Sobre normalização e classificação de polaridade de textos opinativos na web

Lucas Vinicius Avanço

Lucas Vinicius Avanço

Sobre normalização e classificação de polaridade de textos opinativos na web

Dissertação apresentada ao Instituto de Ciências Matemáticas e de Computação - ICMC-USP, como parte dos requisitos para obtenção do título de Mestre em Ciências - Ciências de Computação e Matemática Computacional. *VERSÃO REVISADA*

Área de Concentração: Ciências de Computação e Matemática Computacional

Orientadora: Profa. Dra. Maria das Graças Volpe Nunes

USP – São Carlos
Outubro de 2015

Ficha catalográfica elaborada pela Biblioteca Prof. Achille Bassi
e Seção Técnica de Informática, ICMC/USP,
com os dados fornecidos pelo(a) autor(a)

A946s

Avanço, Lucas Vinicius
Sobre normalização e classificação de polaridade de textos opinativos na web / Lucas Vinicius Avanço; orientadora Maria das Graças Volpe Nunes. -- São Carlos, 2015.
102 p.

Dissertação (Mestrado - Programa de Pós-Graduação em Ciências de Computação e Matemática Computacional) -- Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo, 2015.

1. Processamento de Língua Natural. 2. Análise de Sentimentos. 3. Mineração de Opiniões. 4. Classificação de textos. 5. Normalização de textos de web. I. Volpe Nunes, Maria das Graças, orient. II. Título.

Lucas Vinicius Avanço

On normalization and polarity classification of opinion texts on the web

Master dissertation submitted to the Instituto de Ciências Matemáticas e de Computação - ICMC-USP, in partial fulfillment of the requirements for the degree of the Master Program in Computer Science and Computational Mathematics. *FINAL VERSION*

Concentration Area: Computer Science and Computational Mathematics

Advisor: Profa. Dra. Maria das Graças Volpe Nunes

USP – São Carlos
October 2015

Este trabalho teve o apoio da Samsung Eletrônica da Amazônia Ltda. sob os termos da Lei 8.248/91

Agradecimentos

À minha família, o que tenho de mais importante!

À professora Graça por sua dedicação, orientação e atenção durante todo o desenvolvimento desse trabalho.

À Magali pelo apoio e trabalho em conjunto em diversas questões desse trabalho.

Aos professores Thiago e Sandra, e todos os colegas do NILC, sempre muito atenciosos e prontos para ajudar.

Ao CNPQ e à Samsung Eletrônica da Amazônia Ltda. pelo financiamento desse trabalho.

Resumo

AVANÇO, L. V.; **Sobre normalização e classificação de polaridade de textos opinativos na web**. 2015. 102 p. Dissertação (Mestrado em Ciências - Ciências de Computação e Matemática Computacional) - Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP.

A área de Análise de Sentimentos ou Mineração de Opiniões tem como um dos objetivos principais analisar computacionalmente opiniões, sentimentos e subjetividade presentes em textos. Por conta da crescente quantidade de textos opinativos nas mídias sociais da *web*, e também pelo interesse de empresas e governos em insumos que auxiliem a tomada de decisões, esse tópico de pesquisa tem sido amplamente estudado. Classificar opiniões postadas na *web*, usualmente expressas em textos do tipo "conteúdo gerado por usuários", ou UGC (*user-generated content*), é uma tarefa bastante desafiadora, já que envolve o tratamento de subjetividade. Além disso, a linguagem utilizada em textos do tipo UGC diverge, de várias maneiras, da norma culta da língua, o que impõe ainda mais dificuldade ao seu processamento. Este trabalho relata o desenvolvimento de métodos e sistemas que visam (a) a normalização de textos UGC, isto é, o tratamento do texto com correção ortográfica, substituição de internetês, e normalização de caixa e de pontuação, e (b) a classificação de opiniões, particularmente de avaliações de produtos, em nível de texto, para o português brasileiro. O método proposto para a normalização é predominantemente simbólico, uma vez que usa de forma explícita conhecimentos linguísticos. Já para a classificação de opiniões, que nesse trabalho consiste em atribuir ao texto um valor de polaridade, positivo ou negativo, foram utilizadas abordagens baseadas em léxico e em aprendizado de máquina, bem como a combinação de ambas na construção de um método híbrido original. Constatamos que a normalização melhorou o resultado da classificação de opiniões, pelo menos para métodos baseados em léxico. Também verificamos extrinsecamente a qualidade de léxicos de sentimentos para o português. Fizemos, ainda, experimentos avaliando a confiabilidade das notas dadas pelos autores das opiniões, já que as mesmas são utilizadas para a rotulação de exemplos, e verificamos que, de fato, elas impactam significativamente o desempenho dos classificadores de opiniões. Por fim, obtivemos classificadores de opiniões para o português brasileiro com valores de medida F1 que chegam a 0,84 (abordagem baseada em léxico) e a 0,95 (abordagem baseada em AM), e que são similares aos sistemas para outras línguas, que representam o estado da arte no domínio de avaliação de produtos.

Palavras-chave: análise de sentimentos, classificação de opiniões, normalização de UGC

Abstract

AVANÇO, L. V.; **On normalization and polarity classification of opinion texts on the web**. 2015. 102 p. Dissertation (Master degree - Computer Science and Computational Mathematics) - Instituto de Ciências Matemáticas e de Computação (ICMC/USP), São Carlos - SP.

Sentiment Analysis or Opinion Mining has as a main goal to process opinions, feelings and subjectivity expressed in texts. The large number of opinions in social media has increased the interest of companies and governments, who have changed their decision-making systems. This has caused a great interest in this research area. Opinions are usually expressed by subjective text, and their processing is a hard task. Moreover, reviews posted on the web are of a especial text type, also called user-generated content (UGC), whose processing is a very challenging task, since they differ in many ways from the standard language. This work describes the design of methods and systems aimed at (a) the normalization of UGC texts, through the use of spell checking, substitution of web slangs, case and punctuation correction, and (b) the classification of opinions at document level, especially for reviews of products in Brazilian Portuguese. The method proposed for normalization of UGC is linguistically motivated. For the classification of opinions, which, in this work, consists in assigning a polarity value (positive or negative) to a opinion text, some lexicon-based and machine learning approaches, as well as a combination of both in a new hybrid manner have been implemented and evaluated. We noticed that the text normalization has improved the results of opinion classification for lexicon-based methods. The quality of the sentiment lexicons for Portuguese was extrinsically evaluated. The reliability of the opinions' authors was verified, since they are used for labeling samples. We concluded that they significantly impact the performance of the opinion classifiers. Finally, we proposed some opinion classifiers for Brazilian Portuguese whose F1-measures values reach 0.84 (lexicon-based approach) and 0.95 (machine learning approach), which are analogous to the the similar systems for other languages, which represent the state of the art in the domain of reviews of products.

Keywords: sentiment analysis, opinion classification, UGC normalization

Sumário

Lista de Figuras

Lista de Quadros

Lista de Tabelas

Capítulo

1

Introdução

1.1 Contextualização e Motivação

Buscar opiniões e avaliações de terceiros em relação às mais diversas entidades que nos rodeiam é inerente ao ser humano. Esperamos ter ao menos uma noção dos riscos e ganhos associados a um negócio antes de despender recursos.

A expressão de uma opinião está diretamente ligada ao ato de exteriorizar sentimentos e emoções. Trata-se de uma impressão subjetiva vivenciada por um indivíduo em relação a alguma entidade, física ou não, com a qual interagiu, direta ou indiretamente.

A chamada Computação Afetiva (CA) (*Affective Computing*), desenvolvida a partir da década de 1990, com destaque para o trabalho de Picard (1997), em que se discute acerca de modelos para o reconhecimento computacional de emoções humanas, faz uma ligação entre Computação, Engenharia e emoções.

Apesar de CA estar mais ligada a questões de interação entre humanos e máquinas, uma das formas primárias de reconhecimento de emoções é a que se dá na forma textual. Isso nos remete à área de Processamento de Língua Natural (PLN), que, em linhas gerais,

tem por objetivo lidar com a língua humana, seja escrita ou falada, de forma automática ou semiautomática. PLN também é uma área bastante abrangente, mas podemos pensar que, de sua intersecção com CA, surgiu uma área de estudos mais recente, conhecida como Análise de Sentimentos ou Mineração de Opiniões (MO).

MO tem por objetivo analisar computacionalmente opiniões, sentimentos e subjetividade presentes em textos (Pang e Lee, 2008). Nas últimas décadas, MO passou a ter uma comunidade expressiva e muito ativa de pesquisadores, coincidindo, não por acaso, com a crescente quantidade de textos opinativos produzidos por mídia social na *web* (*blogs*, *microblogs*, redes sociais, fóruns e avaliações de produtos e serviços em portais de *e-commerce*).

O conteúdo que se tem disponível publicamente na *web* dá ao usuário - seja um consumidor ou mesmo um pesquisador de opiniões - a oportunidade de conhecer a experiência de outras pessoas em relação a um determinado produto, serviço ou empresa. Da mesma forma, as empresas desejam conhecer como estão sendo avaliados seus produtos e serviços, assim como os de seus concorrentes.

Outro uso comum que se tem feito de redes sociais e outros recursos da *web* tem relação com questões políticas e sociais. Testemunhamos o potencial que essas ferramentas possuem de disseminar ideias e organizar massas, como nos casos das recentes manifestações em países árabes, em 2011, e também no Brasil, em 2013.

O contexto de aplicação de MO é bastante vasto. A informação obtida tem grande valor para tomada de decisão em inteligência de negócios e de governo. Alguns exemplos de aplicação são: previsão de desempenho de vendas (Liu et al., 2007), análise de postagens no Twitter para entender o comportamento do mercado (Bollen et al., 2011), distribuição estratégica de propagandas em páginas *web* (Jin et al., 2007), elaboração de sistemas de recomendação (Jakob et al., 2009), sumarização automática (Balahur et al., 2012; Kabadjov et al., 2011), previsão de resultados de eleições na política (Tumasjan et al., 2010) e análise de posições políticas (Laver et al., 2003).

Do ponto de vista acadêmico, é importante ressaltar que o estudo de MO traz consigo

muitos problemas de PLN ainda não totalmente resolvidos, tais como: desambiguação de sentido de palavras; tratamento de negação, sarcasmo e ironia; *parsing* (análise sintática); resolução de anáfora, entre outros. Um desafio adicional para as ferramentas de PLN são as características dos textos que formam um córpus compilado da *web*: são textos produzidos por escritores de todo nível cultural e social, alguns deles descompromissados com as regras da língua culta, e que refletem fortemente a oralidade, contendo, assim, inúmeros ruídos que devem ser tratados anteriormente ao uso de outras ferramentas. A esse tipo de texto é comumente atribuído o termo UGC (*user-generated content*) (Krumm et al., 2008), cuja tradução é "conteúdo gerado por usuário".

É difícil estabelecer com clareza qual o estado da arte em classificação de opiniões, por conta de fatores como: domínio (*reviews* sobre filmes, hotéis, restaurantes, produtos em geral, serviços em geral, política, etc.), tipo de texto (*tweets*, *reviews*, postagens em fóruns, *blogs*, e em diferentes tipos de redes sociais), nível textual (documento, sentença ou aspectos), definição do problema de classificação (seis emoções básicas; polaridade positiva ou negativa; utilizar classe neutra ou não; intervalo contínuo - valor numérico de polaridade), natureza do método classificador (baseado em léxico ou em aprendizado de máquina) e língua. Entretanto, podemos dizer que os melhores resultados para o inglês alcançam um valor de acurácia igual a 0,88 (*reviews* sobre produtos e serviços) (Yang et al., 2015), 0,93 (*reviews* sobre filmes) (Nguyen et al., 2014) e 0,95 (*reviews* sobre filmes) (Sharma e Dey, 2012).

Especificamente para a língua portuguesa, nos últimos anos foram criados recursos, como léxicos de sentimentos (Souza et al., 2011; Balage et al., 2013; Silva et al., 2012; Gonçalo Oliveira et al., 2014) e córpus (Freitas et al., 2012; Carvalho et al., 2011; Dosciatti et al., 2013; Hartmann et al., 2014), além de trabalhos que relatam o desenvolvimento de classificadores de opiniões com foco em textos de Twitter (Souza e Vieira, 2012), resenhas de livros (Balage et al., 2013), política (Tumitan e Becker, 2013), manchetes de notícias (Martinazzo et al., 2012; Dosciatti et al., 2013), *reviews* sobre produtos (Anchiêta et al., 2015), entre outros. Os valores de acurácia variam de 0,52, para textos sobre política,

a 0,83, para *reviews* sobre produtos. Embora, recentemente, tenham surgido trabalhos na área para o português, percebemos a falta de trabalhos exploratórios que investiguem diversas formas de classificar opiniões, que vão além dos métodos clássicos, baseados em léxico ou em aprendizado de máquina.

Apesar de termos como objetivo principal deste trabalho a classificação de opiniões (expressas em UGC), ao mesmo tempo, investimos na tarefa de normalização de UGC, que visa beneficiar diversas outras tarefas de PLN que tenham como foco o processamento de textos dessa natureza. É importante destacar que o processo de normalização a que nos referimos neste trabalho consiste em reduzir a quantidade de erros ou ruídos que ocorrem com frequência em UGC. Destacamos dentre eles: erros ortográficos contextuais e não-contextuais, presença de termos típicos do linguajar empregado em textos informais da *web*, o internetês, pontuação incorreta, além do uso indevido ou incorreto de caixa alta ou baixa.

1.2 Objetivos

O projeto aqui descrito tem como objetivos gerais: 1) investigar algoritmos para a classificação de opiniões em textos da *web* escritos em português brasileiro (PB); 2) implementar classificadores de opiniões e avaliar seus desempenhos na classificação de um determinado tipo de opinião; e 3) desenvolver métodos e ferramentas para a normalização de UGC.

Devido à dificuldade em se classificar opiniões sem conhecer o contexto em que se inserem (Aue e Gamon, 2005), para este projeto será utilizado um domínio referente à avaliação de produtos.

Outro ponto importante a ressaltar é que os métodos para classificação de opinião serão restritos ao nível de documento apenas. Não serão abordadas técnicas para tratar opiniões em nível intra-sentencial, considerando sintagmas ou outras partes de uma sentença.

As principais perguntas que este trabalho pretende responder são:

1. Que tipos de desafios os textos UGC em PB oferecem para a tarefa de classificação de opiniões e como vencê-los?

2. A normalização de textos da *web* (UGC - *user-generated content*) melhora o desempenho de classificadores de opiniões?

3. A qualidade dos recursos (p.ex. do léxico de sentimentos) para o PB impacta os resultados dos classificadores de opiniões?

4. Ao aplicar métodos consagrados da literatura - normalmente criados originalmente para o inglês - a textos em PB, serão obtidos resultados semelhantes, a despeito das diferenças de língua e recursos (léxico, córpus, etc.) utilizados?

5. O que acontece com os resultados de um método de classificação de opiniões quando se altera o córpus de treinamento ou mesmo o domínio dos textos opinativos de entrada?

6. É possível melhorar os resultados obtidos por métodos individuais ao combiná-los em uma proposta híbrida?

1.3 Organização da Monografia

O Capítulo 2 introduz os principais conceitos da área de MO, bem como as abordagens e os trabalhos relacionados, tanto para o inglês, quanto para o português. No Capítulo 3 apresentamos os recursos essenciais utilizados em MO. Em seguida, o Capítulo 4 traz um estudo do córpus de *reviews* de produtos utilizado neste trabalho, e também são apresentadas as ferramentas desenvolvidas para a normalização de UGC. O Capítulo 5 contém a descrição dos métodos e classificadores de opiniões desenvolvidos. No Capítulo 6, os classificadores são avaliados e comparados entre si, e, por fim, no Capítulo 7, apresentamos nossas conclusões e algumas ideias para trabalhos futuros.

Capítulo

2

Fundamentos Teóricos e Trabalhos Relacionados

Neste capítulo são apresentadas as principais definições, conceitos, desafios e formas de avaliação que permeiam a atividade de Mineração de Opiniões (MO), principalmente o que se refere à atividade de classificação de opiniões.

2.1 Mineração de Opiniões

Apesar da área de MO abranger os mais diversos tipos de estudo de sentimentos e subjetividade, a tarefa mais investigada no momento se restringe a classificar opiniões (documento, sentenças ou aspectos) em polaridades positiva, negativa ou neutra.

De modo geral, o que é produzido textualmente por humanos pode ser classificado em fato ou opinião (Liu, 2010). Fatos tendem a ser expressos como sentenças objetivas, desprovidas de sentimento, que transmitem alguma informação acerca de um acontecimento, entidade ou seus aspectos (p. ex.: "*O governo decidiu cortar investimentos na educação*").

Já opiniões estão presentes frequentemente em sentenças subjetivas que carregam alguma avaliação, impressão ou sentimento relacionado a algo concreto ou não (p. ex.: "*O produto que comprei é muito bom, superou minhas expectativas.*"). Entretanto, vale lembrar que também pode ocorrer a expressão de uma opinião, de forma implícita, por meio de fatos, de sentenças objetivas (p. ex.: "*Essa máquina de lavar gasta muita água.*").

Há diferentes tipos de opiniões. Segundo Liu (2012), elas podem ser classificadas em: regular ou comparativa; direta ou indireta; implícita ou explícita. Uma opinião é regular se há apenas uma única entidade alvo de opinião, ao contrário de uma comparativa em que duas ou mais entidades se relacionam e são alvos de comparação e qualificação. A opinião é direta se o autor faz referência direta qualificando a entidade alvo de opinião, e indireta se há a presença de entidades auxiliares que não são alvo da opinião, mas recebem uma qualificação, e de forma indireta representam a opinião em relação a entidade alvo (isso ocorre comumente em relações de causa e efeito). E, por último, a opinião é implícita se for estritamente necessário conhecimento de mundo para entender seu significado, já a explícita possui pistas mais claras, frequentemente contendo palavras de sentimento.

Abaixo listamos alguns exemplos, relacionando os diferentes tipos de opinião.

1. Regular, direta e explícita

"*O celular que comprei é ótimo.*"

   - Regular: não há comparação entre entidades, a única entidade alvo de opinião é "*celular*".
   - Direta: não há a presença de outra entidade sendo avaliada, que indiretamente qualificaria a entidade "*celular*".
   - Explícita: a expressão de sentimento é feita explicitamente, com o auxílio da palavra de sentimento "*ótima*".

2. Regular, direta e implícita

"*Essa máquina de lavar gasta muita água.*"

- Regular: não há comparação entre entidades, a única entidade alvo de opinião é "*máquina de lavar*".
- Direta: não há a presença de outra entidade sendo avaliada, que indiretamente qualificaria a entidade "*máquina de lavar*".
- Implícita: é necessário conhecimento de mundo para inferir que o fato de gastar muita água é algo negativo.

3. Regular, indireta e explícita

"*Após o uso da medicação percebi que minha cabeça estava melhor.*"

- Regular: não há comparação entre entidades, a única entidade alvo de opinião é "*medicação*".
- Indireta: a medicação é o principal alvo da opinião, entretanto, a opinião ocorre de forma indireta, já que uma outra entidade, "*cabeça*", é que está sendo qualificada → "*cabeça estava melhor*".
- Explícita: o sentimento é explícito, "*cabeça estava melhor*".

4. Regular, indireta e implícita

"*Após o uso do remédio, meu filho voltou a ser o que era, já até está brincando com os colegas.*"

- Regular: não há comparação entre entidades, a única entidade alvo de opinião é "*remédio*".
- Indireta: o remédio é o principal alvo da opinião, entretanto, a qualificação ocorre de forma indireta, já que a expressão do sentimento ocorre no relato de outros eventos: "*meu filho voltou a ser o que era*" e "*brincando com os colegas*".

- Implícita: é necessário conhecimento de mundo para inferir que o fato da criança estar brincando com os colegas é algo positivo no contexto do uso de um remédio.

5. Comparativa, direta e explícita

"*O sistema X é mais seguro que o Y.*"

- Comparativa: a opinião é feita com base na comparação de duas entidades.
- Direta: não há entidades intermediárias sendo qualificadas.
- Explícita: o sentimento é expresso explicitamente, "*mais seguro*".

6. Comparativa, direta e implícita

"*Preciso carregar a bateria do celular X todo santo dia, já com o celular Y fico a semana toda sem carregar.*"

- Comparativa: a opinião é feita com base na comparação de duas entidades.
- Direta: não há entidades intermediárias sendo qualificadas.
- Implícita: é necessário conhecimento de mundo para inferir que o fato de necessitar de recargas em pequenos intervalos de tempo é algo ruim.

7. Comparativa, indireta e explícita

"*Quando tomei o remédio X meus joelhos ficaram bons, mas ao mudar para o Y, não percebi a mesma melhora*"

- Comparativa: a opinião é feita com base na comparação de duas entidades.
- Indireta: entidades intermediárias são qualificadas, "*joelhos*", o que indiretamente qualifica a entidade alvo da opinião, "*remédio*".
- Explícita: o sentimento é expresso explicitamente, "*joelhos ficaram bons*".

8. Comparativa, indireta e implícita

> "*Após tomar o remédio X consegui voltar a jogar futebol, já com o Y, nem de goleiro pude jogar.*"

- Comparativa: a opinião é feita com base na comparação de duas entidades.
- Indireta: não há nenhuma referência direta que qualifique as entidades alvos da opinião, "*remédio X*" e "*remédio Y*".
- Implícita: é necessário conhecimento de mundo para inferir que "*voltar a jogar futebol*" é algo positivo, pelo menos no contexto de tomar um remédio, enquanto que "*nem de goleiro pude jogar*" é algo negativo.

Opinião é um conceito bastante amplo, porém, no contexto de MO, segundo Liu (2012), é definida como um sentimento positivo, negativo ou neutro, dirigido a uma entidade ou seus aspectos (positivo: "*Esse celular tem uma tela muito boa, e a bateria é excelente*"; negativo: "*Não recomendo, é um celular ultrapassado, a câmera é horrível*"; neutro: "*Comprei esse celular para presentear minha filha, ela está fazendo 12 anos*"). Uma entidade pode ser um bem de consumo, pessoa, organização ou evento, enquanto que aspectos são características ou atributos de uma entidade. Extrair e classificar opiniões são os principais objetos de estudo da área.

Podemos, portanto, dividir MO em dois subproblemas a serem resolvidos: a identificação de textos ou sentenças opinativas e a classificação do sentimento expresso em cada texto. Apesar dos vários trabalhos relacionados à identificação de subjetividade em textos (Riloff e Wiebe, 2003; Wiebe e Mihalcea, 2006; Wiebe et al., 2004), segundo Liu (2010), é mais comum assumir que os textos a serem analisados quanto à polaridade são de fato opinativos. Neste trabalho, segue-se essa abordagem, já que distinguir textos opinativos dos que não expressam opinião constitui um problema muito mais difícil de ser resolvido em comparação com a identificação da polaridade de um texto opinativo (Banea et al., 2008).

