serem treinados. A mesma análise é válida entre F4 e F3.

O tamanho do filtro se traduz no número de palavras que são processadas de maneira conjunta. Filtros menores aparentemente falham na tarefa de capturar algum tipo de contexto que envolve as palavras. Enquanto filtros maiores podem ser mais eficientes neste aspecto. No entanto, o aumento contínuo do tamanho do filtro parece não trazer um desempenho continuamente crescente.

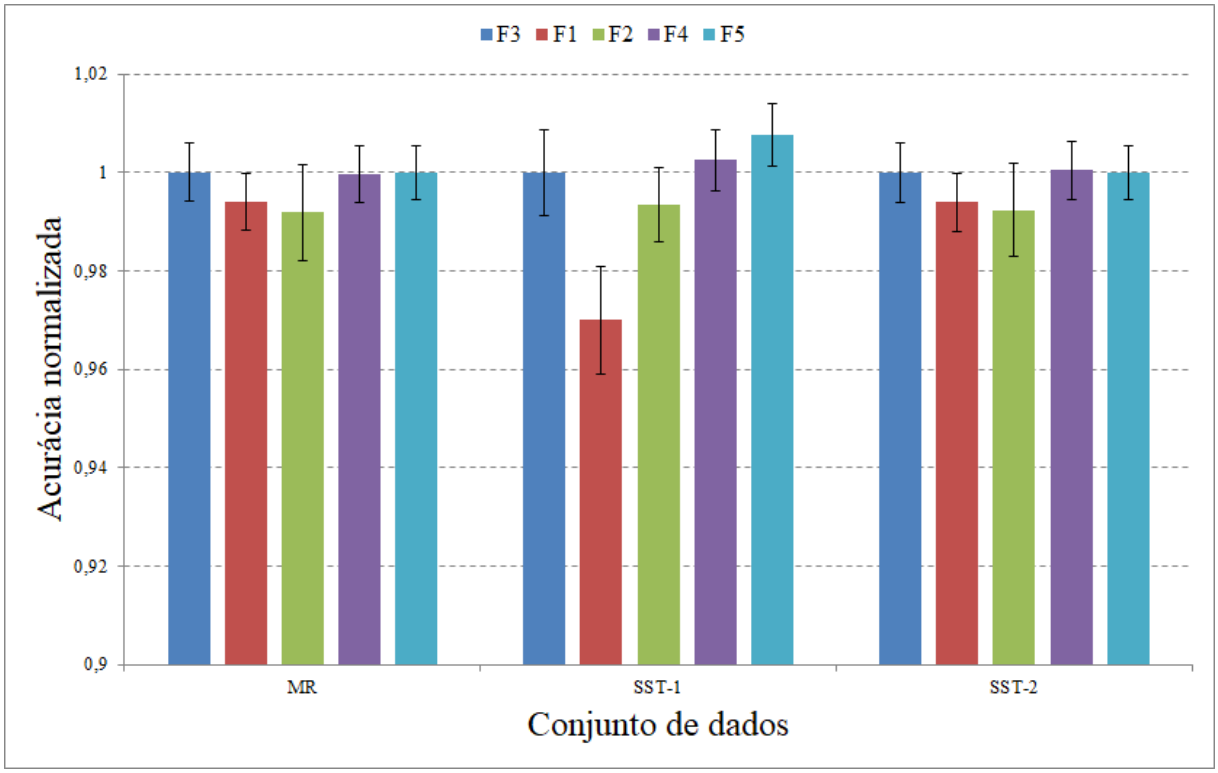


Figura 27: Impacto do Tamanho do Filtro na acurácia

6.1.4 Mapas de Características

Mantendo-se os hiperparâmetros constantes como indicado na Tabela 2 e variando-se o número de mapas de características observa-se pela Tabela 7 que os conjuntos MR e SST-2 são praticamente insensíveis à variação do número de mapas de características pois a hipótese nula se mantém verdadeira na maioria dos testes. Todavia, para SST-1, quão maior o número de mapas, melhor o valor de acurácia. A Figura 28 permite perceber que há, em SST-1, um ganho de 2% em relação à referência quando o número de mapas passa de 100 para 250 e que também há uma perda de aproximadamente 2% quando a quantidade de mapas cai para 50. Os mapas identificam características presentes nos dados. Em um conjunto com múltiplas classes, os mapas podem ajudar a caracterizar alguns atributos e assim melhorar o desempenho de classificação da CNN.

Tabela 7: Resultados do Teste-T (T= 2,262 e α = 95%) considerando os mapas de características

Número de Mapas	MR	SST-1	SST-2
100 - 50	0,466	2,885	0,466
100 - 150	0,270	1,291	0,250
100 - 200	0,617	3,370	0,217
100 - 250	1,784	4,746	1,784
50 - 150	0,106	3,368	0,432
50 - 200	0,133	6,695	0,133
50 - 250	1,592	4,791	1,345
150 - 200	0,259	1,869	0,254
150 - 250	0,954	2,874	0,636
200 - 250	1,682	0,752	1,842

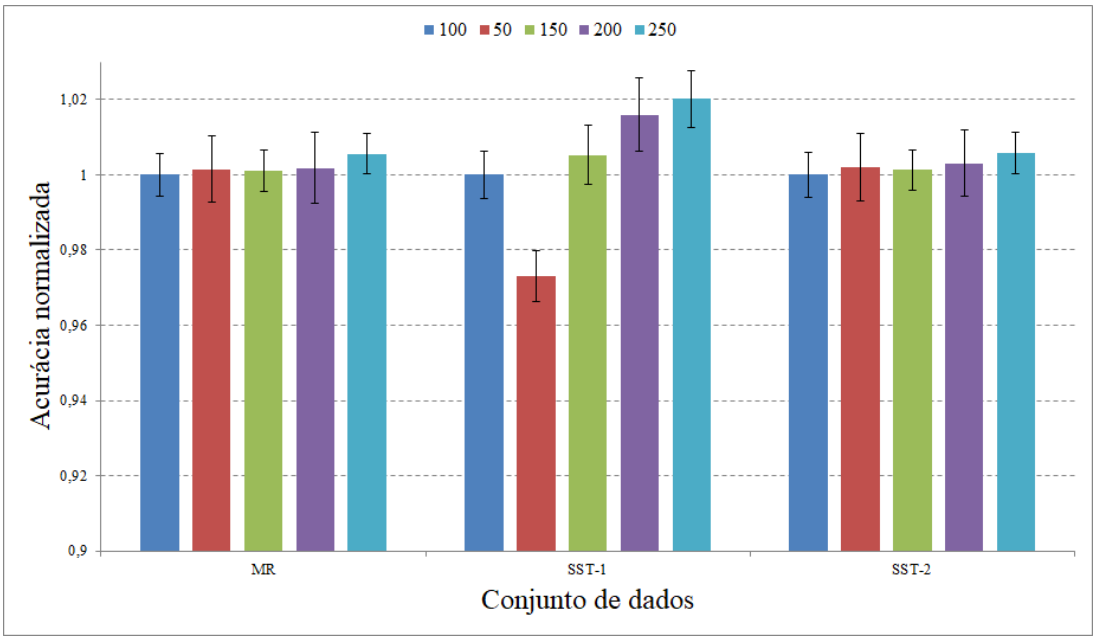


Figura 28: Impacto do número de Mapas de Características na acurácia

6.1.5 *Pooling*

São testados 2 estratégias de *Pooling* nesta seção, *pooling* máximo e o *pooling* médio. Os resultados desta classe de experimentos estão apresentados na Tabela 8 mostram que os conjuntos MR e SST-2 são insensíveis à variação deste hiperparâmetro pois a hipótese nula é verdadeira. Contudo, SST-1 sofre com a escolha de *pooling* médio e a hipótese nula não é verdadeira. Isto provavelmente ocorre pelo fato de as fronteiras entre as classes neste conjunto de dados não estarem bem definidas. Quando a média é usada para fazer o *Pooling*, classes como "positivo"e "muito positivo"podem ser confundidas pela rede pois a média re-

duz a diferença que há entre estas classes. A Figura 30 mostra a diferença de desempenho ao se usar o *pooling* médio no conjunto SST-1. Há uma perda de 30% em relação à referência. A Figura 30 mostra as matrizes de confusão para os 2 tipos de *pooling* e exibe o seu efeito no processo de classificação. É possível ver que o classificador falha principalmente na identificação sentenças de classe Mpo.

Tabela 8: Resultados do Teste-T (T= 2,262 e α = 95%) considerando o *Pooling*

Tipo de Pooling	MR	SST-1	SST-2
Pooling Máximo - Pooling médio	0,430	21,941	0,670

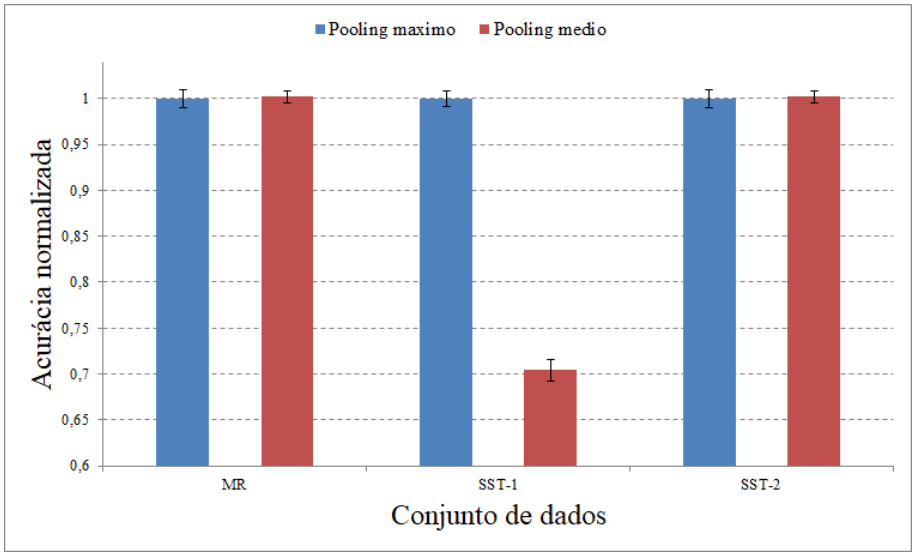
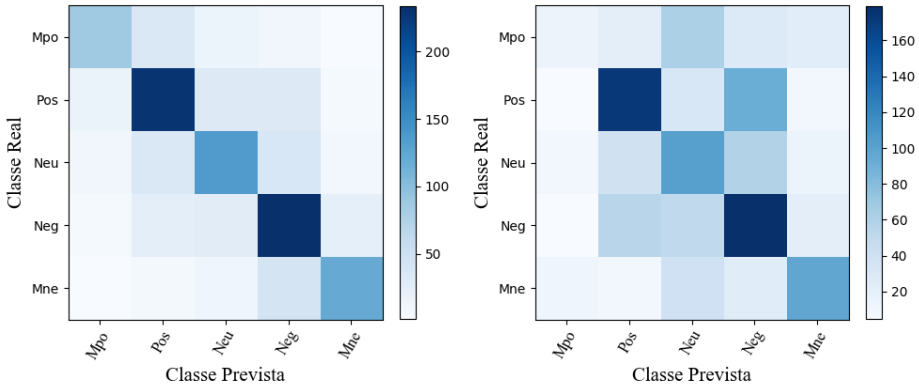


Figura 29: Impacto do tipo de *Pooling* na acurácia



(a) Pooling-máximo (b) Pooling-médio

Figura 30: Matrizes de Confusão para SST-1

6.1.6 Constante de Regularização

Esta seção apresenta os resultados obtidos quando a constante de regularização varia e os outros parâmetros são mantidos fixos. Nesta configuração, a rede neural de convolução se comporta de 2 maneiras diferente dependendo do conjunto de dados. Para SST-1, a Figura 31 e a Tabela 9 indicam que a rede tem melhor desempenho quando a constante de regularização é igual a 0 e que quão maior o valor de CR, pior a acurácia de CNN. A rede ganha cerca de 4% em acurácia para CR igual a 0, mas pode perder mais de 15% com CR igual a 12. Já para MR e SST-2, a rede mostra ser indiferente às trocas de constante de regularização já que a hipótese nula é verdadeira para todos os casos.

Tabela 9: Resultados do Teste-T ($T = 2,262$ e $\alpha = 95\%$) considerando a constante de regularização

Constante de Regularização	MR	SST-1	SST-2
CR 3 - CR 0	1,753	8,926	2,232
CR 3 - CR 6	0,507	5,412	0,507
CR 3 - CR 9	0,239	7,371	0,146
CR 3 - CR 12	1,069	4,508	1,472
CR 0 - CR 6	2,063	11,417	2,063
CR 0 - CR 9	2,101	10,437	1,901
CR 0 - CR 12	1,036	5,712	1,036
CR 6 - CR 9	0,262	2,876	0,262
CR 6 - CR 12	0,871	3,221	0,0871
CR 9 - CR 12	1,050	2,645	1,455

6.1.7 Taxa de *Dropout*

Quando se varia a taxa de *Dropout*, a rede se comporta de maneira extremamente semelhante para os 3 conjuntos de dados. Taxas de *Dropout* altas não possuem diferença estatística entre si de acordo com a Tabela 10. Contudo, quando taxas menores são testadas, há uma queda de performance. A mais significativa ocorre quando a taxa é igual a 0. Com uma taxa tão baixa, a rede começa a sofrer sobreajuste e tem seu treinamento interrompido. Apesar da taxa de regularização estar presente, esta não parece ser o suficiente para evitar a ocorrência deste fenômeno.

