

Universidade Federal do Pampa

Henrico Bertini Brum

**Análise de sentimentos para o português
usando redes neurais recursivas**

Alegrete

2015

Henrico Bertini Brum

Análise de sentimentos para o português usando redes neurais recursivas

Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador: Fábio Natanael Kepler

Alegrete

2015

Henrico Bertini Brum

Análise de sentimentos para o português usando redes neurais recursivas

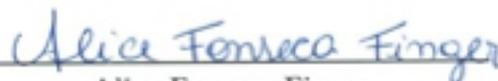
Trabalho de Conclusão de Curso apresentado ao Curso de Graduação em Ciência da Computação da Universidade Federal do Pampa como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Trabalho de Conclusão de Curso defendido e aprovado em 10 de Julho de 15.

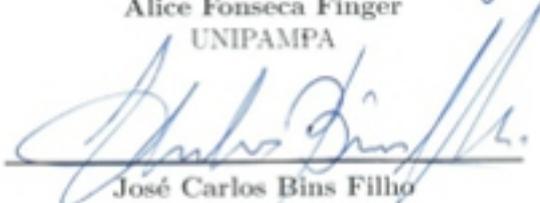
Banca examinadora:



Fábio Natanael Kepler
Orientador



Alice Fonseca Finger
UNIPAMPA



José Carlos Bins Filho
UNIPAMPA

*Dedico esse trabalho à minha mãe que, mesmo não tendo entendido
absolutamente nada da apresentação, disse que foi o
melhor TCC já apresentado.*

:)

Agradecimentos

Esse trabalho não seria possível sem o apoio de três partes fundamentais da minha vida acadêmica e pessoal - meus amigos, meus familiares e meus professores.

Agradeço aos amigos que estavam comigo nas aulas, nos trabalhos e nas provas. Nos que deram aquela sugestão pontual no trabalho de conclusão e nos que me deram carona pra voltar tarde pra casa depois de uma execução cutosa. Agradeço também aos amigos que estavam comigo nos finais de semana, do mate e daquele puxão de orelha quando eu queria desistir de tudo.

Aos meus familiares, agradeço aos meus parentes que sempre me apoiaram e me incentivaram ao estudo. Agradeço também aos que se tornaram parentes ao longo da graduação, à família de Alegrete e à família de Oshawa, que me deu apoio durante o início da definição do tema desse trabalho. Agradeço também à família de Bagé, que teve que aturar grandes depressões e ansiedades na etapa final desse trabalho... hehehe.

Agradeço aos meus professores da graduação que me ensinaram alguma coisa. Alguns de maneira amigável e outros por meio de críticas e algumas notas baixas. Aprendi muito com ambos os estilos e sou grato à todos e especialmente ao meu orientador que conseguiu me guiar pra esse trabalho que aqui apresento.

Nathan: "After a long day of Turing tests you gotta unwind."

Caleb: "What were you doing with Ava?"

Nathan: "What?"

Caleb: "You tore up her picture."

*Nathan: "I'm gonna tear up the f*kin' dance floor, dude."*

Check it out."

(Ex-Machina, 2015)

Resumo

Neste trabalho apresentamos um modelo de treinamento para análise de sentimentos em sentenças para o idioma português brasileiro. Utilizamos uma implementação de Rede Neural Recursiva com Tensor desenvolvida pela Universidade de Stanford que consegue resultados do estado da arte na análise de sentenças para o idioma inglês. Criamos um *treebank* de sentimentos com sentenças em português para o treinamento de um modelo de análise de sentimentos, usando como base um conjunto já existente de resenhas de livros marcadas quanto à polaridade. Esse *treebank* possui 12.512 sentenças em formato *Penn Treebank*, com marcações de polaridades associadas para cada sentença em três classes - Positiva, Neutra e Negativa. Para a geração do *treebank*, extraímos árvores sintáticas das sentenças do córpus utilizando os parsers sintáticos da Universidade de *Stanford* e da Universidade de *Berkeley* treinados com córpus sintáticos em português brasileiro. Nosso trabalho documenta toda a construção do córpus e o uso do mesmo para o treinamento de um modelo de análise de sentimentos para as três classes. O modelo desenvolvido no trabalho foi submetido a análises seguindo a metodologia de *10-Fold Cross-Validation* e obteve 51,18% de acurácia levando em consideração somente sentenças positivas e negativas e 69,08% de acurácia sobre a marcação completa de todas as sentenças.

Palavras-chave: Inteligência artificial. Redes neurais. Análise de sentimentos.

Abstract

In this work we present a trained model for sentiment analysis in Brazilian Portuguese sentences. We used a recursive neural tensor network implementation developed by the University of Stanford team that achieved state-of-the-art results on sentiment analysis on English sentences. A sentiment Treebank for sentiment analysis in Portuguese was formed based on a previous data set of book reviews. The Treebank is formed by 12.512 sentences in Penn Treebank format containing associated polarities for each sentence distributed in three classes - Positive, Neutral and Negative. In order to generate the Treebank, syntactic trees were extracted from the corpus sentences using parsers developed by the University of Stanford and the University of Berkeley trained with a Portuguese data set. Our work describes the construction of the corpus and the use of it on the training of a sentiment analysis model. This model was tested following 10-fold cross-validation method and obtained 51,18% of accuracy on positive and negative prediction and 69,08% of accuracy on the prediction of all the sentences.

Key-words: Artificial intelligence. Neural networks. Sentiment analysis.

Lista de ilustrações

Figura 1 – Círculo Afetivo proposto por Watson e Tellegen	26
Figura 2 – Representação simples de uma Rede Neural (RN)	28
Figura 3 – Representação simplificada do modelo de RNTN para análise de senti- mentos	33
Figura 4 – Representação da análise de um vetor-composição na RNTN.	34
Figura 5 – O modelo recursivo de rede neural para sentimentos aplicado a uma sentença	35
Figura 6 – A camada de tensor da RNTN.	36
Figura 7 – O tensor inserido na RNTN	37
Figura 8 – Representação gráfica dos passos do desenvolvimento de um modelo de análise de sentimentos para Português brasileiro.	39
Figura 9 – Sentença retirada do Córpus Reli analisada sintaticamente em formato PTB.	42
Figura 10 – Número de sentenças com cada polaridade e nota da resenha onde esta se encontra.	46
Figura 11 – Representação de <i>Cross Validation</i> onde $k = 4$	47
Figura 12 – Acurácia sobre todos nós da árvore de polaridades durante treinamento do modelo.	49
Figura 13 – Acurácia sobre o topo das árvores de polaridades durante treinamento do modelo.	50
Figura 14 – Acurácia Combinada sobre todos nós da árvore de polaridades durante treinamento do modelo.	50
Figura 15 – Acurácia Combinada sobre o topo das árvores de polaridades durante treinamento do modelo.	51
Figura 16 – Matriz-confusão da predição de todos os nós-raiz de um caso de teste do treinamento.	52
Figura 17 – Matriz-confusão da predição de todos os nós de um caso de teste do treinamento.	53
Figura 18 – Gráfico de medidas dos experimentos estruturais na RNTN usando o córpus Berkeley.Reli.	55
Figura 19 – Gráfico de medidas dos experimentos estruturais na RNTN usando o córpus Stanford.Reli.	55

Lista de tabelas

Tabela 1 – Resultado final obtido com a média dos dez casos de teste criados. . . .	48
Tabela 2 – Experimentos com modificações estruturais na RNTN no córpus Berkeley.Reli.	54
Tabela 3 – Experimentos com modificações estruturais na RNTN no córpus Stanford.Reli.	54
Tabela 4 – Experimentos variando taxa de aprendizagem na RNTN no córpus Berkeley.Reli.	56
Tabela 5 – Experimentos variando taxa de aprendizagem na RNTN no córpus Stanford.Reli.	56

Lista de siglas

IA Inteligência Artificial

LSA Latent Semantic Analysis

PLN Processamento de Linguagem Natural

PTB Penn Treebank

RN Rede Neural

RNN Rede Neural Recursiva

RNTN Rede Neural Recursiva com Tensor

Sumário

1	Introdução	23
1.1	Organização	24
2	Fundamentação	25
2.1	Análise de Sentimentos	25
2.2	Aprendizado de Máquina	26
2.2.1	Redes Neurais	27
2.3	Trabalhos Relacionados	29
2.4	Objetivos Específicos	31
3	Abordagem e Desenvolvimento	33
3.1	A Rede Neural Recursiva com Tensor	34
3.2	O <i>Treebank</i> de Sentimentos de Stanford	38
3.3	Analisando Sentimentos em Português Brasileiro	38
3.3.1	Córpus de sentimentos	39
3.3.2	A análise sintática	40
3.3.3	A formação de um córpus de sentimento em Português brasileiro	42
4	Experimentos e Análises	45
4.1	Metodologia e Testes	45
4.1.1	O aprendizado na rede neural	49
4.2	Melhorando o modelo para Português brasileiro	53
4.2.1	Modificações estruturais	53
4.2.2	Modificações de aprendizagem	54
5	Conclusões	57
5.1	Contribuições e Possibilidades Futuras	58
	Referências	59

1 Introdução

Diariamente, milhões de informações circulam pelos cabos de fibra ótica e movimentam sites, microblogs e redes sociais. O conteúdo presente nas redes sociais, por exemplo, se estende desde análises e comentários sobre filmes e programas televisivos até conversas e experiências dos seus usuários. A questão é que esses dados não se encontram estruturados em um formato compreensível para um algoritmo ou sistema computacional.

A Inteligência Artificial (IA) provê técnicas que podem facilitar a compreensão de dados em linguagem humana, de maneira que possamos usar como entrada uma sentença em linguagem natural e termos como saída um conjunto de dados extraídos desta. Isso facilita a análise de dados em larga escala, que podem ser coletados de documentos ou redes sociais em linguagem natural.

Hoje em dia são comuns os casos de empresas e consumidores que se relacionam através de redes sociais. Segundo Santos et al. (2011) “essa opinião informal ou formal, dependendo de onde foi postada, é de extrema importância, pois refletirá o sentimento sem censura do usuário”.

Outra área que pode se beneficiar da análise de sentimentos é o campo das finanças. Alguns estudos investigaram a influência de sentimentos em relatórios financeiros e na movimentação do mercado de ações, como Aguiar (2012) e Yoshinaga e Junior (2009)

Ainda existem dificuldades em sistemas computacionais compreenderem exatamente como os consumidores e clientes se sentem em relação aos produtos e serviços de uma companhia. A extração automática do sentimento ou sensação de sentenças também seria um grande avanço para pesquisas de marketing e análise de comportamento do consumidor.

Neste trabalho estudamos técnicas de Processamento de Linguagem Natural (PLN) que nos possibilitam extrair o sentimento, ou a intenção de sentenças no idioma Português brasileiro, tornando possível avaliar a opinião de sentenças e frases em relação a determinados assuntos.

Buscamos analisar sentenças em português brasileiro e as classificar em três classes. As sentenças que possuírem conotação positiva em relação ao assunto tratado, serão marcadas como Sentenças Positivas, enquanto as que possírem conotação negativa serão consideradas Sentenças Negativas. No caso de uma sentença não ser classificada em uma dessas classes, não expressando uma opinião específica, ela será considerada uma Sentença Neutra.

Para classificarmos sentenças, utilizaremos técnicas de aprendizado de máquina e

redes neurais a fim de treinarmos um modelo de classificação que pode ser aplicado em qualquer conjunto de sentenças.

1.1 Organização

Este trabalho está organizado da seguinte maneira - no [Capítulo 2](#) discutimos os fundamentos do trabalho, trazendo definições para os termos utilizados e onde eles se aplicam em nosso contexto. Apresentamos trabalhos relacionados que nos nortearam no início da pesquisa, com discussões acerca de suas vantagens, desvantagens e técnicas empregadas na análise de sentimentos. Ao final do capítulo apresentamos os objetivos específicos do trabalho.

No [Capítulo 3](#) buscamos explicar como abordamos o problema, quais técnicas foram escolhidas para resolvê-lo e quais os passos que foram necessários para desenvolver o trabalho. Além disso, apresentamos os recursos empregados em todo o desenvolvimento e criamos um verdadeiro passo-a-passo da metodologia aplicada, assim como os desafios enfrentados durante toda a pesquisa.

No [Capítulo 4](#) discutimos nossa metodologia de experimentos e apresentamos os resultados obtidos nestes, assim como iniciamos uma reflexão sobre possíveis melhorias que foram aplicadas e seus resultados.

No [Capítulo 5](#) encerramos o trabalho e concluímos nossa pesquisa, apresentando possíveis melhorias e uma reflexão sobre trabalhos futuros, apresentando alternativas de possibilidades futuras e nossas discussões sobre o quanto conseguimos realizar de nosso objetivo inicial.

2 Fundamentação

Um dos grandes desafios na área de IA é a compreensão de informações em formato não estruturado. Quando tratamos de dados estruturados, seja em uma tabela ou base de dados, temos informações dispostas de uma maneira que o computador consegue identificar, comparar e categorizar. De posse de dados não padronizados surge o desafio de torná-los compreensíveis para a máquina, para que as informações relevantes contidas no conjunto possam ser extraídas por um algoritmo.

A linguagem humana é a maneira como os seres humanos se comunicam entre si, composta por palavras que juntas formam sentenças seguindo um padrão de idioma com regras gramaticais e sintáticas. Essas regras, apesar de definirem o idioma, não são suficientes para estruturar a informação que se deseja transmitir, e cabe ao campo do Processamento de Linguagem Natural (PLN) o desafio de extrair informações destas sentenças.