Há dois principais pontos a serem considerados no que se refere à classificação de opiniões: a natureza do método classificador e o nível de granularidade das unidades básicas textuais a serem classificadas. Quanto aos métodos, podem ser implementadas abordagens Baseadas em Léxico (Turney, 2002; Taboada et al., 2011), de Aprendizado de Máquina (Pang et al., 2002; Mohammad et al., 2013) ou a combinação de ambas (Prabowo e Thelwall, 2009). Já com relação às unidades textuais, de acordo com Liu (2012), elas podem ser as seguintes: documento, sentença ou aspecto.

Em nível de documento (Pang et al., 2002; Turney, 2002), um texto opinativo como um todo, que pode ser a avaliação de um produto ou serviço, é rotulado como positivo, negativo ou neutro. No nível de sentenças (Wilson et al., 2004), um documento é subdividido e cada sentença tem sua polaridade identificada. Já no nível de aspecto (Hu e Liu, 2004), são extraídos e agrupados entidade e aspectos e identificados os sentimentos para cada um deles. Neste trabalho investigamos a classificação de polaridade no nível de documento apenas.

Para um melhor entendimento do que se espera de um classificador de opiniões nos diferentes níveis de classificação, considere a seguinte avaliação de um telefone celular, obtida do site Buscapé[1](cada sentença foi numerada para facilitar a descrição dos sentimentos presentes):

"(1) *Design ótimo, um celular moderno e de baixo custo, suas funções são práticas e objetivas.* (2) *A câmera traseira é muito boa mas a frontal deixa um pouco a desejar.* (3) *Android é muito bom e rápido, dificilmente fica lento.* (4) *Em geral um excelente aparelho.*"

- Nível de Documento (opinião completa): Pode-se dizer que a polaridade do texto é positiva, dada a predominância de sentimentos positivos ("*ótimo*", "*moderno*", "*baixo custo*", "*práticas*", "*objetivas*", "*muito boa*", "*excelente*", etc.).

- Nível de Sentença: A sentença (1) expressa uma opinião positiva em relação ao

[1]http://www.buscape.com.br

celular como um todo. Já a sentença (2) possui um sentimento positivo em relação à câmera traseira, entretanto pode ser classificada como negativa já que a presença da conjunção adversativa "mas" intensifica o sentimento da oração no que se refere "a frontal deixa um pouco a desejar". Por fim as sentenças (3) e (4) também são positivas.

- Nível de Aspecto: Na sentença (1) são identificados os seguintes aspectos e sentimentos correspondentes: design (ótimo), custo (baixo) e funções (práticas, objetivas); além da entidade: celular (moderno). Em (2) são obtidos: câmera traseira (muito boa) e câmera frontal (deixa a desejar). Em (3) é identificado: Android (muito bom, rápido). Finalmente em (4) aparece a entidade "aparelho" que pode ser agrupada com a entidade "celular" em (1).

Este exemplo de avaliação de produto (texto opinativo) é constituído apenas por opiniões regulares (Liu, 2011), diretas (sem entidades intermediárias com relações de causa e efeito) e explícitas (sentenças subjetivas – sentimentos explícitos). Este é o cenário mais comum que tem sido considerado pela maior parte dos trabalhos nessa área de pesquisa, sendo este também o caso a ser tratado neste projeto. No entanto, há ainda outras formas de expressão da opinião, como mostramos anteriormente nos exemplos de opiniões regulares/comparativas, diretas/indiretas e explícitas/implícitas.

2.2 Avaliação de Resultados

Não é uma tarefa muito simples estabelecer a qualidade exata de um classificador de opiniões, a começar pela própria dificuldade em se ter um conjunto de dados cujos rótulos (positivo, negativo ou neutro) sejam conhecidos e confiáveis. Quem rotula impõe um viés que acaba definindo ou influenciando fortemente o comportamento do classificador, se estivermos considerando métodos de AM.

A maior parte dos trabalhos de classificação de opiniões que utilizam AM aplica métodos supervisionados, onde cada exemplo da base de dados deve possuir uma classe

determinada por um especialista do domínio. Sendo assim, costuma-se obter *reviews* da *web* e os respectivos rótulos de forma automática, já que muitas *reviews* são acompanhadas de alguma medida objetiva, por exemplo, um número de estrelas, ou uma nota num intervalo de 0 a 5.

No trabalho de Maks e Vossen (2013) é apresentada a discrepância existente entre *reviews* e as respectivas notas dadas pelos seus autores. Um dos problemas detectados pelos autores é o fato de o autor da *review* avaliar aspectos diferentes no texto e na nota. Não é raro encontrar uma avaliação cujo texto é completamente negativo, onde são criticados um ou dois aspectos da entidade em questão, porém a nota é alta, bastante positiva.

Independentemente da abordagem seguida para a construção do classificador, é bastante séria essa questão de trabalhar com um conjunto de dados cujos rótulos são pouco confiáveis. Para classificadores baseados em léxico, esse problema de origem dos rótulos pode tornar a avaliação da acurácia do classificador bastante enganadora. Em AM essa questão é ainda mais crítica, já que precisamos de um conjunto de dados com o mínimo de ruído possível, caso contrário haverá poucas chances de ocorrer aprendizado da real natureza do problema que estamos resolvendo.

Outros pontos que dificultam a avaliação de um classificador de sentimentos são: domínio de aplicação, método de classificação e língua. A maior parte dos trabalhos em análise de sentimentos é desenvolvida para textos de gêneros e domínios bastante específicos, como por exemplo, *reviews* sobre produtos e serviços específicos (filmes, livros, restaurante, hotel, etc.), opiniões no cenário político, *tweets*, demais textos de redes sociais, entre outros. Os textos para cada domínio possuem suas peculiaridades, logo cada um dos classificadores possui diferentes características também, principalmente se pensarmos em classificadores obtidos por métodos de AM.

Comparar classificadores obtidos por métodos de natureza distinta também constitui um problema. Métodos baseados em léxico tendem a ser bastante genéricos, ou seja, alcançam resultados parecidos para diferentes domínios. O mesmo não ocorre ao se cons-

truir um classificador utilizando AM, já que ele aprende o que é positivo e negativo com base nos textos que lhe foram apresentados, aprendendo inclusive termos que são empregados apenas naquele domínio específico. Isso nos permite entender porque, em geral, um classificador obtido com AM tende a gerar melhores resultados do que um baseado em léxico, quando aplicados pontualmente em um único domínio.

A questão de língua sempre impõe alguma dificuldade ao compararmos sistemas de PLN, entretanto, considerando as demais áreas de PLN, já mais consagradas, vemos que os valores que medem o desempenho de sistemas no estado da arte não diferem tanto quando comparamos pares de línguas distintas. Na Análise de Sentimentos, que é uma área ainda nova, supomos que o mesmo também ocorra.

As principais medidas utilizadas para a avaliação da classificação de sentimentos são as mesmas empregadas na classificação tradicional de textos (por tópicos ou assuntos) e na Recuperação de Informação. São elas: Precisão, Cobertura, F-Measure e Acurácia. Cada uma dessas medidas pode ser obtida a partir da matriz de confusão (Tabela 2.1), que consiste simplesmente em uma tabela que discrimina os tipos de acertos e erros cometidos em uma tarefa de classificação (tradicionalmente binária).

Tabela 2.1: Matriz de confusão

		Humano	
		Positivo	Negativo
Máquina	Positivo	VP	FP
	Negativo	FN	VN

- Precisão (P): de todas instâncias classificadas como pertencentes a uma determinada classe, quantas de fato pertencem a essa classe.

  Precisão para a classe "positivo" (P_+):

$$P_+ = \frac{VP}{VP + FP}$$

Precisão para a classe "negativo" (P_-):

$$P_- = \frac{VN}{VN + FN}$$

- Cobertura (C): do total de instâncias que pertencem a uma determinada classe, quantas foram classificadas como tal.

  Cobertura para a classe "positivo" (C_+):

$$C_+ = \frac{VP}{VP + FN}$$

  Cobertura para a classe "negativo" (C_-):

$$C_- = \frac{VN}{VN + FP}$$

- F-Measure (F_β): medida que relaciona precisão e cobertura por meio de uma média harmônica desses valores.

$$F_\beta = (1 + \beta^2) \cdot \frac{P \cdot C}{\beta^2 \cdot P + C}$$

  Costuma-se utilizar $\beta = 1$, o que dá origem à medida conhecida por F1-Measure, F1-Score, ou por simplicidade, F1.

$$F1 = 2 \cdot \frac{P \cdot C}{P + C}$$

  Assim como é feito para as medidas de precisão e cobertura, os valores de F1 são calculados para cada uma das classes ("positivo" e "negativo").

- Acurácia (A): mede a quantidade de acertos em relação ao total de instâncias classificadas.

$$A = \frac{VP + VN}{VP + FN + FP + VN}$$

As medidas mencionadas (Precisão, Cobertura, F-measure e Acurácia) são melhores quanto mais próximas de 1 e piores quanto mais próximas de 0.

2.3 Abordagens baseadas em Léxico

Dado um texto, ou partes menores que compõem o mesmo, nosso objetivo é inferir qual a *orientação semântica* ou a *polaridade* que este texto possui. Assim como na maior parte dos trabalhos em MO, assumimos que um texto pode estar associado a uma polaridade *positiva*, *negativa* ou *neutra*, de acordo com os sentimentos presentes nesse texto. Desejamos, portanto, de alguma forma, identificar quais sentimentos estão expressos no texto, combiná-los, e assim obter um valor que nos diga se o texto corresponde a uma opinião positiva, negativa ou neutra.

Na abordagem baseada em léxico, toma-se como premissa que as palavras, independentemente do contexto em que aparecem, possuem uma polaridade a priori, que pode ser expressa numericamente, indicando tanto o sentido da polaridade (positiva ou negativa) quanto a sua força (muito positiva ou muito negativa). Isso nos remete aos léxicos de sentimentos, que são compostos por uma lista de palavras de sentimentos, ou seja, palavras que possuem uma polaridade a priori.

Um dos principais trabalhos que seguem essa estratégia é o de Turney (2002), cujo objetivo é classificar *reviews* em **recomendado** ou **não recomendado**. O resultado é obtido a partir da média das *orientações semânticas* calculadas para os *bigramas* que possuem adjetivos ou advérbios no texto. Adjetivos e advérbios comumente carregam uma polaridade. A classificação da *review* é feita por meio dos seguintes passos: 1) utiliza-se um *tagger* (etiquetador morfossintático) para a identificação de padrões (bigramas) em que estão presentes adjetivos e advérbios; 2) estima-se a orientação semântica de cada bigrama identificado; 3) rotula-se a *review* como um todo de acordo com a ponderação das orientações semânticas dos bigramas.

Inicialmente são recuperados os bigramas que possuem um adjetivo ou advérbio e um

segundo termo para a definição de um contexto. Em seguida cada um desses bigramas deve ter a orientação semântica calculada, e isso é feito utilizando o algoritmo de PMI-IR (*Pointwise Mutual Information - Information Retrieval*) (Church e Hanks, 1990). Basicamente, o que o algoritmo faz é mensurar a diferença entre a informação mútua de um bigrama com uma palavra bastante positiva (p. ex.: "*excellent*") e a informação mútua desse mesmo bigrama com uma palavra bastante negativa (p. ex.: "*poor*").

A medida pontual de informação mútua (PMI) mede a quantidade de informação que variáveis aleatórias compartilham entre si. Nesse caso, as variáveis são os bigramas e as palavras semente ("*excellent*" e "*poor*"). O cálculo é dado pela Equação 2.1.

$$PMI(bigrama, semente) = \log_2 \frac{P(bigrama \cap semente)}{P(bigrama)P(semente)} \tag{2.1}$$

Utilizando métodos de Recuperação de Informação (IR), como buscadores na *web*, é possível recuperar esses valores de probabilidade, por meio da contagem em que essas variáveis aparecem juntas ($P(bigrama \cap semente)$) e separadas ($P(bigrama)$ e $P(semente)$). Perceba que o valor de $PMI(bigrama, semente)$ é mínimo para o caso em que "bigrama" e "semente" não possuem nenhuma relação de dependência, já que teríamos o seguinte valor $P(bigrama \cap semente) = P(bigrama)P(semente)$, resultando em $\log_2 1 = 0$. Por fim, obtém-se a orientação semântica (SO) pela Equação 2.2.

$$SO(bigrama) = PMI(bigrama, "excellent") - PMI(bigrama, "poor") \tag{2.2}$$

O último passo consiste em calcular a média de todas as orientações semânticas e classificar a *review* em "recomendado", caso a média seja positiva, ou "não recomendado", caso contrário. O resultado reportado por Turney (2002) é de acurácia média igual a 0,75, sendo o melhor resultado para o domínio de veículos com acurácia igual a 0,84, e o pior para filmes, 0,66.

Outro trabalho que classifica opiniões utilizando conhecimento linguístico explícito é relatado em Taboada et al. (2011), que apresenta uma nova forma de cálculo de orientação

semântica (*Semantic Orientation Calculator* - SO-CAL). São utilizadas técnicas para a identificação e o tratamento de fenômenos da língua que modificam a polaridade a priori das palavras de sentimento, como contextos de negação e intensificação.

SO-CAL estabelece inicialmente duas premissas: 1) as palavras possuem uma polaridade conhecida a priori, ou seja, que independe de contexto; 2) as polaridades podem ser expressas como valores numéricos.

O fenômeno de intensificação é modelado por meio de modificadores que possuem uma porcentagem de alteração da palavra de sentimento. Palavras que intensificam um sentimento possuem uma porcentagem positiva, já as que diminuem a intensidade do sentimento possuem porcentagem negativa. Isso permite que alguns modificadores possuam mais força que outros, ou seja, que aumentem muito ou reduzam muito a polaridade de uma palavra de sentimento. Se pensarmos em casos para o português, teríamos, por exemplo: muito (+25%), demais (+50%), pouco (-25%) e pouquíssimo (-50%).

Para tratar negação, primeiro é definida a região de influência de um termo de negação (p. ex.: "não", "nunca", "nem", etc.) sobre uma palavra de sentimento. Com base nas etiquetas morfossintáticas das palavras seguintes é determinado até que ponto uma palavra de negação pode ser influente. A palavra de sentimento que está sob influência de negação é deslocada em 4 unidades (os léxicos de sentimentos utilizados por Taboada et al. (2011) estão em uma escala de -5 a 5).

Também são reconhecidos termos que tendem a tornar a presença de sentimentos nas sentenças pouco confiável para o cálculo de orientação semântica. A lista de termos utilizada é constituída por modais (p. ex.: "*would*" e "*should*"), expressões condicionais (p. ex.: "*if*"), alguns verbos como "espero", "duvido", expressões interrogativas e palavras entre aspas. Em muitos casos esses termos neutralizam o efeito das palavras de sentimento que ocorrem nos textos, logo essas deveriam ser ignoradas. Um exemplo de sentença em que isso ocorre é "*Esse filme deve ser muito bom*", onde o modal "deve" neutraliza a positividade de "muito bom". Em casos como este, as sentenças são marcadas como neutras.

Finalmente, para computar a orientação semântica de um texto, são somadas as polaridades das palavras, eventualmente modificadas por contexto, e o valor obtido é dividido pelo número de sentenças. Se o resultado for superior a um limiar, o texto é classificado como positivo, caso contrário, como negativo.

Os autores apresentam resultados para diferentes versões de classificadores, e com aplicação em diferentes domínios (*reviews* sobre livros, carros, computadores, eletrodomésticos, hotéis, filmes e músicas). As médias dos valores de acurácia (considerando todos os domínios) são: 0,6604, com base apenas nas polaridades de palavras de sentimento; 0,6835, ao adicionar tratamento de negação; 0,7135, ao tratar negação e intensificação; e o melhor resultado, 0,7874, ao tratar os outros fenômenos linguísticos relatados anteriormente.

Para o PB, o trabalho desenvolvido por Souza e Vieira (2012) tem o objetivo de classificar *tweets*[2] utilizando uma abordagem baseada em léxico. Os autores apresentam resultados comparando o impacto de diferentes formas de tratamento da negação, diferentes léxicos de sentimentos, além do efeito que causam diferentes técnicas de pré-processamento. Os léxicos utilizados foram o SentiLex e o OpinionLexicon. Duas formas distintas para tratar negação foram testadas: 1) janela de 5 palavras; 2) escopo de negação considerando a sentença toda. Como pré-processamento foram normalizados alguns casos específicos comuns em textos de *web*, como repetição intencional de vogais, além de ser aplicada uma correção ortográfica baseada em similaridade fonética.

Os resultados reportados pelos autores apontam que: o tratamento de negação por escopo sentencial gera melhor resultado; o léxico OpinionLexicon mostrou-se melhor que o SentiLex; e as técnicas de pré-processamento, por eles empregadas, tiveram um impacto muito pouco significativo no resultado da classificação. Os melhores resultados apresentados pelos autores são de F1-positiva igual a 0,54 e F1-negativa igual a 0,45.

Outro trabalho, também para o PB, é relatado em Balage et al. (2013). Os autores avaliam três léxicos de sentimentos para o português (SentiLex, OpinionLexicon, e LIWC-

[2]Mensagens de até 140 caracteres postadas na rede social Twitter

sentic[3]), utilizando um método baseado em léxico similar ao proposto por Taboada et al. (2011), para um domínio de avaliações de livros (córpus ReLi). O córpus utilizado para avaliação possui 4.210 opiniões positivas e 1.024 negativas. No nível sentencial são: 2.883 sentenças positivas e 596 negativas.

No nível textual, os 3 léxicos tiveram valores de acurácia próximos, entretanto o LIWC-sentic mostrou-se melhor para classificação de textos positivos (F1-positiva igual a 0,7037), e para os negativos o SentiLex teve melhores resultados (F1-negativa igual a 0,6025). Já para o nível sentencial, a acurácia obtida com o LIWC-sentic é superior à obtida utilizando os outros léxicos (acurácia igual a 0,5733). Além disso, manteve-se a mesma característica de melhor desempenho em textos positivos com o LIWC-sentic, e em textos negativos, com o SentiLex.

Em Tumitan e Becker (2013) é apresentado um trabalho de classificação de sentimentos, baseado em léxico, para o PB, no domínio de política. O objetivo do trabalho foi determinar se os comentários postados na *web* (versão on-line do jornal Folha de São Paulo[4]) se refletem em indicadores de intenção de voto.

Os autores conduziram um experimento de classificação de sentimento (positivo ou negativo) para 600 sentenças que continham opinião sobre alguns dos candidatos. Criou-se um córpus *gold* a partir dessas 600 sentenças. Dois anotadores rotularam as sentenças e o resultado foi: 482 sentenças negativas, 72 positivas, e 46 neutras. O experimento levou em consideração 3 métodos de classificação: "*Baseline*", "Léxico Modificado", "Sem acentuação".

O cálculo de orientação semântica para cada sentença é basicamente a diferença entre a soma dos termos positivos e a soma dos termos negativos, utilizando como léxico de sentimentos o SentiLex. O primeiro método "*Baseline*" consiste apenas em aplicar esse cálculo de orientação semântica. O segundo, "Léxico Modificado", utiliza um léxico enri-

[3]LIWC (Tausczik e Pennebaker, 2010) é o nome de uma ferramenta para análise de textos para o inglês, portanto daqui em diante iremos usar "LIWC-sentic" para fazer referência a um subconjunto de palavras de sentimentos que fazem parte de uma tradução do léxico original utilizado pela ferramenta LIWC

[4]http://www.folha.uol.com.br/

quecido com termos específicos do domínio da política no contexto daquela situação. O último, "Sem acentuação", trata os problemas de acentuação nas sentenças, simplesmente removendo todos os acentos das palavras e do léxico de sentimentos.

As acurácias de cada uma das variações foram: "*Baseline*" (0,35), "Léxico Modificado" (0,43), "Sem acentuação" (0,52). O melhor valor de precisão para sentenças positivas foi conseguido com o método "Sem acentuação"; para sentenças negativas, foi com o método "Léxico Modificado". Para os valores de cobertura ocorreu o inverso: a melhor cobertura positiva foi obtida utilizando-se o "Léxico Modificado", e a melhor cobertura negativa, o "Sem acentuação".

Um trabalho que reúne tanto aspectos linguísticos quanto características do ambiente de publicação das opiniões é relatado em Neto e Barros (2014). O objetivo dos autores é fazer análise de sentimentos, para o inglês, em debates polarizados não ideológicos, comumente encontrados em fóruns de discussão. O trabalho trata concessões (opiniões associadas a expectativas), referências anafóricas, e análise do grafo de *replies* (respostas a opiniões). As etapas para a classificação dos *posts* são: resolução de anáfora; atribuição inicial de polaridades; análise de concessões; classificação do *post*; e criação e análise do grafo de *replies*.

A resolução de anáfora é feita simplesmente com a substituição do pronome "*it*" pela entidade mais recentemente citada no texto. Pronomes como "*she*" e "*he*" que se referem a pessoas não são considerados.

O próximo passo, de atribuição inicial de polaridade, monta triplas do tipo <*produto, palavra de sentimento, sentença*> com base em padrões linguísticos que relacionam entidade alvo de opinião e a ordem em que ocorrem palavras com determinadas etiquetas morfossintáticas.

A análise de concessões é utilizada para inverter a polaridade de parte da sentença, que é a concessão, já que foi identificada a contra-expectativa. Comumente a concessão e a contra-expectativa são separadas por conectivos como "*but*" ("mas"), "*however*" ("no entanto"), entre outros.

Para classificar a postura do *post* é feita simplesmente a soma das polaridades calculadas para todas as triplas, sendo que é dada uma importância maior para a última tripla, onde geralmente está resumida a opinião.

Por fim, o grafo de *replies* que conecta *posts* é usado para inferir concordância ou discordância entre participantes do debate, dessa forma a polaridade de um *post* é uma pista para a polaridade de outro *post*.

O córpus utilizado é composto por 807 *posts*, sobre diversos temas, obtidos dos sites *CreateDebate* e *ConvinceMe*. Os experimentos realizados mostraram um ganho na acurácia da classificação ao adicionar conhecimento linguístico e também ao utilizar o grafo de *replies*. O resultado da acurácia média do classificador foi de 0,69.

Um trabalho que faz uma análise linguística mais profunda do texto, para o espanhol, é apresentado em Vilares et al. (2013). A principal ideia deste trabalho está no uso da estrutura sintática para o cálculo da polaridade, levando em conta os seguintes aspectos linguísticos: intensificação, oração subordinada adversativa e negação.

O método consiste em tratar os aspectos linguísticos citados nos diferentes níveis da árvore sintática e subir as polaridades calculadas em direção ao nó raiz. O tratamento de intensificadores é o mesmo relatado em Taboada et al. (2011), que considera diferentes porcentagens para aumentar ou reduzir a polaridade de uma palavra de sentimento.