Quando DO igual a 0 é usado, a rede perde cerca de 40% de acurácia em MR e SST-2 e 60% em SST-1. A Figura 33 mostra, através das matrizes de confusão, como a rede é afetada pela ausência de *dropout* nos 3 conjuntos de dados. O classificador funciona bem para taxas

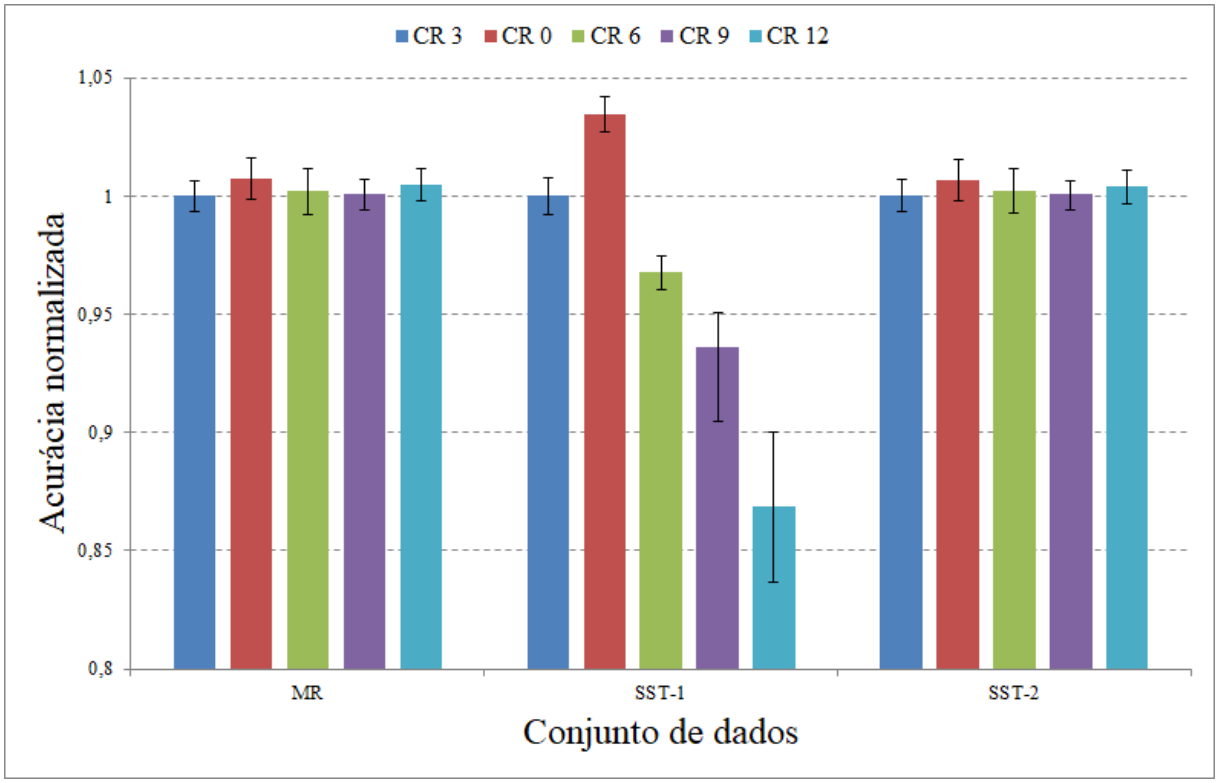


Figura 31: Impacto da variação da Constante de Regularização na acurácia

Tabela 10: Resultados do Teste-T ($T = 2,262$ e $\alpha = 95\%$) considerando a taxa de*Dropout*

Taxa de Dropout	MR	SST-1	SST-2
DO 0,5 - DO 0	64,509	94,938	68,459
DO 0,5 - DO 0,2	5,947	5,947	1,241
DO 0,5 - DO 0,7	1,185	1,318	1,185
DO 0,5 - DO 0,9	1,743	2,249	1,325
DO 0 - DO 0,2	62,213	68,521	1,684
DO 0 - DO 0,7	59,753	82,635	51,753
DO 0 - DO 0,9	61,698	115,188	58,453
DO 0,2 - DO 0,7	0,298	10,865	1,298
DO 0,2 - DO 0,9	0,535	6,687	0,685
DO 0,7 - DO 0,9	0,222	0,468	0,245

maiores que 0. Contudo, falha completamente para este valor. Quando DO é igual a 0, a rede consegue identificar apenas a classe positiva. Este resultado indica a importância do *dropout* no trabalho regularizar uma CNN.

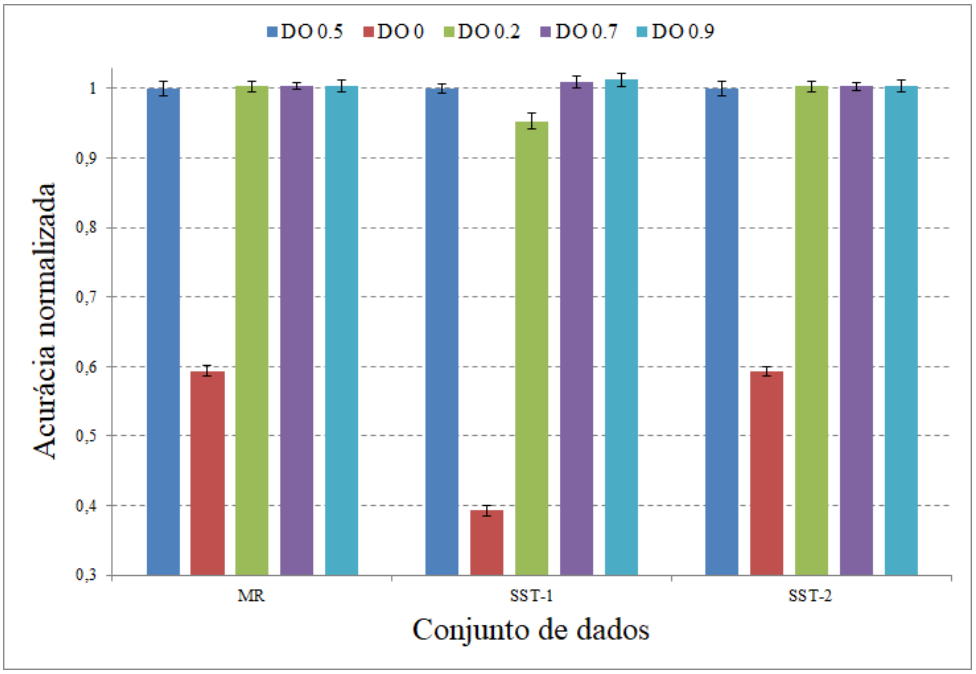
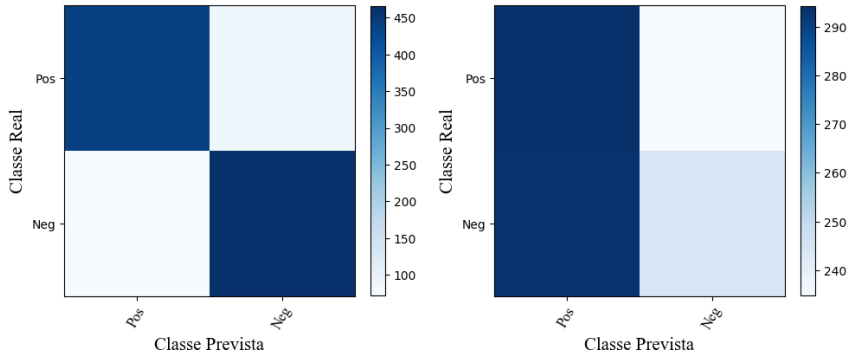
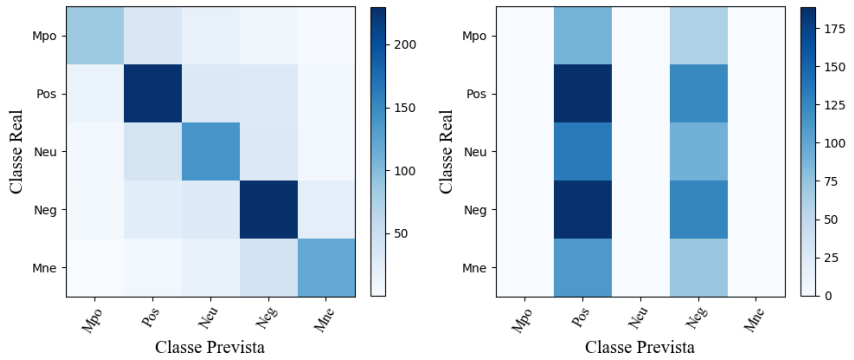


Figura 32: Impacto da taxa de *Dropout* na acurácia



(a) *Dropout* diferente de 0 em MR e SST-2

(b) *Dropout* diferente de 0 em MR e SST-2



(c) *Dropout* diferente de 0 em SST-1

(d) *Dropout* igual 0 em SST-1

Figura 33: Matrizes de Confusão considerando diferentes valores de *Dropout*

6.2 Comparação entre Modelos

Nesta seção, as configurações de CNN com melhores desempenhos são comparadas aos modelos que são frequentemente usados na indústria e aos modelos no estado da arte. Os resultados das simulações realizadas indicam que a CNN com melhor desempenho em MR e SST-2 é a CNN que faz uso da configuração de referência. Para estes conjuntos de dados, a hipótese nula não é verdadeira apenas em casos nos quais a acurácia da rede diminui. A CNN com melhor resultado em SST-1 é a rede que tem CR igual a 0. Estas redes são identificadas nesta seção por CNN-A e CNN-B e suas configurações estão nas Tabelas 11 e 12 respectivamente.

Tabela 11: Configuração da melhor rede para MR e SST-2

Hiperparâmetro	Valor
Embedding	W2VE
Função de Ativação	ReLU
Tamanho do Filtro	F3
Mapas de Características	100
Pooling	Pooling máximo
CR	3
DO	0.5

Tabela 12: Configuração da melhor rede para SST-1

Hiperparâmetro	Valor
Embedding	W2VE
Função de Ativação	ReLU
Tamanho do Filtro	F3
Mapas de Características	100
Pooling	Pooling máximo
CR	0
DO	0.5

Para se ter um referencial para os resultados da CNN, é preciso avaliar o desempenho alcançado usando modelos mais simples de aprendizado de máquinas para classificação de sentenças. Como referência, são usados os seguintes algoritmos: *Naive Bayes* (FRIEDMAN; GEIGER; GOLDSZMIDT, 1997) que é um algoritmo de aprendizado de máquinas baseado na aplicação do teorema de Bayes (RUDNER; LIANG, 2002) com a suposição "ingênua"de independência entre os dados; *Support Vector Machine* (STEINWART; CHRISTMANN, 2008) que é um

modelo não-probabilístico que busca encontrar o hiperplano de máxima distância entre os dados; Regressão Logística (COLLINS; SCHAPIRE; SINGER, 2002a) que é um modelo de regressão não-linear; e Floresta Randômica (COLLINS; SCHAPIRE; SINGER, 2002b) que é um classificador composto por várias árvores de decisão (QUINLAN, 1986). Estes modelos são implementados com o auxílio do *Scikit-Learn* e seus hiperparâmetros foram mantidos com os valores padrão (PEDREGOSA et al., 2011) encontrados na documentação da biblioteca. Apenas para o SVM é feita a mudança de kernel RBF (*Radial Basis Function*) para Linear. Os dados são transformados em entradas para os algoritmos usando o LSA conforme explicado na Seção 4.2.1 do Capítulo 4 considerando apenas uni-gramas e bi-gramas. Outra forma de representar as sentenças foi por meio de vetores médios. Estes foram obtidos através da média aritmética dos vetores das palavras presentes nas sentenças. O conjunto de vetores escolhidos foi *Word2Vec*. Sendo assim, uma sentença pode ser representada por um vetor de $300x1$.