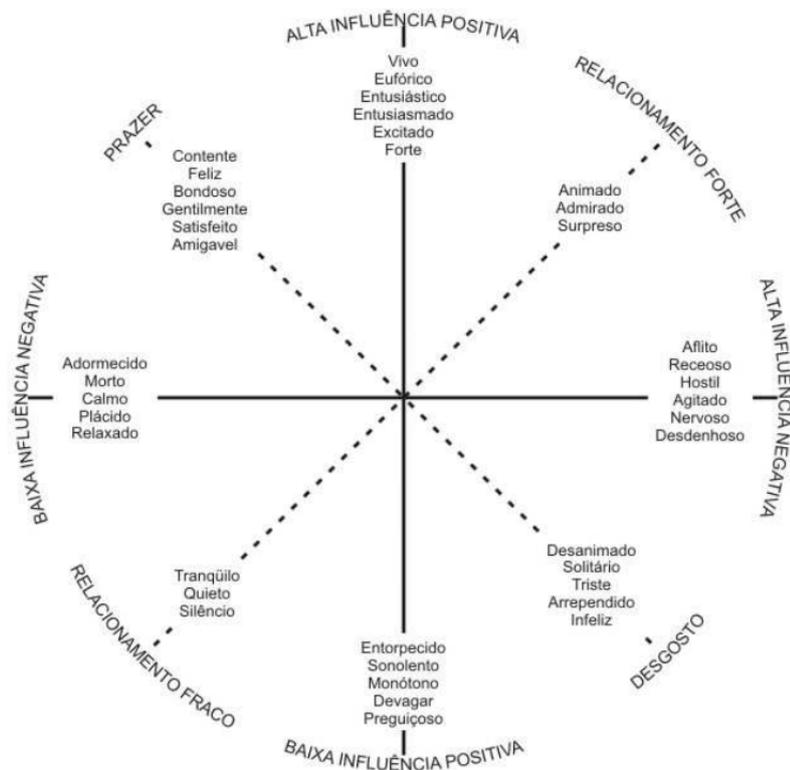
2.1 Análise de Sentimentos

Textos e sentenças contêm diversas informações, algumas delas fáceis de serem encontradas e extraídas e outras demandam relacionamentos complexos de palavras para serem compreendidas em sua totalidade. Ao analisar um conjunto de sentenças em português brasileiro temos a nossa disposição informações tais como a estrutura gramatical usada, o tipo de voz (passiva ou ativa), nomes dos envolvidos no texto ou localidades que foram citadas. O foco deste trabalho é extrair o sentimento ou a intenção de cada frase, informação essa que nem sempre se encontra escrita diretamente nas sentenças analisadas.

“Análise de sentimentos é o campo de estudo que analisa opiniões de pessoas, sentimentos, avaliações, atitudes e emoções de sentenças e textos.” (LIU, 2012). Nosso enfoque nesse trabalho será a sensação ou o sentimento de sentenças. Diferentes abordagens trazem marcações diversas de sentimentos que podem ser extraídos. Algumas abordagens utilizam classes, tais como alegria, tristeza, raiva ou surpresa, enquanto outras utilizam números para indicar polaridades nas sentenças, como por exemplo utilizar o número 0 para negativo e 1 para positivo.

A definição de classes com as quais trabalhar é importante, pois os métodos de classificação devem levar em consideração o conjunto de classes que usaremos. Na Figura 1 é possível ver um exemplo de classes de sentimentos que podem ser analisados em uma sentença. Devemos ressaltar que nem sempre será possível classificar o sentimento correto de uma sentença, pois muitas sentenças não representam um sentimento específico.

Figura 1 – Círculo Afetivo proposto por Watson e Tellegen



Fonte: (MARTINAZZO, 2010)

Nossa abordagem busca a extração da polaridade dos sentimentos, o que reduz a quantidade de classes para três - Positiva, Negativa ou Neutra. Dessa maneira, abstraímos o conceito de sentimentos como surpresa ou medo para sabermos a opinião do autor da sentença a algum tema específico.

Os desafios da área de análise de sentimentos se assemelham a situações comuns na PLN, como sentenças com erros gramaticais e palavras escritas de forma errada, porém especificamente para nosso trabalho, contamos com um desafio a mais que são frases de duplo entendimento, com ironias e sarcasmo, onde podemos ter uma interpretação errada da frase se não analisarmos a sentença corretamente, entre outras situações práticas específicas que serão vistas no decorrer das implementações.

2.2 Aprendizado de Máquina

“Aprendizado de máquina se trata de fazer computadores modificarem ou adaptarem suas ações de tal maneira que estas se tornem mais precisas, onde essa precisão é medida pelo quão bem as decisões tomadas refletem as escolhas corretas” (MARS LAND,

2011). Algoritmos que possuem essa finalidade são implementados com a habilidade de calcular suas ações e as melhorar com o tempo, dessa maneira podem atingir ou ao menos aproximar os resultados esperados a medida que possuem mais entradas de dados.

No contexto de análise de sentenças, um algoritmo receberá um conjunto de palavras, onde diversas características de cada palavra serão apresentadas em formato numérico. Essas características são então submetidas a uma função $f(x)$ que usará pesos para cada característica e retornará um resultado. Ao término da operação, haverá uma saída e o próprio algoritmo deverá avaliar se a saída se aproxima ou não do esperado. Em caso positivo, se mantém os pesos de cada característica, em caso negativo, os pesos são modificados para que a saída se aproxime mais do resultado correto para a respectiva entrada.

Podemos ter duas formas principais de aprendizado de máquina - aprendizado supervisionado e não supervisionado.

No aprendizado não supervisionado, um algoritmo busca resolver um problema sem ter a exata noção dos resultados esperados para tal. Neste esquema de treinamento somente os padrões de entrada estão disponíveis, ao contrário do aprendizado supervisionado (BRAGA; FERREIRA; LUDERMIR, 2007).

O paradigma não supervisionado de aprendizado funciona muito bem para problemas de busca de padrões, onde não sabemos exatamente quais padrões estamos procurando. Como nosso problema se trata de uma classificação e sabemos exatamente quais classes temos, uma abordagem não supervisionada pode comprometer nosso desempenho sem necessidade.

O aprendizado supervisionado é uma abordagem que visa treinar um algoritmo para realizar uma tarefa baseando-se em um conjunto de entradas e saídas já conhecidos e corretos. Esse conjunto de dados pode ser chamado de supervisor, ou professor externo e é responsável por estimular as entradas por meio de padrões e observar a saída calculada pela mesma (BRAGA; FERREIRA; LUDERMIR, 2007).

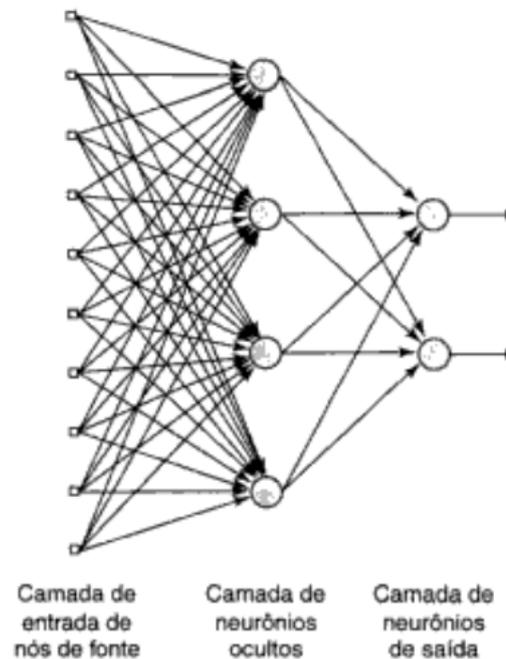
Apesar de muito eficiente, essa abordagem é muito dependente do conjunto de dados de entrada que recebe, havendo a necessidade de um conjunto robusto e que contenha as informações necessárias para o treinamento correto do algoritmo.

2.2.1 Redes Neurais

“Redes Neurais (RNs) são um método geral e prático para o aprendizado de aproximações de funções a partir de exemplos” (NETO; NICOLETTI, 2005). Esses algoritmos são capazes de modificar sua própria função ou os valores aplicados à entrada de dados após uma execução. Essa habilidade as permite aprender por meio de exemplos, fazendo com que sejam muito utilizadas para aprendizado supervisionado.

Uma RN é caracterizada pela utilização de neurônios, que são unidades de processamento de informação (HAYKIN, 2001). A RN recebe diversos *sinais de entrada* que são multiplicados por pesos W , compondo assim um neurônio. Os sinais de entrada podem ser vistos como representações vetoriais de objetos de entrada, em nosso caso, palavras. Em cada neurônio temos a soma de todos os sinais de entrada multiplicados por pesos diferentes que irão acentuar certas características destes sinais.

Figura 2 – Representação simples de uma RN



Fonte: (HAYKIN, 2001)

Chamamos de *camada* um grupo de neurônios que recebem sinais de entradas da camada anterior, enviam sinais de saída para a camada seguinte e não possuem ligações entre si. O número de neurônios em cada camada é variável. Na Figura 2 vemos uma representação de RN. Em alguns casos, as RNs podem possuir diferentes arquiteturas.

$$n_1 = f\left(W \begin{bmatrix} s_1 \\ s_2 \end{bmatrix}\right) \quad (2.1)$$

Na Equação 2.1 podemos identificar uma fórmula genérica de um valor de neurônio. Temos n_1 como o valor do neurônio, enquanto W será o peso multiplicando os sinais de entrada da camada anterior, representados por s_1 e s_2 . A função $f(x)$ será a função de ativação do neurônio.

Uma propriedade das redes neurais multicamadas é a captura de relacionamentos não-lineares entre as características dos sinais entrada da rede. Esses relacionamentos são reconhecidos pela RN por conta das interações existentes entre os neurônios na troca de camada, criando combinações dificilmente observáveis por método empírico.

2.3 Trabalhos Relacionados

A análise de sentimentos em português brasileiro já existe há algum tempo, porém ainda não está consolidada uma ferramenta ou técnica definitiva para a classificação de sentenças quanto à sua polaridade. Dessa maneira, buscamos diversas abordagens que nos auxiliaram a compreender essa área de Processamento de Linguagem Natural (PLN) e nos deram embasamento para desenvolver nosso trabalho.

Nos últimos anos diversos estudos em relação à identificação de sentimentos e intenções em sentenças cresceu consideravelmente. Segundo [Armellini e Rodrigues \(2013\)](#), o aumento significativo e contínuo em que se encontram as redes sociais tem feito delas fontes muito atrativas para avaliação da opinião pública sobre uma determinada marca. Talvez essa seja a maior motivação para empresas e universidades investirem mais em trabalhos no ramo da análise de sentimentos.

Encontramos alguns trabalhos que já haviam proposto técnicas para analisar sentimentos em sentenças. Em [Araújo, Gonçalves e Benevenuto \(2013\)](#), temos oito alternativas de extração de sentimentos em sentenças em português. Foram testadas abordagens envolvendo redes neurais e aprendizado supervisionado especificamente para o microblog *Twitter*, porém esse trabalho, apesar de ter sido realizado no Brasil trabalha com Tweets em inglês. Assim sendo, córpus como o SentiWordNet ([ESULI; SEBASTIANI, 2006](#)) e a ferramenta LIWC ([PENNEBAKER; BOOTH; FRANCIS, 2007](#)) puderam ser utilizadas.

Algumas pesquisas buscaram focar a análise de sentimento em um ramo específico, como é o caso de [Godbole, Srinivasaiyah e Skiena \(2007\)](#) onde foram analisados os sentimentos em notícias e blogs. [Alm, Roth e Sproat \(2005\)](#) analisaram histórias infantis e trazem uma anotação de sentimentos diferenciada, envolvendo as classes raiva, desgosto, medo, felicidade, tristeza, surpresa positiva e surpresa negativa.

Ainda em inglês, [Strapparava e Mihalcea \(2008\)](#) se concentraram na construção de um conjunto de dados anotados para seis emoções: raiva, desgosto, medo, satisfação, tristeza e surpresa. A marcação automática foi feita sem uma técnica complexa, somente uma variação de Latent Semantic Analysis (LSA), onde mantém-se somente o radical de uma palavra e dá-se uma classe a ela.

[Martinazzo \(2010\)](#) também buscou extrair sentimentos de textos pequenos com a ajuda de uma técnica de LSA, agrupando palavras em classes. Ela conseguiu um grupo

com palavras que indicam cada sentimento estudado e conseguiu um acerto de até 70% na identificação de emoções.

Ferreira (2010) utilizou uma abordagem baseada em algoritmos genéticos e uma pesquisa por características das palavras. Foram retiradas postagens do microblog *Twitter* e para a avaliação foram feitas marcações manuais no conjunto de dados a fim de servir como córpus para análise. A técnica levou em consideração características retiradas empiricamente com base na observação dos pesquisadores e de trabalhos anteriores. Esse método tornou necessário uma redução posterior de características irrelevantes para o problema, algo que em nossa abordagem não será necessário por causa da habilidade da rede neural de conseguir aprender características automaticamente.

O uso do *Twitter* como córpus é muito explorado em outros trabalhos. Pak e Paroubek (2010) e Go, Huang e Bhayani (2009) pesquisaram métodos para extração de postagens com marcação de polaridade associada baseando-se em emoticons presentes no texto, supervisionando o treinamento das palavras de cada postagem.

Silva (2010) propôs uma ferramenta baseada na análise de palavras separadamente. A pesquisa traz uma classificação em quatro etapas, as palavras são analisadas quanto a polaridade, então as expressões negativas e adversativas são tratadas e, por fim, é feita sumarização das opiniões. O sistema ainda traz uma opção para que o usuário controle manualmente a polaridade das expressões.

