Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise de sentimento em comentários sobre aplicativos para dispositivos móveis: Estudo do impacto do pré-processamento.

Lucas Braga Ribeiro

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Marcelo Ladeira

Coorientador
Dr. Thiago Veiga Marzagão

Brasília
2015

Universidade de Brasília — UnB
Instituto de Ciências Exatas
Departamento de Ciência da Computação
Bacharelado em Ciência da Computação

Coordenador: Prof. Dr. Homero Luiz Piccolo

Banca examinadora composta por:

Prof. Dr. Marcelo Ladeira (Orientador) — CIC/UnB
Prof. Dr. Flávio de Barros Vidal — CIC/UnB
Prof. Dr. Pedro Henrique Melo Albuquerque — ADM/UnB

CIP — Catalogação Internacional na Publicação

Braga Ribeiro, Lucas.

Análise de sentimento em comentários sobre aplicativos para dispositivos móveis: Estudo do impacto do pré-processamento. / Lucas Braga Ribeiro. Brasília : UnB, 2015.
82 p. : il. ; 29,5 cm.

Monografia (Graduação) — Universidade de Brasília, Brasília, 2015.

1. Análise de Sentimento, 2. Mineração de Dados, 3. Processamento da Linguagem Natural

CDU 004.8

Endereço: Universidade de Brasília
Campus Universitário Darcy Ribeiro — Asa Norte
CEP 70910-900
Brasília–DF — Brasil

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Análise de sentimento em comentários sobre aplicativos para dispositivos móveis: Estudo do impacto do pré-processamento.

Lucas Braga Ribeiro

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Marcelo Ladeira (Orientador)
CIC/UnB

Prof. Dr. Flávio de Barros Vidal
CIC/UnB

Prof. Dr. Pedro Henrique Melo Albuquerque
ADM/UnB

Prof. Dr. Homero Luiz Piccolo
Coordenador do Bacharelado em Ciência da Computação

Brasília, 07 de julho de 2015

Dedicatória

Este trabalho é dedicado a meus pais, Assunção e Rubens, e a meus irmãos, Mariana e Pedro, por toda a dedicação e empenho que tiveram para me proporcionar o suporte familiar, financeiro e emocional necessário, para que pudesse trilhar meu caminho ate aqui, nunca medindo esforços para me garantir a melhor educação possível.

Agradecimentos

Agradeço primeiramente a Deus, fonte de todas as coisas boas na minha vida, por me dar forças, sabedoria e inteligência para trilhar meus caminhos.

Agradeço aos meus pais pelo suporte sem limites que sempre me deram em todos os momentos da minha vida.

Agradeço aos amigos próximos pelos momentos de lazer que foram compartilhados.

Agradeço a compreensão daqueles a quem tive que dar menos atenção nesses últimos momentos da graduação, para me dedicar à monografia.

Agradeço ao governo brasileiro, em especial à CAPES e ao Ministério da Educação, por ter me concedido a oportunidade de cursar a graduação sanduíche na Brown University.

Agradeço ao Professor Marcelo Ladeira pelos anos de acompanhamento como orientador, se mostrando sempre dedicado, profissional e rigoroso, dando orientações que muitas vezes ultrapassavam a barreira acadêmica.

Agradeço ao Dr. Thiago Marzagão pela dedicação e apoio nesse último semestre da minha graduação, tendo um papel essencial como coorientador do meu trabalho.

Agradeço a todos os meus colegas de curso e de universidade, que me acompanharam e me ajudaram de diversas formas no decorrer da minha graduação.

Agradeço ao amigo Fernando Santos, com quem as ideias e a motivação para este trabalho nasceram.

"O coração do que tem discernimento
adquire conhecimento e os ouvidos
dos sábios saem à sua procura."