A Figura 19 do Capítulo 4 permite observar que *word embeddings* criam clusters de palavras semanticamente relacionadas, então outra possível abordagem para representar os dados é explorar a semelhança de palavras dentro de um cluster. O agrupamento de vetores desta forma é chamado de quantização vetorial (ZHARMAGAMBETOV; PAK, 2015). Para aplicá-la, é preciso primeiro encontrar os centros dos clusters das palavras do vocabulário formado pelos dados, o que pode ser feito usando um algoritmo de clusterização como *K-Means* (KANUNGO et al., 2002). Tendo encontrado os clusters para cada conjunto de palavras, as sentenças podem ser representadas como um vetor de ocorrências de clusters de modo semelhante ao BOW. A sentença então fica representada por um vetor de dimensões $Nx1$, onde N é o número de clusters encontrados. As simulações são feitas usando o mesmo tipo de pré-processamento e com 10 pastas para validação cruzada. Os resultados das simulações são apresentados na Tabela 13. Estes indicam que esses modelos têm desempenho significativamente pior que as melhores CNN nos 3 conjuntos de dados. Todavia, uma vantagem destes algoritmos é o tempo de treinamento que não ultrapassa poucos segundos.

Além de modelos simples de aprendizado de máquinas, a Tabela 13 contém os resultados de alguns modelos no estado da arte em NLP. RAE são *autoencoders* recursivos que usam vetores pré-treinados do *corpus* Wikipedia . RNTN são redes neurais tensoriais recursivas que fazem uso de dados pré-processados em árvores de análise sintática. MV-RNN são redes neurais recursivas que também usam dados pré-processados em árvores . CCAE são *autoenconders* com operadores gramaticais de categoria . *Sent-Parser* é classificador for-

Tabela 13: Acurácia (%) das CNNs e dos modelos de referência.

Modelo	MR	SST-1	SST-2
CNN-A	80,8	46,8	86,6
CNN-B	80,7	47,9	86,4
LSA-Regressão Logística	75,9	40,6	78,9
LSA-Support Vector Machine	75,9	39,1	84,9
LSA-Floresta Randômica	68,1	33,6	78,1
Vetor-Médio-Regressão Logística	65,1	34,8	78,0
Vetor-Médio-Support Vector Machine	64,3	35,1	77,5
Vetor-Médio-Floresta Randômica	58,3	25,8	70,6
Cluster-Naive Bayes	70,8	35,0	80,4
Cluster-Regressão Logística	70,0	34,3	79,5
Cluster-Support Vector Machine	68,2	32,4	78,4
Cluster-Floresta Randômica	65,3	31,0	74,7
RAE (SOCHER et al., 2011)	77,7	43,2	82,4
RNTN (SOCHER et al., 2013)	-	45,7	85,4
MV-RNN (SOCHER et al., 2012)	79,0	44,4	82,9
CCAE (HERMANN; BLUNSOM, 2013)	77,8	-	-
Sent-Parser (DONG et al., 2014)	79,5	-	-
G-Dropout (WANG; MANNING, 2013)	79,0	-	-
F-Dropout (WANG; MANNING, 2013)	79,1	-	-

mado por analisadores de sentimentos específicos. *G-Dropout* e *F-Dropout* são classificadores que usam *dropout* Gaussiano e *dropout* rápido. A Figura 34 exibe a acurácia normalizada das CNN-A e CNN-B em relação ao modelo MV-RNN.

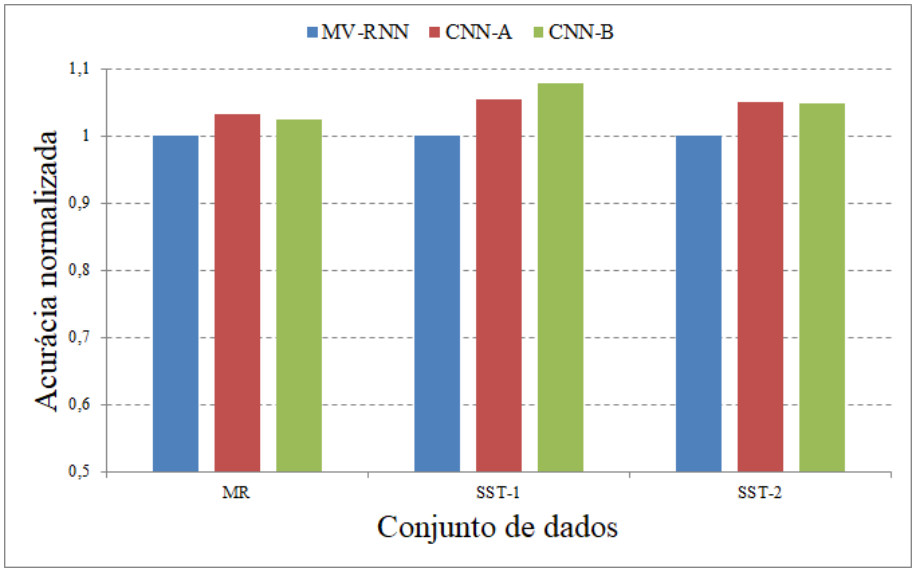


Figura 34: Ganho de acurácia para as configurações propostas em relação a MV-RNN

Nos 3 conjuntos de dados, CNN-A e CNN-B superam o desempenho de MV-RNN, porém não possível afirmar se a hipótese nula é verdadeira por causa da ausência de dados relativos a MV-RNN. Apesar disso, é possível afirmar que a CNN tem vantagem sobre MV-RNN por conta do tipo de dados que exigem. Enquanto a CNN demanda um conjunto de dados formado apenas por sentenças e classes, MV-RNN exige que o conjunto de dados seja formado por sentenças processadas em árvores de análise sintática. Este tipo de dado é geralmente mais caro e mais difícil de ser encontrado.

6.3 Considerações Finais do Capítulo

Neste capítulo, foram apresentados os resultados obtidos na avaliação de diferentes configurações topológicas, baseadas nos diferentes hiperparâmetros de uma rede de convolução. Os resultados indicam que pode haver grande perda de desempenho em alguns casos dependendo do hiperparâmetro e do conjunto de dados avaliado. Na maioria dos caso em que a hipótese nula não se confirmou, ocorria perda de desempenho em relação à configuração de referência. Contudo, houve casos em que ocorreu ganho de desempenho. Isto aconteceu apenas no conjunto SST-1 que possui 5 classes. Os resultados da Seção 6.2 mostraram que uma CNN usada para análise de sentimentos pode superar de maneira significativa os modelos mais comumente usados e que também pode ser competitiva e eventualmente superar o desempenho de modelos no estado da arte.

É importante ressaltar que nos Apêndices estão ilustrados todos os dados correspondentes às simulações citadas. No capítulo seguinte serão apresentadas as conclusões relevantes ao desenvolvimento e resultados desta dissertação, assim como os trabalhos futuros relacionados à proposta de análise de sentimentos por redes de convolução.

Capítulo 7

CONCLUSÕES E TRABALHOS FUTUROS

ESTE capítulo completa esta dissertação, apresentando as principais conclusões obtidas do desenvolvimento do presente trabalho. São apresentadas também as possíveis direções para futuros trabalhos, envolvendo o uso de redes de convolução e análise de sentimentos.

7.1 Conclusões

Esta dissertação abordou o uso de redes neurais de convolução na classificação de sentimentos em sentenças. A arquitetura de rede utilizada foi a mesma proposta em (KIM, 2014). Nesta dissertação, investigou-se a influência de 7 diferentes tipos de hiperparâmetro sobre o comportamento da rede de convolução. O primeiro hiperparâmetro testado é o tipo de *word embedding* utilizado como entrada da rede. O segundo hiperparâmetro é o tipo de função de ativação usada sobre os dados que fluem pela rede. O terceiro é o tamanho dos filtros de convolução que são usados na rede. O quarto é o número de mapas de características que são gerados pelos filtros de convolução. O quinto hiperparâmetro é o tipo de *pooling* usado para condensar a informação que sai dos mapas de características. O sexto hiperparâmetro é a constante de regularização que modifica a função de custo e tenta evitar o sobreajuste da rede. O último hiperparâmetro é a taxa de *dropout* que também é uma forma de regularização.

Este estudo foi motivado pela crescente demanda por sistemas automatizados capazes de analisar com precisão o conteúdo de documentos publicados *online*. Seu objetivo foi investigar as possibilidades de ajustes de configurações de uma rede neural de convolução quando usadas em dados de análise de sentimentos.

Inicialmente, criou-se um ambiente virtual usando Python que é a linguagem principal para análise de dados e aprendizado de máquinas atualmente. O ambiente foi montado em um computador com sistema operacional *Linux*, distribuição *Ubuntu 16.04*, processador *Intel Core i7* com 2,6 *GHz*, *16GB* de memória RAM e placa de vídeo *NVIDIA GeForce GTX 1060 GPU* com 6*GB* de memória dedicada. Os principais módulos Python utilizados foram o *Tensorflow*, *NumPy* e *Scikit-Learn*.

Em seguida, os dados foram coletados e pré-processados seguindo os passos descritos na Seção 5.2 do Capítulo 5. A partir de então, adotou-se a configuração da Tabela 2 do Capítulo 5 como sendo o modelo de referência. A metodologia usada procurou explorar as possibilidades de hiperparâmetros variando-se um tipo de hiperparâmetro por vez em um universo discreto de possibilidades. As simulações foram feitas usando um sistema de validação cruzada com 10 pastas.

Variando-se o tipo de *word embedding* verificou-se que a CNN tem o desempenho prejudicado ao mudar de W2VE para fTD ou GVD nos conjuntos MR e SST-2. Em SST-1, ao contrário dos outros dados, existe melhora de desempenho ao se usar GVE, fTD, W2VE. Os ganhos de acurácia podem chegar a 1% em relação à referência.

A análise sobre a função de ativação não produziu configurações capazes de superar o desempenho da configuração de referência. Os resultados mostraram que as funções de Tanh e Sigmoid não funcionam bem em todos os conjuntos de dados. A queda de desempenho é mais acentuada em SST-1. A rede chega a perder 45% em relação à referência quando se usa a função Sigmoid.

A análise a respeito do tamanho dos filtros de convolução mostrou que há perdas de desempenho quando filtros menores são usados. O filtro F1 tem uma perda de 3% em relação à referência. Os resultados também mostraram que não há ganho significativo ao se aumentar o excessivamente o tamanho dos filtros.

O número de mapas de características mostrou ser relevante apenas no conjunto SST-1. Neste, os resultados indicaram que há ganhos desempenho ao se usar um maior número de mapas. O aumento de acurácia chega a 2% em relação à referência. O maior número de mapas tem grande impacto sobre o desempenho da rede pois é este o responsável por identificar as características subjacentes aos dados. Em um conjunto de múltiplas classes como SST-1, isso é fundamental pois ajuda a definir melhor as fronteiras entre as classes.

A variação de *Pooling* apenas é relevante para SST-1. Contudo, não há ganho de desempenho. A rede sofre uma queda de 30% em relação à referência ao se usar o *Pooling* médio. Usando este tipo de *pooling* a rede fica incapaz de identificar sentenças da classe Mpo.

A constante de regularização mostrou não ter efeito relevante nos conjuntos MR e SST-2. Todavia, em SST-1, os resultados indicam que o aumento deste hiperparâmetro faz com que a rede sofra quedas significativas de desempenho. Para este conjunto, a melhor rede é a que usa CR igual 0. Neste caso, há um aumento de cerca de 4% de acurácia em relação à referência.

O *dropout* mostrou ter grande importância para todos os conjuntos de dados testados. Os resultados mostraram que o valor de taxa de *dropout* não importa desde que esta não seja 0. Em MR e SST-2, a rede perde 40% de acurácia em relação à referência. Em SST-1, essa queda pode chegar a 60% na ausência de *dropout*.

As duas CNNs que obtiveram melhores resultados foram comparadas aos modelos geralmente utilizados para análise de sentimentos e alguns modelos no estado da arte. As CNNs mostraram ser significativamente melhores que algoritmos como *Naive Bayes* e *Support Vector Machine*. Além disso, as duas também foram competitivas com os modelos no estado da arte.

7.2 Trabalhos Futuros

Nesta seção, são citadas algumas possíveis modificações à metodologia proposta, com o intuito de melhorar o desempenho da CNN. Também são levantadas propostas para futuros trabalhos na área de análise de sentimentos.

Um primeiro trabalho a ser feito é implementar os modelos no estado da arte apresentados nesta dissertação a fim de se obter dados suficientes para avaliar por meio de teste estatístico se a hipótese nula é verdadeira ou não quando estes modelos são comparados com as CNNs de melhores desempenho.

Uma outra análise que pode ser feita é usar outros tipos de *word embeddings*. No capítulo 4, há alguns deles listados. Contudo, existem vários outros e a cada dia que passa novos *embeddings* são criados. Também é possível avaliar qual seria o desempenho da CNN obtida com a combinação de *embeddings*, seja por concatenação, soma, ou outra operação qualquer.