Outro trabalho que trata de análise de sentimentos é o de Pang e Lee (2005). Os autores analisaram não só a polaridade associada, mas propuseram uma mudança na avaliação, criando uma escala de zero à quatro estrelas para a avaliação de sentenças. Assim não se trata exatamente da análise *booleana* de uma sentença, mas sim de uma abordagem de graus de similaridade. Em vez de classificar as sentenças em positivas e negativas, busca-se classificá-las em cinco graus relativos de positividade.

Pang e Lee (2008) realizaram um vasto estudo das técnicas e abordagens aplicados à análise de sentimentos. Durante o estudo foram vistas técnicas de aprendizado de máquina, engenharia de características e algumas abordagens não supervisionadas que enriqueceram muito nossa visão acerca da análise de sentimentos.

Um dos trabalhos que trouxe uma enorme contribuição para o campo da Análise de Sentimento é o de Socher et al. (2013a), que foi desenvolvido na Universidade de Stanford. Neste trabalho os pesquisadores propõem uma abordagem que envolve um robusto *Treebank* de Sentimentos e um modelo de rede neural recursivo que utiliza um tensor agregado à estrutura da rede.

Os resultados obtidos nesse trabalho foram comparados a outros modelos, como redes neurais recursivas comuns, classificadores Naive Bayes(NB), bigramas NB e Máquinas de Vetor de Suporte, atingindo acurácia de até 80.7% na previsão de sentimentos em

todos os nós, se tornando o estado da arte para o idioma inglês. Por essa razão, usamos o modelo de [RNTN](#) como base fundamental para nossa pesquisa.

2.4 Objetivos Específicos

Os objetivos específicos desse trabalho são os de analisar sentenças em Português e extrair os sentimentos associados a elas. Para tal, decidimos utilizar o modelo de [RNTN](#) proposto por [Socher et al. \(2013a\)](#).

Nos propusemos a replicar o modelo utilizando uma base de dados em Português brasileiro e o *Treebank* de sentimentos de Stanford, utilizando uma nova base de dados completamente em nosso idioma. O processo deve ser totalmente documentado para que possa ser replicado seguindo-se uma sequência de passos descritos nesse trabalho.

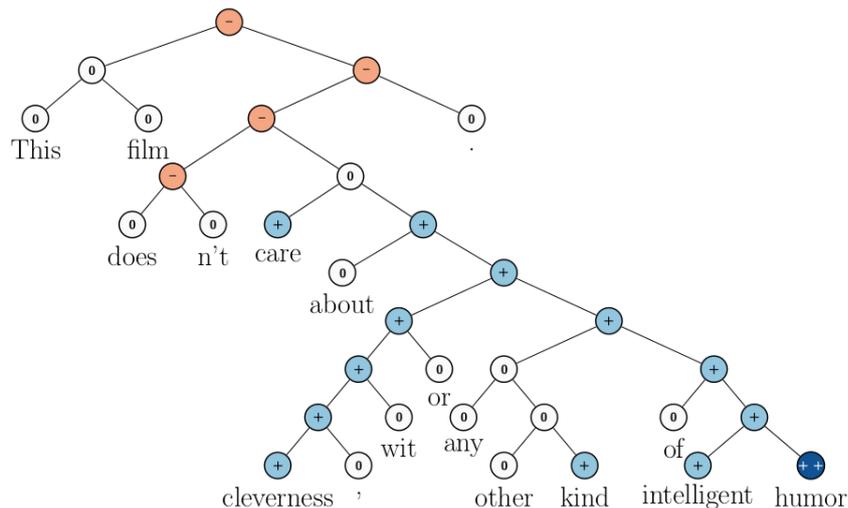
Após a geração do modelo, buscamos testá-lo quanto à acurácia na análise de sentimentos e propor melhorias para o idioma Português brasileiro, assim como discutir futuros trabalhos que possam fazer uso do modelo ou do fluxograma de treino do modelo.

3 Abordagem e Desenvolvimento

O trabalho que nos baseamos para analisar sentenças é o proposto por [Socher et al. \(2013a\)](#). Neste trabalho os autores apresentam um modelo de análise de sentimentos que faz uso de um modelo de rede neural recursiva com tensor, uma abordagem que trouxe melhorias sobre modelos anteriores e se tornou o estado da arte da análise de sentimentos para o idioma inglês. Além disso, também introduzem o *treebank* de sentimentos da Universidade de Stanford que é usado como arquivo de treino para a Rede Neural Recursiva com Tensor ([RNTN](#)).

O funcionamento do modelo proposto por [Socher et al. \(2013a\)](#) de [RNTN](#) se baseia em três fundamentos importantes da [IA](#) - o aprendizado supervisionado por meio de redes neurais, uma modelagem recursiva no uso dessa rede neural ([SOCHER et al., 2011](#)) e um tensor que acentua as características composicionais do modelo.

Figura 3 – Representação simplificada do modelo de [RNTN](#) para análise de sentimentos



Fonte: ([SOCHER et al., 2013a](#))

Na [Figura 3](#) vemos uma sentença sendo analisada pela [RNTN](#). O procedimento da análise divide a sentença em pares de palavras e procede em *bottom-up* classificando os nós intermediários até atingir o topo. O topo da árvore de sentimentos possuirá então a polaridade associada da sentença inteira.

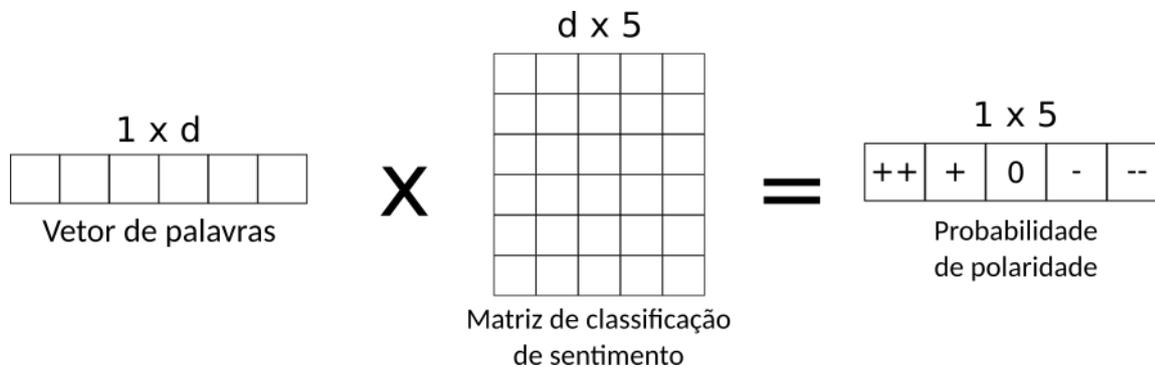
3.1 A Rede Neural Recursiva com Tensor

A análise de polaridades de sentenças é uma tarefa complicada do ponto de vista computacional. Enquanto seres humanos conseguem perceber a sensação de positividade ou negatividade de uma sentença, os algoritmos de aprendizado de máquina que resolvem esse problema tendem a buscar a sensação por meio de análise da combinação de palavras da sentença.

Cada palavra da sentença é representada por um vetor de palavras. Esse vetor possui dimensão d e é composto por valores numéricos que capturam características das palavras. Neste trabalho, o valor d será o espaço vetorial fixo que o modelo irá utilizar em todos os nós das árvores sentimentais usadas. Usaremos a nomenclatura de vetor de palavras para representar não só palavras, mas conjunções e sintagmas de sentenças. Sintagmas são os trechos, ou sub-árvores, das sentenças analisadas pelo algoritmo.

Utilizando um classificador *softmax* podemos calcular a polaridade de um vetor de palavras. Para isso, usaremos uma abordagem neural que multiplica esse vetor d por uma matriz de pesos $W_s \in \mathbb{R}^{d \times 5}$. Os pesos dessa matriz serão alterados durante a fase de treinamento do modelo.

Figura 4 – Representação da análise de um vetor-composição na RNTN.

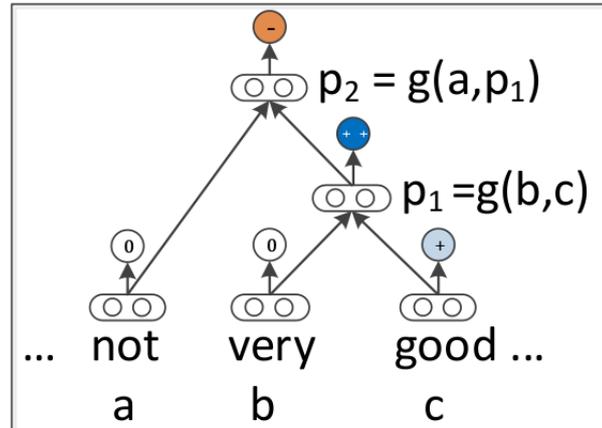


Na Figura 4 temos uma representação que mostra a geração da probabilidade de marcação sentimental. A função *softmax* será aplicada ao vetor gerado, normalizando seus valores e escolhendo o maior dentre eles como a polaridade associada ao vetor-composição.

A composicionalidade desse modelo segue uma estratégia de cálculo de nós em *bottom-up*. Inicialmente são calculados os nós mais abaixo na árvore da sentença e então são calculados os sintagmas superiores até que se atinja o topo da árvore.

O vetor-composição deve possuir dimensões iguais em todos os nós, para que uma mesma função composicional seja aplicada em toda a árvore. Essa função composicional possuirá uma matriz de treinamento muito semelhante a matriz de classificação de sentimentos.

Figura 5 – O modelo recursivo de rede neural para sentimentos aplicado a uma sentença



Fonte: (SOCHER et al., 2013a)

Na Figura 5 podemos ver um exemplo de trecho de árvore de sentimentos sendo analisado. Neste exemplo podemos entender mais facilmente a recursão na rede neural.

Temos p_1 como o nó formado por dois vetores, b e c , e p_2 como sendo a composição de a e p_1 . Para calcular o valor de p_1 e p_2 teremos a seguinte equação:

$$p_1 = f\left(W \begin{bmatrix} b \\ c \end{bmatrix}\right), p_2 = f\left(W \begin{bmatrix} a \\ p_1 \end{bmatrix}\right) \quad (3.1)$$

Novamente teremos uma matriz de treinamento, porém desta vez ela é descrita por $W \in \mathbb{R}^{d \times 2d}$. Essa matriz será multiplicada pela matriz formada pelos vetores de entrada concatenados. Esse cálculo resultará em um vetor de dimensão d , ficando assim apto a ser usado como entrada da camada seguinte da rede neural, possibilitando a recursão do modelo.

Um classificador *softmax* é aplicado tanto nos vetores b e c como nos vetores p_1 e p_2 para calcular o sentimento dos sintagmas da sentença.

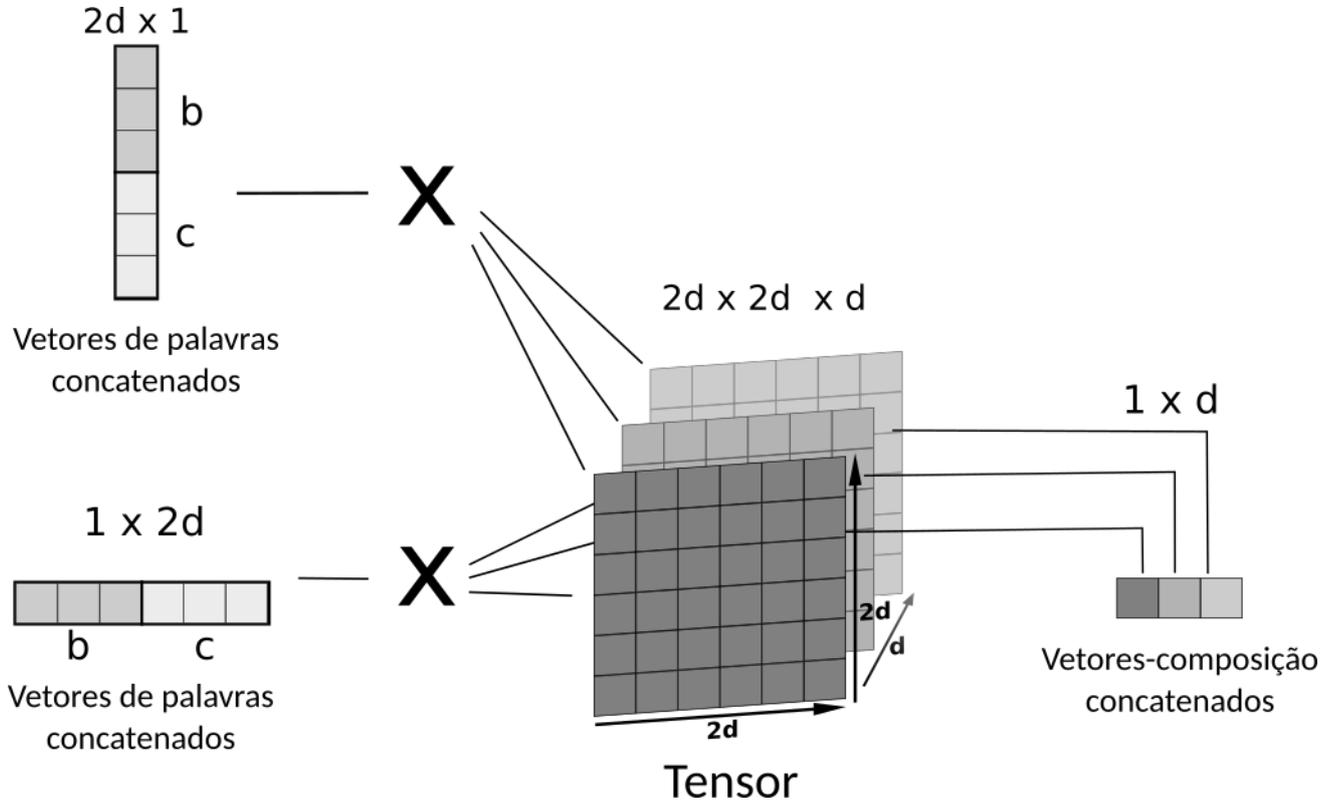
A função aplicada ao vetor-composição será $f = \tanh$ para linearizar as produções desse vetor entre 0 e 1.