Provérbios 18:15

Resumo

Este trabalho apresenta a análise de sentimento em comentários em português e inglês e mostra os impactos do pré-processamento dos comentários nos resultados. A tarefa de identificar o sentimento expresso em um texto por seu autor é chamada análise de sentimento. Dentre as muitas fases da análise de sentimento destaca-se a etapa de pré-processamento. No decorrer do trabalho é analisado o impacto, na fase de pré-processamento do texto, da remoção de *stop-words*, remoção de repetições de letras nas palavras e pontuações, da correção de gírias e palavras escritas de maneira errada, da aplicação de uma ferramenta de *stemming* e ainda da representação do texto em unigramas, bigramas ou uma combinação de ambos. As técnicas são executadas sobre dois *corpora* com comentários sobre aplicativos móveis extraídos da *Google Play*, um contendo 2.031.480 comentários em português e outro contendo 4.843.110 comentários em inglês. É analisada, ainda, a curva de aprendizagem dos classificadores *Support Vector Machine* e *Naive Bayes* a fim de averiguar qual é a quantidade mínima de comentários para que os classificadores atinjam níveis aceitáveis de performance. Observa-se empiricamente que não existe uma sequência de pré-processamento que se destaque das demais de forma conclusiva. Averigua-se, ainda, que a remoção de *stop-words* não melhora os resultados em nenhum caso estudado, que a a representação dos atributos em Unigrama + Bigrama mostrou-se melhor que as demais quando utilizado SVM, mas não houve evidência conclusiva para este aspecto quando se utiliza *Naive Bayes*, e que uma quantidade suficiente de comentários no *corpus* para resultados satisfatórios em português varia entre 182 mil e 510 mil, e em inglês varia entre 242 mil e 871 mil, de acordo com o classificador utlizado.

Palavras-chave: Análise de Sentimento, Mineração de Dados, Processamento da Linguagem Natural

Abstract

This document shows the sentiment analysis of reviews in Porgutuese and English and shows the impacts of preprocessing the texts. The task of identifying the sentiment expressed in a text by its author is called sentiment analysis. Among many steps to perform sentiment analysis we can emphasize the text preprocessing. In this document we analyze the impact, within the text preprocessing step, of the stop words removal, the elimination of repeated characters, the spell checking and correction of misspellings and slang, the stemming technique and the role of text representation in unigrams, bigrams or a combination of both. The techniques are performed over two corpora with reviews of mobile applications extracted from Google Play, one containing 2.031.480 reviews in Portuguese and another containing 4.843.110 reviews in English. Furthermore the learning curves of Support Vector Machines and Naive Bayes classifiers are analyzed in order to verify if it is possible to determine a minimum amount of reviews that is sufficient to reach an acceptable performance. We can empirically observe that there is no sequence of text preprocessing that is better than all others in a conclusive way. Moreover, the stop words removal did not improve the results in any of the studied cases, the Unigram + Bigram representation demonstrated to be the best option when we use the SVM but there is no conclusive evidence about this aspect when we use Naive Bayes classifiers. The amount of reviews that is sufficient for the analysis in Portuguese is between 182.000 and 507.000 and in English is between 242.000 and 871.000, accordingly to the classifier used.

Keywords: Sentiment Analysis, Data Mining, Natural Language Processing

Sumário

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Com a facilidade de comunicação que a internet proporciona, as pessoas não estão mais apenas consumindo informação, mas também produzindo conteúdo, escrevendo em blogs, mídias sociais e textos em geral. Do meio dessa quantidade enorme de conteúdo produzido diariamente podemos retirar informações valiosas, dentre elas, a opinião (ou sentimento) dos autores acerca do assunto descrito em seus textos.

Segundo estimativas do website w3techs [4] a maioria dos textos em páginas na Internet é escrito em inglês, representando uma fatia de 55,4% do conteúdo indexado. Textos escritos em português representam uma fatia considerável na quantidade de conteúdo disponível na rede, em 8º lugar com um percentual de 2,5% das páginas. O Brasil, país que possui a maior população lusófona do mundo, está na 5ª posição no ranking de quantidade de internautas por país, atrás apenas da China, Estados Unidos, Índia e Japão[1].