Além de aumentar a variedade de *word embeddings*, é possível estudar a influência de outros hiperparâmetros em intervalos ainda maiores que os analisados nesta dissertação e com mais dados e em outros idiomas. Também é possível avaliar outros hiperparâmetros como taxa de aprendizagem, número de épocas e quantidade de filtros que não foram abordados neste trabalho. Obviamente esta é uma busca subótima pois analisa apenas pequenas variações da configuração de referência. Todavia, por conta do tempo de treinamento deste tipo de rede ainda ser razoavelmente alto, a metodologia adotada pode ser considerada como sendo uma exploração inicial do espaço de hiperparâmetros. Em breve, uma nova geração de TPUs (*Tensor Processing Unit*) será lançada pelo Google™ como serviço na nuvem. A empresa afirma que o tempo de treinamento de redes neurais vai cair consideravelmente permitindo análise de um universo muito maior de combinações de hiperparâmetros. Desse modo, os melhores resultados obtidos neste trabalho podem servir como valores de inicialização para buscas mais eficientes como PSO ou Algoritmo Genético .

REFERÊNCIAS

ABADI, M. et al. *TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems*. 2015. Software available from tensorflow.org.

ARCHAK, N.; GHOSE, A.; IPEIROTIS, P. G. Show me the money!: Deriving the pricing power of product features by mining consumer reviews. In: *Proceedings of the 13th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2007. (KDD '07), p. 56–65.

ASUR, S.; HUBERMAN, B. A. Predicting the future with social media. In: *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*. Washington, DC, USA: IEEE Computer Society, 2010. (WI-IAT '10), p. 492–499.

BALBAERT, I. *Getting Started with Julia Programming Language*. USA: Packt Publishing, 2015.

BAR-HAIM, R. et al. Identifying and following expert investors in stock microblogs. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (EMNLP '11), p. 1310–1319.

BARBOSA, L.; FENG, J. Robust sentiment detection on twitter from biased and noisy data. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (COLING '10), p. 36–44.

BHATIA, P.; GUTHRIE, R.; EISENSTEIN, J. Morphological priors for probabilistic neural word embeddings. 2015.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *J. Mach. Learn. Res.*, JMLR.org, v. 3, p. 993–1022, mar. 2003.

BOJANOWSKI, P. et al. Enriching word vectors with subword information. *arXiv preprint arXiv:1607.04606*, 2016.

BOLLEN, J.; MAO, H.; ZENG, X. Twitter mood predicts the stock market. *CoRR*, abs/1010.3003, 2010.

CAMP, M. van de; BOSCH, A. van den. The socialist network. *Decision Support Systems*, v. 53, n. 4, p. 761 – 769, 2012. ISSN 0167-9236. 1) Computational Approaches to Subjectivity and Sentiment Analysis 2) Service Science in Information Systems Research : Special Issue on {PACIS} 2010.

CASTELLANOS, M. et al. Lci: A social channel analysis platform for live customer intelligence. In: *Proceedings of the 2011 ACM SIGMOD International Conference on Management of Data*. New York, NY, USA: ACM, 2011. (SIGMOD '11), p. 1049–1058.

CHEN, C. C.; TSENG, Y.-D. Quality evaluation of product reviews using an information quality framework. *Decision Support Systems*, v. 50, n. 4, p. 755 – 768, 2011. ISSN 0167-9236. Enterprise Risk and Security Management: Data, Text and Web Mining.

CHEN, Y.; XIE, J. Online consumer review: Word-of-mouth as a new element of marketing communication mix. *Management Science*, v. 54, n. 3, p. 477–491, 2008.

COLLINS, M.; SCHAPIRE, R. E.; SINGER, Y. Logistic regression, adaboost and bregman distances. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 48, n. 1-3, p. 253–285, set. 2002.

COLLINS, M.; SCHAPIRE, R. E.; SINGER, Y. Logistic regression, adaboost and bregman distances. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 48, n. 1-3, p. 253–285, set. 2002.

CORTES, C.; VAPNIK, V. Support-vector networks. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 20, n. 3, p. 273–297, set. 1995.

DAS, S. R.; CHEN, M. Y. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Manage. Sci.*, INFORMS, Institute for Operations Research and the Management Sciences (INFORMS), Linthicum, Maryland, USA, v. 53, n. 9, p. 1375–1388, set. 2007.

DATA. 06 2002. http://www.cs.cornell.edu/people/pabo/movie-review-data/.

DAV. *word2vec*. 10 2016. https://github.com/dav/word2vec.

DAVE, K.; LAWRENCE, S.; PENNOCK, D. M. Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. In: *Proceedings of the 12th International Conference on World Wide Web*. New York, NY, USA: ACM, 2003. (WWW '03), p. 519–528.

DELLAROCAS, C.; ZHANG, X. M.; AWAD, N. F. Exploring the value of online product reviews in forecasting sales: The case of motion pictures. *Journal of Interactive Marketing*, Elsevier BV, v. 21, n. 4, p. 23 – 45, 2007.

DHILLON, P. S.; FOSTER, D. P.; UNGAR, L. H. Eigenwords: Spectral word embeddings. *J. Mach. Learn. Res.*, JMLR.org, v. 16, n. 1, p. 3035–3078, jan. 2015.

DONG, L. et al. A statistical parsing framework for sentiment classification. *CoRR*, abs/1401.6330, 2014.

DUMAIS, S. et al. Inductive learning algorithms and representations for text categorization. In: *Proceedings of the Seventh International Conference on Information and Knowledge Management*. New York, NY, USA: ACM, 1998. (CIKM '98), p. 148–155.

ELMAN, J. L. Distributed representations, simple recurrent networks, and grammatical structure. *Machine Learning*, v. 7, p. 195–225, 1991.

ESULI, A.; SEBASTIANI, F. *SENTIWORDNET: A Publicly Available Lexical Resource for Opinion Mining*. 2006.

FELDMAN, R. et al. The stock sonar - sentiment analysis of stocks based on a hybrid approach. In: SHAPIRO, D. G.; FROMHERZ, M. P. J. (Ed.). *IAAI*. San Francisco, USA: AAAI, 2011.

FRIEDMAN, N.; GEIGER, D.; GOLDSZMIDT, M. Bayesian network classifiers. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 29, n. 2-3, p. 131–163, nov. 1997.

GLOVE Global Vectors for Word Representation. 08 2014. https://nlp.stanford.edu/projects/glove/.

GOSSET, W. S. The probable error of a mean. *Biometrika*, n. 1, p. 1–25, March 1908. Originally published under the pseudonym “Student”.

GROH, G.; HAUFFA, J. Characterizing social relations via nlp-based sentiment analysis. In: ADAMIC, L. A.; BAEZA-YATES, R. A.; COUNTS, S. (Ed.). *ICWSM*. The AAAI Press, 2011. Disponível em: <http://dblp.uni-trier.de/db/conf/icwsm/icwsm2011.htmlGrohH11>.

HATZIVASSILOGLOU, V.; MCKEOWN, K. R. Predicting the semantic orientation of adjectives. In: *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1997. (ACL ’98), p. 174–181.

HAYKIN, S. *Redes Neurais - 2ed.* USA: BOOKMAN COMPANHIA ED, 2001.

HE, Y.; ZHOU, D. Self-training from labeled features for sentiment analysis. *Information Processing*
Management, v. 47, n. 4, p. 606 – 616, 2011.

HERMANN, K. M.; BLUNSOM, P. The role of syntax in vector space models of compositional semantics. In: *In Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*. Sofia, Bulgaria: Association for Computational Linguistics, 2013.

HOFMANN, T. Probabilistic latent semantic indexing. In: *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*. New York, NY, USA: ACM, 1999. (SIGIR ’99), p. 50–57.

HONG, Y.; SKIENA, S. The wisdom of bookies – sentiment analysis versus. the nfl point spread. In: COHEN, W. W.; GOSLING, S. (Ed.). *ICWSM*. Atlanta, USA: The AAAI Press, 2010.

HU, M.; LIU, B. Mining and summarizing customer reviews. In: *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: ACM, 2004. (KDD ’04), p. 168–177.

HU, N.; PAVLOU, P. A.; ZHANG, J. Can online reviews reveal a product’s true quality?: Empirical findings and analytical modeling of online word-of-mouth communication. In: *Proceedings of the 7th ACM Conference on Electronic Commerce*. New York, NY, USA: ACM, 2006. (EC ’06), p. 324–330.

HU, Z. et al. Review sentiment analysis based on deep learning. In: *e-Business Engineering (ICEBE), 2015 IEEE 12th International Conference on*. Taipei, Taiwan: IEEE, 2015. p. 87–94.

JI, S. et al. Wordrank: Learning word embeddings via robust ranking. *CoRR*, abs/1506.02761, 2015.

JOLLIFFE, I. *Principal Component Analysis*. USA: Springer Verlag, 1986.

JONES, E. et al. *SciPy: Open source scientific tools for Python*. 2001–. [Online; accessed 24/06/2017].

JOSHI, M. et al. Movie reviews and revenues: An experiment in text regression. In: *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (HLT '10), p. 293–296.

KALCHBRENNER, N.; GREFENSTETTE, E.; BLUNSOM, P. A convolutional neural network for modelling sentences. *CoRR*, abs/1404.2188, 2014.

KANG, H.; YOO, S. J.; HAN, D. Senti-lexicon and improved naïve bayes algorithms for sentiment analysis of restaurant reviews. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 39, n. 5, p. 6000–6010, abr. 2012.

KANUNGO, T. et al. An efficient k-means clustering algorithm: Analysis and implementation. *IEEE Trans. Pattern Anal. Mach. Intell.*, IEEE Computer Society, Washington, DC, USA, v. 24, n. 7, p. 881–892, jul. 2002.

KERNIGHAN, B. W. *The C Programming Language*. 2nd. ed. USA: Prentice Hall Professional Technical Reference, 1988.

KIM, Y. Convolutional neural networks for sentence classification. *CoRR*, abs/1408.5882, 2014.

KO, Y.; SEO, J. Automatic text categorization by unsupervised learning. In: *Proceedings of the 18th Conference on Computational Linguistics - Volume 1*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2000. (COLING '00), p. 453–459.

KOHONEN, T.; SCHROEDER, M. R.; HUANG, T. S. *Self-Organizing Maps*. 3rd. ed. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2001.

LANE, P. C.; CLARKE, D.; HENDER, P. On developing robust models for favourability analysis: Model choice, feature sets and imbalanced data. *Decision Support Systems*, v. 53, n. 4, p. 712 – 718, 2012. 1) Computational Approaches to Subjectivity and Sentiment Analysis 2) Service Science in Information Systems Research : Special Issue on {PACIS} 2010.

LECUN, Y.; BENGIO, Y. The handbook of brain theory and neural networks. In: ARBIB, M. A. (Ed.). Cambridge, MA, USA: MIT Press, 1998. cap. Convolutional Networks for Images, Speech, and Time Series, p. 255–258.

LEVY, O.; GOLDBERG, Y. Neural word embedding as implicit matrix factorization. In: *Proceedings of the 27th International Conference on Neural Information Processing Systems*. Cambridge, MA, USA: MIT Press, 2014. (NIPS'14), p. 2177–2185.

LI, C. et al. Recursive deep learning for sentiment analysis over social data. In: *Web Intelligence (WI) and Intelligent Agent Technologies (IAT), 2014 IEEE/WIC/ACM International Joint Conferences on*. Warsaw,Poland: IEEE, 2014. v. 2, p. 180–185.

LI, Y.-M.; LI, T.-Y. Deriving market intelligence from microblogs. *Decision Support Systems*, v. 55, n. 1, p. 206 – 217, 2013.

LIU, J. et al. Low-quality product review detection in opinion summarization. In: *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*. Prague, Czech Republic: The Association for Computational Linguistics, 2007. p. 334–342. Poster paper.

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

MARTIN, G. *A Game of Thrones*. USA: Bantam Books, 1997. (A Bantam spectra book: Fantasy).

MARTÍN-VALDIVIA, M.-T. et al. Sentiment polarity detection in spanish reviews combining supervised and unsupervised approaches. *Expert Syst. Appl.*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 40, n. 10, p. 3934–3942, ago. 2013.

MCCULLOCH, W. S.; PITTS, W. Neurocomputing: Foundations of research. In: ANDERSON, J. A.; ROSENFELD, E. (Ed.). Cambridge, MA, USA: MIT Press, 1988. cap. A Logical Calculus of the Ideas Immanent in Nervous Activity, p. 15–27.

MCGLOHON, M.; GLANCE, N.; REITER, Z. Star quality: Aggregating reviews to rank products and merchants. In: *Proceedings of Fourth International Conference on Weblogs and Social Media (ICWSM)*. Washington, USA: Association for the Advancement of Artificial Intelligence, 2010.

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

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *CoRR*, abs/1301.3781, 2013.

MILLER, M. et al. Sentiment flow through hyperlink networks. In: ADAMIC, L. A.; BAEZA-YATES, R. A.; COUNTS, S. (Ed.). *ICWSM*. San Francisco, USA: The AAAI Press, 2011.

MITCHELL, T. M. *Machine Learning*. 1. ed. New York, NY, USA: McGraw-Hill, Inc., 1997.

MOHAMMAD, S. M.; KIRITCHENKO, S.; ZHU, X. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. *CoRR*, abs/1308.6242, 2013.