Esse modelo de composição é muito eficaz, porém se torna dependente desta composição vetorial. Para sentenças muito longas, essa composição deixa de ser eficiente e muita informação pode ser perdida. Torna-se necessário o uso de uma representação mais atenuante para capturar composições mais complexas.

Por essa razão foi proposto o uso de uma camada extra na rede neural que, segundo os autores, pode acentuar ainda mais as composições formadas no modelo. Essa camada possui um tensor, uma matriz tridimensional contendo valores numéricos, que

será treinado em conjunto com o modelo.

Figura 6 – A camada de tensor da RNTN.



O tensor, definido por $V \in \mathbb{R}^{2d \times 2d \times d}$ é uma matriz tridimensional que será dividida em d fatias. Cada fatia irá capturar um tipo de influência que um nó filho tem em um nó pai. Essa composição pode ser vista na [Figura 6](#).

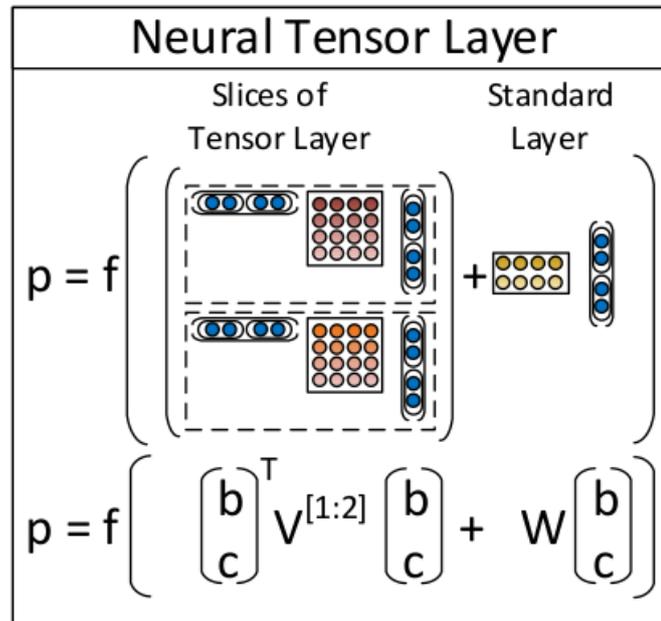
Cada fatia do tensor será multiplicada pela concatenação dos vetores-composição dos filhos do nó e pela matriz transposta destes, gerando um valor numérico. Teremos então d valores resultantes, um para cada fatia do tensor. Esses valores serão concatenados formando um vetor de dimensão d que será somado ao vetor resultante da camada padrão da rede neural, vista na [Equação 3.1](#).

A RNTN é composta destas duas camadas, enquanto a análise de sentimentos será feita por meio do classificador *softmax* aplicado nos vetores-composição da árvore de sentimentos. Podemos conferir a equação completa da rede neural na [Figura 7](#).

A [Equação 3.2](#) mostra o cálculo de p_1 na RNTN:

$$p_1 = f \left(\begin{bmatrix} b \\ c \end{bmatrix}^T V^{[1:d]} \begin{bmatrix} b \\ c \end{bmatrix} + W \begin{bmatrix} b \\ c \end{bmatrix} \right) \quad (3.2)$$

Figura 7 – O tensor inserido na RNTN



Fonte: (SOCHER et al., 2013a)

O autor ainda cita como possível melhoria a inserção de uma segunda camada na rede neural, porém seus experimentos iniciais mostraram que a otimização de um modelo usando mais uma camada seria muito custosa.

O treinamento na RNTN ocorre por *backpropagation* (HECHT-NIELSEN, 1989). Esse algoritmo consegue analisar as saídas do algoritmo e compará-las com o cópulo de treinamento, gerando assim uma taxa de erro. Esse procedimento é muito comum em técnicas de aprendizado de máquina.

Como sabemos, ao término da rede neural teremos um vetor-composição do topo da árvore. A matriz de classificação irá gerar o vetor de probabilidades e o *softmax* irá escolher a polaridade indicada. O *backpropagation* irá comparar esse resultado com o resultado contido no cópulo de treino para calcular a taxa de erro.

O erro será propagado por toda a estrutura e a classificação de cada nó da árvore será comparada com o arquivo de treino, para que o algoritmo saiba a partir de qual nó não há erro de marcação.

O algoritmo irá modificar seus pesos, aumentando os pesos que deveriam levar à marcação correta e diminuindo os pesos que levariam ao contrário. A modificação será feita em três partes da estrutura - na matriz de classificação de sentimentos, na matriz de pesos da camada padrão da RN e no tensor.

3.2 O *Treebank* de Sentimentos de Stanford

Um elemento essencial do trabalho original é a presença do *Treebank* de sentimento de Stanford (SOCHER et al., 2013b), que foi usado nos testes da RNTN.

Esse *treebank* de sentimentos é um conjunto de 11.855 sentenças em inglês marcadas quanto à sua polaridade e em formato Penn Treebank (PTB). A formatação desse cópús é importante pois define o padrão que o algoritmo aceita para treinamento de um novo modelo. Para fins de replicação é necessário que se utilize o formato PTB com as marcações de polaridade no cópús de treino.

O *treebank* surgiu da manutenção de um cópús formado a partir do site *Rotten Tomatoes*¹ e publicado por Pang e Lee (2005). O conjunto original é composto por críticas de filmes e possui marcações de polaridades sobre as sentenças, ou seja, possuem uma única nota atribuída para cada sentença.

Os pesquisadores utilizaram o Stanford Parser para extrair as árvores sintáticas destas frases e desenvolveram uma interface de análise de polaridades manual. Essa interface serviu como ferramenta para que anotadores realizassem a marcação dos sintagmas do cópús. Esse processo foi possível por meio do Amazon Mechanical Turk², um *web-service* da Amazon onde é possível contratar serviços por meio de *crowdsourcing* (onde muitos usuários trabalham de maneira transparente ao requisitante do serviço) para realizar tarefas subjetivas que não poderiam ser confiadas a um computador.

Diversos usuários receberam sentenças aleatórias de trechos do cópús e puderam marcar em até 25 categorias diferentes, desde *Very Negative* até *Very Positive*. Após a marcação completa, os resultados foram normalizados para as cinco classes usadas no modelo.

3.3 Analisando Sentimentos em Português Brasileiro

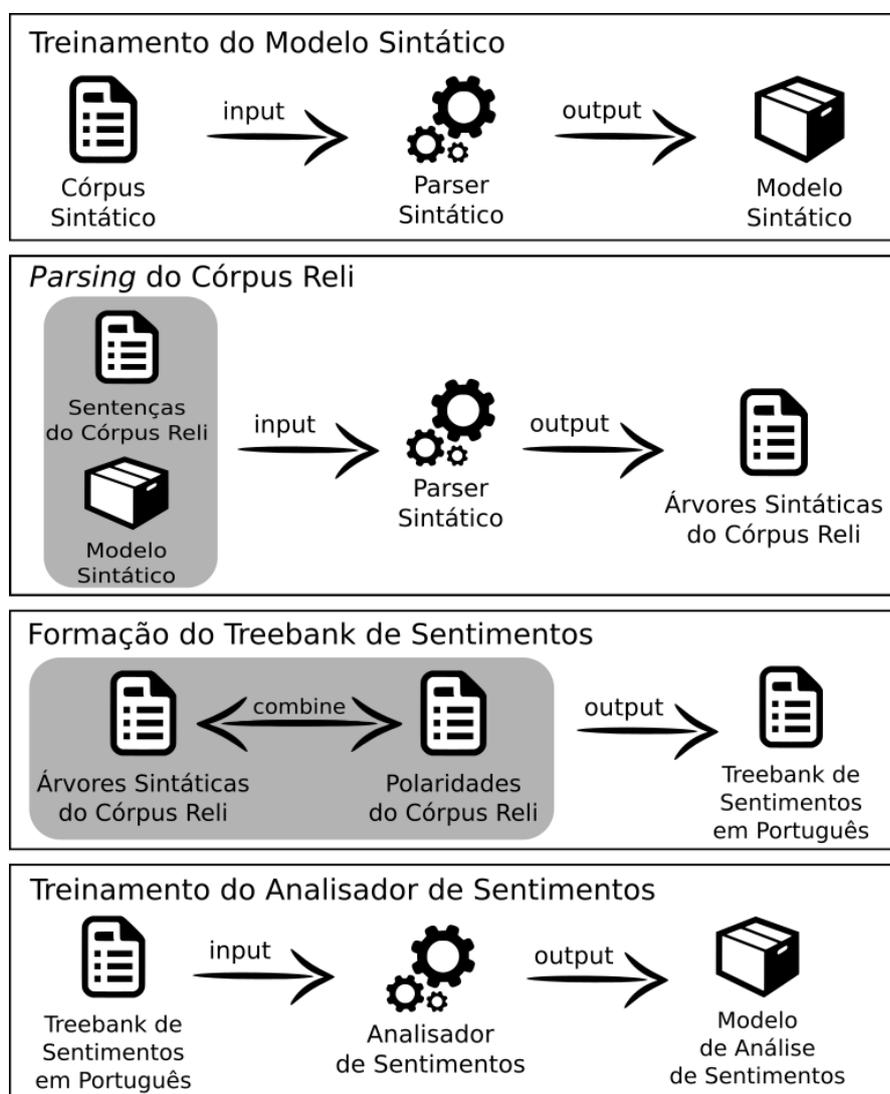
Nesta seção mostraremos como replicamos o modelo proposto por Socher et al. (2013a) e criamos um modelo de análise de sentimentos para Português brasileiro usando a RNTN.

Na Figura 8 podemos ver a sequência de procedimentos que formamos para o desenvolvimento do modelo.

¹ Disponível em <<http://rottentomatoes.com>>. Acessado em 21 de Junho de 2015.

² Acessível em <<https://www.mturk.com/mturk/welcome>>. Acessado em 21 de Junho de 2015.

Figura 8 – Representação gráfica dos passos do desenvolvimento de um modelo de análise de sentimentos para Português brasileiro.



3.3.1 Cópus de sentimentos

O primeiro passo para se replicar o modelo proposto é conseguir um cópus de sentimentos equivalente ou semelhante ao *Treebank* de sentimentos de Stanford, utilizado no trabalho original.

Buscamos na literatura por cópus em Português brasileiro que tivessem marcação de polaridade associada. Nossa busca nos levou a dois cópus - o Cópus Reli e o Cópus SentiLex.

O cópus Sentilex (SILVA; CARVALHO; SARMENTO, 2012) está em sua terceira versão e apresenta um conjunto de 84.562 palavras e flexões destas com anotações referentes a classe gramatical, flexão e polaridade associada.

O *cópus Reli* (FREITAS et al., 2012) apresenta um conjunto de 1.600 resenhas de livros anotados manualmente. O conjunto é composto por treze documentos (cada um relativo a um livro) contendo comentários retirados da rede social de livros e leitores Skoob³.

O conjunto completo do Reli possui 12.508 sentenças e 259.978 palavras. As resenhas são em relação à livros de sete autores variados, com comentários sobre livros populares ou clássicos, o que nos dará registros formais, com um vocabulário mais rebuscado, até registros mais informais contendo gírias e abreviações.

Por se tratar de um *cópus* de palavras e não sentenças, descartamos o *Cópus Sentilex* neste estágio do trabalho e trabalhamos com o *cópus Reli*, que já apresenta as sentenças que precisamos.

O *cópus Reli* é essencial para nossa pesquisa, porém possui dois desafios para o seu uso no modelo. O primeiro é que suas marcações são feitas sobre sentenças inteiras, ou seja, no caso de uma sentença ser positiva, teremos marcação positiva para todas as palavras da sentença. Isso prejudica no treino do modelo para sentenças que possuem mais de uma polaridade em suas sub-árvores, ou que possuem ironia ou sarcasmo em sua composição.

O segundo desafio é o seu formato. O *Cópus Reli* se encontra em formato próprio e para que sirva de treino na RNTN deve ser transformado para o formato PTB.

3.3.2 A análise sintática

Para atingir a representação PTB, foi necessário buscarmos uma ferramenta que nos permitisse extrair a árvore sintática de sentenças em Português brasileiro. A ferramenta que nos proporciona essa tarefa é chamada de *parser* sintático.

A busca por um *parser* em nosso idioma se mostrou insatisfatória, visto que outras ferramentas poderiam nos dar resultados melhores se pudéssemos ensiná-las a extraírem árvores sintáticas para o Português. Fizemos, portanto uma busca por *parsers* que nos permitissem treinar modelos sintáticos para outros idiomas.

Tal busca nos levou a considerar o *parser* da Universidade de Stanford⁴ (CHEN; MANNING, 2014) e o *parser* da Universidade de Berkeley⁵ (PETROV et al., 2006).

Ambos os *parsers* podem ser treinados a partir de um *cópus* sintático para que aprendam a dividir sentenças em Português, portanto, nosso próximo passo foi buscar um *cópus* sintático. Dessa maneira encontramos o Bosque e o Tycho Brahe.