Outro aspecto, a presença de oração subordinada adversativa, é considerado também como uma espécie de intensificação, já que a presença de conjunções adversativas possuem a função de contrastar a oração principal com a subordinada, geralmente dando maior ênfase à ideia expressa pela oração subordinada. Sendo assim, os autores simulam esse efeito dando maior peso à polaridade calculada para essa oração e um menor valor para a principal.

Para realizar o tratamento de negação, primeiramente é identificado o contexto ou escopo de um termo que denota negação e em seguida é feita a modificação da polaridade de palavras de sentimento pertencentes a este contexto. Para a definição do escopo de negação são utilizadas diversas regras levando em conta a estrutura sintática onde são

identificados termos de negação. A modificação da polaridade é feita seguindo também o método de Taboada et al. (2011), adicionando uma quantidade fixa na direção contrária a da polaridade original. Por exemplo, "*problemas*" tem uma polaridade igual a -2, e "*no da problemas*" resulta em uma polaridade igual a $-2 + 4 = +2$.

Os autores reportam resultados para diferentes versões do classificador. Primeiramente um *baseline* que classifica com base na combinação de polaridades das folhas ao nó raiz; depois tratando intensificação; depois orações subordinadas adversativas; e depois negação. Com o tratamento de negação melhorou-se muito a acurácia na classificação de sentenças negativas.

Para a versão final do classificador, os autores reportam a acurácia obtida sobre diferentes domínios (tipos de produtos ou serviços): hotel, computador, máquina de lavar, celulares, carros, música, livros, filmes. O melhor resultado foi para hotel com acurácia de 0,90, e o pior para filmes, com 0,66 de acurácia. Os piores resultados (música, livros, filmes), segundo os autores, se deve ao fato da opinião ser expressa de forma muito sutil e subjetiva, ou seja, a polaridade não fica clara mesmo sob olhar humano.

Todos os trabalhos apresentados até aqui utilizam uma abordagem baseada em léxico para a construção de classificadores de opiniões. Alguns utilizam mais conhecimento linguístico, explorando e analisando mais profundamente aspectos e estrutura dos textos, enquanto outros são mais superficiais e aplicam apenas algum cálculo sobre as palavras de sentimentos presentes na opinião.

2.4 Abordagens baseadas em Aprendizado de Máquina (AM)

Em MO tem-se investigado muito o uso de AM, sendo a principal motivação os bons resultados obtidos em classificação de textos por tópicos ao utilizar técnicas desse tipo. Entretanto, como é discutido em Pang e Lee (2008), mesmo tornando o problema de classificação de opiniões um problema de classificação binária (apenas 2 classes: positiva e negativa), o problema é mais desafiador que a classificação de textos em tópicos ou

assuntos.

Ao contrário do que ocorre na classificação de textos por tópicos, as palavras presentes em um texto, apesar de serem bons indicadores, em geral não são suficientes para classificar a polaridade da opinião. Basicamente, o que torna mais difícil a classificação de sentimento é a grande sobreposição que existe entre as classes, ou seja, há opiniões positivas com palavras que denotam sentimento negativo e o caso contrário também é frequente. Apenas isso já impõe grande dificuldade em se obter um classificador que discrimine bem opiniões positivas e negativas.

Técnicas de AM permitem que sejam construídos modelos a partir de dados. Em técnicas supervisionadas, como Naive Bayes, ID3, SVM, MLP, Máxima Entropia, entre outros, com um conjunto de exemplos rotulados é possível criar um classificador que seja capaz de classificar novas instâncias nunca vistas. Alguns passos são importantes ao utilizar técnicas desse tipo, um dos mais importantes consiste em determinar quais *features*[5] são representativas para o problema a ser resolvido.

Conforme é relatado em Liu (2012), algumas *features* comumente utilizadas para classificação de opiniões são: termos e as respectivas frequências (*Bag-of-Words*); etiquetas morfossintáticas (*pos - part-of-speech*); palavras de sentimento; palavras modificadoras de sentimento (*sentiment shifters*); e dependência sintática entre as palavras.

Um dos primeiros trabalhos a investigar o uso de AM em classificação de sentimentos foi desenvolvido por Pang et al. (2002). Os autores utilizaram 3 diferentes algoritmos de AM para a construção de classificadores: Máxima Entropia, Naive Bayes e SVM. Para cada um deles foram testadas diferentes *features* para o aprendizado, e o domínio de aplicação considerado foi o de *reviews* de filmes. O conjunto de dados utilizado é composto por *reviews* obtidas do site especializado em descrição e avaliação de filmes, o IMDb[6].

As combinações de *features* investigadas foram: unigramas; unigramas + bigramas; bigramas; unigramas + PoS (*part-of-speech*); adjetivos; unigramas + posição. Para os

[5] *Features* são os atributos que compõem o vetor de características de cada exemplo
[6] http://www.imdb.com

atributos unigramas e bigramas, também foi avaliado o desempenho ao utilizar apenas a informação de presença/ausência em comparação com a informação da frequência.

Os resultados obtidos pelos autores mostraram que, em média, para todas as combinações de *features* utilizadas, comparando-se os três algoritmos, Naive Bayes teve o pior resultado, e SVM, o melhor, apesar das diferenças não serem grandes. Os autores também constatam que utilizar apenas unigramas (presença de termos) produz um resultado melhor em comparação com as outras combinações que adicionam bigramas e PoS (*part-of-speech*). O melhor classificador foi obtido com o algoritmo SVM utilizando como *features* apenas *Bag-of-Words* indicando presença ou ausência do termo. O melhor valor de acurácia reportado, utilizando validação cruzada com 3 *folds*, é de 0,829.

Em Moraes et al. (2013) é apresentado um trabalho de classificação de polaridade para textos curtos (conhecidos pelo nome de *tips*), postados em um tipo de rede social baseada em geolocalização, o *Foursquare*. Nesse tipo de rede social os usuários interagem, compartilham e recomendam locais que frequentaram. Para classificar os *tips* em positivo ou negativo foram criados três classificadores distintos utilizando as seguintes técnicas de AM: Naive Bayes, SVM e Máxima Entropia. Um quarto classificador, baseado em léxico, foi construído utilizando o léxico SentiWordNet (Esuli e Sebastiani, 2006), para efeito de comparação com os métodos de AM.

Para os três classificadores baseados em AM, os dados de treinamento (as *tips*) foram modelados como *Bag-of-Words*, porém o valor para cada atributo (palavra) consiste na medida *tf-idf*. Isto é, cada atributo representa a frequência de uma determinada palavra na *tip*, ponderada pela frequência dessa mesma palavra em todas as *tips* da base.

O classificador baseado em léxico primeiramente aplica um *tagger* (etiquetador morfossintático) e lematiza cada *tip*. O tratamento de negação é realizado invertendo-se a polaridade de palavras que são precedidas por termos que denotam negação. Consultando-se o léxico SentiWordNet, é computada uma pontuação final positiva e negativa. A pontuação final superior define a polaridade que deve ser atribuída à *tip*.

Os resultados obtidos por Moraes et al. (2013) indicam que, em termos da medida F1

média (classe positiva e negativa), o classificador baseado em léxico é, estatisticamente, no mínimo tão bom quanto os métodos baseados em AM. Entretanto, caso o foco seja detectar *tips* negativas, os métodos supervisionados, principalmente SVM e Naive Bayes, são os mais indicados. Os valores de F1-positiva, F1-negativa e F1-média são (para cada um dos classificadores): baseado em léxico (0,85; 0,54; 0,69), SVM (0,80; 0,53; 0,66), Máxima Entropia (0,79; 0,51; 0,65) e Naive Bayes (0,81; 0,55; 0,68).

Tem sido bastante comum também o desenvolvimento de classificadores utilizando AM voltados para a classificação de *tweets*. Esse tipo de texto possui diversas particularidades, a começar pelo uso excessivo de expressões típicas da *web* ("internetês"), além de símbolos, como `#` e `@`, que possuem a semântica de fazer referência a assuntos ou usuários dentro da rede social. Em Mohammad et al. (2013) é apresentado um trabalho para a classificação de *tweets* utilizando o algoritmo SVM e uma combinação de *features* que leva em conta diversas características do texto (algumas dessas características podem ser estendidas a textos de outra natureza).

Antes da extração do vetor de características, foi aplicada uma normalização aos *tweets*: transformação de todas URLs para o padrão `http://someurl` e todas menções a usuários para `@someuser`. Em seguida os textos foram tokenizados e etiquetados com um *tagger* de PoS (*part-of-speech*).

Após a normalização, cada *tweet* passou a ser representado por um vetor de características contendo: n-gramas (presença ou ausência para n = 1,2,3,4); quantidade de palavras totalmente em caixa alta; quantidade de ocorrências de cada *tag* PoS; quantidade de *hashtags* - `#`; pontuação (repetição de sinais de exclamação e interrogação, separados ou combinados); emoticons; palavras alongadas ("loooove") e quantidade de contextos de negação.

Os resultados obtidos com o classificador construído utilizando SVM e todas essas *features* foram: 0,69 de F1-Score para *tweets* e para textos de mensagens por celular (SMS) foi obtido 0,68 de F1-Score.

Para o português, um trabalho que utiliza AM para a classificação de sentimentos,

porém com um objetivo um pouco diferente do que foi apresentado até o momento, é relatado em Dosciatti et al. (2013). Neste trabalho os autores propõem a construção de um classificador das seis emoções básicas (alegria, tristeza, raiva, medo, desgosto e surpresa) em textos curtos.

Os autores compilaram um córpus constituído de 1.750 textos (250 para cada uma das seis emoções e 250 para uma classe "neutra"). Para isso, recuperaram manchetes de notícias do site *globo.com*. O pré-processamento dos textos consistiu em: transformação para minúsculas; remoção de acentos, pontuação e demais caracteres especiais; remoção de *stopwords*; e aplicação de *stemmer*.

Para construção do vetor de características utilizaram a modelagem de *bag-of-words* e *tf-idf (term-frequency - inverse document frequency)*. A partir dessa representação os autores construíram um classificador utilizando SVM, efetuando treinamento e teste com 10-*fold cross validation*.

O valor médio de acurácia obtido foi de 0,61. As classes (emoções) com piores resultados foram "Desgosto" (0,39), "Alegria" (0,45), e "Tristeza" (0,54). Os melhores resultados foram para as classes "Raiva" (0,75), "Medo" (0,81), e "Surpresa" (0,81). A precisão média foi de 0,58 e a cobertura média foi 0,61. Os autores ainda mostraram que os classificadores obtidos com Naive Bayes e KNN produziram resultados piores. Para Naive Bayes a acurácia média foi de 0,49 e para KNN foi de 0,54.

Um outro trabalho, que na verdade não utiliza exatamente AM, mas sim um método estatístico, o LSA (*Latent Semantic Analysis*), é relatado em Martinazzo et al. (2012). O objetivo dos autores foi o de classificar textos curtos (manchetes de notícias), escritos em português, em uma das seis emoções básicas (alegria, tristeza, raiva, medo, desgosto e surpresa).

O método LSA é um método estatístico que permite estabelecer associações entre termos presentes em textos. De forma bem resumida, o método primeiramente constrói uma matriz onde cada linha representa um termo e cada coluna, um texto/documento, assim cada célula corresponde à frequência do termo no documento correspondente. Os

valores armazenados nessa matriz geralmente são normalizados a partir da importância do termo no documento em si e também em toda coleção de documentos, ou seja, costuma-se utilizar a medida *tf-idf*. Por fim, é aplicado o teorema SVD (*Single Value Decomposition*), que produz uma representação da matriz em uma dimensão menor, dando maior ênfase às relações mais "fortes" entre os termos.

Após a aplicação do método LSA, tem-se a representação vetorial dos diversos termos que denotam cada uma das seis emoções básicas. A partir disso é calculado um ponto médio para cada emoção. Por fim, para classificar um texto em uma das emoções, é calculada a similaridade no espaço vetorial, utilizando a medida de cosseno (quanto menor o ângulo entre os vetores, maior a similaridade), entre cada ponto representativo de cada emoção e o texto em análise.

Para avaliar o método, os autores conduziram um experimento com 700 notícias curtas, escritas em PB, extraídas do site *globo.com*. Essas notícias foram rotuladas manualmente em uma das seis emoções. Os resultados não são muito detalhados, relatam apenas uma acurácia geral para a identificação das emoções, que foi em torno de 0,67.

Uma abordagem que também utiliza AM, porém de uma forma distinta das apresentadas até o momento, é proposta por Balage Filho e Pardo (2013) e Balage Filho et al. (2014). Os autores utilizam uma arquitetura híbrida para a classificação de *tweets* em inglês. O método consiste em aplicar classificadores de naturezas distintas, com base em regras, léxico e AM. Os classificadores são aplicados em uma ordem específica, sendo que a execução de um depende do resultado produzido pelo anterior. Conforme relatado nesses trabalhos, a combinação de métodos distintos produz melhores resultados em comparação com cada método aplicado isoladamente. O melhor valor de F1-média (classes positiva e negativa) reportado pelos autores é de 0,6539.

Métodos de AM são construídos com uma quantidade de esforço razoavelmente menor quando comparados a métodos que incorporam explicitamente conhecimento linguístico, ficando a maior parte do trabalho concentrada na escolha de um bom conjunto de *features*. No entanto, não se deve desprezar o esforço de preparação (obtenção, normalização e

rotulação) do córpus de treinamento. O motivo para os bons resultados em domínios específicos para o qual o classificador foi treinado é o mesmo que explica a sua fragilidade ao ser aplicado em domínios distintos: como é discutido em Pang e Lee (2008)[7], uma mesma sentença pode possuir uma conotação positiva em um determinado domínio e negativo em outro.

2.5 Síntese dos trabalhos discutidos

Seguem tabelados os resumos dos trabalhos discutidos nesse capítulo. Na Tabela 2.2 estão os trabalhos para o inglês, e na Tabela 2.3 para o português.

[7] Seção 4.4

Tabela 2.2: Resumos dos principais trabalhos em MO para o inglês

	Turney (2002)	Pang et al. (2002)	Taboada et al. (2011)	Moraes et al. (2013)	Mohammad et al. (2013)	Neto e Barros (2014)
Objetivos	Classificar polaridade em nível de texto	Classificar polaridade em nível de texto	Classificar polaridade em nível de sentença	Classificar polaridade em nível de texto	Classificar polaridade em nível de texto	Classificar polaridade em nível de texto
Técnicas	Baseado em léxico: PMI-IR	AM: NB, ME, SVM	Baseado em léxico	Baseado em léxico e AM	AM: SVM	Baseado em léxico
Dados	*reviews* (filmes, veículos, bancos, viagens)	*reviews* de filmes	*reviews* (livros, carros, filmes, etc.)	tips (Foursquare)	twitter	*posts* em debates polarizados sites: CreateDebate ConvinceMe
Avaliação	240 (+), 170 (-)	700 (+), 700 (-) 3-fold cross validation	400 (+), 400 (-)	3.454 tips: 3.014 (+), 440 (-), 5-fold cross validation	3.813 tweets 2.094 SMS	807 *posts*
Precisão	-	-	-	(+) 0,92 (-) 0,48	-	-
Cobertura	-	-	-	(+) 0,82 (-) 0,75	-	-
F1	-	-	(+) 0,81 (-) 0,79	(+) 0,85 (-) 0,55	0,69 (tweets) 0,68 (SMS)	-
Acurácia	0,66 - 0,84	0,77 - 0,83	0,80	-	-	0,69 (média entre 5 debates)

Tabela 2.3: Resumos dos principais trabalhos em MO para o português

	Souza e Vieira (2012)	Balage et al. (2013)	Tumitan e Becker (2013)	Dosciatti et al. (2013)	Martinazzo et al. (2012)
Objetivos	Classificar polaridade em nível de texto	Classificar polaridade em nível de sentença e texto	Classificar polaridade em nível de sentença	Classificar textos curtos em uma das seis emoções básicas	Classificar textos curtos em uma das seis emoções básicas
Técnicas	Baseado em léxico	Baseado em léxico	Baseado em léxico	AM: SVM	LSA
Dados	twitter	resenhas de livros (ReLi)	opiniões sobre política	manchetes de notícias (site *globo.com*)	manchetes de notícias (site *globo.com*)
Avaliação	1700 tweets	textos: 4.210 (+), 1.024 (-) sentenças: 2.883 (+), 596 (-)	600 sentenças: 482 (-) 72 (+) 46 neutras	1.700 manchetes 250 para cada emoção 250 para classe neutra	700 manchetes
Precisão	(+): 0,66 (-): 0,74	textos: (+): 0,96 (-): 0,72 sentenças: (+): 0,92 (-): 0,46	(+): 0,27 (-): 0,90	0,58 (média)	-
Cobertura	(+): 0,46 (-): 0,33	textos: (+): 0,58 (-): 0,52 sentenças: (+): 0,65 (-): 0,48	(+): 0,65 (-): 0,51	0,61 (média)	-
F1	(+): 0,55 (-): 0,45	textos: (+): 0,70 (-): 0,60 sentenças: (+): 0,74 (-): 0,47	(+): 0,37 (-): 0,65	0,77 (média)	-
Acurácia	-	textos: 0,52 sentenças: 0,57	0,52	0,61 (média)	0,67

Capítulo

3

Recursos utilizados em Mineração de Opiniões

Os recursos básicos para a tarefa de MO são, em geral: córpus e léxico de sentimentos. Apesar de algumas limitações, principalmente se compararmos com o inglês, é notório o esforço recente da comunidade em prover tais recursos para o PB, permitindo o desenvolvimento de mais trabalhos na área para o PB.

3.1 Córpus

Um dos recursos mais importantes para diversas tarefas de PLN é o córpus: um conjunto de textos que podem estar pré-processados e, geralmente, anotados, dependendo do fim para o qual o córpus foi compilado. Em MO precisamos de um córpus de textos opinativos para (a) levantar conhecimento linguístico para embasar as propostas; (b) servir de recurso para treinar algoritmos baseados em AM, e, nesse caso, o ideal é que contenha anotações de polaridade (classe discreta ou contínua) para cada texto opinativo, ou para suas partes (nível de sentença ou aspecto, por exemplo).

Para o PB, um córpus de opiniões anotado e disponível para uso é o ReLi (Freitas

et al., 2012). O córpus contém avaliações de livros obtidas do site *skoob.com*, local onde leitores comentam e opinam sobre os livros que leram. São 1.600 resenhas sobre 13 livros. A anotação manual do córpus indica as polaridades em nível de sentença e de sintagma. Para o português europeu, há o SentiCórpus-PT (Carvalho et al., 2011), um córpus relacionado à política de Portugal, formado por comentários de eleitores sobre um debate político para eleição do Parlamento Português. A anotação foi feita indicando a polaridade [-2,-1,0,1,2] em nível de sentença, sendo -2 muito negativa, 0 neutra e 2 muito positiva.

Outro córpus para o PB, disponível para uso, é apresentado no trabalho de Dosciatti et al. (2013). O córpus é formado por notícias obtidas do site *globo.com*, relacionadas à categorias como: política, policial, economia, entre outros. São 1.750 textos, sendo 250 para cada uma das seis emoções básicas (alegria, tristeza, raiva, medo, desgosto e surpresa), e mais 250 para uma classe "neutra", significando que não há predominância de nenhuma dessas emoções.

Todas as demais referências encontradas na literatura, para o PB, apenas citam a construção de córpus específicos para o trabalho desenvolvido: Siqueira e Barros (2010) utilizam um córpus de avaliações de serviços prestados por lojas online (*reviews* obtidas do site "E-bit"), e Ribeiro Jr et al. (2012) trabalham com um córpus no domínio de veículos (*reviews* obtidas do site "Carrosnaweb").

Neste trabalho o foco está em classificar opiniões acerca de produtos em geral, portanto temos utilizado o córpus construído por Hartmann et al. (2014), compilado da *web*. Os detalhes desse córpus são apresentados no Capítulo 4. Como se trata de postagens de usuários em um portal de avaliações de produtos, os textos que compõem o córpus têm os mais diversos tipos de erro e ruído, desde simples erros de digitação até o uso de expressões próprias da *web*, o "internetês", passando por abreviaturas e construções fortemente influenciadas pela oralidade. Essas características tendem a degradar os resultados obtidos com ferramentas de PLN, que são usualmente criadas com base em textos bem escritos. Logo, uma tarefa de normalização do córpus pode ser necessária. Essa questão

e o trabalho de normalização também são detalhados no Capítulo 4.

3.2 Léxico de Sentimentos

Léxicos de sentimentos são formados por um conjunto de palavras de uma língua usualmente utilizadas para expressar sentimento. Em PB temos palavras que denotam um sentimento positivo, como por exemplo: "bom", "legal", "funciona", "adorei", "perfeição", "maravilhoso", "perfeitamente"; outras, sentimento negativo: "ruim", "imperfeição", "defeito", "terrível", "negativamente", "detestei"; e aquelas que, dependendo daquilo que qualificam ou do contexto, podem denotar um ou outro sentimento: "barato", "pouco", "pequeno", "qualidade". Chamamos essas palavras de: *palavras de sentimento*. Nota-se que podem pertencer a diferentes classes gramaticais (adjetivos, nomes, verbos, advérbios), mas encontram-se sobretudo entre os adjetivos e advérbios.

Pode-se construir um léxico de sentimentos de forma manual ou automática. A primeira, por extração de dicionários e gramáticas, é bastante custosa e, muitas vezes, incompleta, e costuma ser utilizada apenas como um refinamento do resultado obtido com formas automáticas. Os principais métodos automáticos são: Baseado em Dicionário (Hu e Liu, 2004) e Baseado em Córpus.

O método baseado em dicionário trabalha na expansão de uma lista de palavras-semente selecionadas manualmente, cujas polaridades (positiva ou negativa) são conhecidas. O algoritmo resume-se aos seguintes passos: dada uma palavra da lista inicial de palavras-semente, são buscados os sinônimos e antônimos da palavra em questão. As novas palavras encontradas são adicionadas à lista, e o processo iterativo se repete. Após a obtenção da lista, um processo de inspeção e limpeza manual pode ser empregado.

O léxico obtido atribui uma polaridade a princípio genérica e independente de contexto, entretanto, muitas palavras assumem uma polaridade ou outra dependendo do contexto em que são empregadas. Por exemplo, a palavra "inesperado" poderia ser positiva para um livro ou filme, e negativa se estiver se referindo a algo que deveria ser confiável e estável,

como um veículo por exemplo. Um método baseado em córpus poderia ser utilizado para ajudar nesse sentido.