MOHAMMAD, S. M.; YANG, T. Tracking sentiment in mail: How genders differ on emotional axes. *CoRR*, abs/1309.6347, 2013.

MORAES, R.; VALIATI, J. F.; NETO, W. P. G. Document-level sentiment classification: An empirical comparison between {SVM} and {ANN}. *Expert Systems with Applications*, v. 40, n. 2, p. 621 – 633, 2013. ISSN 0957-4174.

NARAYANAN, V.; ARORA, I.; BHATIA, A. Fast and accurate sentiment classification using an enhanced naive bayes model. *CoRR*, abs/1305.6143, 2013. Disponível em: <http://arxiv.org/abs/1305.6143>.

NASUKAWA, T.; YI, J. Sentiment analysis: Capturing favorability using natural language processing. In: *Proceedings of the 2Nd International Conference on Knowledge Capture*. New York, NY, USA: ACM, 2003. (K-CAP '03), p. 70–77.

NICKOLLS, J. et al. Scalable parallel programming with cuda. *Queue*, ACM, New York, NY, USA, v. 6, n. 2, p. 40–53, mar. 2008.

NIELSEN, M. A. *Neural Networks and Deep Learning*. .: Determination Press, 2015.

O'CONNOR, B. et al. From Tweets to Polls: Linking Text Sentiment to Public Opinion Time Series. In: *Proceedings of the International AAAI Conference on Weblogs and Social Media*. Washington, USA: Association for the Advancement of Artificial Intelligence, 2010.

ODERSKY, M.; SPOON, L.; VENNERS, B. *Programming in Scala: A Comprehensive Step-by-step Guide*. 1st. ed. USA: Artima Incorporation, 2008.

ORTIGOSA-HERNáNDEZ, J. et al. Approaching sentiment analysis by using semi-supervised learning of multi-dimensional classifiers. *Neurocomput.*, Elsevier Science Publishers B. V., Amsterdam, The Netherlands, The Netherlands, v. 92, p. 98–115, set. 2012.

OSGOOD, C. E. The nature and measurement of meaning. *Psychological Bulletin*, v. 49, n. 3, p. 197–237, 1952.

OUYANG, X. et al. Sentiment analysis using convolutional neural network. In: *Computer and Information Technology; Ubiquitous Computing and Communications; Dependable, Autonomic and Secure Computing; Pervasive Intelligence and Computing (CIT/IUCC/DASC/PICOM), 2015 IEEE International Conference on*. Allahabad, India: IEEE, 2015. p. 2359–2364.

PAK, A.; PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining. In: CHAIR), N. C. C. et al. (Ed.). *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*. Valletta, Malta: European Language Resources Association (ELRA), 2010.

PANG, B.; LEE, L.; VAITHYANATHAN, S. Thumbs up: Sentiment classification using machine learning techniques. In: *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (EMNLP '02), p. 79–86.

PAPADIMITRIOU, C. H. et al. Latent semantic indexing: A probabilistic analysis. In: *Proceedings of the Seventeenth ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*. New York, NY, USA: ACM, 1998. (PODS '98), p. 159–168.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011.

PENNINGTON, J.; SOCHER, R.; MANNING, C. D. Glove: Global vectors for word representation. In: *Empirical Methods in Natural Language Processing (EMNLP)*. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1532–1543.

PRESS, W. H. et al. *Numerical Recipes in FORTRAN; The Art of Scientific Computing*. 2nd. ed. New York, NY, USA: Cambridge University Press, 1993.

QIU, G. et al. Dasa: Dissatisfaction-oriented advertising based on sentiment analysis. *Expert Systems with Applications*, v. 37, n. 9, p. 6182 – 6191, 2010. ISSN 0957-4174.

QUINLAN, J. R. Induction of decision trees. *Mach. Learn.*, Kluwer Academic Publishers, Hingham, MA, USA, v. 1, n. 1, p. 81–106, mar. 1986.

RAMOS, J. *Using TF-IDF to Determine Word Relevance in Document Queries*. 1999.

ŘEHŮŘEK, R.; SOJKA, P. Software Framework for Topic Modelling with Large Corpora. In: *Proceedings of the LREC 2010 Workshop on New Challenges for NLP Frameworks*. Valletta, Malta: ELRA, 2010. p. 45–50. `http://is.muni.cz/publication/884893/en`.

ROSSUM, G. *Python Reference Manual*. Amsterdam, The Netherlands, The Netherlands, 1995.

RUDNER, L. M.; LIANG, T. Automated essay scoring using bayes' theorem. eScholarship at Boston College, 2002. Disponível em: <http://escholarship.bc.edu/jtla/vol1/2>.

SADIKOV, E.; PARAMESWARAN, A. G.; VENETIS, P. Blogs as predictors of movie success. In: ADAR, E. et al. (Ed.). *ICWSM*. Arlington, USA: The AAAI Press, 2009.

SAKUNKOO, P.; SAKUNKOO, N. Analysis of social influence in online book reviews. In: ADAR, E. et al. (Ed.). *ICWSM*. Arlington, USA: The AAAI Press, 2009.

SALTON, G.; MCGILL, M. J. *Introduction to modern information retrieval*. New York: McGraw - Hill Book Company, 1983.

SOCHER, R. et al. Semantic Compositionality Through Recursive Matrix-Vector Spaces. In: *Proceedings of the 2012 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Seattle, USA: Association for Computational Linguistics, 2012.

SOCHER, R. et al. Semi-supervised recursive autoencoders for predicting sentiment distributions. In: *Proceedings of the Conference on Empirical Methods in Natural Language Processing*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2011. (EMNLP '11).

SOCHER, R. et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: *EMNLP*. Seattle, USA: Association for Computational Linguistics, 2013.

STEINWART, I.; CHRISTMANN, A. *Support Vector Machines*. 1st. ed. USA: Springer Publishing Company, Incorporated, 2008.

STEWART, G. W. On the early history of the singular value decomposition. *SIAM Rev.*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 35, n. 4, p. 551–566, dez. 1993.

STONE, P. J.; HUNT, E. B. A computer approach to content analysis: Studies using the general inquirer system. In: *Proceedings of the May 21-23, 1963, Spring Joint Computer Conference*. New York, NY, USA: ACM, 1963. (AFIPS '63 (Spring)), p. 241–256.

TUMASJAN, A. et al. Predicting elections with twitter: What 140 characters reveal about political sentiment. In: *Proceedings of the Fourth International AAAI Conference on Weblogs and Social Media*. Washington, USA: Association for the Advancement of Artificial Intelligence, 2010. p. 178–185.

TURNEY, P. D. Thumbs up or thumbs down: Semantic orientation applied to unsupervised classification of reviews. In: *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2002. (ACL '02), p. 417–424.

UIJLINGS, J.; SMEULDERS, A. Visualising bag-of-words. In: *demo at ICCV*. Trento, Italy: ., 2011.

VAPNIK, V. N. *The Nature of Statistical Learning Theory*. New York, NY, USA: Springer-Verlag New York, Inc., 1995.

VINCENT, P. et al. Stacked denoising autoencoders: Learning useful representations in a deep network with a local denoising criterion. *J. Mach. Learn. Res.*, JMLR.org, v. 11, p. 3371–3408, dez. 2010. ISSN 1532-4435.

WALKER, M. A. et al. That is your evidence: Classifying stance in online political debate. *Decision Support Systems*, v. 53, n. 4, p. 719 – 729, 2012. 1) Computational Approaches to Subjectivity and Sentiment Analysis 2) Service Science in Information Systems Research : Special Issue on {PACIS} 2010.

WALLACH, H. M. Topic modeling: Beyond bag-of-words. In: *Proceedings of the 23rd International Conference on Machine Learning*. New York, NY, USA: ACM, 2006. (ICML '06), p. 977–984.

WALT, S. v. d.; COLBERT, S. C.; VAROQUAUX, G. The numpy array: A structure for efficient numerical computation. *Computing in Science and Engg.*, IEEE Educational Activities Department, Piscataway, NJ, USA, v. 13, n. 2, p. 22–30, mar. 2011.

WANG, S.; MANNING, C. Fast dropout training. In: DASGUPTA, S.; MCALLESTER, D. (Ed.). *Proceedings of the 30th International Conference on Machine Learning (ICML-13)*. Atlanta, USA: JMLR Workshop and Conference Proceedings, 2013. v. 28, n. 2, p. 118–126.

WASSERMAN, P.; SCHWARTZ, T. Neural networks, ii: What are they and why is everybody so interested in them now? *IEEE Intelligent Systems*, IEEE Computer Society, Los Alamitos, CA, USA, v. 3, p. 10–15, 1988.

WIEBE, J. M.; BRUCE, R. F.; O'HARA, T. P. Development and use of a gold-standard data set for subjectivity classifications. In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics on Computational Linguistics*. Stroudsburg, PA, USA: Association for Computational Linguistics, 1999. (ACL '99), p. 246–253.

XIANGHUA, F. et al. Multi-aspect sentiment analysis for chinese online social reviews based on topic modeling and hownet lexicon. *Knowledge-Based Systems*, v. 37, p. 186 – 195, 2013.

YANO, T.; SMITH, N. A. What's worthy of comment? content and comment volume in political blogs. In: COHEN, W. W.; GOSLING, S. (Ed.). *ICWSM*. Washington, USA: The AAAI Press, 2010.

ZHANG, W.; SKIENA, S. *Trading Strategies to Exploit Blog and News Sentiment*. 2010.

ZHANG, Y.; WALLACE, B. C. A sensitivity analysis of (and practitioners' guide to) convolutional neural networks for sentence classification. *CoRR*, abs/1510.03820, 2015. Disponível em: <http://arxiv.org/abs/1510.03820>.

ZHARMAGAMBETOV, A. S.; PAK, A. A. Sentiment analysis of a document using deep learning approach and decision trees. In: *2015 Twelve International Conference on Electronics Computer and Computation (ICECCO)*. Almaty, Kazakhstan: IEEE, 2015. p. 1–4.

ZHOU, S.; CHEN, Q.; WANG, X. Active deep networks for semi-supervised sentiment classification. In: *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010. (COLING '10), p. 1515–1523.

ÖSTERLUND, A.; ÖDLING, D.; SAHLGREN, M. Factorization of latent variables in distributional semantic models. In: MàRQUEZ, L. et al. (Ed.). *EMNLP*. Beijing, China: The Association for Computational Linguistics, 2015. p. 227–231.

APÊNDICE – Matrizes de Confusão

Este apêndice apresenta as matrizes de confusão médias obtidas para cada uma das topologias de rede testadas no Capítulo 6.

Tabela 14: Matriz de confusão MR considerando GloVe dinâmico

444,3	84,8
76,8	461,1

Tabela 15: Matriz de confusão MR considerando GloVe estático

436,9	92,2
73,3	464,6

Tabela 16: Matriz de confusão MR considerando fastText dinâmico

444,6	84,5
78,3	459,6

Tabela 17: Matriz de confusão MR considerando fastText estático

451,2	77,9
80,9	457

Tabela 18: Matriz de confusão MR considerando Word2Vec dinâmico

449,2	79,9
77,8	460,1

Tabela 19: Matriz de confusão MR considerando Word2Vec estático

445,8	83,3
72,6	465,3

Tabela 20: Matriz de confusão SST-1 considerando fastText estático

88,5	31,5	21,3	7,8	1,2
19,5	222,6	39,7	26,6	4,4
9,7	34,2	141,7	33,3	6,8
4,8	24,8	30,1	230,7	22,3
0,7	5,9	17,1	41,2	119,6

Tabela 21: Matriz de confusão SST-1 considerando fastText dinâmico

90,7	30,4	17,6	10,6	1
20,7	225,9	32	30,4	3,8
9,7	36,7	136,4	36,6	6,3
4,3	23,9	28,7	236,7	19,1
1,7	5,9	14,9	45,5	116,5

Tabela 22: Matriz de confusão SST-1 considerando GloVe dinâmico

90,9	33	15	9,1	2,3
19	227,3	30,1	31,2	5,2
10,8	36,3	134,9	35,9	7,8
4,8	26,3	27,7	230,3	23,6
2,1	6,5	14,4	41	120,5

Tabela 23: Matriz de confusão SST-1 considerando GloVe estático

90	32,3	16,9	8,9	2,2
17,8	226,5	35,5	26,5	6,5
8,3	37,1	142,2	31	7,1
4,3	24	33,2	227	24,2
1,6	6,1	15,8	38,6	122,4

Tabela 24: Matriz de confusão SST-1 considerando Word2Vec estático

88,7	29,6	19,9	10,1	2
21,9	218,3	32,3	34	6,3
12	35,1	135,4	35,5	7,7
8,5	19,9	29,1	234	21,2
1,8	4,4	14,2	42,9	121,2