A automação da análise do sentimento expresso por um texto acerca de determinado assunto, pessoa ou produto pode ser útil em diversas áreas como marketing, vendas e até política. O estudo da opinião expressa pelos autores em forma de texto, no âmbito da ciência da computação, é denominado análise de sentimento[19].

Este trabalho descreve um estudo de análise de sentimento em comentários sobre aplicativos para dispositivos móveis. Nele são efetuadas diversas análises, como a aplicação de diferentes técnicas de pré-processamento em textos informais e a comparação dos resultados obtidos de forma a identificar quais das etapas efetivamente impactam a análise em textos escritos em português e inglês. Tais análises nos permitem identificar se as técnicas utilizadas na literatura podem ser aplicadas em ambos os idiomas estudados, se cada idioma tem sua peculiaridade nas etapas e, uma vez identificados as reais contribuições de tais técnicas na classificação de textos informais, analisar a necessidade da aplicação das mesmas quando da análise em outras situações em *corpora* semelhantes.

Outra análise conduzida no decorrer do trabalho é acerca da quantidade de comentários

necessária na partição utilizada para o treinamento dos classificadores Como a etapa de coleta de dados é crucial e custosa na análise de sentimento, identificar uma quantidade de comentários suficiente ao aprendizado de máquina irá poupar tempo e recursos em futuras pesquisas.

A base de dados aqui utilizada, posteriormente discutida, consiste em 2.031.480 comentários em português e 4.843.110 comentários em inglês, extraídos de uma lista com 27.198 aplicativos da categoria "Jogos"disponíveis na loja de aplicativos oficial do Google, a *Google Play*. Dentre as contribuições deste trabalho inclui-se a disponibilização de tal base para futuros estudos em mineração de textos. Este trabalho também se propõe a validar a métrica de satisfação dos usuários utilizada pela loja, baseada em uma nota de 1 a 5 estrelas, utilizando uma abordagem estatística para delimitar intervalos das médias das notas em estrelas para cada classe de sentimento: positivo, negativo ou neutro.

No decorrer da pesquisa os comentários foram classificados utilizando dois modelos de classificadores: *Naive Bayes* e *Support Vector Machines*. Ambos os modelos são amplamente utilizados na literatura para realizar classificação de texto e análise de sentimento e seus resultados frequentemente superam os resultados de outras técnicas consideradas estado-da-arte[25].

1.1 Definição do Problema

Na análise de sentimento uma das etapas que mais gasta tempo computacional e de pesquisa é o pré-processamento dos dados. Averiguar a importância e a necessidade das etapas de pré-processamento usualmente aplicadas a tal tipo de texto, a fim de diminuir o tempo total necessário para a análise de sentimento, é o principal problema a ser analisado neste trabalho. Concomitantemente a essa análise, temos um objetivo secundário de identificar quantidades de comentários suficientes à etapa de treinamento dos classificadores e averiguar empiricamente se existe uma quantidade suficiente de comentários que pode ser determinada para se atingir resultados aceitáveis de uma forma geral.

Assumimos por hipótese, e iremos testar durante esta pesquisa, as seguinte assertivas:

- O pré-processamento adotado na literatura contribui para a melhoria do desempenho na classificação utilizando classificadores *Support Vector Machines* e *Naive Bayes* no domínio de análise de sentimento em comentários sobre jogos;
- As etapas de pré-processamento têm o mesmo impacto tanto para comentários em inglês quanto para comentários em português, nesse domínio;

Para testar as hipóteses será conduzida uma avaliação empírica, que não necessariamente se generaliza para todos os estudos de análise de sentimento. Ambas as inves-

tigações visam diminuir o tempo de processamento necessário para a análise como um todo. As etapas de pré-processamento que são analisadas são particularmente custosas por precisarem percorrer todo o corpus para efetuar mudanças nas palavras. O tamanho do conjunto de treinamento também influencia diretamente neste tempo, pois quanto maior o corpus, maior o tempo gasto para a sua coleta e maior a quantidade de palavras para serem processadas.