A abordagem baseada em córpus utiliza também uma lista inicial de palavras-semente a serem utilizadas para descobrir outras palavras de sentimento e as respectivas polaridades com base em um córpus. No que é considerado ainda hoje um dos principais trabalhos nesse sentido, Hatzivassiloglou e McKeown (1997) utilizam, para o inglês, um córpus e um conjunto de adjetivos-semente a fim de derivar novos adjetivos que carregam sentimento no córpus. Seu método aplica um conjunto de regras linguísticas. Uma dessas regras, por exemplo, identifica a ocorrência de um adjetivo-semente, mais uma conjunção "e" seguida de outro adjetivo. O novo adjetivo identificado é anotado com a mesma polaridade do adjetivo-semente. Por exemplo, para uma sentença do tipo "O celular é muito bom e barato", o adjetivo barato seria anotado com a mesma polaridade de "bom". Outras regras similares com conjunções "mas" e "ou" são também aplicadas.

No entanto, vale lembrar que, mesmo dentro de um único domínio, uma palavra pode assumir diferentes polaridades dependendo do contexto em que ocorre. Por exemplo, considere o verbo "demorar" nas sentenças seguintes: "A bateria demora para acabar" e "O touch-screen demora para capturar o toque". A primeira ocorrência é positiva enquanto que a segunda é negativa. Logo, independentemente do método empregado, ainda é um grande desafio a construção de léxico de sentimentos.

Para a língua portuguesa, temos disponíveis os seguintes léxicos de sentimentos até o momento: SentiLex (Silva et al., 2012), OpinionLexicon (Souza et al., 2011), um subconjunto dos *synsets* do recurso OntoPT com polaridades associadas (Gonçalo Oliveira et al., 2014), e uma tradução, do inglês, do léxico utilizado pelo software LIWC (Balage et al., 2013).

O SentiLex foi construído para o português europeu, para o propósito específico de mineração de opiniões relacionadas a entidades humanas. A primeira etapa de sua construção utiliza um conjunto de padrões léxico-sintáticos (3-grama, 4-grama e 5-grama) a fim de identificar os candidatos a adjetivos que caracterizam entidades humanas. Em

seguida, para a atribuição de polaridade às palavras de sentimento identificadas, é construído um grafo, onde os nós são os lemas e as arestas representam relação de sinonímia. Os nós possuem uma das seguintes polaridades -1, 0, 1 ou *null*. Um nó com polaridade *null* terá uma nova polaridade atribuída de acordo com a informação de sua vizinhança. O léxico é constituído por 7.014 lemas (82.347 formas flexionadas). São 4.779 (16.863) adjetivos, 1.081 (1.280) nomes, 489 (29.504) verbos e 666 (34.700) expressões idiomáticas.

A construção do OpinionLexicon, para o PB, utiliza três diferentes métodos (Souza et al., 2011): baseado em dicionário, baseado em córpus e tradução. No método baseado em dicionário, é atribuída a polaridade de acordo com a relação entre as distâncias mínimas da palavra a ser anotada e cada palavra do conjunto de sementes positivas e de sementes negativas. O método baseado em córpus aplica a medida pontual de informação mútua (mede o grau de dependência entre duas palavras, como é mostrado em Turney (2002)) em um córpus de *reviews* de filmes e jornalístico sobre diferentes temas. Por fim, também utilizou a tradução de um léxico de sentimentos feito para o inglês (Hu e Liu, 2004). A composição final do léxico resultou em 30.678 entradas, sendo 30.236 palavras e 442 expressões.

Outro recurso que pode ser usado como léxico de sentimentos é o conjunto de *synsets* com polaridades associadas, construído com base no léxico de sentimentos SentiLex e nos *synsets* do recurso OntoPT, que é uma base de conhecimento léxico-semântica para o Português, estruturada de forma semelhante à WordNet de Princeton (Gonçalo Oliveira et al., 2014). Os autores desse recurso descrevem sua construção em dois passos: 1) atribuição inicial de polaridade e 2) propagação de polaridade. No primeiro passo é calculada uma polaridade para o *synset* como um todo, sendo que a polaridade de cada lema do *synset* contribui para esse cálculo. O passo seguinte consiste em transmitir a polaridade entre *synsets* que estão diretamente conectados por uma relação semântica. Esse recurso contém 13.843 *synsets* polarizados.

O último léxico de sentimentos que temos para o português é na verdade um subconjunto da tradução do léxico inglês da ferramenta LIWC (Balage et al., 2013). O léxico

original possui 127.161 entradas, cada uma delas classificadas com uma ou mais das 64 etiquetas semânticas estabelecidas (exemplos de etiquetas são: *family*, *friend*, *feel*, *health*, *anger*). Para formar um léxico de sentimentos, costuma-se recuperar apenas as entradas etiquetadas com "posemo" (emoção positiva) ou "negemo" (emoção negativa). Após esse filtro temos: 12.878 palavras de sentimento positivo, e 15.115, de negativo.

No Capítulo 5 será apresentada uma avaliação extrínseca desses léxicos (LIWC-sentic[1], OpinionLexicon, SentiLex e OntoPT-sentic[2]), indicando os resultados obtidos ao serem utilizados em um classificador de opiniões (*reviews* de produtos) baseado em léxico.

[1]http://www.nilc.icmc.usp.br/portlex/index.php/pt/projetos/liwc

[2]Daqui em diante usaremos "OntoPT-sentic" para fazer referência ao subconjunto de *synsets* polarizados do recurso maior OntoPT

Capítulo

4

Proposta de um Normalizador de UGC

Neste capítulo discutimos os desafios de se processar textos do tipo UGC (*user-generated content*), bem como apresentamos propostas para sua normalização. Discutimos primeiramente o córpus de trabalho, suas características e principais problemas. Em seguida, propomos um conjunto de ferramentas para minimizar os efeitos desses problemas.

4.1 Córpus de *Reviews* de Produtos

Assume-se, neste trabalho, um domínio específico de aplicação dos classificadores de opiniões a serem desenvolvidos, que é o de *reviews* de produtos. Necessitamos, portanto, de um córpus de *reviews* desse domínio. O córpus que utilizamos foi construído por meio de *crawling* de um dos mais conhecidos *websites* destinados à comparação de produtos, o Buscapé[1]. Uma descrição detalhada do córpus pode ser encontrada em Hartmann et al. (2014).

[1] http://www.buscape.com.br

O córpus é caracterizado por textos não muito longos, cada um deles variando bastante quanto ao nível de qualidade referente às normas da língua. O ambiente em que esses textos são postados, a *web*, tradicionalmente é aberto à publicação de textos produzidos pelos mais diversos tipos de autores. Há textos que refletem um descompromisso total com a escrita bem formada e correta, mas também há textos bem escritos, sem erros. O córpus é composto por 85.910 *reviews*, 4.088.718 tokens e 837.866 tipos.

A título de exemplo, seguem dois textos obtidos do córpus (na forma como ocorrem originalmente, com os erros preservados):

- `muita tequinologia é demais depois de adiqirir vc não vai querer outro`
  `O que gostei: exelente`
  `O que não gostei: nada declarar`

- `Quando decidi que era este o produto, eu já estava satisfeita com as suas funcionalidades e passei a comparar preço nas lojas. Quando recebi em casa, o produto me encantou ainda mais. É mais compacto do que eu imaginava, seus botões e imagens são realmente intuitivos e o manual é tão completo que é preciso conter a vontade de partir logo para a utilização.`
  `O que gostei: Barato e fácil de usar.`
  `O que não gostei: Nada.`

Não é uma tarefa muito fácil processar textos com uma grande quantidade de ruídos, principalmente ao tentar utilizar ferramentas como *taggers* e *parsers*, que, naturalmente, são construídas com base em textos que obedecem às normas da língua. Percebe-se uma grande variedade de tipos de erros cometidos em textos de *web*, o que dificulta ainda mais a tarefa de normalização dos mesmos.

4.2 Pré-processamento do Córpus

Para se ter conhecimento das tarefas de pré-processamento necessárias, e quanto cada uma delas é importante, Hartmann et al. (2014) apresentam uma análise do córpus com-

pilado da *web* que estamos usando neste trabalho. A metodologia consistiu em: primeiramente todos os tokens foram buscados no léxico Unitex-PB (Muniz et al., 2005), e os não identificados foram selecionados e submetidos ao corretor ortográfico Aspell com o dicionário do PB. Em seguida, foram selecionados 9 textos e submetidos ao *parser* Palavras (Bick, 2000), com o objetivo de avaliar o impacto que a correção de erros produz na precisão do *parser*. A precisão passou de 0,8373 para 0,8428.

Como o ganho não foi muito significativo, uma análise mais detalhada foi realizada, separando as palavras desconhecidas (5.775 tokens) em diferentes categorias para que uma tarefa de anotação fosse empregada. As categorias definidas são: X (erros ortográficos), SI (siglas), NP (nomes próprios), AB (abreviações), IN (internetês), ES (estrangeirismo), UM (unidades de medida) e SC (sem categoria). Na Figura 4.1, algumas ocorrências podem ser vistas com as respectivas anotações de erros em textos do córpus.

ela e **[X: é]** muito escura quando vc **[AB: você]** esta [X: está] deitado se vc **[AB: você]** tiver **[IN: estiver]** sentado ela e **[X: é]** boa mas deitada nao **[X: não]** e **[X: é]** muito nao **[X: não]**. A Samsung inova o mercado de tv's **[AB: televisões]** com uma grande obra de arte que se adequa **[X: adéqua]** à qualquer ambiente. Esta tv **[AB: televisão]** possui excelente imagem quando ligada a uma fonte de dvd **[SI: DVD]** com hdmi **[SI: HDMI]** e na tv **[AB: televisão]** a cabo (Digital). O som é perfeito quando é personalizado pelo usuário. Ou seja, MENU, SOUD **[ES: SOUND]**, EQUALIZAR E ENTER **[ES]**.

Gostei dimais **[X: demais]** dessa câmera, comprei outra!Além de uma excelente e reconhecida marca, essa câmera tem um design **[ES]** super inovador e mtu **[IN: muito]** atraente...uma resolução mtu **[IN: muito]** boa e **[X: é]** td **[AB: tudo]** o que uma boa câmera SONY tem que ter!! Até hj **[AB: hoje]** nunca me deixou na mão... recomendo!!!

muto **[X: muito]** bom para manuziar **[X:manusear]** quando vc **[AB: você]** ta **[AB: está]** trabalhano **[X: trabalhando]** com este produto ,nao **[X: não]** tenho que recramar **[X: reclamar]** gostei mesmo parabéns. RECOMENDO O PRODUTO, FÁCIL DE USAR, ADOREI !

Figura 4.1: Textos do córpus com anotação de erros (Hartmann et al., 2014)

A tarefa de anotação de erros em categorias teve como resultado a seguinte distribuição mostrada na Figura 4.2.

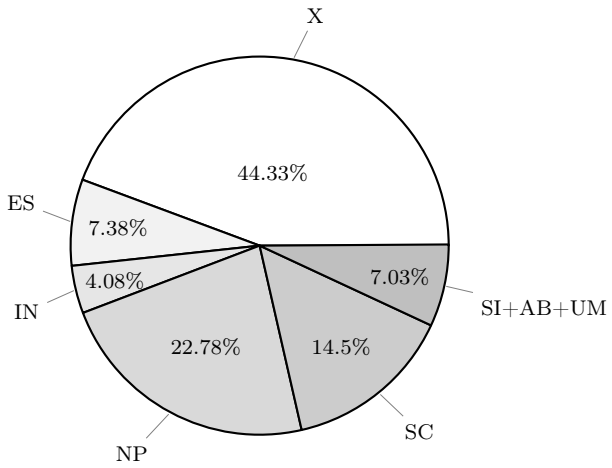

Figura 4.2: Distribuição de erros por categorias.

Notamos a predominância de erros ortográficos, o que já era de se esperar devido à informalidade do meio em que são publicados esses textos. Percebe-se também que a segunda categoria de erros mais frequente é a de Nomes Próprios. Isso ocorre devido ao descuido dos autores em escreverem, principalmente nomes de marcas, lojas, ou nomes de tecnologias, com a primeira letra minúscula. Os outros tipos de erros, como estrangeirismo e internetês, ocorrem com uma frequência um pouco menor.

Outro trabalho dirigido à normalização deste mesmo córpus foi desenvolvido e é detalhado em Duran et al. (2014). Nele apresenta-se uma análise linguística detalhada dos erros encontrados no córpus, além de uma avaliação do impacto da correção de cada tipo de erro nos resultados de um *tagger*.

Esse experimento utilizou uma amostra de 10 *reviews* do córpus Buscapé, um total de 1.226 tokens, que foi submetida à anotação do *tagger* MXPOST[2] (Ratnaparkhi et al., 1996). A precisão obtida pela ferramenta (treinada com um córpus jornalístico) nessa amostra foi de 0,8874, enquanto que o melhor valor relatado para textos do domínio jornalístico é de 0,9698.

Em seguida, criou-se uma amostra *gold* (revisada por um humano especialista) para as

[2]http://www.inf.ed.ac.uk/resources/nlp/local_doc/MXPOST.html

seguintes categorias de erros: erros ortográficos, caixa alta/baixa, pontuação e expressões da *web* (internetês). O ganho obtido para cada tipo de correção é mostrado na Tabela 4.1.

Tabela 4.1: Ganho de desempenho do *tagger* corrigindo cada tipo de erro (Duran et al., 2014)

Caixa	+15,94%
Pontuação	+4,34%
Ortografia	+2,90%
Internetês	+1,45%

Estes resultados mostram como é importante a correção de caixa. Isso faz sentido já que ferramentas como *taggers* e *parsers*, em geral, têm seu processo de anotação fortemente guiado pela informação de caixa das palavras.

4.3 Tokenizador de textos de *Web* escritos em PB

Textos publicados na *web*, principalmente *reviews*, frequentemente estão em uma linguagem própria da internet, além de possuir muitos termos relativos à tecnologia. O uso de um tokenizador tradicional acaba acarretando a perda de termos importantes, como é o caso de unidades de medida (30Mb, 1kb, 2gb, etc.) e também emoticons (:) , :-(, =) , etc.). Desenvolvemos, portanto, um tokenizador capaz de identificar emoticons, unidades de medida, e URL's. Segue abaixo um exemplo típico de *review* com o respectivo resultado produzido pelo tokenizador que desenvolvemos.

- Texto de entrada

```
Oi, esse cel tem 15gb de storage, mto bom:-).
Isso custa R$1299,99 reais, tem um bom
custo-benefício(na minha opinião):))!!!E pode
pedir 30% de desconto.
```

- Texto tokenizado

```
Oi , esse cel tem 15 gb de storage , mto bom :-) .
Isso custa  R$ 1299,99 reais , tem um bom
custo-benefício ( na minha opinião ) :)) ! ! ! E pode
pedir 30 % de desconto .
```

Como pode ser visto nesse exemplo, estão identificadas partes importantes do texto, como emoticons, números, unidades de medida, moeda e pontuação. Todos esses tokens dão um nível de conhecimento em relação ao texto bastante útil para a posterior classificação de sua polaridade.

4.4 *Spell-checker* baseado em fonética para o português brasileiro

Como pode ser visto nos resultados apresentados por Hartmann et al. (2014), erros ortográficos são muito frequentes no córpus de *reviews* de produtos, e, apesar de termos bons *spell-checkers* para o PB, como o Aspell[3] (livre) e o do ReGra (proprietário) (Nunes e Oliveira Jr, 2000), especificamente para esses tipos de texto eles não se saíram tão bem. O principal motivo para isso está na natureza dos erros ortográficos presentes nessas *reviews*, a maioria deles ocorre pela forte influência da língua falada. Além disso, o pré-processamento de um córpus, ao contrário de uma revisão de texto, requer a correção automática sem interferência humana. *Spell-checkers* tradicionais produzem como resultado uma lista de candidatos à correção do erro detectado, e cabe ao usuário escolher o mais apropriado. Aqui visamos encontrar o melhor candidato que vai substituir a palavra em questão, sem interferência humana.

Temos, portanto, desenvolvido um *spell-checker* para o PB, especialmente orientado a corrigir erros motivados pela similaridade fonética, e que deve efetuar a correção automática dos erros identificados sem necessitar de uma interferência humana. Para isso precisamos de boas heurísticas para o ranqueamento dos possíveis candidatos à solução.

[3]http://aspell.net

Nossa estratégia consiste, primeiramente, em obter uma lista de candidatos à correção, recuperando as palavras do léxico que estão a uma distância de edição igual a 1 ou 2 da palavra errada. Em seguida são aplicados algoritmos de checagem de proximidade fonética; um consiste de uma modificação que fizemos do Soundex (Rusell, 1918); o outro consiste de regras fonéticas formuladas especificamente para o PB.

A distância de edição (ou distância de Levenshtein) consiste de uma medida da quantidade de alterações (inserção, remoção ou substituição de uma letra) necessárias para que uma palavra se transforme em outra. Dada uma palavra não encontrada no léxico, geramos uma lista de candidatos e ordenamos essa lista decrescentemente por frequência, com base na lista de frequências do Corpus Brasileiro[4]. Quanto maior a frequência, mais provável que esta seja a substituição mais plausível. Este é um primeiro filtro que utilizamos para ranquear os candidatos.

Entretanto, apenas a informação de frequência não é suficiente. O passo seguinte consiste em aplicar regras fonéticas do PB que atribuem códigos iguais a consoantes que compartilham o mesmo fonema (por exemplo: chinelo - xinelo; jiló - giló; casa - caza; entre outros casos típicos de uso da oralidade na escrita). Assim, são gerados códigos numéricos para a palavra original e para cada um dos candidatos. Se algum candidato tiver o mesmo código que a palavra original, esse candidato é então retornado como correção. Três, das vinte e uma regras fonéticas para PB, utilizadas pelo *spell-checker*, são apresentadas na Tabela 4.2. O conjunto completo está descrito no Apêndice A.

Na Tabela 4.2, a coluna "Condição" indica o contexto e a(s) letra(s) (em negrito) que deve ser substituída pelo número correspondente na coluna "Código". Por exemplo, a palavra "carro" satisfaz a condição "**c** seguido por a", logo um dos códigos a serem gerados para essa palavra será "1arro".

A seguir são exemplificados os códigos gerados para algumas das palavras e regras apresentadas na Tabela 4.2:

[4]http://corpusbrasileiro.pucsp.br/cb/Downloads/wl_cb_full_1gram_sketchengine.txt.zip

Tabela 4.2: Algumas regras fonéticas para PB utilizadas pelo *spell-checker*

Código	**Condição**	**Palavras Corretas**	**Palavras Erradas**
1	**c** (seguido por 'a', 'o' ou 'u'); **k**; **qu**; **q**;	casa, quero, cobre	kasa, kero, kobre
2	**ç**; **c** (seguido por 'e' ou 'i'); **s** (inicial seguido por 'e' ou 'i'); **s** (final ou seguido por consoante); **ss**; **sc** (seguido por 'e' ou 'i'); **xc** (seguido por 'e' ou 'i'); **z** (final); **x** (seguido por consoante);	cachaça, nascer, exceção, extremo, cebola	caxassa, nasser, esseção, estremo, sebola
3	**ch**; **sh**; **x** (seguido por vogal);	chuva, show, peixe	xuva, xou, peiche

- casa (1asa), kasa (1asa)
- cachaça (1a3a2a), caxassa (1a3a2a)
- extremo (e2tremo), estremo (e2tremo)

Caso não tenha sido encontrado nenhum candidato com o mesmo código que a palavra original ao serem aplicadas as regras fonéticas do PB, utilizamos o algoritmo Soundex (Rusell, 1918), na tentativa de encontrar um candidato foneticamente similar à palavra errada. O algoritmo Soundex (independente de língua) gera códigos numéricos iguais para letras que possuem o mesmo ponto de articulação (considera aspectos fisiológicos e articulatórios da produção da fala). Da mesma forma, novamente, buscamos por candidatos que tenham o mesmo código da palavra original. Se ainda assim não for encontrado um candidato que satisfaça essa condição, é retornada a palavra mais frequente que esteja a uma distância de edição igual a 1 ou 2 da palavra original. No Quadro 4.1 são apresentados os passos do algoritmo utilizado pelo *spell-checker*.

Para avaliar a qualidade de cada método de ranqueamento, ajustamos o *spell-checker*

Quadro 4.1: Algoritmo utilizado pelo *spell-checker*

- Entrada: palavra do texto não encontrada no léxico
- Saída: palavra que substituirá a entrada
    1. Gerar candidatos recuperando palavras do léxico que estão a uma distância de 1 ou 2 em relação à palavra errada
    2. Ordenar por frequência a lista de candidatos
    3. Aplicar Regras Fonéticas do português:
        (a) Gerar código fonético para a palavra dada como entrada
        (b) Para cada palavra da lista de candidatos gerar um código fonético
        (c) Se for encontrada alguma palavra (a mais frequente) com o mesmo código calculado para a entrada, é retornado o candidato como correção
    4. Se ainda não foi retornado um candidato, aplicar o algoritmo Soundex:
        (a) Gerar código soundex para a palavra dada como entrada
        (b) Para cada palavra da lista de candidatos gerar um código soundex
        (c) Retorna a palavra mais frequente que tiver o mesmo código soundex da entrada, se houver
    5. Se ainda não foi retornado nenhum candidato, retorna a palavra mais frequente da lista de candidatos

para funcionar de 3 maneiras diferentes (fazendo a correção automática ao escolher o primeiro candidato da lista de sugestões): I) aplica-se apenas a distância de edição 1 e 2; II) aplica-se a distância de edição 1 e 2, e, em seguida, aplica-se o algoritmo Soundex; III) gera-se a lista com distância de edição 1 e 2, e aplicam-se as regras fonéticas para PB (como as que são mostradas na Tabela 4.2).

A Figura 4.3 mostra que a intersecção da contribuição dos métodos não é muito grande, ou seja, cada um dos métodos corrige erros diferentes, assim, com a combinação de todos alcança-se um resultado melhor do que com o uso das técnicas de forma isolada. Para fazer essa avaliação, utilizamos um total de 1.323 palavras com erros ortográficos. Esse conjunto de palavras foi obtido a partir da tarefa de anotação realizada por Hartmann et al. (2014), tal como foi descrito na Seção 4.2.