Tabela 25: Matriz de confusão SST-1 considerando Word2Vec dinâmico

88,5	32,9	17,7	8,6	2,6
18,5	227,3	32,7	29,4	4,9
8,7	37,5	139,5	32,7	7,3
4,7	26,6	30,5	229,9	21
2	8,1	14,6	39,3	120,5

Tabela 26: Matriz de confusão SST-2 considerando GloVe dinâmico

444,3	84,8
72,8	465,1

Tabela 27: Matriz de Confusão SST-2 considerando GloVe estático

433,9	95,2
73,3	464,6

Tabela 28: Matriz de confusão SST-2 considerando fastText dinâmico

440,6	88,5
78,3	459,6

Tabela 29: Matriz de confusão SST-2 considerando fastText estático

450,2	78,9
80,9	457

Tabela 30: Matriz de confusão SST-2 considerando Word2Vec dinâmico

449,2	79,9
70,8	467,1

Tabela 31: Matriz de confusão SST-2 considerando Word2Vec estático

443,8	85,3
72,6	465,3

Tabela 32: Matriz de confusão MR considerando Tangente Hiperbólica

442,2	86,9
80,9	457

Tabela 33: Matriz de confusão MR considerando Linear

448,3	80,8
77,9	460

Tabela 34: Matriz de confusão MR considerando ReLU

444,8	84,3
75,5	462,4

Tabela 35: Matriz de confusão MR considerando Sigmoid

428,4	100,7
68,9	469

Tabela 36: Matriz de confusão SST-1 considerando Tangente Hiperbólica

31,3	30,9	71,8	10,7	5,6
7,8	218,1	39,5	41,1	6,3
6,5	39,7	132,9	34,2	12,4
4	31,4	31,8	221,6	23,9
2,7	8,9	16	42,1	114,8

Tabela 37: Matriz de confusão SST-1 considerando Sigmoid

0	19,9	113	17,4	0
0	151,4	62,9	98,5	0
0	34,2	155,4	36,1	0
0	107	67,6	138,1	0
0	18,1	137,8	28,6	0

Tabela 38: Matriz de confusão SST-1 considerando Linear

92,5	31,3	14,4	10	2,1
19,9	224,7	31,3	31,1	5,8
10,5	35,8	137,6	32,9	8,9
4,8	25,3	27,8	230,8	24
1,9	5,9	13,9	40,1	122,7

Tabela 39: Matriz de confusão SST-1 considerando ReLU

87,6	32,5	19,4	8,8	2
17,2	224	39,1	27,6	4,9
8,7	36,2	141,5	31,2	8,1
3,9	24,9	34,9	225,1	23,9
2,1	6,5	15,8	37,6	122,5

Tabela 40: Matriz de confusão SST-2 considerando Tangente Hiperbólica

441,2	87,9
80,9	457

Tabela 41: Matriz de confusão SST-2 considerando Linear

448,3	80,8
72,9	465

Tabela 42: Matriz de confusão SST-2 considerando ReLU

442,8	86,3
75,5	462,4

Tabela 43: Matriz de confusão SST-2 considerando Sigmoid

425,4	103,7
68,9	469

Tabela 44: Matriz de confusão MR considerando filtro (1, 2, 3)

443,5	85,6
75	462,9

Tabela 45: Matriz de confusão MR considerando filtro (2, 3, 4)

442,7	86,4
76,1	461,8

Tabela 46: Matriz de confusão MR considerando filtro (3, 4, 5)

447,3	81,8
73,3	464,6

Tabela 47: Matriz de confusão MR considerando filtro (4, 5, 6)

450,7	78,4
77	460,9

Tabela 48: Matriz de confusão MR considerando filtro (5, 6, 7)

444,9	84,2
71	466,9

Tabela 49: Matriz de confusão SST-1 considerando filtro (1, 2, 3)

84,6	37,3	13,2	10,3	4,9
20,7	219,9	31,4	33,6	7,2
10,6	39,1	129,4	38	8,6
9,7	24,8	25,3	228,6	24,3
5,1	6,7	8,3	45,4	119

Tabela 50: Matriz de confusão SST-1 considerando filtro (2, 3, 4)

86,5	36,3	14,7	8,6	4,2
17,7	230,1	31,2	26,5	7,3
9,4	40,8	132,2	33,9	9,4
5,2	27,5	24	229,7	26,3
2	7,5	10,4	42,6	122

Tabela 51: Matriz de confusão SST-1 considerando filtro (3, 4, 5)

90,6	31,9	16,5	8	3,3
17,5	226,7	33,6	29,9	5,1
8,7	38,1	138,4	33	7,5
4,4	25,1	28,8	231,9	22,5
1,5	6,5	14,8	43,6	118,1

Tabela 52: Matriz de confusão SST-1 considerando filtro (4, 5, 6)

92,6	31,1	17,8	7,6	1,2
19,6	227,5	34,7	26,2	4,8
11,4	36,1	140,5	31,3	6,4
4,4	25,7	36,5	226,1	20
1,3	6,8	17,6	37,8	121

Tabela 53: Matriz de confusão SST-1 considerando filtro (5, 6, 7)

93,4	30,8	16,9	6,9	2,3
18,3	228,3	33,1	27,7	5,4
10,5	35,9	139,1	31,3	8,9
3,2	25,7	31,9	227,6	24,3
1,3	7,4	14,6	37,8	123,4

Tabela 54: Matriz de confusão SST-2 considerando filtro (1, 2, 3)

443,5	85,6
71	466,9

Tabela 55: Matriz de confusão SST-2 considerando filtro (2, 3, 4)

442,7	86,4
74,1	463,8

Tabela 56: Matriz de confusão SST-2 considerando filtro (3, 4, 5)

443,3	85,8
73,3	464,6

Tabela 57: Matriz de confusão SST-2 considerando filtro (4, 5, 6)

450,7	78,4
74	463,9

Tabela 58: Matriz de confusão SST-2 considerando filtro (5, 6, 7)

444,9	84,2
70	467,9

Tabela 59: Matriz de confusão MR considerando 50 mapas de características

446,4	82,7
75,2	462,7

Tabela 60: Matriz de confusão MR considerando 100 mapas de características

450,7	78,4
80,9	457

Tabela 61: Matriz de confusão MR considerando 150 mapas de características

444,3	84,8
73,5	464,4

Tabela 62: Matriz de confusão MR considerando 200 mapas de características

443,1	86
71,6	466,3

Tabela 63: Matriz de confusão MR considerando 250 mapas de características

450	79,1
75,1	462,8

Tabela 64: Matriz de confusão SST-1 considerando 50 mapas de características

78,7	35,6	23,9	10	2,1
16,2	222,4	36,8	33,2	4,2
10,9	37,2	134,6	36,8	6,2
2	23,6	34,6	230,2	22,3
0,6	6,1	17,4	50	110,4

Tabela 65: Matriz de confusão SST-1 considerando 100 mapas de características

87,4	32	21	8,6	1,3
18	221,8	37,6	32,5	2,9
8,9	36,4	137,5	36,7	6,2
3,8	24,1	32,8	236,6	15,4
1,5	5,6	16,8	46,1	114,5

Tabela 66: Matriz de confusão SST-1 considerando 150 mapas de características

88,8	32,1	18,1	8,9	2,4
18,4	222,6	37	29,3	5,5
8	37,7	138,8	33,7	7,5
4,5	24	32,7	230,2	21,3
1,3	7	15,6	39	121,6

Tabela 67: Matriz de confusão SST-1 considerando 200 mapas de características

93,1	32,3	13,1	9,2	2,6
22,4	229,2	28,2	27	6
11,1	36,3	135,9	34	8,4
9,1	24,7	26,2	229,1	23,6
4,3	7,1	11,6	38,3	123,2

Tabela 68: Matriz de confusão SST-1 considerando 250 mapas de características

95,4	29,5	14,8	8,2	2,4
24,2	224,5	30,8	26,1	7,2
11,2	35	140,2	29,1	10,2
7,2	24,5	27,8	227,1	26,1
3,3	7,2	13	34,3	126,7

Tabela 69: Matriz de confusão SST-2 considerando 50 mapas de características

443,4	85,7
75,2	462,7

Tabela 70: Matriz de confusão SST-2 considerando 100 mapas de características

454,7	74,4
80,9	457

Tabela 71: Matriz de confusão SST-2 considerando 150 mapas de características

440,3	88,8
73,5	464,4

Tabela 72: Matriz de confusão SST-2 considerando 200 mapas de características

440,1	89
71,6	466,3

Tabela 73: Matriz de confusão SST-2 considerando 250 mapas de características

450	79,1
73,1	464,8

Tabela 74: Matriz de confusão MR considerando Pooling máximo

446,9	82,2
73,8	464,1

Tabela 75: Matriz de confusão MR considerando Pooling médio

446,9	82,2
73,8	464,1

Tabela 76: Matriz de confusão SST-1 considerando Pooling máximo

87,3	35	16,3	9,6	2,1
16,6	228,6	31,1	31,1	5,4
9,4	35,1	135,8	37,8	7,6
5,3	24	28,1	233,2	22,1
2	6,1	12,6	43,3	120,5

Tabela 77: Matriz de confusão SST-1 considerando Pooling médio

15	21,3	62,1	28,1	23,8
5,6	173,2	32	92,1	9,9
10,3	39,3	102,4	59,7	14
5	55,5	50,8	178,8	22,6
12,2	9,2	40,3	25,1	97,7

Tabela 78: Matriz de confusão SST-2 considerando Pooling máximo

445,9	81,2
73,8	464,1

Tabela 79: Matriz de confusão SST-2 considerando Pooling médio

446,9	82,2
70,8	467,1

Tabela 80: Matriz de confusão MR considerando a constante de regularização 0

448,9	80,2
74,6	463,3

Tabela 81: Matriz de confusão MR considerando a constante de regularização 3

439,6	89,5
72	465,9

Tabela 82: Matriz de confusão MR considerando a constante de regularização 6

445,6	83,5
76,1	461,8

Tabela 83: Matriz de confusão MR considerando a constante de regularização 9

444,4	84,7
75,8	462,1

Tabela 84: Matriz de confusão MR considerando a constante de regularização 12

454	75,1
82,1	455,8

Tabela 85: Matriz de confusão SST-1 considerando a constante de regularização 0

101,2	27,8	11	7,4	2,9
21,6	227,4	30,6	25,2	8
11,7	31,1	143,6	31,1	8,2
6,4	22,2	23,6	234,1	26,4
3,7	6,4	6,9	37,7	129,8

Tabela 86: Matriz de confusão SST-1 considerando a constante de regularização 3

87,9	34,5	18	7,6	2,3
16,7	225,5	38,1	25,8	6,7
8,4	35,1	142,5	31,6	8,1
4,1	25,5	32	228,5	22,6
1,4	6,1	15,6	37,7	123,7

Tabela 87: Matriz de confusão SST-1 considerando a constante de regularização 6

78,5	35,1	24,3	10,3	2,1
15,7	223,7	32,7	34,3	6,4
9,4	39,5	135,7	33	8,1
4,4	29,9	29,8	226,5	22,1
1,3	7,7	17,9	40,1	117,5

Tabela 88: Matriz de confusão SST-1 considerando a constante de regularização 9

75,4	31,8	29,7	9,7	3,7
19,1	217,5	36,3	33,2	6,7
15,1	39,3	128,6	33,4	9,3
5	30,1	33,7	222,5	21,4
3,4	7,1	19,6	41,9	112,5

Tabela 89: Matriz de confusão SST-1 considerando a constante de regularização 12

59,3	32,1	38,3	13,8	6,8
17,5	208,7	22,5	54,6	9,5
18,7	36,6	107,3	44,8	18,3
6,4	39,3	21,8	223,3	21,9
3,2	12,8	25,5	39,8	103,2

Tabela 90: Matriz de confusão SST-2 considerando a constante de regularização 0

442,9	86,2
74,6	463,3

Tabela 91: Matriz de confusão SST-2 considerando a constante de regularização 3

439,6	89,5
71	464,9

Tabela 92: Matriz de confusão SST-2 considerando Regularização 6

440,6	88,5
76,1	461,8

Tabela 93: Matriz de confusão SST-2 considerando a constante de regularização 9

444,4	84,7
71,8	466,1

Tabela 94: Matriz de confusão SST-2 considerando a constante de regularização 12

454	75,1
81,1	456,8

Tabela 95: Matriz de confusão MR considerando taxa de Dropout 0

294,2	234,9
293,4	244,5

Tabela 96: Matriz de confusão MR considerando taxa de Dropout 0,2

447	82,1
74,5	463,4

Tabela 97: Matriz de confusão MR considerando taxa de Dropout 0,5

442,2	86,9
72,4	465,5

Tabela 98: Matriz de confusão MR considerando taxa de Dropout 0,7

447,2	81,9
74,1	463,8

Tabela 99: Matriz de confusão MR considerando Taxa de taxa de Dropout 0,9

453	76,1
79,3	458,6

Tabela 100: Matriz de confusão SST-1 considerando taxa de Dropout 0

0	89,7	0	60,6	0
0	188,6	0	124,2	0
0	134,5	0	91,2	0
0	186,6	0	126,1	0
0	111,3	0	73,2	0