³ <<http://www.skoob.com.br/>>. Acessado em 24 de Maio de 2015.

⁴ Disponível em <<http://nlp.stanford.edu/software/lex-parser.shtml>>

⁵ Disponível em <<https://code.google.com/p/berkeleyparser/>>

O Bosque conta com 9.368 frases retiradas de dois corpúscos distintos. Um deles chamado CETENFolha (SANTOS; SARMENTO, 2002), que conta com as 365 edições do jornal A Folha de S. Paulo do ano de 1994, e o CETEMPúblico (ROCHA; SANTOS, 2000), que conta com aproximadamente 180 milhões de palavras retiradas do jornal português Público.

O Tycho Brahe é um corpúscos anotado com 65 textos contendo 2.792.217 palavras de textos de autores nascidos entre os anos de 1380 e 1845.

Testes iniciais sugeriram que ambos os corpúscos seriam suficientes para nossa pesquisa, porém o Bosque apresentou problemas na marcação de sentenças. Suas marcações entraram em conflito com ambos os *parsers* o que inviabilizou seu uso no trabalho, portanto usamos o Tycho Brahe para o treinamento dos *parsers*.

Realizamos os treinamentos usando os dois *parsers* sobre o Corpúscos Tycho Brahe e geramos um arquivo serializado de formato *.ser.gz* usando o *Stanford Parser* e um arquivo serializado *.gr* gerado pelo *Berkeley Parser*.

Compreendemos que o processo de *parsing* pode ser medido e seus resultados comparados, porém nosso objetivo é o de analisar sentimentos de sentenças, acima de sua estrutura sintática, portanto mantivemos ambos os resultados e continuamos a pesquisa usando os dois modelos serializados. Deste momento em diante, sempre que falarmos sobre *parser* no singular, estaremos nos referindo a ambos os *parsers* - Stanford Parser e Berkeley Parser.

O processo de divisão do corpúscos Reli em dois arquivos, um contendo as sentenças em texto e outro contendo a polaridade de cada sentença, foi feito por meio de *scripts* produzidos na linguagem de programação *Python*⁶. A linguagem foi escolhida por comodidade, visto que já possuímos experiência com essa linguagem e por ela proporcionar maior praticidade no trabalho com *strings*.

Um problema enfrentado foi o uso dos caracteres parênteses nas sentenças do Reli. Em ambos os *parsers*, sua presença confundiu a árvore sintática final, pois como esta se encontra em formato PTB os parênteses são usados como demarcadores de hierarquia nas árvores. Por essa razão, antes de aplicar o *parser* nas sentenças, substituímos os parênteses por hífen.

Após dividirmos todas as sentenças, teremos como saída suas árvores sintáticas, contendo marcações sintáticas das palavras e conjunções. Na Figura 9 podemos ver um exemplo de sentença do Corpúscos Reli após aplicação de um *parser* sintático.

⁶ Os *scripts* descritos nesse trabalho estão disponíveis no endereço <<https://bitbucket.org/HBrum/sentipampa>>

Figura 9 – Sentença retirada do C rpus Reli analisada sintaticamente em formato PTB.

Meu livro preferido!	
Sentença original	
(FRAG (@FRAG (NP (PRO\$ Meu) (N livro)) (ADJP (VB preferido))) (. !))	<pre> graph TD FRAG --> @FRAG FRAG --> P1["."] @FRAG --> NP NP --> PRO\$ NP --> N PRO\$ --> Meu N --> livro P1 --> P2["!"] </pre>
�rvore sint�tica em formato Penn Treebank	�rvore sint�tica da sentena

3.3.3 A formao de um c rpus de sentimento em Portugu s brasileiro

Para a formao do c rpus final de sentimento em Portugu s brasileiro no formato PTB,   necess rio trocar as marcaes sint ticas das  rvores geradas na etapa anterior por polaridades.

Utilizamos *scripts* para agregar os valores das polaridades nas  rvores das sentenas, formando assim um c rpus de sentimentos em formato PTB.

Durante o processo de atribuio de polaridades, observamos comportamentos diferentes nas extraes resultantes dos dois *parsers*. Enquanto o *Berkeley Parser* possui em sua composio uma opo de binarizao, no conseguimos localizar algo equivalente no *Stanford parser*. Essa binarizao faz com que cada n  da  rvore gerada possua dois filhos.

Como vimos na seo 3.1, a RNTN analisa sempre em pares de palavras e sintagmas, portanto precisamos binarizar as  rvores de sentimentos do c rpus antes de treinarmos o modelo.

Para as  rvores geradas pelo *Stanford Parser*, aplicamos um algoritmo de binarizao (transformao na forma normal de *Chomsky*   esquerda) utilizando a biblioteca NLTK(BIRD; KLEIN; LOPER, 2009) antes de atribu rmos as polaridades aos n s.

O modelo original utiliza cinco classes de marcao de polaridades - *Very Positive*, *Positive*, *Neutral*, *Negative* e *Very negative*. Reduzimos nossas classes para somente tr s, pois o c rpus Reli no possui marcaes para “Muito Positivo” ou “Muito Negativo”.

Ap s a formao do c rpus de sentimento em formato PTB, podemos treinar um

modelo de análise de sentimentos. Dessa maneira atingimos nosso objetivo de replicar o modelo de [Socher et al. \(2013a\)](#). No próximo capítulo mostraremos os experimentos realizados com o modelo criado.

4 Experimentos e Análises

Neste capítulo apresentaremos os testes realizados sobre o modelo criado para analisar sentenças em Português brasileiro. Mostraremos as técnicas e medidas utilizadas nos experimentos e discutiremos gráficos que demonstram como o aprendizado funcionou na [RNTN](#).

Traçamos alguns paralelos dos resultados obtidos neste trabalho em relação ao trabalho original e mostramos algumas propostas e modificações realizadas no modelo para prover ganho para sentenças em Português brasileiro.

4.1 Metodologia e Testes

Para nossos experimentos, utilizamos o *córpus* de sentimentos produzido com base no *Córpus Reli* ([FREITAS et al., 2012](#)) e nos dois *parsers* descritos no [Capítulo 3](#) - o *Stanford Parser* ([CHEN; MANNING, 2014](#)) e o *Berkeley Parser* ([PETROV et al., 2006](#)). Para facilitar a leitura e o desenvolvimento das ideias, trataremos o primeiro de *córpus Stanford.Reli* e o segundo de *Berkeley.Reli*.

Desenvolvemos uma série de passos que nos permitiram criar um modelo analisador de sentimentos para Português brasileiro. Usaremos duas medidas de acurácia sobre os dados obtidos nos testes nos modelos criados - a acurácia simples e a acurácia combinada.

A acurácia é calculada a partir do total de sentenças corretamente marcadas pelo analisador sobre o número total de sentenças do conjunto de testes. Essa medição nos mostrará como o modelo acerta a análise de sentenças quanto aos seus sentimentos - positivo, negativo ou neutro.

$$\text{Acurácia} = \frac{\text{Acertos}}{\text{Total}}$$

A acurácia combinada é calculada somente sobre as marcações positivas e negativas, ou seja, não leva em consideração as marcações de neutralidade nas sentenças. Essa medição nos mostra como o algoritmo consegue adivinhar sentimentos relevantes para serem extraídos, levando em consideração que encontrar neutralidade em uma sentença não é uma informação tão relevante quanto saber se ela é positiva ou negativa.

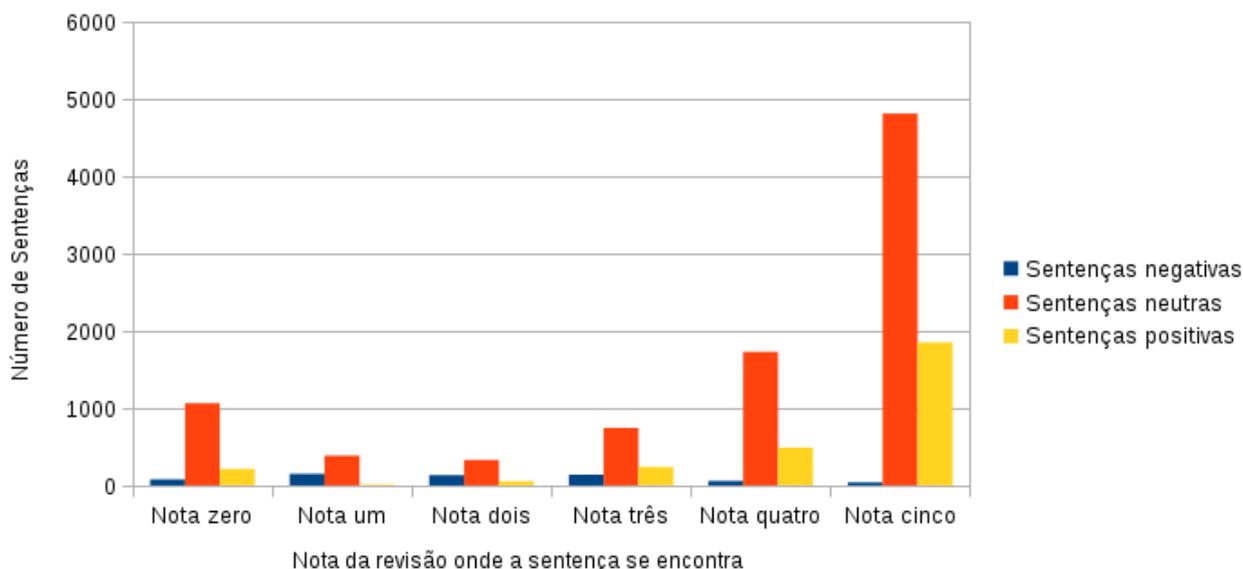
$$\text{Acurácia Combinada} = \frac{\text{Acertos Positivos} + \text{Acertos Negativos}}{\text{Total de Sentenças Positivas} + \text{Total de Sentenças Negativas}}$$

Quanto ao conjunto de dados aos quais essas medidas serão aplicadas, trazemos duas abordagens de observação - sobre somente o topo da árvore de sentimentos, contendo o valor atribuído à sentença como um todo, e sobre todos os nós da árvore de sentimentos, representando todos os sintagmas presentes nos documentos.

Dessa maneira esperamos testar a capacidade do algoritmo calcular composições complexas nas sentenças testadas, como árvores de sentenças positivas que possuem sub-árvores mistas.

O cópús Reli é composto por resenhas de livros e conta com a marcação das polaridades das sentenças e da nota total de cada resenha. Cada resenha possui uma nota discreta que varia de zero a cinco. Em cada resenha, sentenças podem ser marcadas com polaridades diferentes.

Figura 10 – Número de sentenças com cada polaridade e nota da resenha onde esta se encontra.



Na figura [Figura 10](#) vemos o gráfico de sentenças marcadas nas três polaridades agrupadas pela nota da resenha completa onde se encontram. O que podemos notar disso é uma ocorrência muito grande de sentenças neutras em resenhas que obtiveram notas altas e baixas, o que pode nos fazer questionar se a marcação do cópús retrata fielmente o sentimento correto de cada sentença.

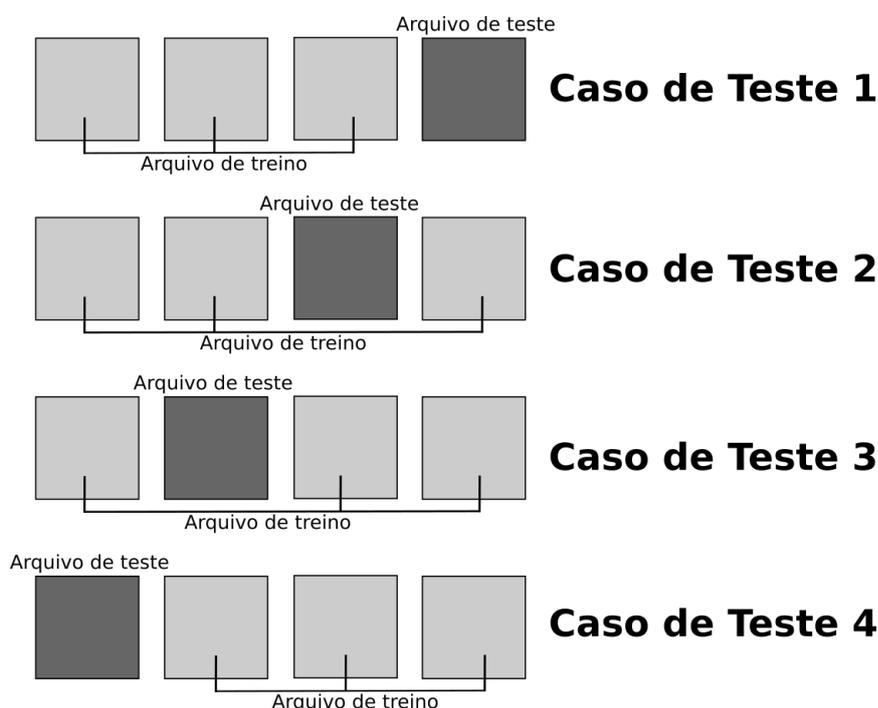
Como podemos observar, a distribuição da polaridade de sentenças não é a ideal para nossos testes. Do total de 12.512 sentenças, temos 2.854 sentenças positivas, 593 sentenças negativas e 9.065 sentenças neutras.

Como podemos ver, temos um número muito alto de sentenças neutras, estas

representam 72% do nosso córpus. Isso poderá fazer com que o algoritmo tenha problemas para analisar sentenças de diferentes polaridades, pois a grande ocorrência de sentenças de uma mesma classe faz com que ele aprenda a escolher essa classe com mais ocorrência. Isso justifica o uso da acurácia combinada nas nossas medições, pois ela descarta as sentenças neutras em sua análise.