Na Tabela 4.3 são apresentados alguns exemplos de palavras incorretas e as respectivas

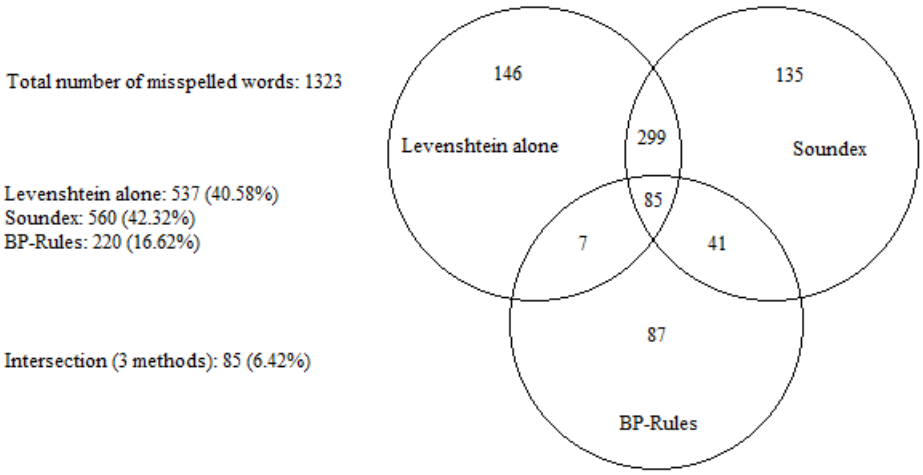

Figura 4.3: Acertos na primeira sugestão para cada método de ranqueamento (Avanço et al., 2014)

sugestões (primeira da lista de sugestões). Na segunda coluna (Frequência), as sugestões foram obtidas com base apenas na distância de edição e no ranqueamento por frequência do Córpus Brasileiro. Na terceira coluna (Soundex), o ranqueamento foi obtido com base na frequência e também na aplicação dos códigos fonéticos determinados pelo algoritmo Soundex. Na última coluna (PB-Regras), o ranqueamento utilizou informação de frequência e de aplicação dos códigos fonéticos definidos por regras fonéticas específicas do PB.

Tabela 4.3: Sugestões considerando cada técnica de ranqueamento isoladamente (as sugestões corretas aparecem sublinhadas)

Palavra Original	**Frequência**	**Soundex**	**PB-Regras**
amarzenamento	armazenamento	SEM SUGESTÃO	SEM SUGESTÃO
espectativa	expectativa	expectativa	expectativa
keijo	meio	queijo	queijo
fasia	casa	fase	fazia
xave	deve	sabe	chave

Avaliou-se também o resultado do *spell-checker* completo, do Quadro 4.1. (indicado por "*Spell-checker* Fonético"), em comparação ao Aspell. A Tabela 4.4 mostra o ganho ao se utilizar informação fonética específica da língua.

Tabela 4.4: Número de acertos (Correto), de erros (Incorreto), ou de falta de sugestão para ambos *spell-checkers*, considerando o experimento para 1.323 palavras

	Correto	Incorreto	Sem sugestão
Spell-checker Fonético	**866 (65,46%)**	**452 (34,16%)**	**5 (0,38%)**
Aspell	621 (46,94%)	654 (49,43%)	48 (3,63%)

Como pode ser visto na Tabela 4.4 nosso *spell-checker* foi capaz de corrigir cerca de 18% de palavras a mais em comparação com o Aspell, o que é um resultado bastante interessante.

Entretanto, ainda com o objetivo de tentar melhorar esse resultado, adicionamos um módulo de conversão grafema-fonema desenvolvido por Mendonça e Aluisio (2014). O conversor produz uma transcrição para a palavra de entrada e também para os candidatos a correção, em seguida esses resultados são utilizados de modo semelhante ao que é feito com as técnicas descritas anteriormente (regras fonéticas para PB e Soundex). No cenário do mesmo experimento anterior, houve um aumento de apenas 1% na acurácia.

Outras modificações realizadas no funcionamento do *spell-checker*, visando sua aplicação em um sistema de normalização, são descritas na seção seguinte.

4.5 *Pipeline* de normalização: UGCNormal

A partir das ferramentas desenvolvidas e de outras existentes, desenvolvemos um sistema para normalização de textos do tipo UGC (*user-generated content*), UGCNormal (Duran et al., 2015). Esse sistema é composto por módulos que desempenham as seguintes funções: delimitação de sentenças; tokenização; correção ortográfica; normalização de internetês e correção de caixa envolvendo acrônimos, unidades de medida e nomes próprios. Na Figura 4.4 apresenta-se a arquitetura do sistema de normalização completo.

O primeiro passo consiste em delimitar o texto em sentenças utilizando a ferramenta desenvolvida por Condori e Pardo (2015). Em seguida as sentenças são submetidas ao

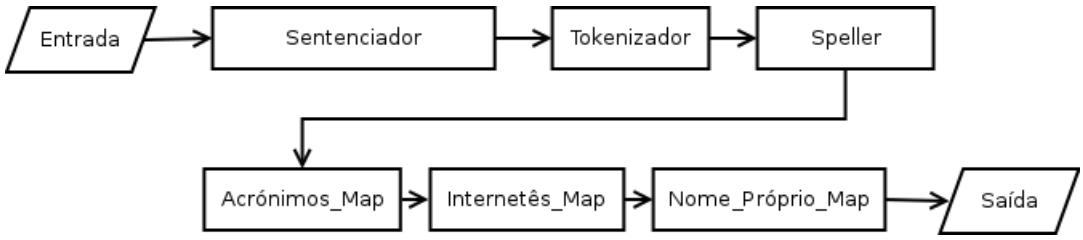

Figura 4.4: *Pipeline* de normalização: UGCNormal

tokenizador, descrito na Seção 4.3, e ao *spell-checker* (Avanço et al., 2014) apresentado na Seção 4.4. Os módulos seguintes utilizam dicionários para efetuar a normalização de siglas, internetês e nomes próprios.

O módulo que aparece na Figura 4.4 como "Acrónimos_Map" tem a função de identificar acrônimos no texto e transformá-los em caixa alta. O módulo "Internetês_Map" realiza a substituição de formas do internetês para as formas esperadas da língua, como por exemplo, "vc" para "você", "tb" para "também", entre outras. O módulo "Nome_Próprio_Map" realiza a capitalização dos nomes identificados, já que nesse tipo de texto é muito comum encontrarmos nomes de marcas, produtos e tecnologias em caixa baixa. Vale lembrar que testamos reconhecedores de entidades nomeadas em uma amostra do córpus, porém o resultado não foi satisfatório.

Os dicionários construídos possuem as seguintes características:

- 21.669 nomes próprios
- 432 formas de internetês
- 248 palavras estrangeiras
- 18 siglas para unidades de medida
- 156 tipos de acrônimos genéricos

Todos esses dicionários são utilizados durante a execução do *spell-checker*, e também cada um deles separadamente em seu módulo de normalização correspondente. Entretanto, apenas os dicionários de nomes próprios e de palavras estrangeiras são considerados para a correção de palavras do texto; o internetês, as unidades de medida e outros

acrônimos, nessa etapa apenas servirão para que o *spell-checker* não tente corrigi-las. Seguimos essa estratégia já que observamos uma alta taxa de ocorrência de nomes próprios e também de palavras estrangeiras erradas (por exemplo: "*desing*" em vez de "*design*", "*blutooth*" em vez de "*bluetooth*", entre outras).

Outra modificação importante no funcionamento do *spell-checker* consiste no uso de uma lista de palavras válidas da língua, que raramente ocorrem em texto do tipo UGC, e que, quando ocorrem, na verdade são decorrências de erros de acentuação. Por exemplo, muitas palavras que se encaixam nessa categoria, e portanto estão nessa lista, são verbos conjugados na segunda ou primeira pessoa do singular muito pouco usadas, como "abundancias", "alugueis", "autenticas", "veiculo", "capitulo", entre outras. Durante a execução, todas as palavras que estão nessa lista não são consideradas como palavras válidas e portanto são corrigidas para a forma correta no contexto em que ocorrem. Essa lista é composta por 561 palavras.

Segue abaixo um exemplo dos resultados, parciais e final, produzidos pelos passos do normalizador (cada erro/ruído está sublinhado):

- Texto de entrada:

  eleh eh mtt daora para vc mecher entao eu recomendu conprarem ese sangsung eleh otimo alem de ser barato eh dahora

- Após sentenciador:

  Eleh eh mtt daora para vc mecher entao eu recomendu conprarem ese sangsung eleh otimo alem de ser barato eh dahora.

- Após tokenizador:

  Eleh eh mtt daora para vc mecher entao eu recomendu conprarem ese sangsung eleh otimo alem de ser barato eh dahora .

- Após *spell-checker*:

Ele eh mtt daora para vc mexer então eu recomendo comprarem esse samsung ele ótimo além de ser barato eh dahora .

- Após “Acrónimos_Map”:

  Ele eh mtt daora para vc mexer então eu recomendo comprarem esse samsung ele ótimo além de ser barato eh dahora .

- Após “Internetês_Map”:

  Ele é muito da hora para você mexer então eu recomendo comprarem esse samsung ele ótimo além de ser barato é da hora .

- Após “Nome_Próprio_Map”:

  Ele é muito da hora para você mexer então eu recomendo comprarem esse Samsung ele ótimo além de ser barato é da hora .

Na seção seguinte é apresentada uma avaliação intrínseca e outra extrínseca da ferramenta UGCNormal.

4.6 Avaliação intrínseca e extrínseca

Foram compiladas duas amostras de textos opinativos (*reviews*) extraídos de sites de *e-commerce*, Buscapé e Mercado Livre. De cada córpus foi obtida uma amostra de 60 *reviews*, que foram utilizadas para a avaliação intrínseca do UGCNormal. Ambas as amostras foram anotadas manualmente por um humano especialista, que anotou os seguintes tipos de erros: erros ortográficos independentes de contexto (p. ex.: “exelente”, “pessimo”, etc.), erros ortográficos dependentes de contexto (p. ex.: “pratico”/“prático”, “esta”/“está”, etc.), internetês (p. ex.: “vc”/“você”, “q”/“que”, etc.), uso inapropriado de caixa (nomes próprios, acrônimos e início de sentença), palavras aglutinadas (p. ex.: “muitobom”/“muito bom”) e pontuação (apenas ponto final).

Na Tabela 4.5 é apresentada para cada tipo de erro/ruído, e para cada amostra de cada córpus, a taxa de correções no seguinte formato **X / Y = Z**, sendo que **Y** corresponde ao total de erros anotados na amostra para um dado tipo de erro/ruído, **X**, ao total de correções, e **Z**, à cobertura da tarefa de normalização.

Tabela 4.5: Taxa de acerto para cada tipo de erro/ruído em cada amostra (Buscapé e Mercado Livre)

Tipo de erro/ruído	**Buscapé**	**Mercado Livre**
Ortográfico Não-contextual	50 / 56 = 0,89	87 / 108 = 0,80
Ortográfico Contextual	15 / 39 = 0,38	24 / 76 = 0,31
Internetês	4 / 6 = 0,67	15 / 25 = 0,60
Caixa (Nomes próprios)	11 / 12 = 0,92	13 / 19 = 0,68
Caixa (início de sentença)	14 / 14 = 1,00	7 / 12 = 0,58
Palavras aglutinadas	0 / 2 = 0	2 / 6 = 0,33
Pontuação	44 / 47 = 0,94	58 / 79 = 0,73

Embora ambas as amostras contenham o mesmo número de *reviews*, percebe-se uma maior quantidade de erros/ruídos na amostra do córpus Mercado Livre, e acreditamos que a diferença nas taxas de acerto entre ambas as amostras ocorra também por conta dessa diferença entre as quantidades de erros. Observa-se uma boa taxa de correção de erros ortográficos não-contextuais, algo entre 0,80 e 0,89, assim como para erros de pontuação (diretamente relacionado à correção de caixa de início de sentença). Como citado anteriormente, os erros ortográficos contextuais são tratados com base em uma lista de palavras analisadas manualmente (palavras pouco frequentes que diferem em apenas um acento de palavras mais frequentes). Com isso corrigimos de 0,31 a 0,38 das ocorrências desse tipo de erro.

Não tratamos erros de palavras aglutinadas, entretanto, há poucos casos em que isso é resolvido pelo próprio tokenizador. Os casos observados foram de ocorrências como "8mb" → "8 mb". Por fim, de 0,60 a 0,67 dos casos de internetês foram resolvidos. Os casos não resolvidos são por limitação do dicionário de formas de internetês. Se uma forma de internetês não consta no dicionário, a mesma não é substituída pela forma padrão da língua.

Também avaliamos a precisão do sistema de normalização, já que um sistema desse

tipo está sujeito também à inserção de novos erros no texto. Verificamos que na amostra do Buscapé foram feitas 149 modificações pelo UGCNormal, sendo que desse total, apenas 11 foram intervenções indevidas (modificações em palavras que estavam corretas), o que corresponde a uma precisão de 0,93. Para a amostra do Mercado Livre, foram feitas 14 modificações indevidas, de um total de 220, o que também corresponde a uma precisão de 0,93.

Como avaliação extrínseca, investigamos o impacto causado pela normalização no resultado produzido por um *tagger*, o MXPOST (Ratnaparkhi et al., 1996), treinado no córpus MAC-Morpho (Aluísio et al., 2003). Para essa avaliação, utilizamos uma amostra menor, apenas 10 *reviews*, já que precisamos de amostras *gold* (corrigidas por humano) que envolvem tanto a correção de *tags* morfossintáticas quanto a correção de erros diretamente no texto. Ao aplicarmos o *tagger* nessa amostra de 10 textos, foi percebida uma acurácia de 0,9135, enquanto que a acurácia reportada na literatura para textos jornalísticos é de cerca de 0,97. Para a amostra *gold* (10 textos corrigidos por um humano), a acurácia do *tagger* foi de 0,9339, e para a amostra normalizada pelo UGCNormal, foi de 0,9315.

Os valores de acurácia, para a amostra *gold* e para a normalizada, são muito próximos, e ambos são melhores que o valor obtido para a amostra original (sem normalização). Realizamos o teste de Wilcoxon (*Signed-rank*) a fim de verificar a chance da melhora na acurácia do *tagger* ter sido obtida ao acaso. Ao nível de significância de 5%, rejeitamos a hipótese da melhora não ser significativa (*p-value* obtido igual a 0,02249).

Ainda como forma de avaliação extrínseca, no Capítulo 6, Seção 6.6, apresentamos resultados obtidos com os classificadores de opinião, ao serem aplicados no córpus não normalizado, e também em um normalizado com o UGCNormal.

CAPÍTULO

5

Métodos de Classificação de Opiniões

Neste capítulo são apresentados métodos de classificação de opiniões, investigados neste trabalho, com foco no nível de texto. Algumas técnicas empregam conhecimento linguístico explícito, enquanto outras se baseiam em estatística e aprendizado de máquina. As avaliações para os classificadores serão apresentadas apenas no Capítulo 6.

5.1 Classificação de Opiniões

Neste trabalho investigamos e desenvolvemos técnicas para classificar opiniões em textos em português do Brasil. Como já mencionamos, isso pode ser realizado em diferentes níveis (aspectos, sentenças e texto). Dedicamos maior atenção ao nível de textos, utilizando abordagens baseadas em léxico e em AM. Assumimos também, em princípio, um domínio específico (*reviews* de produtos) para a aplicação dos métodos desenvolvidos.

Outra decisão importante que tomamos foi a de tratar o problema de classificação de opiniões como um problema de classificação binária, positivo ou negativo, já que a classe neutra adiciona uma complexidade considerável ao problema pelo fato de haver grande

sobreposição dessa classe em relação as demais (positivo e negativo). Na literatura é possível encontrar diferentes abordagens como: considerar a classe neutra, além das classes positiva e negativa (Koppel e Schler, 2006); considerar como um problema de regressão (busca-se por uma nota que varia em uma escala de intensidade) (Pang e Lee, 2005); classificar um texto em emoções (alegria, tristeza, raiva, etc.) e não apenas em negativo ou positivo (Martinazzo et al., 2012; Dosciatti et al., 2013).

Nas seções seguintes iremos apresentar os classificadores desenvolvidos. Na Seção 5.2 é apresentado um classificador *baseline*; na Seção 5.3, um classificador baseado em léxico; na Seção 5.4, classificadores baseados em AM; na Seção 5.5, um classificador híbrido, ou seja, que é formado pela combinação dos anteriores (baseado em léxico e AM); por último, na Seção 5.6 são apresentados outros classificadores de opinião (alguns são variações dos anteriores), em que é empregado o conhecimento semântico obtido pela modelagem de palavras em um espaço vetorial, utlizando o método de Mikolov et al. (2013).

5.2 Classificador de Opiniões - *baseline*

A fim de nos fornecer subsídios a uma avaliação dos classificadores desenvolvidos, criamos um classificador *baseline*, que consiste na implementação de um método bastante simples. Este classificador informa se um texto (uma opinião) é positivo ou negativo, com base apenas nas palavras de sentimento encontradas no texto, utilizando um léxico de sentimentos. As polaridades são simplesmente somadas (valores positivos e negativos), sendo o valor final da soma o indicativo da polaridade do texto. Se a soma resultante for maior que zero, o texto é classificado como positivo; se menor ou igual a zero, como negativo.

Considere o seguinte exemplo de opinião e de funcionamento do classificador:

Ex. 1:

"Não gostei, a bateria não é muito boa"

$$\text{gostei } (+1) + \text{boa } (+1) = +2 \quad \rightarrow \quad \text{texto positivo}$$

Embora palavras de sentimento, principalmente adjetivos e verbos, sejam bons indicadores para a classificação da polaridade de um texto, é claramente perceptível a fragilidade do método ao ignorar contextos de negação ou intensificação/redução de sentimentos. O método seguinte, na Seção 5.3, explora esses fenômenos linguísticos.

5.3 Classificador de Opiniões baseado em Léxico - CBL

Desenvolvemos um classificador de opiniões baseado em léxico (Avanço e Nunes, 2014), CBL, seguindo a proposta de Taboada et al. (2011) (detalhado no Capítulo 3). Parte-se do princípio de que palavras de sentimento possuem uma polaridade a priori (determinada pelo léxico de sentimentos). Caso essas palavras ocorram em um contexto de negação ou de intensificação (ou redução), as respectivas polaridades são modificadas. Em seguida, para a sentença ou texto, é calculada a orientação semântica geral por meio da soma das polaridades encontradas, eventualmente modificadas pelo contexto.

Foram utilizados os seguintes léxicos de sentimentos, cada um deles de forma independente: OpinionLexicon (Souza et al., 2011), SentiLex (Silva et al., 2012), LIWC-sentic (Balage et al., 2013), e um subconjunto do OntoPT com polaridades (Gonçalo Oliveira et al., 2014). Para o tratamento de negação e intensificação (ou redução) são usadas as palavras listadas na Tabela 5.1.

Tabela 5.1: Conjuntos de palavras de Negação, Intensificação e Redução

Negação	Intensificação	Redução
jamais, nada, nem, nenhum, ninguém, nunca, não, tampouco	*mais, muito, demais, completamente, absolutamente, totalmente, definitivamente, extremamente, frequentemente, bastante*	*pouco, quase, menos, apenas*

Definimos como contexto uma janela de 4 palavras. Se uma palavra de sentimento estiver a uma distância de até 3 palavras à frente de uma palavra que indique negação ou

intensificação (ou redução), é modificada a polaridade original da palavra de sentimento. Esse tamanho de janela foi escolhido empiricamente; janelas de 3 ou 5 palavras produziram resultados piores.

Se uma palavra de sentimento estiver presente em um contexto em que ocorra apenas negação, sua polaridade é invertida; se ocorrer em um contexto que possui palavra de negação e de intensificação, a força da polaridade é reduzida; se se tratar de um contexto com negação e uma palavra de redução, a força da polaridade é aumentada. O fator de intensificação, sem negação, triplica a polaridade da palavra; o de redução, sem negação, divide a polaridade por 3. O fator de intensificação e redução igual a 3 foi definido utilizando-se o mesmo critério para a definição do tamanho de janela, por experimentos. No Quadro 5.1 é apresentado o algoritmo utilizado pelo classificador de opiniões.

Quadro 5.1: Algoritmo do classificador de opiniões baseado em léxico - CBL

```
1:  sentimento_texto ← 0
2:  enquanto houver palavra_sentimento no texto faça
3:      polaridade ← ler lexico(palavra_sentimento)
4:      se palavra de intensificação no contexto então
5:          se palavra de negação no contexto então
6:              polaridade ← polaridade/3
7:          senão
8:              polaridade ← polaridade * 3
9:          fim se
10:     senão se palavra de redução no contexto então
11:         se palavra de negação no contexto então
12:             polaridade ← polaridade * 3
13:         senão
14:             polaridade ← polaridade/3
15:         fim se
16:     senão se palavra de negação no contexto então
17:         polaridade ← −1 * polaridade
18:     fim se
19:     sentimento_texto = sentimento_texto + polaridade
20: fim enquanto
```

A seguir é dado um exemplo de cálculo de orientação semântica para uma sentença:

Ex. 2:

"O celular é bom, apesar da bateria não ser muito boa"

bom (+1) + não ser muito boa (+1/3) = +1.33

Há duas palavras de sentimento na sentença: *bom* e *boa*. A primeira ocorre sem influência de negação, intensificação ou redução. A segunda sofre influência de negação e intensificação combinados, o que equivale a termos apenas uma palavra de redução agindo sobre a palavra de sentimento.

Este método apresenta uma melhora significativa na correta identificação das polaridades dos sentimentos que aparecem no texto, todavia ainda assim possui algumas limitações. Podemos citar como principal fragilidade a forte dependência em relação ao léxico de sentimentos. Há o problema do léxico não ser completo, de não possuir certas palavras que carregam sentimento, e também o problema de definir polaridade para palavras que são positivas ou negativas dependendo do contexto, como por exemplo: grande, pequeno, rápido, lento, leve, pesado, etc. . Esse tipo de problema tende a não ocorrer em métodos que se baseiam em AM.

5.4 Classificadores de opiniões baseados em aprendizado de máquina - C-SVM e C-NB

Nosso ponto de partida aqui consiste no desenvolvimento de classificadores que utilizam os algoritmos de aprendizado de máquina (AM) que a literatura recente aponta como os melhores para a classificação de opinião, bem como as *features*[1] que têm se mostrado mais promissoras para essa tarefa. A partir disso buscamos acrescentar *features* relatadas em trabalhos do estado da arte e também outras que pudessem melhorar o desempenho dos classificadores atuais.

[1] atributos ou características que descrevem cada exemplo da base de dados

A abordagem de AM seguida foi a de aprendizado supervisionado, justamente a que tem sido mais comumente utilizada para identificar se uma opinião é positiva ou negativa. Para que este tipo de aprendizado seja possível é necessário que se tenha um conjunto de dados de treinamento e teste rotulados, ou seja, cada exemplo deve estar anotado com a respectiva classe à qual pertence. Em nosso cenário, precisamos de textos (*reviews* de produtos), e para cada um deles, um rótulo indicando se é positivo ou negativo. Para isso, costuma-se utilizar a própria nota dada pelo autor da *review* em relação à entidade que ele estava avaliando. É comum encontrar *websites* de *e-commerce* com uma interface que permite ao usuário atribuir uma nota, refletindo, a princípio, a opinião expressa no texto. Neste trabalho, em que temos usado o Buscapé, também temos acesso a esse tipo de informação: para cada *review* há uma nota de 0 a 5. Usamos essa informação para separar *reviews* em positivas e negativas.