Tabela 101: Matriz de confusão SST-1 considerando taxa de Dropout 0,2

71,3	38	27,7	9	4,3
13,8	226,4	35,1	31,3	6,2
12,6	43,2	127,5	33,2	9,2
4,9	33,9	28,8	221,2	23,9
2,4	8,4	13,5	42,7	117,5

Tabela 102: Matriz de confusão SST-1 considerando taxa de Dropout 0,5

86	35	18,2	9,4	1,7
15,7	227,4	32,6	31,3	5,8
7,7	40,5	139,2	32,3	6
4,6	25,2	31,1	229,3	22,5
0,6	6,5	15,9	42	119,5

Tabela 103: Matriz de confusão SST-1 considerando taxa de Dropout 0,7

92,3	32,4	15	9,2	1,4
21,8	228,1	28,6	28,1	6,2
10,7	37,8	137	33,9	6,3
5,5	27,8	26,7	232,1	20,6
4,4	6,8	12,6	41,2	119,5

Tabela 104: Matriz de confusão SST-1 considerando taxa de Dropout 0,9

91,9	31,5	15,6	9,2	2,1
22,4	223,3	34,6	27,2	5,3
9,9	33,6	142,5	32	7,7
6,4	21,6	31,6	231,9	21,2
2,9	5,5	17,1	37	122

Tabela 105: Matriz de confusão SST-2 considerando taxa de Dropout 0

291,2	237,9
293,4	244,5

Tabela 106: Matriz de confusão SST-2 considerando taxa de Dropout 0,2

447	82,1
73,5	464,4

Tabela 107: Matriz de confusão SST-2 considerando taxa de Dropout 0,5

440,2	88,9
72,4	465,5

Tabela 108: Matriz de confusão SST-2 considerando taxa de Dropout 0,7

447,2	81,9
72,1	465,8

Tabela 109: Matriz de confusão SST-2 considerando Taxa de taxa de Dropout 0,9

450	79,1
79,3	458,6

APÊNDICE B – Médias e Desvios-padrão

Este apêndice apresenta os valores numéricos para média e desvio-padrão da acurácia obtidos nas simulações realizadas no Capítulo 6.

Tabela 110: Média e desvio padrão considerando Word Embeddings em MR

-	**fastText dinâmico**	**GloVe dynamic**	**word2vec dinâmico**	**fastText estático**	**GloVe estático**	**word2vec estático**
Média	0,808547	0,804892	0,807423	0,801172	0,803889	0,806202
Desvio Padrão	0,0116319	0,0114956	0,0123541	0,0111192	0,0104485	0,0107108

Tabela 111: Média e desvio padrão considerando Word Embeddings em SST-1

-	**fastText dinâmico**	**GloVe dinâmico**	**word2vec dinâmico**	**fastText estático**	**GloVe estático**	**word2vec estático**
Média	0,472764	0,471825	0,472342	0,41715	0,471366	0,467513
Desvio Padrão	0,00989639	0,0106283	0,0159805	0,0122271	0,0138878	0,0127104

Tabela 112: Média e desvio padrão considerando Word Embeddings em SST-2

-	**fastText dinâmico**	**GloVe dinâmico**	**word2vec dinâmico**	**fastText estático**	**GloVe estático**	**word2vec estático**
Média	0,852005	0,847208	0,853817	0,844917	0,85148	0,86631
Desvio Padrão	0,0109873	0,012531	0,0103122	0,0118903	0,0108601	0,0118483

Tabela 113: Média e desvio padrão considerando a função de ativação em MR

-	Tanh	ReLU	Sigmoid	Linear
Média	0,801737	0,806234	0,80105	0,805265
Desvio Padrão	0,0116742	0,0114232	0,00833219	0,011676

Tabela 114: Média e desvio padrão considerando função de ativação em SST-1

-	Tanh	ReLU	Sigmoid	Linear
Média	0,405987	0,468126	02375126	0,466535
Desvio Padrão	0,0243267	0,0172759	0,0542224	0,0134435

Tabela 115: Média e desvio padrão considerando a função de ativação em SST-2

-	Tanh	ReLU	Sigmoid	Linear
Média	0,850266	0,866363	0,843284	0,864537
Desvio Padrão	0,0116825	0,0115644	0,0118487	0,00834265

Tabela 116: Média e desvio padrão considerando o tamnho do filtro em MR

-	(1, 2, 3)	(2, 3, 4)	(3, 4, 5)	(4, 5, 6)	(5, 6, 7)
Média	0,795485	0,803704	0,808639	0,814358	0,814545
Desvio Padrão	0,0113877	0,00981386	0,0118062	0,0114262	0,0109521

Tabela 117: Média e desvio padrão considerando o tamanho de filtro em SST-1

-	(1, 2, 3)	(2, 3, 4)	(3, 4, 5)	(4, 5, 6)	(5, 6, 7)
Média	0,458938	0,464958	0,469342	0,471029	0,474486
Desvio Padrão	0,0218127	0,0151219	0,0174642	0,012461	0,0127137

Tabela 118: Média e desvio padrão considerando o tamanho do filtro em SST-2

-	(1, 2, 3)	(2, 3, 4)	(3, 4, 5)	(4, 5, 6)	(5, 6, 7)
Média	0,854518	0,859388	0,865991	0,854864	0,854388
Desvio Padrão	0,012023	0,0118578	0,00947658	0,0115503	0,0108594

Tabela 119: Média e desvio padrão considerando mapas de características em MR

-	50 Mapas	100 Mapas	150 Mapas	200 Mapas	250 Mapas
Média	0,802015	0,805703	0,81064	0,812296	0,812483
Desvio Padrão	0,00885599	0,0114788	0,0109139	0,00947646	0,0108473

Tabela 120: Média e desvio padrão considerando mapas de características em SST-1

-	50 Mapas	100 Mapas	150 Mapas	200 Mapas	250 Mapas
Média	0,454553	0,467681	0,471223	0,47139	0,471256
Desvio Padrão	0,0133905	0,0126913	0,0158058	0,0097695	0,0151698

Tabela 121: Média e desvio padrão considerando mapas de características em SST-2

-	50 Mapas	100 Mapas	150 Mapas	200 Mapas	250 Mapas
Média	0,840276	0,865916	0,861404	0,862902	0,86513
Desvio Padrão	0,0119566	0,00899635	0,0106801	0,00890676	0,0109862

Tabela 122: Média e desvio padrão considerando Pooling em MR

-	Pooling máximo	Pooling médio
Média	0,805796	0,800389
Desvio Padrão	0,00953006	0,0140334

Tabela 123: Média e desvio padrão considerando Pooling em SST-1

-	Pooling máximo	Pooling médio
Média	0,457089	0,278162
Desvio Padrão	0,0170654	0,0238855

Tabela 124: Média e desvio padrão considerando Pooling em SST-2

-	Pooling máximo	Pooling médio
Média	0,865858	0,855458
Desvio Padrão	0,0095184	0,0137773

Tabela 125: Média e desvio padrão considerando taxa de regularização em MR

-	CR 0	CR 3	CR 6	CR 9	CR 12
Média	0,80692	0,808441	0,800422	0,801578	0,802671
Desvio Padrão	0,00852601	0,0131293	0,00966914	0,0112952	0,00683326

Tabela 126: Média e desvio padrão considerando taxa de regularização em SST-1

-	CR 0	CR 3	CR 6	CR 9	CR 12
Média	0,478975	0,468366	0,459275	0,437858	0,391737
Desvio Padrão	0,0152819	0,0153098	0,013723	0,0287081	0,063143

Tabela 127: Média e desvio padrão considerando taxa de regularização em SST-2

-	CR 0	CR 3	CR 6	CR 9	CR 12
Média	0,864178	0,865808	0,85096	0,849939	0,852547
Desvio Padrão	0,0134662	0,00891595	0,0095634	0,0109223	0,00699255

Tabela 128: Média e desvio padrão considerando taxa de Dropout em MR

-	DO 0	DO 0,2	DO 0,5	DO 0,7	DO 0,9
Média	0,504873	0,803233	0,807703	0,803796	0,804358
Desvio Padrão	0,0142261	0,00747654	0,00999829	0,0110575	0,0087437

Tabela 129: Média e desvio padrão considerando taxa de Dropout em SST-1

-	DO 0	DO 0,2	DO 0,5	DO 0,7	DO 0,9
Média	0,165346	0,464098	0,467717	0,462125	0,464317
Desvio Padrão	0,0151351	0,0223032	0,0137682	0,0186948	0,0186079

Tabela 130: Média e desvio padrão considerando taxa de Dropout em SST-2

-	DO 0	DO 0,2	DO 0,5	DO 0,7	DO 0,9
Média	0,504914	0,850756	0,866277	0,8532	0,853498
Desvio Padrão	0,00976483	0,0137898	0,00753446	0,0108699	0,00886532

APÊNDICE C – Diagramas de Caixa

Neste apêndice são apresentados os diagramas de caixa obtidos para as simulações apresentadas no Capítulo 6.