O método escolhido para testar o modelo de sentimentos é chamado *Cross-Validation*, ou Validação Cruzada.

Figura 11 – Representação de *Cross Validation* onde $k = 4$.



Cross-Validation é o processo de criar pares distribuídos de arquivos de “treino” e “teste” partindo de um único conjunto de dados (SAMMUT; WEBB, 2011). Para isso estabelecemos um número k de casos de testes e dividimos nosso conjunto de dados para testá-los separadamente, como podemos observar na Figura 11. A média dos valores de todos os casos de teste será aceita como o caso médio do algoritmo. Nosso procedimento de teste será chamado 10-Fold Cross-Validation, pois estabeleceremos como 10 o k de nossa validação cruzada.

Criamos uma rotina de testes que dividiu nosso córpus em 10 arquivos de treino e 10 arquivos de teste. Cada arquivo de treino contém 90% das sentenças, enquanto nossos arquivos de teste contém os 10% restantes. Não utilizaremos arquivos de validação para o treino, pois não buscamos uma solução final para nosso problema, somente o teste do modelo.

Nossos córpus contam com 12.508 sentenças, sendo que cada arquivo de treino terá

em média 11.260 sentenças e cada arquivo de teste cerca de 1.250.

O treinamento dos modelos durou cerca de 60 horas para 20 casos de teste (10 para o Reli.Berkeley e 10 para o Reli.Stanford). Todas as execuções de casos de testes foram repetidas para confirmar a integridade dos resultados obtidos.

Tabela 1 – Resultado final obtido com a média dos dez casos de teste criados.

	Acurácia Simples		Acurácia Combinada	
	Todos os Nós	Topo da Árvore	Todos os nós	Topo da Árvore
Reli.Berkeley	70,19%	69,08%	17,82%	51,18%
Reli.Stanford	69,54%	67,47%	17,51%	51,36%

Na [Tabela 1](#) vemos os resultados obtidos na 10-Fold Cross-Validation. No trabalho original foi atingido o valor de 80,7% de acurácia simples e 87,6% de acurácia combinada em polaridades de todos os nós da árvore. Nossos resultados não atingiram esses valores e isso se deve a qualidade do córpus utilizado no trabalho original. Como descrito previamente, uma marcação manual foi feita no *Treebank* de Sentimentos de Stanford, tal marcação não pôde ser replicada em nossos experimentos, o que fez com que nosso córpus não se comportasse tão bem na análise das conjunções intermediárias das árvores.

Essas variações se apresentam em sentenças onde uma pequena conjunção pode alterar seu sentido completamente, como por exemplo na sentença “*O filme não é bom, apesar de ser inteligente e divertido.*” onde teríamos uma sentença totalmente positiva, se não fosse a presença de “*não é bom*”, no início da frase. Para um humano, é simples dizer que esse trecho negativo é o determinante da sentença, porém para o algoritmo, são necessárias várias sentenças sendo analisadas separadamente com uma marcação manual de seus nós intermediários.

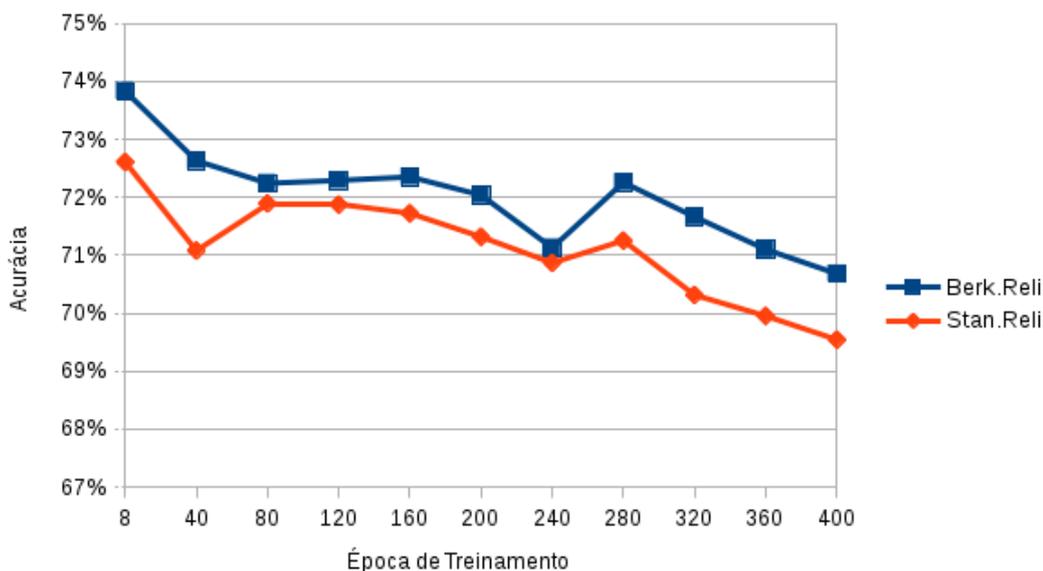
Para a marcação somente do topo da árvore de sentimentos, o trabalho original apresentou 45,7% na acurácia e 85,4% na acurácia combinada para sentenças em inglês. Nossos resultados superaram a marcação no topo da árvore na acurácia. Isso se deve ao fato de nosso córpus de treino e teste não possuir tanta variedade nas polaridades. Enquanto no trabalho original existia um córpus mais robusto e com uma maior taxa de sentenças positivas e negativas, nosso conjunto de dados possuía mais sentenças neutras, o que aumenta a chance do algoritmo acertar em marcações neutras.

Nem sempre o acerto sobre marcações neutras é ruim, mas é claramente mais vantajoso possuir um modelo treinado para encontrar sentenças positivas ou negativas do que um que reconheça sentenças neutras a maioria das vezes. Novamente atentamos para a qualidade e cobertura superiores do córpus de sentimentos de Stanford em relação aos córpus gerados nesse trabalho.

4.1.1 O aprendizado na rede neural

Os testes em RNs são realizados em “Épocas”. Uma época representa uma passagem da rede neural sobre todas as sentenças do conjunto de dados uma vez. Para o treinamento dos modelos usamos 400 épocas, valor padrão do algoritmo. Os gráficos apresentados a seguir representam a acurácia e acurácia combinada do modelo durante o processo de treinamento.

Figura 12 – Acurácia sobre todos nós da árvore de polaridades durante treinamento do modelo.

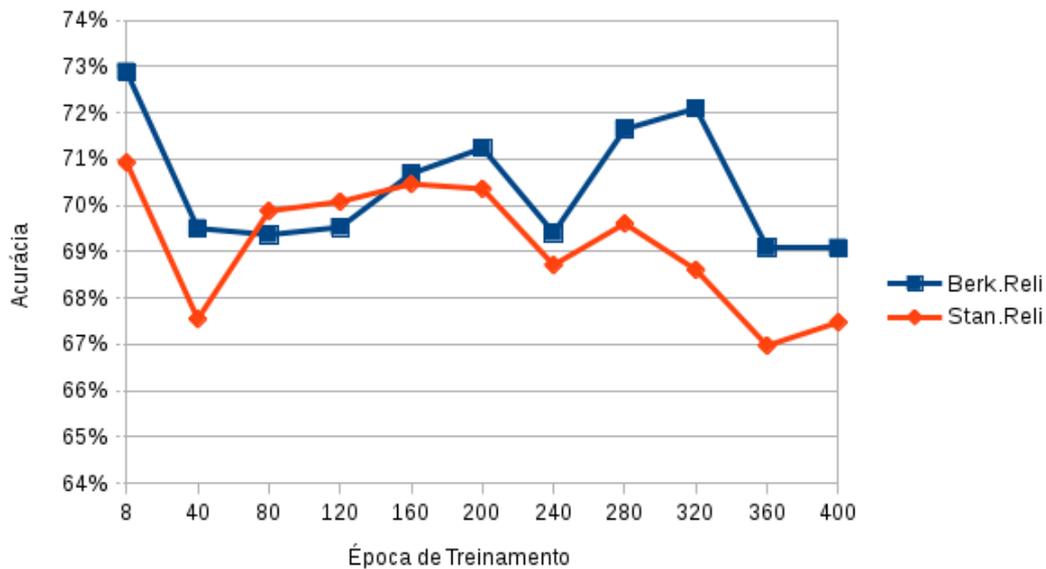


Na [Figura 12](#) vemos a acurácia que o modelo atinge em diferentes momentos do treinamento. Essa medida foi calculada sobre todos os nós da árvore de sentimentos e podemos notar uma queda quase constante no aprendizado. Uma das razões possíveis para essa queda é o número elevado de sentenças neutras, o que não é ideal, visto que um número mais equilibrado de sentenças nos daria um resultado mais significativo.

O comportamento nos dois corpúscos também pode ser observado, visto que ambos se comportam de maneira semelhante, apesar de possuírem variações nos valores atingidos. Obtivemos um resultado melhor no corpúsculo extraído sintaticamente com o parser de Berkeley, mas quanto à acurácia na análise de sentimentos notamos que o comportamento das duas linhas é semelhante. Isso mostra que a divisão sintática do corpúsculo pode influenciar na eficiência da análise de sentimentos, mas não altera o comportamento do aprendizado da RNTN.

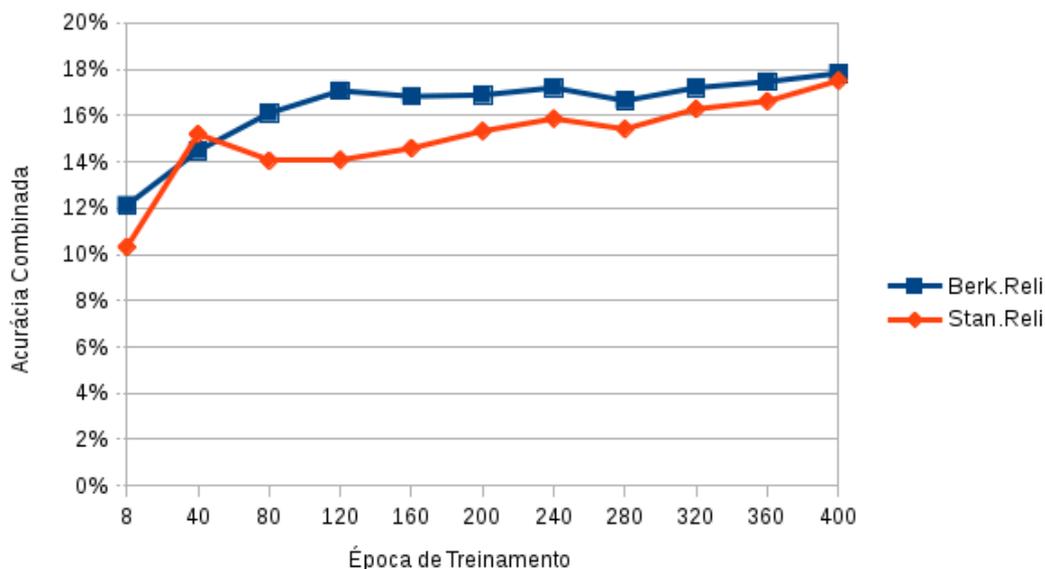
Na [Figura 13](#) vemos a acurácia sobre somente o topo da árvore, ou seja, somente na marcação de sentenças inteiras. Essa medida é diretamente ligada à capacidade do algoritmo de analisar sentenças, ainda que cometa erros analisando suas sub-árvores. O

Figura 13 – Acurácia sobre o topo das árvores de polaridades durante treinamento do modelo.



aprendizado não diminui tanto a acurácia do modelo, visto que este diminui em cerca de somente 4% ao longo do aprendizado. Novamente temos um comportamento muito semelhante nos testes de ambos os corpúss utilizados.

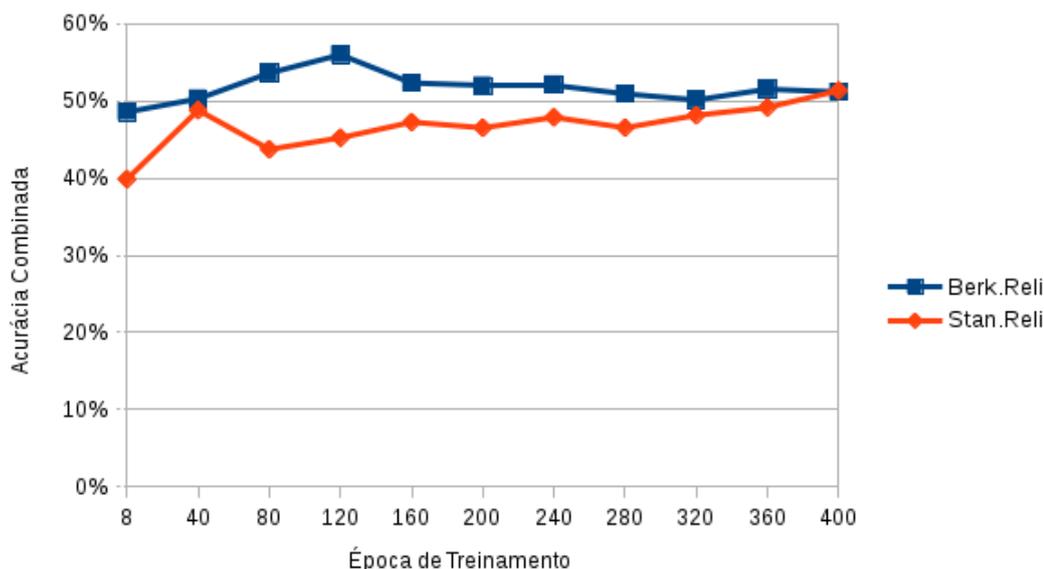
Figura 14 – Acurácia Combinada sobre todos nós da árvore de polaridades durante treinamento do modelo.