5.4.1 Pré-processamento e definição de *features*

Anteriormente ao processo de extração de *features*, foram removidos dos textos sinais de pontuação e também *stopwords*[2]. Em seguida, para a composição das *features*, modelamos cada texto como uma *bag-of-words*. Como o próprio nome indica, consiste em representar um texto como uma "sacola" de palavras, ou seja, um texto passa a ser um conjunto de palavras sem importar a ordem em que originalmente elas ocorreram no texto.

A forma mais simples é representar um texto como um vetor, onde cada posição corresponde a uma palavra (ou *stem* da palavra) e o valor correspondente, binário (0 ou 1), indica a presença ou ausência do termo no texto. Porém, podem ser utilizadas outras medidas mais informativas quanto ao termo, que levam em conta informações estatísticas (Salton e Buckley, 1988). As mais comuns são a *tf - term frequency*, que corresponde à frequência com que o termo ocorre no texto; e a *tfidf - term frequency - inverse document frequency*, uma modificação da *tf*, ponderada por um fator que indica a representatividade

[2]Lista de palavras com alta frequência e "baixo valor semântico": *a(s), o(s), de, para, com*, etc.

do termo (frequência em relação a uma coleção de textos).

Nesse trabalho preferimos usar essa modelagem mantendo os itens como palavras, ou seja, não utilizamos *stemmer*[3], e também usamos apenas a informação de presença ou ausência de cada termo (0 ou 1). Essa escolha é motivada por análises comparativas de modelagens e seus resultados, como pode ser visto em Pang et al. (2002). Para a escolha de palavras que devem compor a *bag-of-words*, é feito um corte por frequência de todo vocabulário visto no conjunto de textos. Em nossos experimentos, selecionar apenas palavras com frequência maior que 4 foi o que gerou melhor resultado.

Outras *features* consideradas para o aprendizado foram: presença/ausência de palavras de negação (ver Tabela 5.1); quantidade de emoticons positivos e negativos; resultado da classificação do texto utilizando o classificador CBL (apresentado na seção anterior); quantidade de palavras positivas e negativas utilizando um léxico de sentimentos; e a quantidade de ocorrência de algumas classes de palavras (ADJ - adjetivo; ADV - advérbio; N - substantivo; e V - verbo), utilizando o *tagger nlpnet* (Fonseca e Rosa, 2013).

5.4.2 Algoritmos de aprendizado de máquina utilizados

Os algoritmos de AM utilizados foram: Naive Bayes e SVM. Ambos são bastante utilizados em classificação de opiniões (Pang et al., 2002). Naive Bayes é um classificador probabilístico que basicamente aplica o teorema de Bayes, assumindo que não há dependência condicional entre cada uma das *features*. Apesar de ser bastante ingênuo ("*naive*") ao assumir isso, é possível obter resultados bastante interessantes ao aplicá-lo, principalmente, em problemas de categorização de textos (Lewis, 1998).

SVM, de forma resumida, constrói um classificador ao definir um ou vários hiperplanos em um espaço de dimensão qualquer, por meio da otimização de uma função objetivo, maximizando as distâncias entre as instâncias (as mais próximas ao hiperplano, para cada classe) ao hiperplano. Algumas decisões importantes ao utilizar esse algoritmo para a

[3]Ferramenta que transforma as palavras reduzindo-as a uma forma comum (*stem*). Essa forma não necessariamente equivale a raiz morfológica da palavra. Por exemplo as palavras "escrever", "escrevendo", "escreve" e "escreveria", todas são mapeadas para "escrev".

construção do classificador referem-se à escolha de valores para os parâmetros que definem questões como: qual fator de penalização utilizar (interfere no tamanho da margem entre os vetores de suporte), ou qual função *kernel* aplicar para mapear as instâncias em um espaço de maior dimensão, permitindo separar de forma linear as instâncias.

No classificador que construímos utilizando SVM, escolhemos utilizar um *kernel* linear, já que, para problemas envolvendo categorização de textos, essa é, geralmente, uma boa escolha (Joachims, 1998); para problemas em que a quantidade de *features* é grande, pode não ser interessante mapear para um espaço de dimensão ainda maior (Hsu et al., 2003). Empiricamente definimos o parâmetro de penalização de erro igual a 1.0, sendo que, quanto maior o valor desse parâmetro, maior o rigor em não cometer erros de classificação, e portanto, menor a margem. Por outro lado, valores pequenos tendem a produzir classificadores com margens grandes.

5.4.3 Seleção de *features*

Algumas formas conhecidas para seleção de *features* vão desde métodos simples como a remoção de *features* com baixa variância (atributos cujos valores variam pouco nos exemplos) até a construção de classificadores que utilizam SVM ou árvores de decisão, por exemplo, para avaliar a importância de cada *feature*. Neste trabalho fizemos experimentos com três métodos: seleção por baixa variância; classificador utilizando árvores de decisão; e classificador utilizando SVM.

O melhor resultado foi obtido utilizando-se um classificador construído com SVM, estabelecendo como forma de penalização a norma L1, produzindo, assim, soluções esparsas, ou seja, muitos coeficientes estimados são iguais a zero. O que fizemos foi remover todas as *features* cujos coeficientes resultaram em zero. Com isso conseguimos reduzir bastante a dimensão do vetor de características, aproximadamente de 7.000 para 1.600, e ainda como efeito dessa redução foi observado um aumento bastante significativo na acurácia do classificador.

5.5 Classificadores de Opiniões utilizando Modelo de Espaço Vetorial

Nesta seção, apresentamos o uso de um modelo de espaço vetorial (MEV), com o objetivo de melhorar alguns dos classificadores apresentados anteriormente, principalmente no que se refere ao uso do léxico de sentimentos. Os problemas que enfrentamos ao depender de um léxico de sentimentos são: 1) o léxico pode ser muito grande, associando polaridades a palavras cuja real polaridade só pode ser determinada por contexto; 2) o léxico pode ser muito reduzido, e com isso muitas palavras que assumem polaridade no texto não são identificadas. Motivados por essas limitações, pensamos em outro modo de obter a polaridade de palavras. Utilizamos o método de Mikolov et al. (2013) para construir um modelo de representação numérica (vetorial) de palavras, capaz de capturar semelhanças sintáticas e semânticas.

MEVs são modelos bastante atrativos por serem capazes de extrair conhecimento a partir de um córpus sem nenhum tipo de anotação. MEV é um conceito que surgiu concomitantemente ao desenvolvimento de um sistema de recuperação de informação, SMART (Salton, 1971), e apesar de tradicionalmente ser aplicado no contexto de recuperação de informação, MEV tem sido usado com sucesso em diversas aplicações de PLN que envolvem medir similaridade entre palavras, *phrases* e documentos. Em análise de sentimentos, alguns trabalhos que fazem uso dessa modelagem são apresentados em Turney e Littman (2003); Velikovich et al. (2010); Maas et al. (2011); Socher et al. (2013); Alghunaim et al. (2015).

Em Turney e Littman (2003) é proposto um método para aprendizado de léxico de sentimentos, utilizando conjuntos de palavras-semente de sentimento e o método LSA (*Latent Semantic Analysis*), que obtém relações semânticas entre palavras por meio da aplicação do teorema SVD (*Single-Value Decomposition*) em uma matriz termo-documento. Nessa mesma linha, o trabalho de Velikovich et al. (2010) também propõe um método para criação de léxicos de sentimentos, por meio de um algoritmo de propagação de polaridades

em uma estrutura de grafo. O trabalho de Maas et al. (2011) descreve um modelo que utiliza aprendizado supervisionado e não-supervisionado capaz de capturar tanto informações semânticas entre termos, quanto conteúdo referente a sentimento. A proposta de Socher et al. (2013) consiste em utilizar uma rede neural recorrente, tomando como entrada a representação numérica de palavras além da árvore sintática de sentenças, para a classificação de sentenças em positivo ou negativo. Já em Alghunaim et al. (2015) é explorado MEV com o objetivo de identificar, categorizar e classificar aspectos em positivo ou negativo.

Utilizando o método de Mikolov et al. (2013), obtemos a representação vetorial de palavras do córpus Buscapé. Para obter a polaridade de palavras de sentimento, primeiramente obtemos algumas palavras-semente positivas e negativas, buscando as palavras mais similares numericamente às palavras "*ótimo*" e "*péssimo*", e calculando o cosseno entre os vetores. As dez palavras do córpus mais similares a "*ótimo*" formam o conjunto de palavras-semente positivas, e as dez mais similares a "*péssimo*", o conjunto de sementes negativas.

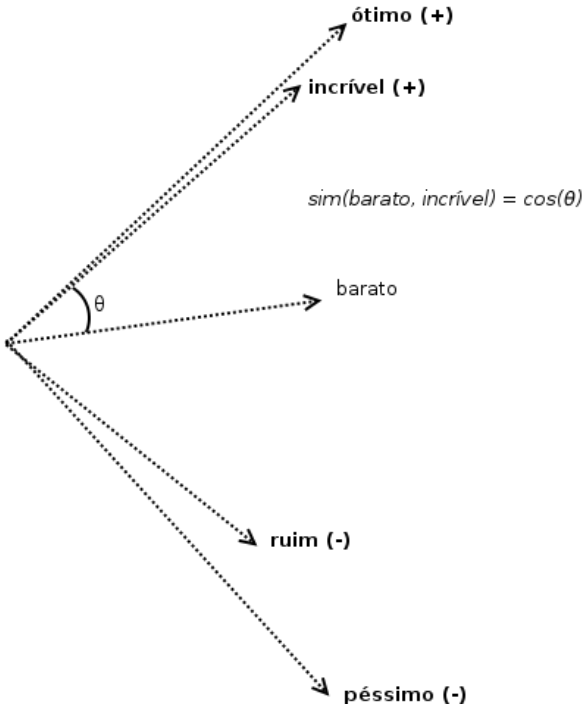

Figura 5.1: Representação vetorial de palavras e cálculo de similaridade

A Figura 5.1 exemplifica a representação vetorial de palavras-semente positivas (*ótimo* e *incrível*), palavras-semente negativas (*péssimo* e *ruim*), e uma palavra cuja polaridade deverá ser calculada, *barato*. Nota-se também na figura, que a função de similaridade entre palavras é dada pelo cosseno entre os vetores. É a diferença entre a similaridade de *barato* com sementes positivas, e a similaridade de *barato* com sementes negativas, que determina um valor de polaridade, positivo ou negativo, para a palavra *barato*.

Na expressão 5.1, a polaridade da palavra w, indicada por $pol(w)$, é calculada pela diferença entre a soma de similaridades com palavras-semente positivas, $\sum_{P_i} sim(w, P_i)$, e a soma de similaridades com palavras-semente negativas, $\sum_{N_i} sim(w, N_i)$. Os termos P_i e N_i indicam, respectivamente, uma palavra-semente positiva e uma negativa. A função $sim(x, y)$ calcula a similaridade entre as palavras x e y.

$$pol(w) = \sum_{P_i} sim(w, P_i) - \sum_{N_i} sim(w, N_i) \tag{5.1}$$

Modificamos o classificador CBL no sentido de utilizar o léxico de sentimentos apenas para a identificação de palavras de sentimento no texto. Entretanto, a polaridade indicada no léxico é ignorada, e o valor de polaridade a ser utilizado para a palavra de sentimento é calculado pela expressão 5.1. Chamamos este classificador de CBL-MEV.

Criamos ainda um novo classificador, C-MEV, que não utiliza o léxico de sentimentos para identificação de palavras de sentimento. Simplesmente removemos do texto sinais de pontuação e *stopwords* e calculamos para todas as palavras restantes uma polaridade. Todas as polaridades são somadas, e o resultado determina a polaridade de uma sentença. Da mesma forma, a soma das polaridades das sentenças resulta na polaridade do texto.

5.6 Classificador de Opiniões Híbrido - CH

Considerando que os dois principais tipos de classificadores de opiniões, baseado em léxico e em AM, possuem diferentes características, pensamos na possibilidade de combiná-los tentando obter um classificador melhor que cada um deles isoladamente. A estratégia

seguida foi de considerar primeiramente o resultado do classificador obtido com SVM e, caso o texto a ser classificado esteja a uma certa distância arbitrariamente "próxima" ao hiperplano que separa as classes, a classificação final é dada pelo classificador baseado em léxico.

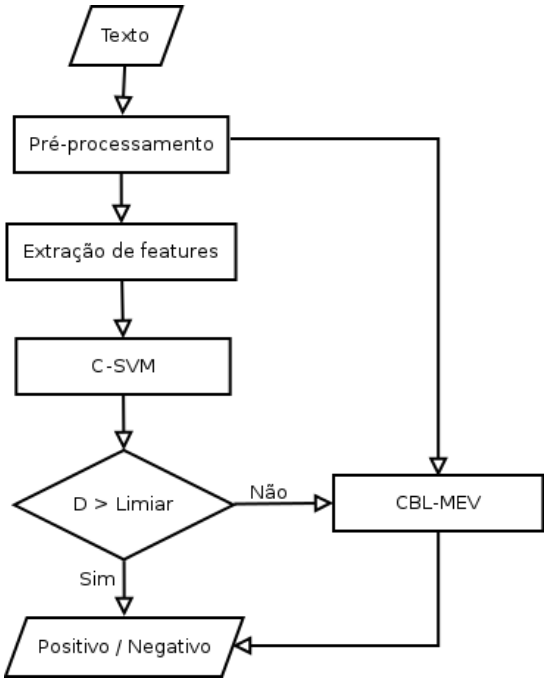

Figura 5.2: Fluxo de execução do classificador híbrido

O processo para a classificação pode ser visto na Figura 5.2. Após o pré-processamento e extração de *features*, a função de decisão do C-SVM calcula a distância (D) da instância (texto) ao hiperplano que separa as classes "Positivo" e "Negativo". Caso a instância esteja distante o suficiente do hiperplano ($D > Limiar$), a classificação final a ser considerada é a calculada por C-SVM. Se o contrário ocorrer, a classificação final é dada pelo classificador CBL-MEV.

No capítulo seguinte apresentamos a avaliação dos classificadores propostos neste capítulo.

Capítulo

6

Avaliação dos Classificadores de Opiniões

Este capítulo é dedicado à avaliação dos classificadores apresentados no capítulo anterior. Os aspectos considerados incluem os paradigmas em que se enquadram os classificadores, os córpus de referência e treinamento, os léxicos de sentimento, a normalização dos textos, entre outros.

Listamos a seguir os classificadores propostos e que serão avaliados.

- *Baseline*: utiliza léxico de sentimentos e a somatória de polaridades para classificar opinião em nível de texto.

- CBL: baseado no método de Taboada et al. (2011), utiliza léxico de sentimentos, tratamento de negação e intensificação/redução de sentimento para classificar opinião em nível de texto.

- C-SVM: utiliza AM, algoritmo SVM, para classificar opinião em nível de texto.

- C-NB: utiliza AM, algoritmo Naive-Bayes, para classificar opinião em nível de texto.

- C-MEV: utiliza um MEV (modelo de espaço vetorial) para a determinação de polaridade das palavras e classificação de opinião em nível de texto. Em contraste com CBL-MEV, este classificador não utiliza léxico de sentimentos.

- CBL-MEV: mesmo método de CBL, porém é utilizado MEV para cálculo de polaridade lexical.

- CH: classificador híbrido, combina CBL-MEV e C-SVM, para classificar opinião em nível de texto.

6.1 Córpus utilizados

A fim de avaliar a aderência dos resultados ao córpus de referência, utilizamos quatro córpus diferentes para avaliarmos a classificação de opiniões em positivo e negativo. Três deles são compostos por *reviews* de produtos, resultados de *crawling* da *web* de portais de *e-commerce*, Buscapé[1] e Mercado Livre[2]. Além disso, avaliamos também os classificadores em um córpus de domínio distinto ao nosso propósito, o córpus de avaliações de livros, ReLi, que foi apresentado no Capítulo 2 (Seção 2.3). O Quadro 6.1 resume as descrições dos córpus.

Os rótulos de positivo e negativo no córpus ReLi são resultados de anotação manual (Freitas et al., 2012). Já os de *reviews* de produtos, Buscapé-1 e Mercado Livre, são obtidos com base nas notas de 0 a 5 dadas pelos próprios autores dos textos. Com base em uma análise manual de amostras de ambos os córpus, decidimos considerar *reviews* com notas 1 e 2 como negativas, e as que possuíam nota 5, como positivas. Desconsideramos as de nota 0 após verificarmos que não correspondiam de fato a textos negativos.

O córpus Buscapé-2[3] possui duas versões, a 2A em que os rótulos são definidos em positivo, caso o autor da *review* tenha marcado o campo "*Eu recomendo*", ou negativo,

[1] http://www.buscape.com.br

[2] http://www.mercadolivre.com.br

[3] Compilação e avaliação feitas por equipe da UFPI, sob coordenação do Prof. Dr. Raimundo Santos Moura, que nos cedeu o direito de uso do córpus.

Quadro 6.1: Descrições dos córpus para avaliação dos classificadores

- Buscapé-1:
  - 6.812 *reviews* negativas
  - 6.873 *reviews* positivas
- Buscapé-2:
  - Buscapé-2A: rotulação feita pelos escritores (autores das *reviews*)
    - 2.000 *reviews* negativas
    - 2.000 *reviews* positivas
  - Buscapé-2B: rotulação feita por leitores
    - 748 *reviews* negativas
    - 1.085 *reviews* positivas
    - 71 *reviews* neutras
    - 96 *reviews* mistas
- Mercado Livre:
  - 21.499 *reviews* negativas
  - 21.819 *reviews* positivas
- ReLi:
  - 593 sentenças negativas
  - 2.852 sentenças positivas

caso o autor tenha marcado "*Eu não recomendo*"; e a 2B possui os mesmos textos que a 2A, porém os rótulos foram dados por dois leitores (anotadores) que concordaram em 80,6% dos casos. Com a anotação manual, mais duas classes surgiram, *mista* (mistura de positivo e negativo) e *neutro* (nem positivo, nem negativo), entretanto ignoramos essas duas classes, e utilizamos apenas as classes positivo e negativo.

6.2 Avaliação de classificadores baseados em léxico

Nesta seção são avaliados os classificadores que empregam o cálculo de polaridade lexical para classificar opiniões. São eles: *Baseline*, CBL, CBL-MEV e C-MEV. Os primeiros resultados, apresentados na Tabela 6.1, dizem respeito à aplicação do classificador *Baseline*, para o qual utilizamos o SentiLex como léxico de sentimentos. Esse classificador foi aplicado nos córpus: Buscapé-1, Mercado Livre e ReLi.

Tabela 6.1: Avaliação do classificador *Baseline*, utilizando o léxico SentiLex, para os córpus Buscapé-1, Mercado Livre e ReLi

Baseline	F1-Positivo	F1-Negativo	F1-Média	Acurácia
Buscapé-1	0.7011	0.5416	0.6213	0.6381
Mercado Livre	**0.7402**	**0.7351**	**0.7376**	**0.7377**
ReLi	0.3612	0.3083	0.3348	-

Analisando a Tabela 6.1, percebe-se a grande diferença entre os resultados para os córpus de *reviews* de produtos (Buscapé-1 e Mercado Livre) e o ReLi. Apesar dos tamanhos dos córpus serem bastante diferentes, o córpus ReLi parece possuir opiniões mais difíceis de serem classificadas. Observa-se também a diferença entre a medida F1 para positivo e negativo em todos os córpus listados. Não podemos afirmar, entretanto, uma possível explicação para os valores mais altos para a classe positiva, em comparação com a classe negativa, seria o fato de opiniões negativas também serem expressas com palavras positivas em contextos de negação.

Não é apresentado o valor de acurácia para o córpus ReLi, já que é um córpus desbalanceado (mais opiniões positivas que negativas - veja Quadro 6.1) e essa medida é influenciada pelos resultados da classe majoritária, o que a torna pouco confiável para avaliação do classificador. As avaliações em relação aos demais classificadores, para o córpus ReLi, também não apresentam valores de acurácia.

Conforme relatado no Capítulo 5, nosso próximo passo foi desenvolver o CBL (Seção 5.3). Na Tabela 6.2 são apresentados os resultados para o classificador CBL, para cada léxico de sentimentos utilizado, em cada um dos córpus (Buscapé-1, Mercado Livre e

ReLi).

Tabela 6.2: Avaliação do CBL para cada léxico e córpus

CBL	Léxico de Sentimentos	F1-Positivo	F1-Negativo	F1-Média	Acurácia
Buscapé-1	LIWC-sentic	0.7586	0.6430	0.7008	0.7119
	OpinionLexicon	0.7145	0.6346	0.6745	0.6794
	SentiLex	**0.7510**	**0.7210**	**0.7360**	**0.7369**
	OntoPT-sentic	0.6717	0.5478	0.6098	0.6196
Mercado Livre	LIWC-sentic	0.7295	0.6583	0.6939	0.6981
	OpinionLexicon	0.7368	0.6918	0.7143	0.7161
	SentiLex	**0.7638**	**0.7650**	**0.7644**	**0.7644**
	OntoPT-sentic	0.7080	0.6485	0.6782	0.6810
ReLi	**LIWC-sentic**	**0.6955**	**0.3200**	**0.5078**	**-**
	OpinionLexicon	0.6328	0.3564	0.4946	-
	SentiLex	0.4567	0.3330	0.3948	-
	OntoPT-sentic	0.6194	0.3391	0.4793	-

Novamente, percebemos que os resultados são melhores para a classe positiva em comparação com a classe negativa. Entretanto, é interessante notar que essa diferença é menor para o CBL que utiliza como léxico de sentimentos, o SentiLex. Isso ocorre para avaliação nos três córpus, como pode ser visto na Tabela 6.2.

Comparando os resultados obtidos com o classificador *Baseline* em relação aos produzidos pelo CBL, é bastante clara a superioridade do CBL, para todos os três córpus. O classificador *Baseline* aplicado no córpus Mercado Livre obteve 0,7376 de F1-média, já o CBL, utilizando o mesmo léxico de sentimentos, aplicado no mesmo córpus, obteve 0,7644. A diferença entre os classificadores é ainda maior quando comparamos os resultados para o córpus Buscapé e ReLi. Ao aplicar o teste de *t-student* verificamos que a diferença é estatisticamente significativa, com nível de significância (α) igual a 0,05 e *p-value* menor que 10^{-5} para os córpus Buscapé-1 e ReLi, e *p-value* igual a 0,0079 para o córpus Mercado Livre.

Em relação aos resultados para cada léxico de sentimentos, para os córpus de *reviews* de produtos, Buscapé-1 e Mercado Livre, o léxico SentiLex foi o que alcançou melhores

resultados. Porém, o mesmo não ocorreu quando avaliado no córpus ReLi, o pior resultado foi obtido com este léxico (0,3948 de F1-média).