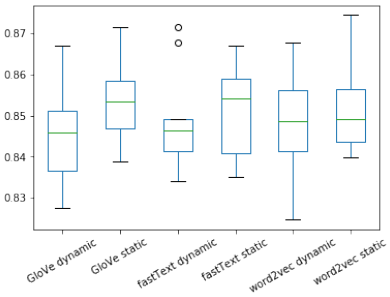


Figura 35: Diagrama de caixa em MR considerando word embeddings

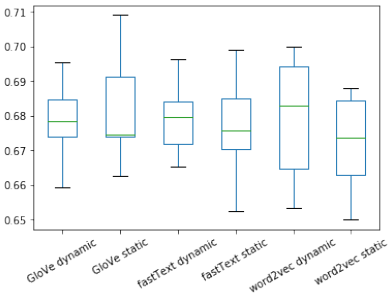


Figura 36: Diagrama de caixa em SST-1 considerando word embeddings

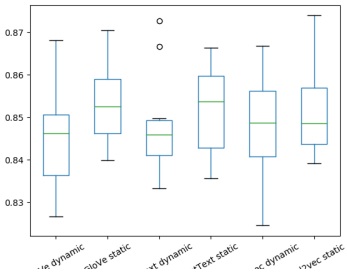


Figura 37: Diagrama de caixa em SST-2 considerando word embeddings

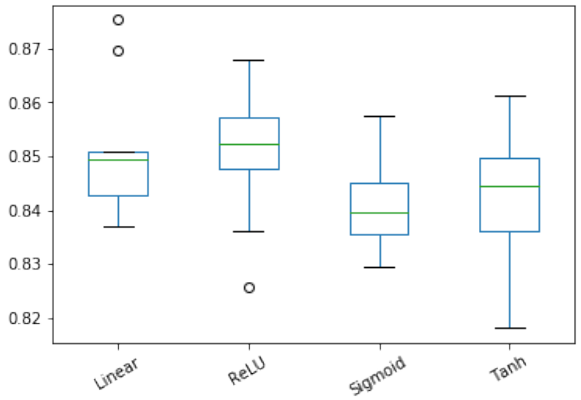


Figura 38: Diagrama de caixa em MR considerando função de ativação

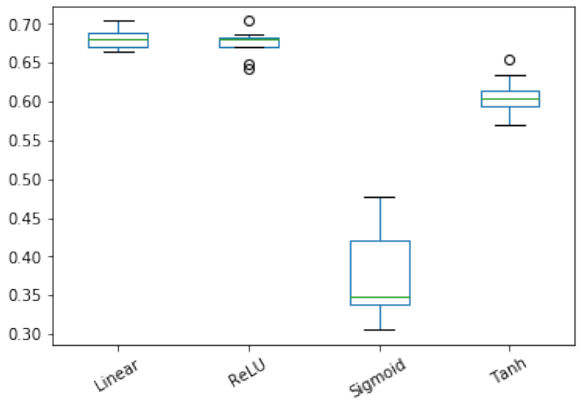


Figura 39: Diagrama de caixa em SST-1 considerando função de ativação

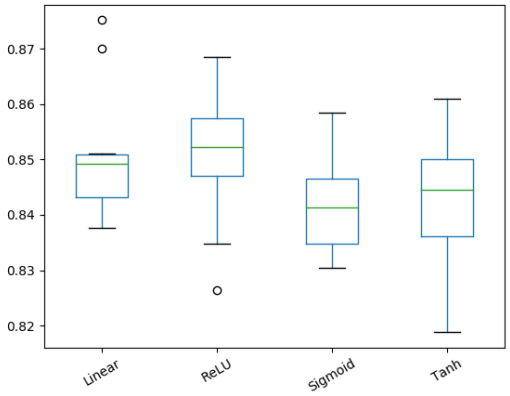


Figura 40: Diagrama de caixa em SST-2 considerando função de ativação

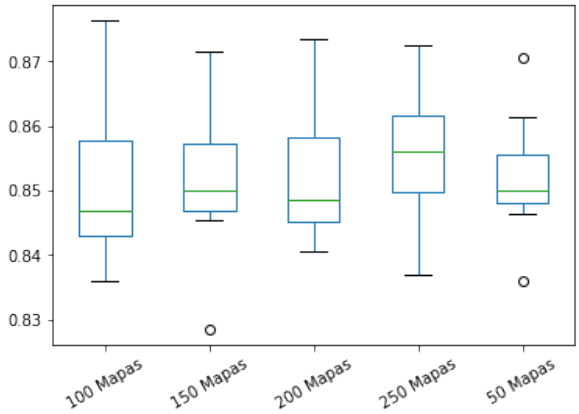


Figura 41: Diagrama de caixa em MR considerando número de mapas

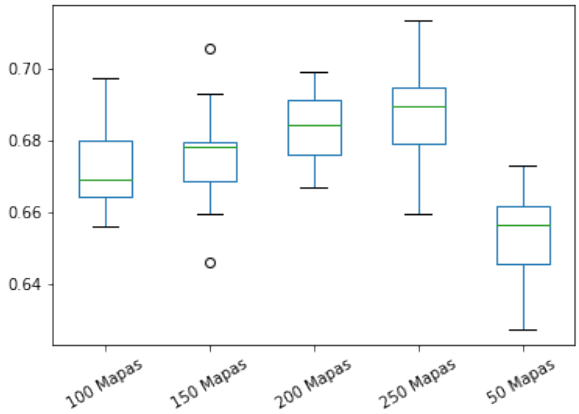


Figura 42: Diagrama de caixa em SST-1 considerando número de mapas

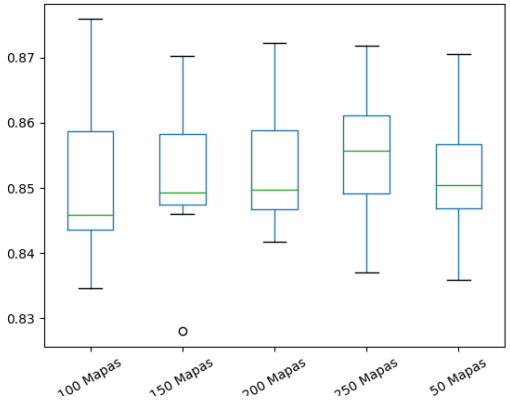


Figura 43: Diagrama de caixa em SST-2 considerando número de mapas

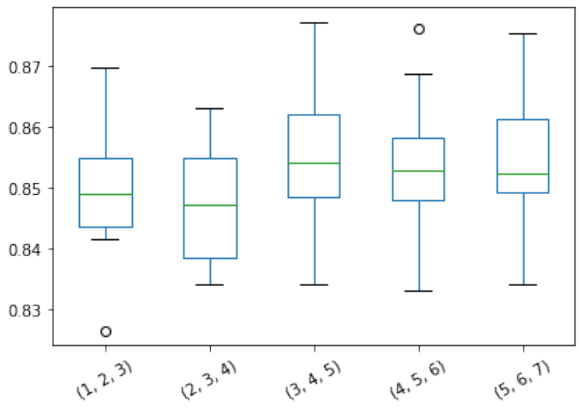


Figura 44: Diagrama de caixa em MR considerando tamanho de filtro

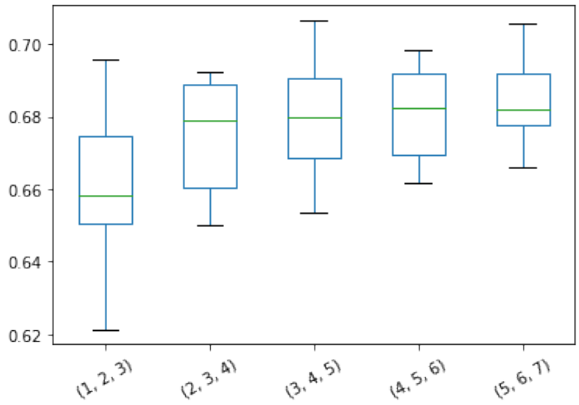


Figura 45: Diagrama de caixa em SST-1 considerando tamanho de filtro

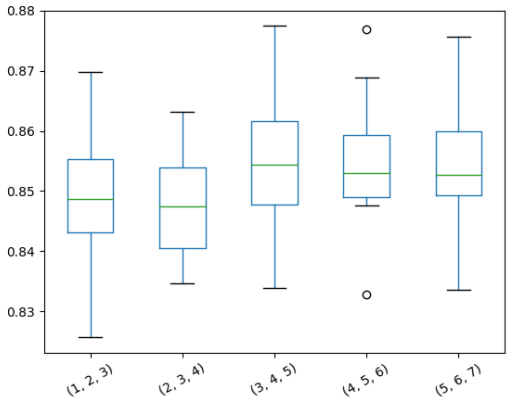


Figura 46: Diagrama de caixa em SST-2 considerando tamanho de filtro

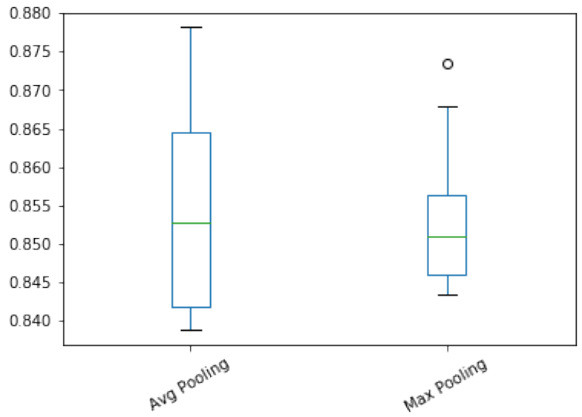


Figura 47: Diagrama de caixa em MR considerando pooling

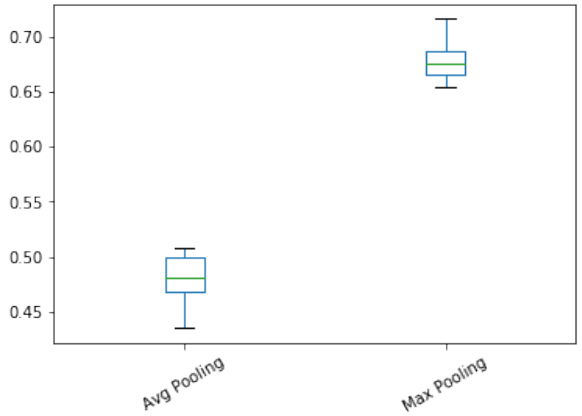


Figura 48: Diagrama de caixa em SST-1 considerando pooling

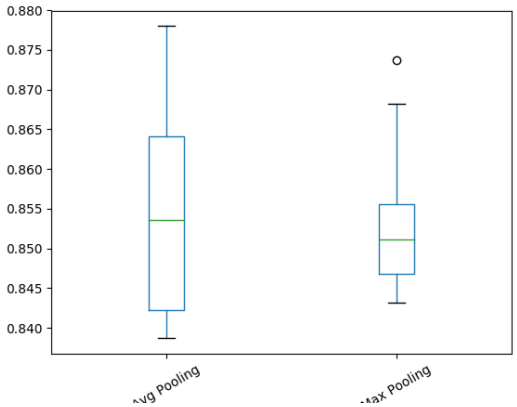


Figura 49: Diagrama de caixa em SST-2 considerando pooling

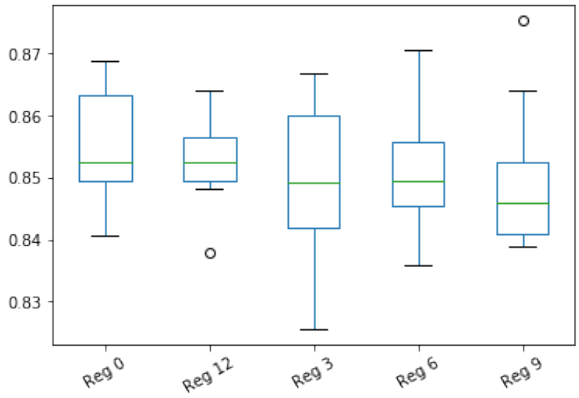


Figura 50: Diagrama de caixa em MR considerando taxa de regularização

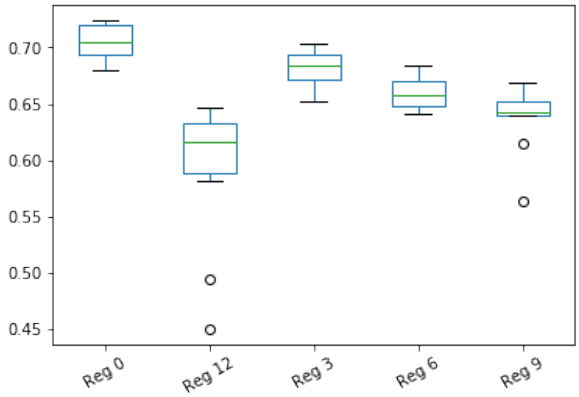


Figura 51: Diagrama de caixa em SST-1 considerando taxa de regularização

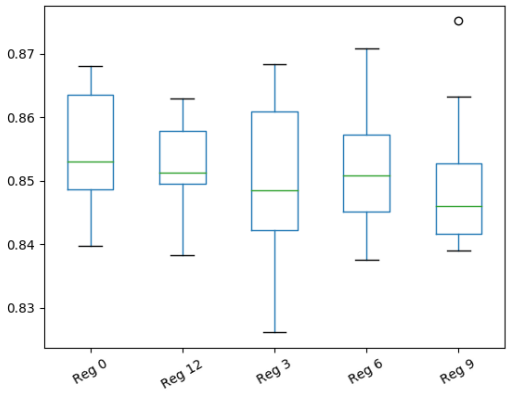


Figura 52: Diagrama de caixa em SST-2 considerando taxa de regularização

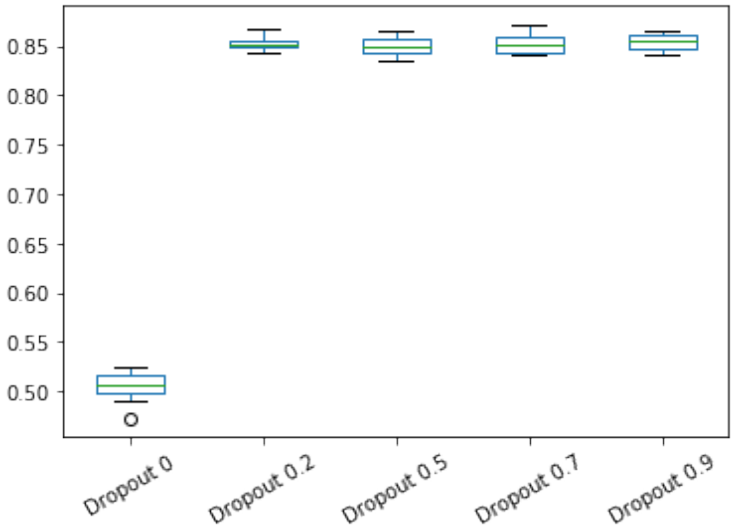


Figura 53: Diagrama de caixa em MR considerando taxa de Dropout

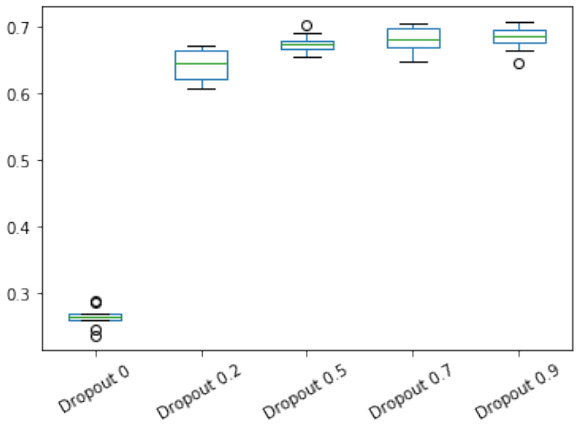


Figura 54: Diagrama de caixa em SST-1 considerando taxa de Dropout

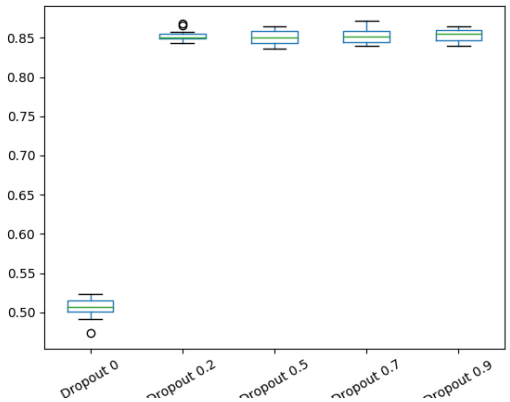


Figura 55: Diagrama de caixa em SST-2 considerando taxa de Dropout