Na Figura 14 vemos a acurácia combinada sobre todos os nós da árvore de sentimentos. Essa acurácia combinada leva em consideração somente as sentenças positivas e

negativas no *córpus* de treino. Podemos observar que ao longo do treinamento, as quedas ocorridas nas análises anteriores mostram que o modelo começa a avaliar melhor as sentenças, fazendo com que diminua a ocorrência de marcações neutras e aumente as marcações de ocorrências positivas e negativas, o que faz com que essa categoria seja beneficiada na análise, mesmo que tenha atingido valores tão baixos.

Figura 15 – Acurácia Combinada sobre o topo das árvores de polaridades durante treinamento do modelo.



Na [Figura 15](#) vemos o comportamento do treinamento na acurácia combinada somente sobre os topos das árvores de sentimentos, ou seja, sobre a avaliação sobre a sentença inteira, desprezando nós intermediários. O valor é bem expressivo e mostra uma convergência desde cedo dos valores calculados.

O comportamento dos dois *córpus* no treinamento foi semelhante e ambos atingiram resultados praticamente iguais ao término da execução, porém esses valores ainda estão muito próximos para afirmarmos que um *córpus* é melhor ou pior que o outro. Em geral os valores do *córpus* Reli.Berkeley foram superiores desde o início do treinamento, talvez o estudo dos *parsers* e de suas composições nos trouxesse maiores resultados quanto à análise.

Para podermos entender melhor o comportamento do aprendizado, extraímos a matriz-confusão de um dos testes ao término da oitava época de treinamento e ao término do treinamento completo para comparar os acertos.

A matriz-confusão é usada para se avaliar o acerto e o erro sobre um conjunto de predições. Nas colunas temos as polaridades extraídas (valores numéricos de 1 a 3), enquanto as linhas representam as classificações corretas contidas no *córpus*. Na

matriz podemos ver quantas marcações foram corretas e quantos erros o modelo cometeu. Mostramos as matrizes do início e do fim para que se possa observar quais as mudanças de uma época remota do treinamento para o fim deste.

Figura 16 – Matriz-confusão da predição de todos os nós-raiz de um caso de teste do treinamento.

Extraído na época 8				
Predição/Classificação Correta	1	2	3	Total
1	6	16	5	27
2	28	558	55	641
3	57	291	234	582
Total	91	865	294	

Extraído na época 400				
Predição/Classificação Correta	1	2	3	Total
1	23	34	12	69
2	48	659	120	827
3	20	172	162	354
Total	91	865	294	

Na [Figura 16](#) vemos as matrizes-confusão do acerto dos nós do topo das árvores de sentimentos. Percebemos que a marcação se foca mais nas sentenças neutras, por causa da alta incidência dessas sentenças no nosso cópulo. Na primeira análise o algoritmo marca 641 sentenças negativamente, ou seja, 51% das marcações sobre as 1.250 sentenças foi neutra. Ao término do treinamento, esse valor sobe pra 66%, com 827 marcações para sentença neutras.

A análise binária (levando em consideração somente sentenças positivas e negativas) também pode ser observada. Após o treinamento existe uma distribuição maior nas predições de sentenças positivas e negativas. A marcação negativa sobe de 4% na época 8 para 11% das predições do algoritmo ao término do treinamento.

Na [Figura 17](#) temos as matrizes-confusão sobre todos os nós das árvores de sentimentos de um caso de teste.

Da mesma maneira que na análise somente sobre o topo da árvore, a distribuição das marcações para sentenças negativas é muito baixa em relação às demais polaridades. Percebemos também que ao longo do treinamento temos uma distribuição maior das predições entre positividade e negatividade, porém uma queda na predição de neutralidade.

O aumento na predição de sentenças negativas é de 8%, nas positivas é de 1%, enquanto temos uma queda de 1% em sentenças neutras. O número de queda é baixo, mesmo assim são classificadas 22% de sentenças neutras a mais de sentenças do que existem no arquivo de testes. Esse excesso de marcações neutras pode justificar o gráfico apresentados na [Figura 12](#).

Figura 17 – Matriz-confusão da predição de todos os nós de um caso de teste do treinamento.

Extraído na época 8				
Predição/Classificação Correta	1	2	3	Total
1	128	450	203	781
2	2442	29692	8783	40917
3	573	3095	2364	6033
Total	3143	33237	11350	

Extraído na época 400				
Predição/Classificação Correta	1	2	3	Total
1	233	441	174	848
2	2399	29568	8791	40758
3	511	3228	2385	6124
Total	3143	33237	11350	

4.2 Melhorando o modelo para Português brasileiro

Após testar o modelo e realizar as medidas, nosso objetivo final foi o de buscar alternativas e modificações no treinamento para atingir resultados melhores para o nosso idioma.

Os experimentos realizados na seção anterior foram realizados utilizando os parâmetros padrões do algoritmo. Nesta sessão mostraremos algumas modificações e testes sobre modelos gerados com parâmetros diferentes dos utilizados no trabalho original.

Em vez de usarmos a *10-Fold Cross-Validation* para esse caso, medimos os 10 casos de teste da *Cross-Validation* e fizemos a média dos resultados nas quatro medidas utilizadas. Como todos os casos se comportaram de maneira similar, escolhemos o caso que mais se aproximou da média para realizar os testes de melhoria.

Dividimos nossos casos de teste em dois tipos de modificação - modificações estruturais e modificações de aprendizagem.

4.2.1 Modificações estruturais

Modificações estruturais são parâmetros onde a estrutura do modelo é alterado. Seja modificando padrões que irão gerar vetores de palavras diferentes ou removendo etapas da [RNTN](#).

O primeiro experimento que realizamos foi usando palavras normalizadas nas sentenças. Sabemos que as palavras são recebidas pela [RN](#) como um vetor numérico, então palavras que possuem características diferentes terão vetores diferentes e serão analisadas de maneira diferente. Normalizando as palavras, ou seja, trocando letras maiúsculas para minúsculas, fazemos com que palavras que antes seriam vistas como diferentes, agora

sejam vistas como iguais.

O segundo experimento foi a retirada da camada de tensor da [RNTN](#). Retirando o tensor, teremos uma Rede Neural Recursiva ([RNN](#)) analisando as sentenças. Segundo os autores, o tensor ajuda nos resultados por que suas camadas acentuam as composicionalidades na [RNTN](#). Não utilizando o tensor poderemos ver seu impacto na [RNN](#).

Os resultados obtidos nos experimentos com modificação estrutural usando o [córpus Berkeley.Reli](#) estão descritos na [Tabela 2](#). Os resultados usando o [córpus Stanford.Reli](#) estão apresentados na [Tabela 3](#). Na [Figura 18](#) e na [Figura 19](#) temos os gráficos das tabelas apresentadas, para facilitar a comparação e visualização destes dados.

Tabela 2 – Experimentos com modificações estruturais na [RNTN](#) no [córpus Berkeley.Reli](#).

	Acurácia		Acurácia Combinada	
	Todos os nós	Topo da árvore	Todos os nós	Topo da árvore
Palavras normalizadas	69.84%	69.54%	18.57%	62.17%
Rede neural sem Tensor	49.04%	61.48%	19.74%	53.67%
Modelo Normal	70.19%	69.08%	17.82%	51.18%

Tabela 3 – Experimentos com modificações estruturais na [RNTN](#) no [córpus Stanford.Reli](#).

	Acurácia		Acurácia Combinada	
	Todos os nós	Topo da árvore	Todos os nós	Topo da árvore
Palavras normalizadas	68.98%	68.75%	16.53%	53.80%
Rede neural sem Tensor	52.92%	60.92%	20.31%	51.46%
Modelo Normal	69.54%	67.47%	17.51%	51.36%

Podemos ver que a normalização das palavras mostrou certo resultado no modelo, os experimentos mostraram valores semelhantes ou superiores aos obtidos nos experimentos anteriores. Isso pode ser a confirmação de que as características do nosso idioma realmente se beneficiam de uma normalização para vetores de palavras. Além disso podemos observar que nas medições de acurácia combinada no topo das árvores de sentimentos no [córpus Berkeley.Reli](#) temos um ganho de mais de 10% sobre o modelo usando vetores de palavras normais.

Analisando a ausência do tensor no modelo, podemos constatar que ele é importante para a análise composicional. Os resultados de [RNTNs](#) executando sem o tensor foram consideravelmente inferiores aos obtidos no modelo original quanto a acurácia. Na acurácia combinada notamos um certo ganho de acurácia, não tão alto quanto o obtido com a normalização das palavras.

4.2.2 Modificações de aprendizagem

O algoritmo nos permite a troca da taxa de aprendizado da rede neural. Essa taxa tem o valor padrão de 0,01 e é a base do cálculo do *backtracking* na alteração dos pesos

Figura 18 – Gráfico de medidas dos experimentos estruturais na RNTN usando o córpus Berkeley.Reli.

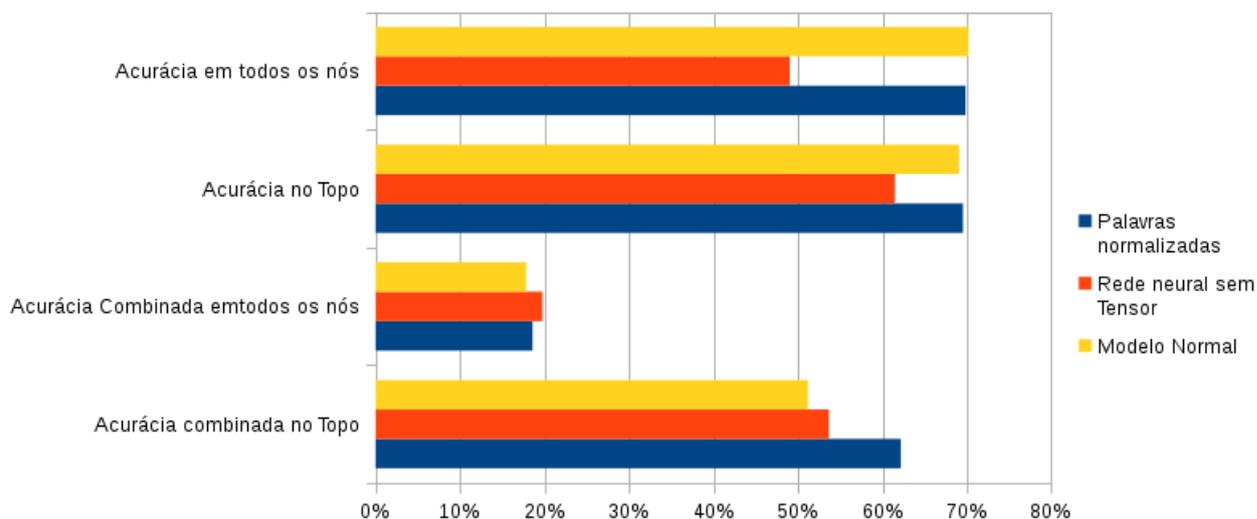
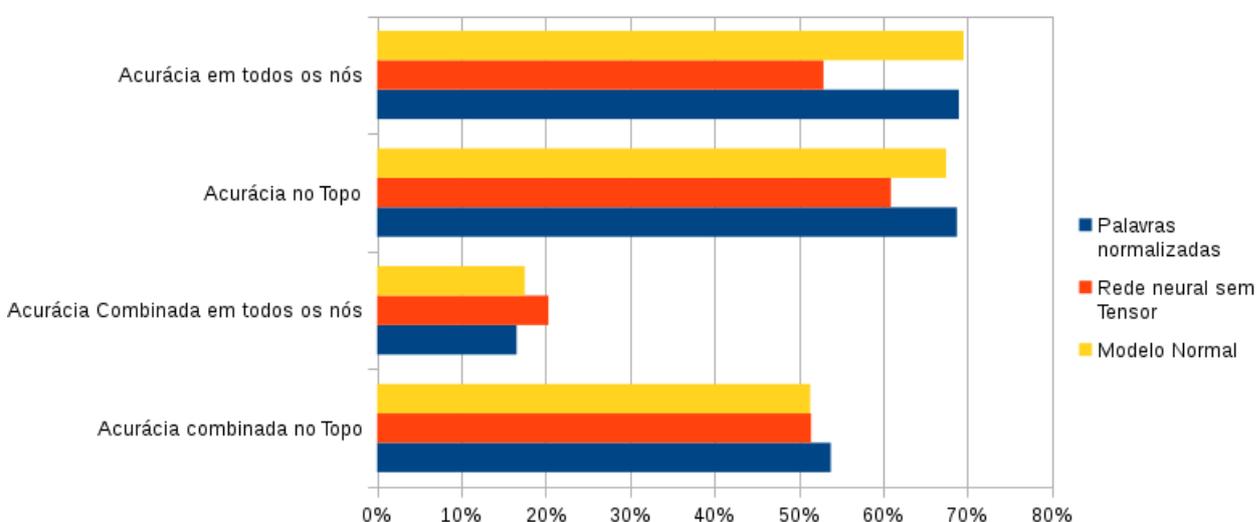


Figura 19 – Gráfico de medidas dos experimentos estruturais na RNTN usando o córpus Stanford.Reli.



da rede. A cada iteração do algoritmo, o mesmo irá calcular o erro sobre as marcações, esse erro afetará cada uma das matrizes de peso da rede e a taxa de aprendizado define quanto desses valores será aumentado ou diminuído no aprendizado.