Em seguida, conforme relatado no capítulo anterior, na Seção 5.5, propomos utilizar um modelo de espaço vetorial (MEV) na tentativa de melhorar os resultados obtidos pelo classificador CBL, e também criamos o C-MEV, que não utiliza léxico de sentimentos para a identificação de palavras de sentimento. Como pode ser visto na Tabela 6.3, os resultados obtidos pelo C-MEV, para o córpus Buscapé-1 e Mercado Livre, são melhores que os obtidos com o classificador *Baseline* (Tabela 6.1), entretanto, para o córpus ReLi, o resultado é pior.

Tabela 6.3: Avaliação do C-MEV e CBL-MEV (léxico SentiLex) para cada um dos córpus

Classificador	Córpus	F1-Positivo	F1-Negativo	F1-Média	Acurácia
C-MEV	Buscapé-1	0.5645	0.7482	0.6563	0.6809
	Mercado Livre	**0.6591**	**0.7926**	**0.7258**	**0.7421**
	ReLi	0.1847	0.3085	0.2466	-
CBL-MEV	Buscapé-1	0.8102	0.8077	0.8090	0.8090
	Mercado Livre	**0.8387**	**0.8466**	**0.8426**	**0.8427**
	ReLi	0.5862	0.3528	0.4695	-

O resultado é mais interessante para o CBL-MEV, evidenciando a necessidade do uso de um léxico de sentimentos para a identificação de palavras de sentimento, embora a polaridade lexical seja determinada com o auxílio do MEV. Fizemos experimentos utilizando apenas o léxico SentiLex, para os córpus Buscapé-1, Mercado Livre e ReLi. Analisando a Tabela 6.3, em comparação com a Tabela 6.2, vemos que o uso do MEV melhorou a acurácia do classificador em cerca de 7 pontos (de $\sim 0,74$ para $\sim 0,81$), quando aplicado no córpus Buscapé-1. Para o Mercado Livre, vemos um aumento de 8 pontos na acurácia (de $\sim 0,76$ para $\sim 0,84$). Também observamos uma melhora para o córpus ReLi, o CBL obteve o valor de F1-média igual a 0,3948, enquanto que o CBL-MEV obteve F1-média igual a 0,4695.

A diferença entre CBL e CBL-MEV é estatisticamente significativa, já que, considerando o teste de *t-student* com nível de significância (α) igual a 0,05, é obtido um *p-value*

menor que 10^{-5}.

6.3 Avaliação dos classificadores obtidos com Aprendizado de Máquina

Nesta seção avaliamos os dois classificadores baseados em Aprendizado de Máquina (AM): C-SVM (utiliza SVM) e C-NB (utiliza Naive Bayes). Verificamos tanto a influência da seleção de *features* quanto a variação do córpus (Buscapé-1 e Mercado Livre). Na Tabela 6.4 temos os resultados para ambos os classificadores.

Tabela 6.4: Avaliação do C-SVM e C-NB, com e sem seleção de *features*, para cada córpus

Classificador	Córpus	Seleção de *features*	F1-Positivo	F1-Negativo	F1-Média	Acurácia
C-SVM	Buscapé-1	SEM	0.8347	0.8278	0.8312	0.8313
		COM	0.8935	0.8886	0.8910	0.8911
	Mercado Livre	SEM	0.9306	0.9291	0.9299	0.9299
		COM	**0.9564**	**0.9557**	**0.9560**	**0.9560**
C-NB	Buscapé-1	SEM	0.8147	0.7618	0.7883	0.7916
		COM	0.8306	0.7925	0.8116	0.8135
	Mercado Livre	SEM	0.9082	0.8989	0.9036	0.9038
		COM	**0.9205**	**0.9148**	**0.9177**	**0.9178**

O que percebemos, tanto para C-SVM quanto para C-NB, sem ou com seleção de *features*, é a diferença dos resultados para o córpus Buscapé-1 e para o Mercado Livre. Nossa única explicação para a superioridade no desempenho no Mercado Livre é seu tamanho, cerca de três vezes maior que o Buscapé-1. Entretanto se olharmos para a Tabela 6.1, já percebemos que até mesmo o classificador *Baseline* se saiu muito bem classificando textos do córpus Mercado Livre, e o mesmo ocorreu para CBL (Tabela 6.2).

Outro resultado que percebemos muito claramente é o ganho obtido com seleção de *features*. Esse processo melhora o classificador em vários sentidos: reduzimos drasticamente a quantidade de *features*, o que é interessante do ponto de vista de custo computacional, e ao mesmo tempo, os valores de F1 e acurácia melhoraram muito.

É importante citar que esses resultados apresentados na Tabela 6.4 são de classificado-

res treinados levando-se em conta apenas *bag-of-words* como *feature*, e a forma de avaliação foi feita por validação cruzada com 10 *folds*[4]. Como é descrito no Capítulo 5, tentamos acrescentar diversas outras *features* que pudessem trazer mais conhecimento para facilitar a separação das classes positivo e negativo, como por exemplo: quantidade de emoticons positivos e negativos, resultado da classificação do texto utilizando o classificador CBL, quantidade de palavras positivas e negativas utilizando um léxico de sentimentos, e a quantidade de ocorrência de algumas classes de palavras (ADJ - adjetivo; ADV - advérbio; N - substantivo; e V - verbo). Entretanto, nenhuma delas trouxe qualquer ganho que melhorasse o resultado que já tínhamos com apenas *bag-of-words*. O mesmo resultado é relatado em Pang et al. (2002).

Se compararmos os resultados obtidos com CBL-MEV (classificador baseado em léxico com melhor resultado) e C-NB, ao aplicarmos o teste de *t-student* com nível de significância (α) igual a 0,05, para o córpus Buscapé-1 obtemos um *p-value* igual a 0,6210, o que nos permite concluir que não houve diferença estatística significativa. Já para o córpus Mercado Livre o *p-value* obtido é menor que 10^{-5}, ou seja, a diferença foi estatisticamente significativa. Aplicando o mesmo teste com o mesmo nível de significância para comparar os classificadores CBL-MEV e C-SVM, os valores obtidos para o *p-value*, nos resultados para os córpus Buscapé-1 e Mercado Livre, foram menores que 10^{-5}.

Entre C-NB e C-SVM, as diferenças são estatisticamente significativas, segundo o teste de *t-student* com nível de significância (α) igual a 0,05. Os valores para o *p-value* foram: menor que 10^{-5} para o córpus Buscapé-1, e igual a 0,0005 para o córpus Mercado Livre.

Devido aos diferentes desempenhos nos dois córpus, decidimos avaliar os resultados do C-SVM (melhor classificador para ambos os córpus), ao treiná-lo no córpus Buscapé-1 e testá-lo no córpus Mercado Livre, e vice-versa, treiná-lo no córpus Mercado Livre e testá-lo no córpus Buscapé-1. Os resultados são apresentados na Tabela 6.5.

Levando em conta os resultados da Tabela 6.5, e também os resultados obtidos com

[4]O método de validação cruzada consiste em particionar todo o conjunto de dados em k *folds*, ou k grupos, do mesmo tamanho. É repetido k vezes o processo, de treinar o classificador com $k-1$ grupos, e testar com um único grupo, o que não foi utilizado durante o treinamento.

Tabela 6.5: Avaliação de C-SVM quando treinado no córpus Mercado Livre e testado no Buscapé-1, e vice-versa

Córpus (Treinamento)	Córpus (Teste)	F1-Positivo	F1-Negativo	F1-Média	Acurácia
Mercado Livre	Buscapé-1	0,8377	0,8398	0,8387	0,8387
Buscapé-1	Mercado Livre	0,8412	0,8128	0,8270	0,8281

CBL-MEV, investigamos se seria possível obter um melhor resultado com o classificador híbrido, CH, que combina os melhores classificadores obtidos: C-SVM e CBL-MEV. Na seção seguinte mostramos também que CH é o classificador com os melhores resultados para a classificação de textos do córpus ReLi.

6.4 Avaliação do Classificador Híbrido

Nesta seção apresentamos os resultados para o classificador híbrido, CH, que foi criado, principalmente, com o objetivo de obter melhores resultados para casos de mudança de domínio. Para avaliá-lo fizemos os seguintes experimentos:

- (1) Combinação de C-SVM treinado com o córpus Mercado Livre e CBL-MEV utilizando o léxico SentiLex: teste de classificação no córpus Buscapé-1.

- (2) Combinação de C-SVM treinado com o córpus Buscapé-1 e CBL-MEV utilizando o léxico SentiLex: teste de classificação no córpus Mercado Livre.

Não conseguimos nenhuma melhora com a primeira configuração. Já para o segundo cenário, em que temos um classificador treinado com SVM utilizando o córpus Buscapé-1, é possível melhorar o resultado do classificador quando testado no córpus Mercado Livre. Vemos na Tabela 6.5 que C-SVM obtém 0,8270 de F1-média e 0,8281 de acurácia. Pela Tabela 6.3 também vemos que CBL-MEV, quando aplicado sob o córpus Mercado Livre, obtém 0,8426 de F1-média e 0,8427 de acurácia. Já o classificador CH obtém 0,8578 de F1-média (0,8645 de F1-Positivo; 0,8512 de F1-Negativo), e 0,8582 de acurácia.

Essas diferenças entre CH e CBL-MEV, e CH e C-SVM, são estatisticamente significativas, considerando o teste de *t-student* com nível de significância (α) igual a 0,05. Entre CH e C-SVM o *p-value* obtido é igual 0,0034, e entre CH e CBL-MEV o *p-value* é 0,0329.

Tabela 6.6: Melhores resultados para o córpus ReLi

Classificador	Córpus (Treinamento)	F1-Positivo	F1-Negativo	F1-Média	Acurácia
C-SVM	Mercado Livre	0,7455	0,4304	0,5880	-
	Buscapé-1	0,8985	0,3199	0,6092	-
CBL-MEV	-	0,5862	0,3528	0,4695	-
CH	**Mercado Livre**	**0,7607**	**0,4433**	**0,6020**	-
	Buscapé-1	**0,8745**	**0,3863**	**0,6304**	-

Também avaliamos se o classificador CH poderia melhorar os resultados obtidos para a classificação de textos do ReLi, que é, na verdade, nossa principal motivação ao propormos a construção deste classificador. A Tabela 6.6 apresenta os melhores resultados para o córpus ReLi e também o resultado obtido com o classificador CH.

Primeiramente, os resultados mostram que, em termos de F1-média, combinar um classificador baseado em aprendizado de máquina, C-SVM, com o classificador CBL-MEV, gera melhores resultados. A F1-média de C-SVM (treinado com Mercado Livre) é igual a 0,5880, enquanto que após combiná-lo com CBL-MEV o valor de F1-média foi igual a 0,6020. Observa-se a mesma melhora para o caso em que C-SVM é treinado com Buscapé-1. O valor de F1-média foi de 0,6092 para 0,6304. Esses resultados mostram que um classificador híbrido pode ser uma alternativa interessante em cenários de mudança de domínio.

Novamente, verificamos se a diferença entre os resultados é significativa, utilizando o teste de *t-student* com nível de significância (α) igual a 0,05. A diferença entre CH (C-SVM treinado no Mercado Livre e CBL-MEV) e C-SVM (treinado no Mercado Livre) não é significativa (*p-value* igual a 0,0514). As diferenças entre as demais configurações de CH e os classificadores CBL-MEV e C-SVM são significativas: *p-value* igual a 0,0292 (diferença entre CH e C-SVM), e *p-value* menor que 10^{-5} (diferença entre CH e CBL-MEV).

As seções seguintes, Seção 6.5 e Seção 6.6, apresentam avaliações de alguns fatores externos que impactam nos resultados produzidos pelos classificadores de opinião.

6.5 Avaliação do impacto da rotulação dada pelo autor na classificação

Realizamos este experimento com o objetivo de avaliar o impacto na classificação de opinião, ao assumirmos a nota dada pelo autor da *review* para a rotulação dos exemplos. Conforme descrito no início deste capítulo, utilizamos o córpus Buscapé-2 a fim de avaliar quão confiáveis são as notas dadas pelos autores das *reviews*, já que as utilizamos para separar em positivas ou negativas. Avaliamos os classificadores CBL, C-SVM e C-NB para as duas versões do córpus Buscapé-2 (Buscapé-2A e Buscapé-2B). Na Tabela 6.7 estão os resultados obtidos pelo classificador CBL para os diferentes léxicos de sentimentos, e na Tabela 6.8, para os classificadores C-SVM e C-NB. Lembramos que a versão 2B é composta por *reviews* anotadas manualmente, enquanto que a 2A mantém a recomendação (positiva ou negativa) do próprio autor.

Tabela 6.7: Avaliação do CBL, utilizando cada léxico separadamente, para os córpus Buscapé-2A e Buscapé-2B

CBL	Léxico	F1-Positivo	F1-Negativo	F1-Média	Acurácia
Buscapé-2A	LIWC-sentic	0,6875	0,5114	0,5994	0,6189
	OpinionLexicon	0,6512	0,4781	0,5646	0,5820
	SentiLex	**0,6879**	**0,6524**	**0,6701**	**0,6712**
	OntoPT-sentic	0,6103	0,4875	0,5489	0,5573
Buscapé-2B	LIWC-sentic	0,7799	0,5768	0,6783	0,7105
	OpinionLexicon	0,7272	0,4983	0,6127	0,6467
	SentiLex	**0,7751**	**0,6945**	**0,7348**	**0,7410**
	OntoPT-sentic	0,6923	0,5025	0,5974	0,6197

Vemos que, da versão 2A para a 2B, considerando todos os léxicos, os aumentos nos valores para acurácia vão de $\sim 0,06$ até $\sim 0,09$. Isso nos mostra que a nota dada pelo autor da *review* é de fato pouco confiável, já que as diferenças são expressivas nos valores obtidos de uma versão para outra. Para os classificadores C-SVM e C-NB essa diferença

também ocorre, como pode ser visto na Tabela 6.8.

Tabela 6.8: Avaliação de C-SVM e C-NB, para os córpus Buscapé-2A e Buscapé-2B

	Córpus	F1-Positivo	F1-Negativo	F1-Média	Acurácia
C-SVM	Buscapé-2A	0.8934	0.8850	0.8892	0.8894
	Buscapé-2B	**0.9325**	**0.8978**	**0.9152**	**0.9187**
C-NB	Buscapé-2A	0.8376	0.8235	0.8306	0.8309
	Buscapé-2B	**0.8686**	**0.8091**	**0.8389**	**0.8445**

Verificamos, utilizando o teste *t-student* com nível de significância (α) igual a 0,05, que as diferenças entre ambas versões de córpus, para CBL e C-SVM, são significativas. Para CBL o *p-value* obtido é menor que 10^{-5}, e para o C-SVM, o *p-value* é igual a 0,0042. Já para o classificador C-NB, as diferenças entre ambas versões de córpus não são significativas (*p-value* igual a 0,1458).

6.6 Avaliação do efeito da normalização

Nosso último experimento, cujos resultados são apresentados na Tabela 6.9, refere-se ao uso do normalizador de UGC, UGCNormal, apresentado no Capítulo 4. Nosso objetivo é avaliar quanto esse pré-processamento pode ajudar a melhorar a classificação das *reviews*, em positivo ou negativo. Utilizamos o córpus Buscapé-1 original e sua versão pré-processada pelo UGCNormal, Buscapé-1 Normalizado.

Percebemos os melhores resultados obtidos com a classificação do córpus normalizado para os classificadores que não utilizam aprendizado de máquina, *Baseline*, CBL, e CBL-MEV. Este resultado é de fato esperado, já que tais classificadores são altamente dependentes da identificação de palavras de sentimentos no texto, e em textos do tipo UGC, publicados na *web*, ocorrem com certa frequência erros ortográficos, tais como "*otimo*", "*exelente*", "*pessimo*", entre outros. Para os classificadores C-SVM e C-NB os valores são muito próximos para ambas versões do córpus. Também é um resultado esperado, pois, pelo menos intuitivamente, para essa tarefa de classificar opiniões, erros dessa natureza parecem não interferir no aprendizado, principalmente se tais erros forem frequentes.

Tabela 6.9: Avaliação dos classificadores para o córpus Buscapé-1 e sua versão pré-processada pelo normalizador de UGC

	Córpus	F1-Positivo	F1-Negativo	F1-Média	Acurácia
Baseline	Buscapé-1	0,7011	0,5416	0,6213	0,6381
	Buscapé-1 Normalizado	0,7179	0,5599	**0,6389**	**0,6562**
CBL	Buscapé-1	0,7510	0,7210	0,7360	0,7369
	Buscapé-1 Normalizado	0,7782	0,7376	**0,7579**	**0,7596**
CBL-MEV	Buscapé-1	0,8102	0,8077	0,8090	0,8090
	Buscapé-1 Normalizado	0,8208	0,8154	**0,8181**	**0,8181**
C-SVM	Buscapé-1	0,8935	0,8886	**0,8910**	**0,8911**
	Buscapé-1 Normalizado	0,8909	0,8856	0,8883	0,8883
C-NB	Buscapé-1	0,8306	0,7925	0,8116	0,8135
	Buscapé-1 Normalizado	0,8328	0,7963	**0,8145**	**0,8164**

Avaliamos se as diferenças entre as versões de córpus para cada classificador é significativa, utilizando o teste de *t-student* com nível de significância (α) igual a 0,05. Os valores de *p-value* para cada classificador foram os seguintes: *Baseline* (0,0581); CBL (0,0206); CBL-MEV (0,3224); C-SVM (0,7578) e C-NB (0,7495). Vemos, portanto, que a diferença é significativa apenas para o CBL.

Neste capítulo avaliamos todos os classificadores desenvolvidos, desde o *Baseline* até os que empregam técnicas mais sofisticadas. Comparamos as diferenças entre métodos e córpus, assim como o impacto que pode ser causado pela existência de ruídos, sejam rótulos errados, ou aqueles tratados na normalização dos textos. No próximo capítulo apresentamos nossas conclusões, assim como algumas linhas de trabalhos futuros.

Capítulo

7

Conclusões e Trabalhos Futuros

Nesse capítulo apresentamos uma visão geral sobre o trabalho desenvolvido e discutimos alguns pontos importantes que observamos nos experimentos realizados, tanto para a tarefa de normalização de UGC (*user-generated content*), quanto para a classificação de opiniões, que foram os dois tópicos de interesse investigados por este trabalho.

A normalização de UGC, no sentido de correção de ruídos (correção ortográfica, tratamento de internetês e capitalização de entidades nomeadas), é uma tarefa que cada vez mais se mostra necessária, dado o interesse crescente em processar textos publicados pelos usuários da *web* (blogs, fóruns, redes sociais). Entretanto, ao mesmo tempo em que há uma riqueza de informação muito grande disponível na *web*, o desafio em processar esse conteúdo é bastante grande.

Atacamos o problema de normalização de UGC por meio da combinação de ferramentas aplicadas em uma determinada sequência, sendo cada módulo responsável pela correção de um tipo de ruído. Destacamos o projeto e implementação de um *spell-checker* dedicado à correção sumária (sem intervenção do usuário) e particularmente voltado à correção de ruídos devidos à semelhança fonética. Até então não se conhecia trabalho

similar para língua portuguesa. A ferramenta de normalização obtida foi nomeada como UGCNormal. Conseguimos bons resultados, entretanto há ainda diversos pontos a serem explorados. Uma forma de melhorar o sistema como um todo consiste em evoluir cada módulo separadamente. Por exemplo, poderiam ser investigadas formas de tornar o *spell-checker* capaz de corrigir erros ortográficos que dependem de contexto para serem identificados (*real-word errors*). É possível também adicionar algumas heurísticas ao *spell-checker*, como a verificação de igualdade do conjunto de letras entre uma palavra errada e um candidato (p. ex.: "ótima" é um melhor candidato que "ótimo" para o erro "ótiam").

Outro tipo de ruído frequente em textos do tipo UGC, e difícil de ser corrigido, refere-se à aglutinação de palavras (p. ex.: "gosteida" → "gostei da"). Esse tipo de ruído é ainda mais frequente em textos de SMS (mensagens de celular) e os publicados no Twitter. Aliás, este seria um outro caminho a se seguir no aprimoramento do sistema UGCNormal: torná-lo capaz de lidar com textos do Twitter, que possuem características bem particulares.

Apesar das limitações, podemos afirmar que o sistema UGCNormal mostrou-se capaz de reduzir consideravelmente a quantidade de ruídos e, como consequência, permitiu que tarefas de PLN subsequentes (*tagger* e classificador de opiniões) obtivessem melhores resultados. Por meio do estudo feito dos tipos de ruídos, de quanto e como cada um deles afeta o processamento do texto, e também tendo em vista a implementação de um sistema normalizador, numa arquitetura de *pipeline*, que combina diversos módulos e recursos, respondemos a nossa primeira pergunta introduzida no Capítulo 1.

Em relação à tarefa de classificação de opinião, investigamos diversos métodos e aplicamos diferentes recursos. Inicialmente desenvolvemos classificadores de opinião utilizando as técnicas conhecidas atualmente que se mostram mais eficientes, métodos baseados em léxico e em aprendizado de máquina (AM). É um pouco difícil estabelecer o estado da arte em análise de sentimentos, mesmo restringindo a classificação de opinião ao nível de texto apenas. A grande variedade de formas de se atacar o problema e, principalmente, a

diversidade de dados (*reviews*, textos do *Twitter*, e outros) e de domínios (opiniões sobre política, hotel, restaurante, produtos eletrônicos, notícias de jornais, etc.) impõem certa dificuldade ao compararmos os resultados obtidos pelos diferentes trabalhos.

Os classificadores de opinião baseados em léxico cumpriram bem sua tarefa, alcançando valores de F1-média de 0,51 (córpus ReLi, classificador CBL) a 0,84 (córpus Mercado Livre, classificador CBL-MEV), o que é compatível com os resultados da literatura para outras línguas (quarta questão introduzida no Capítulo 1). Entretanto, esses classificadores possuem algumas limitações e não atingem os mesmos resultados alcançados por métodos que utilizam AM. Uma primeira questão importante que impacta o resultado desse tipo de classificador refere-se ao principal recurso utilizado, o léxico de sentimentos. Nossos experimentos mostraram que a qualidade do léxico impacta diretamente no resultado da classificação de opiniões, o que responde a terceira questão que propomos na introdução dessa dissertação. É uma tarefa muito complicada a construção de tal recurso, que consiste em atribuir polaridade (positiva ou negativa) a palavras isoladas, sem contexto. Tentamos superar a dificuldade de determinar polaridade lexical por meio de um modelo de espaço vetorial e, de fato, a melhora para este tipo de classificador foi bastante expressiva, ao utilizarmos esse recurso. Outra dificuldade que impacta na qualidade do classificador está na forma de tratar a negação. As formas como o fenômeno ocorre na língua não obedecem a padrões previsíveis, e merecem um estudo aprofundado.

Os melhores resultados foram obtidos com os classificadores que utilizam SVM e Naive Bayes, algoritmos que sempre produziram bons resultados na classificação de textos em tópicos ou assuntos. Quanto à definição do vetor de características, uma forma simples, mas eficiente, consiste em representar o texto como *bag-of-words*. Nos classificadores desenvolvidos utilizamos essa técnica e, embora tenhamos investigado o uso de diversas outras *features*, não observamos nenhuma que produzisse resultados melhores do que ela. Os valores de F1-média obtidos com esses classificadores variam de 0,61 a 0,95, dependendo do córpus de treinamento e de teste, sendo os piores valores referentes à classificação de textos do córpus ReLi, e os melhores, do córpus Mercado Livre. Novamente, os valores

estão em concordância com os melhores sistemas para outras línguas, o que responde a quarta questão proposta no Capítulo 1.

Numa tentativa de melhorar os resultados de um classificador quando submetido a tarefa de classificar textos de um domínio distinto ao qual foi treinado, desenvolvemos um classificador híbrido, que combina os dois melhores classificadores obtidos (de métodos distintos: AM e baseado em léxico). De fato este classificador foi o que obteve melhores resultados na classificação de textos do ReLi, o que valida nossa hipótese de que a mudança de domínio pode ser mais bem resolvida com o uso de classificadores híbridos. Esses resultados respondem as questões 5 e 6, que introduzimos no Capítulo 1.

Também avaliamos se há confiança na nota dada pelo autor da opinião (*review*), já que ela é comumente utilizada para rotular automaticamente a opinião em positiva ou negativa, e isso pode influenciar tanto a avaliação quanto o aprendizado ao utilizarmos métodos de AM. Os resultados mostraram que, de fato, elas são pouco confiáveis, entretanto, quanto maior a quantidade de exemplos (textos rotulados), menor é o impacto causado por este tipo de ruído.

Outra avaliação que fizemos refere-se ao uso da ferramenta desenvolvida, UGCNormal, numa etapa anterior à classificação das *reviews*, com o objetivo de medir o quanto o tratamento de ruídos presentes em UGC pode impactar na tarefa de classificação de opiniões. Para os classificadores baseados em léxico, percebemos uma melhora expressiva após o tratamento dos textos com a ferramenta UGCNormal, o que já era esperado dada a dependência desses métodos em relação à identificação de palavras de sentimento. Já, para os classificadores obtidos com AM, não há melhora aparente, independente do algoritmo utilizado, Naive Bayes ou SVM. A provável explicação talvez esteja no fato de a modelagem e a construção da solução obtida por aprendizado de máquina depender apenas da existência de padrões em exemplos que permitam a distinção de classes, e isso, não necessariamente, para este problema de classificar opinião, depende da expressão da opinião na forma padrão da língua. Esse resultado responde a segunda questão introduzida no Capítulo 1.

Como trabalho futuro seria interessante avaliar um método de comitê de classificadores, ou seja, um sistema que decida quanto à polaridade de uma opinião com base nos resultados de diferentes classificadores, obtidos a partir de diferentes métodos. Uma forma simples seria a de utilizar a ideia de votação por maioria, que classifica um exemplo em uma determinada classe, se essa classe foi a escolhida pela maioria dos classificadores. Outra linha a se seguir seria a de refinar a classificação de opinião, isto é, classificar opinião em nível de aspecto, o que produz uma riqueza de informação muito maior.

Realizamos alguns experimentos preliminares para a classificação de opiniões em nível de aspecto, utilizando o método de Hu e Liu (2004) para a extração de aspectos com base em frequência, e o método de Ding et al. (2008) para determinar a polaridade associada a cada aspecto. Como trabalho futuro, seria intressante investigar o uso de um modelo de espaço vetorial, tanto para as etapas de extração, categorização e agrupamento de aspectos, quanto para a fase de classificação de polaridade. Algo nesse sentido é relatado no trabalho de Alghunaim et al. (2015).

Referências Bibliográficas

Alghunaim, A., Mohtarami, M., Cyphers, S., e Glass, J. (2015). A vector space approach for aspect based sentiment analysis. In *Proceedings of the 1st Workshop on Vector Space Modeling for Natural Language Processing*, páginas 116–122, Denver, Colorado. Association for Computational Linguistics.

Aluísio, S., Pelizzoni, J., Marchi, A. R., de Oliveira, L., Manenti, R., e Marquiafável, V. (2003). An account of the challenge of tagging a reference corpus for brazilian portuguese. In *Computational Processing of the Portuguese Language*, páginas 110–117. Springer.

Anchiêta, R. T., Neto, F. A. R., de Sousa, R. F., e Moura, R. S. (2015). Using stylometric features for sentiment classification. In *Computational Linguistics and Intelligent Text Processing*, páginas 189–200. Springer.

Avanço, L. V., Duran, M. S., e Nunes, M. G. V. (2014). Towards a Phonetic Brazilian Portuguese Spell Checker. *ToRPorEsp - Workshop on Tools and Resources for Automatically Processing Portuguese and Spanish. PROPOR-2014, October, 9 2014*, páginas 24–31.

Avanço, L. V. e Nunes, M. G. V. (2014). Lexicon-based sentiment analysis for reviews of products in brazilian portuguese. *Proceedings of the Brazilian Conference on Intelligent*

Systems (BRACIS) - 2014, October 18-23, 2014, in São Carlos, SP, Brazil, páginas 277–281.

Balage, P. P., Pardo, T. A., e Aluısio, S. M. (2013). An evaluation of the brazilian portuguese liwc dictionary for sentiment analysis. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, páginas 215–219.

Balage Filho, P. P., Avanço, L. V., Pardo, T., e Nunes, M. G. V. (2014). NILC_USP: An Improved Hybrid System for Sentiment Analysis in Twitter Messages. In *Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014)*, páginas 428–432, Dublin, Ireland. Association for Computational Linguistics.

Balage Filho, P. P. e Pardo, T. A. (2013). NILC_USP: A hybrid system for sentiment analysis in twitter messages. In *Seventh International Workshop on Semantic Evaluation (SemEval 2013), 2013, Atlanta, Georgia. Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Proceedings of the Seventh International Workshop on Semantic Evaluation (SemEval 2013), 2013. v. 2*, páginas 568–572. Citeseer.

Balahur, A., Kabadjov, M., Steinberger, J., Steinberger, R., e Montoyo, A. (2012). Challenges and solutions in the opinion summarization of user-generated content. In *Journal of Intelligent Information Systems*, volume 39, páginas 375–398. Springer.

Banea, C., Mihalcea, R., Wiebe, J., e Hassan, S. (2008). Multilingual subjectivity analysis using machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, páginas 127–135. Association for Computational Linguistics.

Bick, E. (2000). The parsing system palavras. *Automatic Grammatical Analysis of Portuguese in a Constraint Grammar Framework, 505p.*

Bollen, J., Mao, H., e Zeng, X. (2011). Twitter mood predicts the stock market. In *Journal of Computational Science*, volume 2, páginas 1–8. Elsevier.

Carvalho, P., Sarmento, L., Teixeira, J., e Silva, M. J. (2011). Liars and saviors in a sentiment annotated corpus of comments to political debates. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, páginas 564–568. Association for Computational Linguistics.

Church, K. W. e Hanks, P. (1990). Word association norms, mutual information, and lexicography. volume 16, páginas 22–29. MIT Press.

Condori, R. E. L. e Pardo, T. A. S. (2015). Experiments on sentence boundary detection in user-generated web content. In *Computational Linguistics and Intelligent Text Processing - 16th International Conference, CICLing 2015, Cairo, Egypt, April 14-20, 2015, Proceedings, Part I*, páginas 227–237.

Ding, X., Liu, B., e Yu, P. (2008). A holistic lexicon-based approach to opinion mining. In *Proceedings of the 2008 International Conference on Web Search and Data Mining*, páginas 231–240. ACM.

Dosciatti, M. M., Ferreira, L. P. C., e Paraiso, E. C. (2013). Identificando emoções em textos em português do brasil usando máquina de vetores de suporte em solução multiclasse. *X Encontro Nacional de Inteligência Artificial e Computacional, Fortaleza-CE, Brasil*, páginas 1–12.

Duran, M. S., Avanço, L. V., Aluísio, S., Pardo, T., e Volpe Nunes, M. G. (2014). Some issues on the normalization of a corpus of products reviews in portuguese. In *Proceedings of the 9th Web as Corpus Workshop (WaC-9)*, páginas 22–28, Gothenburg, Sweden. Association for Computational Linguistics.

Duran, M. S., Avanço, L. V., e Nunes, M. G. V. (2015). UGCNormal: A Normalizer for UGC in Brazilian Portuguese. In *Proceedings of the ACL 2015, Workshop on Noisy User-generated Text (W-NUT), Beijing, China. Association for Computational Linguistics*, páginas 38–47.

Esuli, A. e Sebastiani, F. (2006). Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of the 5th Language Resources and Evaluation Conference (LREC 2006)*, volume 6, páginas 417–422.

Fonseca, E. R. e Rosa, J. L. G. (2013). Mac-morpho revisited: Towards robust part-of-speech tagging. In *Proceedings of the 9th Brazilian Symposium in Information and Human Language Technology*, páginas 98–107.

Freitas, C., Motta, E., Milidiú, R., e Cesar, J. (2012). Vampiro que brilha... rá! desafios na anotação de opinião em um corpus de resenhas de livros. In *ANAIS DO XI ENCONTRO DE LINGUÍSTICA DE CORPUS - ELC 2012*, páginas 1–13, Instituto de Ciências Matemáticas e de Computação da USP, em São Carlos/SP.

Gonçalo Oliveira, H., Paulo-Santos, A., e Gomes, P. (2014). Assigning polarity automatically to the synsets of a wordnet-like resource. In *3rd Symposium on Languages, Applications and Technologies (SLATE 2014) - Bragança, Portugal*, OASICS, páginas 169–184. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik.

Hartmann, N., Avanço, L., Balage, P., Duran, M., Nunes, M. D. G. V., Pardo, T., e Aluísio, S. (2014). A large corpus of product reviews in portuguese: Tackling out-of-vocabulary words. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, páginas 3865–3871, Reykjavik, Iceland. European Language Resources Association (ELRA).

Hatzivassiloglou, V. e McKeown, K. R. (1997). 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*, páginas 174–181. Association for Computational Linguistics.

Hsu, C.-W., Chang, C.-C., Lin, C.-J., et al. (2003). A practical guide to support vector

classification. páginas 1–16. Relatório Técnico - Departamento de Ciências de Computação, Universidade Nacional de Taiwan.

Hu, M. e Liu, B. (2004). Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, páginas 168–177. ACM.

Jakob, N., Weber, S. H., Müller, M. C., e Gurevych, I. (2009). Beyond the stars: exploiting free-text user reviews to improve the accuracy of movie recommendations. In *Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion*, páginas 57–64. ACM.

Jin, X., Li, Y., Mah, T., e Tong, J. (2007). Sensitive webpage classification for content advertising. In *Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising*, páginas 28–33. ACM.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. In *ECML'98 Proceedings of the 10th European Conference on Machine Learning*, páginas 137–142. Springer Berlin Heidelberg.

Kabadjov, M., Balahur, A., e Boldrini, E. (2011). Sentiment intensity: Is it a good summary indicator? In *Human Language Technology. Challenges for Computer Science and Linguistics*, páginas 203–212. Springer.

Koppel, M. e Schler, J. (2006). The importance of neutral examples for learning sentiment. In *Computational Intelligence*, volume 22, páginas 100–109. Wiley Online Library.

Krumm, J., Davies, N., e Narayanaswami, C. (2008). User-generated content. In *IEEE Pervasive Computing*, number 4, páginas 10–11.

Laver, M., Benoit, K., e Garry, J. (2003). Extracting policy positions from political texts using words as data. In *American Political Science Review*, volume 97, páginas 311–331. Cambridge University Press.

Lewis, D. D. (1998). Naive (bayes) at forty: The independence assumption in information retrieval. In *Machine learning: ECML-98*, páginas 4–15. Springer.

Liu, B. (2010). Sentiment analysis and subjectivity. In *Handbook of natural language processing*, volume 2, páginas 627–666. Chapman & Hall.

Liu, B. (2011). *Web Data Mining: Exploring Hyperlinks, Contents, and Usage Data*. 2nd ed, Springer, 624p.

Liu, B. (2012). Sentiment analysis and opinion mining. In *Synthesis Lectures on Human Language Technologies*, volume 5, páginas 1–167. Morgan & Claypool Publishers.

Liu, Y., Huang, X., An, A., e Yu, X. (2007). Arsa: a sentiment-aware model for predicting sales performance using blogs. In *Proceedings of the 30th annual international ACM SIGIR conference on Research and development in information retrieval*, páginas 607–614. ACM.

Maas, A. L., Daly, R. E., Pham, P. T., Huang, D., Ng, A. Y., e Potts, C. (2011). Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies-Volume 1*, páginas 142–150. Association for Computational Linguistics.

Maks, I. e Vossen, P. (2013). Sentiment analysis of reviews: Should we analyze writer intentions or reader perceptions? In *Proceedings of Recent Advances in Natural Language Processing (RANLP-2013), Hissar, Bulgaria*, páginas 415–419.

Martinazzo, B., Dosciatti, M. M., e Paraiso, E. C. (2012). Identifying emotions in short texts for brazilian portuguese. *IV International Workshop on Web and Text Intelligence, Curitiba-PR, Brazil*, páginas 1–12.

Mendonça, G. e Aluisio, S. (2014). Using a hybrid approach to build a pronunciation dictionary for brazilian portuguese. In *INTERSPEECH 2014, 15th Annual Conference*

of the International Speech Communication Association. ISCA. Singapore, September 25-29, 2014, páginas 1278–1282.

Mikolov, T., Chen, K., Corrado, G., e Dean, J. (2013). Efficient estimation of word representations in vector space. *ICLR Workshop - 2013; arXiv preprint arXiv:1301.3781*.

Mohammad, S. M., Kiritchenko, S., e Zhu, X. (2013). Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. In *Proceedings of the seventh international workshop on Semantic Evaluation Exercises (SemEval-2013), Atlanta, Georgia, USA, 2013*.

Moraes, F., Vasconcelos, M., Prado, P., Almeida, J., e Gonçalves, M. (2013). Polarity analysis of micro reviews in foursquare. In *Proceedings of the 19th Brazilian symposium on Multimedia and the web*, páginas 113–120. ACM.

Muniz, M. C., Nunes, M. G. V., Laporte, E., et al. (2005). Unitex-pb, a set of flexible language resources for brazilian portuguese. In *Proceedings of the Workshop on Technology on Information and Human Language (TIL)*, páginas 2059–2068.

Neto, F. A. R. e Barros, F. A. (2014). Asdp: um processo para análise de sentimento em debates polarizados. *XI Encontro Nacional de Inteligência Artificial e Computacionaln(ENIAC 2014), 2014, Sao Carlos*.

Nguyen, D. Q., Nguyen, D. Q., Vu, T., e Pham, S. B. (2014). Sentiment classification on polarity reviews: An empirical study using rating-based features. In *Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, páginas 128–135, Baltimore, Maryland. Association for Computational Linguistics.

Nunes, M. G. V. e Oliveira Jr, O. (2000). O processo de desenvolvimento do revisor gramatical regra. In *Anais do XXVII SEMISH (XX Congresso Nacional da Sociedade Brasileira de Computação)*, volume 1, páginas 1–6.

Pang, B. e Lee, L. (2005). Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, páginas 115–124. Association for Computational Linguistics.

Pang, B. e Lee, L. (2008). Opinion mining and sentiment analysis. In *Foundations and trends in information retrieval*, volume 2, páginas 1–135. Now Publishers Inc.

Pang, B., Lee, L., e Vaithyanathan, S. (2002). Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, páginas 79–86. Association for Computational Linguistics.

Picard, R. W. (1997). *Affective Computing*. MIT Press, 1997.

Prabowo, R. e Thelwall, M. (2009). Sentiment analysis: A combined approach. In *Journal of Informetrics*, volume 3, páginas 143–157. Elsevier.

Ratnaparkhi, A. et al. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the conference on empirical methods in natural language processing*, volume 1, páginas 133–142. Philadelphia, USA.

Ribeiro Jr, S. S., Junior, Z., Meira Jr, W., e Pappa, G. L. (2012). Positive or negative? using blogs to assess vehicles features. *Proceedings of the IX Encontro de Inteligência Artificial (ENIA). Curitiba-PR, Brazil.*

Riloff, E. e Wiebe, J. (2003). Learning extraction patterns for subjective expressions. In *Proceedings of the 2003 conference on Empirical methods in natural language processing*, páginas 105–112. Association for Computational Linguistics.

Rusell, R. C. (1918). US Patent 1261167 issued 1918-04-02.

Salton, G. (1971). The smart retrieval system—experiments in automatic document processing. Prentice- Hall, Englewood Cliffs, N.J., 1971.

Salton, G. e Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. In *Information processing & management*, volume 24, páginas 513–523. Elsevier.

Sharma, A. e Dey, S. (2012). A document-level sentiment analysis approach using artificial neural network and sentiment lexicons. *ACM SIGAPP Applied Computing Review*, 12(4):67–75.

Silva, M. J., Carvalho, P., e Sarmento, L. (2012). Building a sentiment lexicon for social judgement mining. In *Computational Processing of the Portuguese Language*, páginas 218–228. Springer.

Siqueira, H. e Barros, F. (2010). A feature extraction process for sentiment analysis of opinions on services. In *Proceedings of International Workshop on Web and Text Intelligence*.

Socher, R., Perelygin, A., Wu, J. Y., Chuang, J., Manning, C. D., Ng, A. Y., e Potts, C. (2013). Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, volume 1631, páginas 1642–1654.

Souza, M. e Vieira, R. (2012). Sentiment analysis on twitter data for portuguese language. In *Computational Processing of the Portuguese Language*, páginas 241–247. Springer.

Souza, M., Vieira, R., Busetti, D., Chishman, R., Alves, I. M., et al. (2011). Construction of a portuguese opinion lexicon from multiple resources. *8th Brazilian Symposium in Information and Human Language Technology - STIL*.

Taboada, M., Brooke, J., Tofiloski, M., Voll, K., e Stede, M. (2011). Lexicon-based methods for sentiment analysis. In *Computational linguistics*, volume 37, páginas 267–307. MIT Press.

Tausczik, Y. R. e Pennebaker, J. W. (2010). The psychological meaning of words: Liwc

and computerized text analysis methods. *Journal of language and social psychology*, 29(1):24–54.

Tumasjan, A., Sprenger, T. O., Sandner, P. G., e Welpe, I. M. (2010). Predicting elections with twitter: What 140 characters reveal about political sentiment. In *ICWSM*, volume 10, páginas 178–185.

Tumitan, D. e Becker, K. (2013). Tracking sentiment evolution on user-generated content: A case study on the brazilian political scene. In *Brazilian Symposium on Databases (SBBD)*, páginas 1–6.

Turney, P. D. (2002). 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*, páginas 417–424. Association for Computational Linguistics.

Turney, P. D. e Littman, M. L. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems (TOIS)*, 21(4):315–346.

Velikovich, L., Blair-Goldensohn, S., Hannan, K., e McDonald, R. (2010). The viability of web-derived polarity lexicons. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, páginas 777–785. Association for Computational Linguistics.

Vilares, D., Alonso, M. A., e Gómez-rodríguez, C. (2013). A syntactic approach for opinion mining on spanish reviews. *Natural Language Engineering*, 21(01):139–163.

Wiebe, J. e Mihalcea, R. (2006). Word sense and subjectivity. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, páginas 1065–1072. Association for Computational Linguistics.

Wiebe, J., Wilson, T., Bruce, R., Bell, M., e Martin, M. (2004). Learning subjective language. In *Computational linguistics*, volume 30, páginas 277–308. MIT Press.

Wilson, T., Wiebe, J., e Hwa, R. (2004). Just how mad are you? finding strong and weak opinion clauses. In *AAAI 2004*, volume 4, páginas 761–769.

Yang, M., Tu, W., Lu, Z., Yin, W., e Chow, K.-P. (2015). Lcct: A semi-supervised model for sentiment classification. In *Proceedings of the 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, páginas 546–555, Denver, Colorado. Association for Computational Linguistics.

Apêndice

A

Apêndice: Regras fonéticas utilizadas pelo *spell-checker*

Código	**Condição**	**Palavras corretas**	**Palavras erradas**
1	c (seguido por a, o ou u); k; qu; q	casa; quero; cobre	kasa; kero; kobre

2	ç; c (seguido por e ou i); s (inicial seguido por e ou i; final ou seguido por consoante); ss; sc (seguido por e ou i); xc (seguido por e ou i); z (final); x (seguido por consoante)	cachaça; nascer; exceção; extremo; cebola	caxassa; nasser; esseção; estremo; sebola
3	ch; sh; x (seguido por vogal)	chuva; show; peixe	xuva; xou; peiche
4	s (não inicial, seguido por vogal e não precedido por n, l ou s); x (seguido por vogal); z	tesouro; fazenda; exame	tezouro; fasenda; ezame
5	g (seguido por e ou i); j (seguido por e ou i)	monge; canjica	monje; cangica
6	e (final); es (final); i (final); is (final)	contente; pontes	contenti; pontis

7	o (final); os (final); u (final); us (final)	beijo; muitos	beiju; muitus
8	l (precedido por vogal e seguido por consoante); u (precedido por vogal e seguido por consoante); o (final)	alto; solto; fralda	altu; soltu; frauda
9	r (não inicial e seguido por vogal); rr	torre	tore
10	pi (seguido por consoante); p (seguido por consoante); pe (seguido por consoante)	opção; opinião	opição; opnião
11	d (seguido por consoante); de; di	advogado; adquirir; adicional	adevogado; adiquirir; adcional
12	ei (seguido por r, j ou x); e (seguido por r, j ou x)	queijo; leiteiro; peixe	quejo; leitero; pexe
13	g (seguido por consoante); gui	ignição	iguinição
14	b (seguido por consoante); bi	obstruir	obistruir
15	n (não seguido por vogal); m (não seguido por vogal)	tanto; também	tamto; tanbém

16	x (final ou seguido por vogal); cs; cç; quis; quiç; ques	durex; hexa; taxi; facção	durequis; hecsa; tacsi; faquição
17	a,e,i,o,u inicial; ha, he, hi, ho, hu inicial	haste; hóstia; umidade	aste; óstia; humidade
18	li; lh seguido por vogal	família; partilha	famílha; partilia
19	n; nh	companhia	compania
20	am; ão	fizeram; queriam	fizerão; querião
21	c não seguido por vogal; qui	pacto; séquito	paquito; sécto

Tabela A.1: Regras baseadas em fonética implementadas pelo *spell-checker*