Exploramos duas abordagens no experimento - utilizar uma taxa de aprendizagem menor e usar uma taxa de aprendizagem. Empiricamente, definimos quatro valores para testes, usaremos 0,001 e 0,003 para ver como a rede neural se comporta usando valores

abaixo do padrão e 0,3 e 0,1 para testar valores acima do padrão do algoritmo. Os dados para os parsers Berkeley.Reli e Stanford.Reli estão presentes na [Tabela 4](#) e [Tabela 5](#), respectivamente.

Tabela 4 – Experimentos variando taxa de aprendizagem na [RNTN](#) no cópús Berkeley.Reli.

	Acurácia		Acurácia Combinada	
	Todos os nós	Topo da árvore	Todos os nós	Topo da árvore
Aprendizado em 0,03	71.68%	70.61%	14.27%	39.59%
Aprendizado em 0,1	74.08%	73.95%	7.80%	18.77%
Aprendizado em 0,003	61.12%	64.13%	20.13%	51.32%
Aprendizado em 0,001	59.68%	65.18%	20.77%	55.43%
Modelo Normal (0.01)	70.19%	69.08%	17.82%	51.18%

Tabela 5 – Experimentos variando taxa de aprendizagem na [RNTN](#) no cópús Stanford.Reli.

	Acurácia		Acurácia Combinada	
	Todos os nós	Topo da árvore	Todos os nós	Topo da árvore
Aprendizado em 0.03	67.15%	70.01%	13.09%	32.75%
Aprendizado em 0.1	73.22%	73.96%	5.97%	13.74%
Aprendizado em 0.003	51.00%	63.14%	19.57%	54.09%
Aprendizado em 0.001	53.16%	63.64%	17.25%	44.44%
Modelo Normal (0.01)	69.54%	67.47%	17.51%	51.36%

Vemos que o modelo normal, treinado com a taxa de aprendizagem 0,01 se mantém no meio termo entre as acurácias observadas. Nas análises de acurácia, ele se mantém pouco abaixo dos aprendizados maiores, enquanto nas análises de acurácia combinada, se mantém pouco abaixo dos testes realizados com taxas de aprendizagem menores.

Podemos observar os experimentos com taxa de aprendizagem 0,001 e 0,003 quanto à acurácia combinada. Ambas superam o modelo original, enquanto os experimentos com taxa de aprendizagem 0,03 e 0,1 caem muito.

5 Conclusões

O treinamento de um modelo para analisar sentimentos em Português brasileiro foi alcançado. A proposta inicial foi atingida com sucesso, assim como a geração de um *treebank* de sentimentos para nosso idioma.

As tentativas de melhorar o modelo ou encontrar parâmetros que trouxessem melhorias para a análise específica do idioma Português brasileiro, porém, não foram tão positivas.

A questão de trabalhar com um idioma não tão disseminado mundialmente quanto o Português brasileiro também trouxe desafios durante o período de trabalho. Foi possível encontrar inúmeras bibliografias e corpús em inglês enquanto para Português pouca informação era descoberta. Mesmo quando encontrávamos corpús e materiais na área de [PLN](#) para Português, nos deparávamos com formatos não flexíveis e inconsistências nos dados, como foi o caso do Corpús Bosque discutido no [Capítulo 3](#).

O corpús encontrado possui uma distribuição baixa de polaridades de sentenças, onde 72% das sentenças são neutras. Essa distribuição não ideal faz com que os testes não mostrem resultados tão expressivos para o campo da [PLN](#) e cria dúvida quanto à efetividade do modelo. O uso de técnicas como a *Cross-Validation* foi essencial para nossos experimentos, pois traz mais credibilidade para os resultados atingidos.

Usando casos de teste mais equilibrados, com valores de sentenças positivas, negativas e neutras mais distribuídos poderia possibilitar uma exploração maior das composicionalidades do modelo de [RNTN](#).

A ausência de uma marcação nos nós intermediários das árvores fez com que as análises não pudessem explorar diretamente a composicionalidade, resultando em baixos valores para as acurácias envolvendo número total de nós das sentenças.

Nossos resultados finais ficaram dependentes do funcionamento dos *parsers* utilizados no início do processo. Isso pode ter impactado muito na análise de sentimentos, porém demandaria um trabalho excessivo nos testes para garantir árvores sintáticas eficientes, o que fugiria completamente do escopo do trabalho e traria atrasos para o desenvolvimento da pesquisa.

Apesar de não ultrapassar o estado-da-arte na análise de sentimentos, aceitamos nossos resultados obtidos e acreditamos que conseguimos cumprir com tudo o que havíamos nos proposto. Conseguimos trazer um fluxograma de geração de modelo de análise de sentimentos para [RNTN](#) e para a produção de um *treebank* de sentimentos em Português, portanto esperamos que esse trabalho sirva como ponto de partida para futuras pesquisas

na área de [PLN](#) e de análise de sentimentos.

5.1 Contribuições e Possibilidades Futuras

As contribuições desse trabalho para o meio acadêmico são os resultados apresentados na [Capítulo 4](#). A partir desses valores é possível fazer comparações com outros modelos como *Naive-Bayes* ou *Support Vector Machines*.

Uma melhoria nesse próprio modelo também pode ser proposta a partir desse trabalho, analisando quanto aos testes já executados e preparando novos valores de aprendizado, ou variando os testes feitos. O trabalho apresentado abre possibilidades para o desbravamento do modelo [RNTN](#) em sua totalidade.

Observamos que o *córpus* desenvolvido poderia ter sido mais elaborado, algo como marcações manuais semelhantes às executadas por [Socher et al. \(2013a\)](#). O desenvolvimento de uma interface de usuário e uma marcação manual para os nós intermediários das árvores seria muito útil para melhorar o *córpus* gerado nesse trabalho.

O modelo de criação de *treebank* de sentimentos pode ser utilizado sobre outros conjuntos de dados. O *córpus* Reli pode ser alimentado e novos *córpus* podem surgir, o que abriria espaço para que os passos apresentados aqui sejam parte de um *treebank* ainda maior para a língua portuguesa. Essa seria a maior contribuição, visto que atingiria todo o campo da [PLN](#).

O modelo de [RNTN](#) possui uma capacidade composicional imensa que, para ser aproveitada em sua totalidade, se torna muito dependente de seu conjunto de dados para treino. Um desenvolvimento mais em conjuntos de dados e marcações para nosso idioma ajudaria não só pesquisas como essa como uma série de outras ideias que pesquisadores têm todos os dias e não conseguem executar por falta de uma base de dados sólida e robusta que consiga apresentar casos de teste e abrangência de diversos aspectos da nossa linguagem.

Esperamos que esse trabalho tenha sido objetivo e informativo, assim como motivacional para todos os estudantes, pesquisadores e professores nas áreas de Inteligência Artificial, Aprendizado de Máquina ou Processamento de Linguagem Natural.

Referências

- AGUIAR, M. O. *Sentiment analysis em relatórios da administração divulgados por firmas brasileiras*. Dissertação (Mestrado) — Fundação Instituto Capixaba de Pesquisas em Contabilidade, Economia e Finanças, 2012. Citado na página 23.
- ALM, C. O.; ROTH, D.; SPROAT, R. Emotions from text: machine learning for text-based emotion prediction. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*. [S.l.], 2005. p. 579–586. Citado na página 29.
- ARAÚJO, M.; GONÇALVES, P.; BENEVENUTO, F. Measuring sentiments in online social networks. In: ACM. *Proceedings of the 19th Brazilian symposium on Multimedia and the web*. [S.l.], 2013. p. 97–104. Citado na página 29.
- ARMELINI, D. A.; RODRIGUES, E. C. Análise de sentimento dos dados do twitter através de uma abordagem espaço-temporal. *Matemática e Estatística em Foco*, v. 1, n. 2, 2013. Citado na página 29.
- BIRD, S.; KLEIN, E.; LOPER, E. *Natural Language Processing with Python*. [S.l.]: O'Reilly Media, 2009. Citado na página 42.
- BRAGA, A. de P.; FERREIRA, A. C. P. de L.; LUDERMIR, T. B. *Redes neurais artificiais: teoria e aplicações, 2ª Edição*. [S.l.]: LTC Editora, 2007. Citado na página 27.
- CHEN, D.; MANNING, C. D. A fast and accurate dependency parser using neural networks. In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [S.l.: s.n.], 2014. p. 740–750. Citado 2 vezes nas páginas 40 e 45.
- ESULI, A.; SEBASTIANI, F. Sentiwordnet: A publicly available lexical resource for opinion mining. In: *Proceedings of LREC*. [S.l.: s.n.], 2006. v. 6, p. 417–422. Citado na página 29.
- FERREIRA, E. d. B. A. Análise de sentimento em redes sociais utilizando influência das palavras. *Trabalho de Graduação-Universidade Federal de Pernambuco-UFPE. Departamento de Ciências da Computação*, 2010. Citado na página 30.
- FREITAS, C. et al. Vampiro que brilha... rá! desafios na anotação de opinião em um corpus de resenhas de livros. *ENCONTRO DE LINGÜÍSTICA DE CORPUS*, v. 11, 2012. Citado 2 vezes nas páginas 40 e 45.
- GO, A.; HUANG, L.; BHAYANI, R. Twitter sentiment analysis. *Entropy*, v. 17, 2009. Citado na página 30.
- GODBOLE, N.; SRINIVASIAH, M.; SKIENA, S. Large-scale sentiment analysis for news and blogs. *ICWSM*, v. 7, 2007. Citado na página 29.
- HAYKIN, S. S. *Redes neurais*. [S.l.]: Bookman, 2001. Citado na página 28.

- HECHT-NIELSEN, R. Theory of the backpropagation neural network. In: IEEE. *Neural Networks, 1989. IJCNN., International Joint Conference on*. [S.l.], 1989. p. 593–605. Citado na página 37.
- LIU, B. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, Morgan & Claypool Publishers, v. 5, n. 1, p. 1–167, 2012. Citado na página 25.
- MARSLAND, S. *Machine learning: an algorithmic perspective*. [S.l.]: CRC Press, 2011. Citado na página 27.
- MARTINAZZO, B. *Um Método de Identificação de Emoções em Textos Curtos para o Português do Brasil*. Tese (Doutorado) — Pontifícia Universidade Católica do Paraná, 2010. Citado 2 vezes nas páginas 26 e 29.
- NETO, L. G. P.; NICOLETTI, M. do C. *Introdução às redes neurais construtivas*. [S.l.]: EDUFSCAR, 2005. Citado na página 27.
- PAK, A.; PAROUBEK, P. Twitter as a corpus for sentiment analysis and opinion mining. In: *LREC*. [S.l.: s.n.], 2010. Citado na página 30.
- PANG, B.; LEE, L. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*. [S.l.], 2005. p. 115–124. Citado 2 vezes nas páginas 30 e 38.
- PANG, B.; LEE, L. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, Now Publishers Inc., v. 2, n. 1-2, p. 1–135, 2008. Citado na página 30.
- PENNEBAKER, J. W.; BOOTH, R. J.; FRANCIS, M. E. Liwc2007: Linguistic inquiry and word count. *Austin, Texas*, 2007. Citado na página 29.
- PETROV, S. et al. Learning accurate, compact, and interpretable tree annotation. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*. [S.l.], 2006. p. 433–440. Citado 2 vezes nas páginas 40 e 45.
- ROCHA, P. A.; SANTOS, D. Cetempúblico: Um corpus de grandes dimensões de linguagem jornalística portuguesa. *Actas do V Encontro para o processamento computacional da língua portuguesa escrita e falada, PROPOR*, v. 2000, p. 131–140, 2000. Citado na página 41.
- SAMMUT, C.; WEBB, G. I. *Encyclopedia of machine learning*. [S.l.]: Springer Science & Business Media, 2011. Citado na página 47.
- SANTOS, D.; SARMENTO, L. O projecto ac/dc: acesso a corpora/disponibilização de corpora. *Actas do XVIII Encontro da Associação Portuguesa de Linguística*, p. 705–717, 2002. Citado na página 41.
- SANTOS, L. M. et al. Twitter, análise de sentimento e desenvolvimento de produtos: Quanto os usuários estão expressando suas opiniões? *Revista PRISMA. COM*, n. 13, 2011. Citado na página 23.

- SILVA, M. J.; CARVALHO, P.; SARMENTO, L. Building a sentiment lexicon for social judgement mining. In: *Computational Processing of the Portuguese Language*. [S.l.]: Springer, 2012. p. 218–228. Citado na página 39.
- SILVA, N. G. R. da. Bestchoice: Classificação de sentimento em ferramentas de expressão de opinião. 2010. Citado na página 30.
- SOCHER, R. et al. Parsing natural scenes and natural language with recursive neural networks. In: *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*. [S.l.: s.n.], 2011. p. 129–136. Citado na página 33.
- SOCHER, R. et al. Recursive deep models for semantic compositionality over a sentiment treebank. In: CITeseer. *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*. [S.l.], 2013. p. 1631–1642. Citado 8 vezes nas páginas 30, 31, 33, 35, 37, 38, 43 e 58.
- SOCHER, R. et al. Supplementary material: Recursive deep models for semantic compositionality over a sentiment treebank. 2013. Citado na página 38.
- STRAPPARAVA, C.; MIHALCEA, R. Learning to identify emotions in text. In: ACM. *Proceedings of the 2008 ACM symposium on Applied computing*. [S.l.], 2008. p. 1556–1560. Citado na página 29.
- YOSHINAGA, C. E.; JUNIOR, F. H. F. d. C. Sentimento do mercado influencia as taxas de retorno futuras das ações? um estudo empírico com anova de dois fatores. *XII Seminário em Administração - SemeAd - FEA-USP*, 2009. Citado na página 23.