1.2 Estrutura do Documento

O presente trabalho é apresentado com a seguinte estrutura:

- Capitulo 2: Fundamentação teórica. Apresenta os conceitos teóricos das técnicas que foram necessárias ao desenvolvimento dessa pesquisa e de cujo entendimento depende a leitura desse documento.
- Capitulo 3: Metodologia. Discorre sobre a metodologia aplicada no trabalho. Cita todos os passos necessários à reprodução da pesquisa e faz um paralelo dos passos seguidos com as etapas sugeridas no modelo de referência CRISP-DM.
- Capitulo 4: Experimentação e Resultados. No decorrer deste capítulo é apresentada a justificativa de escolha de cada ferramenta bem como os resultados obtidos na analise de sentimento para ambos os idiomas.
- Capitulo 5: Conclusões. Aqui as conclusões do trabalho são expostas.
- Capitulo 6: Trabalhos Futuros. Nesta ultima seção são discutidas quais questões levantadas na pesquisa podem ser mais aprofundadas em trabalhos futuros.

Capítulo 2

Fundamentação Teórica

Esta sessão apresenta a base teórica necessária à reprodução deste estudo e ao entendimento do mesmo. Inicialmente é apresentada uma revisão de trabalhos recentes e relevantes sobre temas correlatos ao do estudo proposto. A seguir são apresentados os conceitos de mineração de dados, algoritmos de classificação, análise de sentimento, ferramentas de processamento da linguagem natural e avaliação de resultados.

2.1 Estado Da Arte

A análise de sentimento pode ser executada sobre quaisquer domínios onde a opinião de um sujeito é representada em forma de texto. Entretanto cada domínio onde as técnicas podem ser aplicadas tem suas próprias singularidades. Em um estudo pioneiro, Pang, Lee e Vaithyanathan[20] analisaram o sentimento expresso pelos autores de resenhas de filmes escritas em inglês disponíveis no site IMDb[1]. As resenhas, entretanto, eram assumidamente escritas de forma correta, seguindo as regras gramaticais do inglês, portanto o trabalho não enfrentou algumas das situações aqui encontradas como, por exemplo, a recorrência de palavras escritas de forma errada e presença de gírias.

Os autores dos comentários extraídos da loja *Google Play* frequentemente não se preocupam em seguir as regras gramaticais dos idiomas nos quais escrevem. Alguns trabalhos similares que também enfrentaram esse empecilho são estudos que utilizam dados extraídos do *Twitter* como fonte de informação [15][13][5]. Os *Tweets* também possuem gírias e palavras escritas de forma errada em seu texto, e algumas etapas no pré-processamento adotadas nos citados estudos foram importantes para determinar quais técnicas poderiam ser utilizadas no pré-processamento de nossos *corpora*.

Moore et. al.[15] , em seu trabalho, anotava a repetição de caracteres e palavras escritas em letras maiúsculas, assumindo que ambas as situações carregavam algum tipo

[1] http://www.imdb.com/

de sentimento com elas. No estudo eles também contavam as ocorrências das etiquetas de *Part-of-Speech*(POSTags) e acrescentavam essa informação à análise. Entretanto o estudo concluiu que a adição das informações de frequência das *POSTags* não contribuiu para melhoria dos resultados dos classificadores. Outros estudos conduzidos por Govardhan et. al. [13] e por Pak e Paroubek[5] executaram um preprocessamento similar em seus trabalhos antes de aplicarem os classificadores: removendo URL's, removendo repetições de letras, *stop-words* e símbolos especiais e representando os comentários como N-gramas.

Um trabalho que utilizou um corpus similar ao que aqui estudamos foi conduzido por Liu et. al. [18]. Em seu estudo os pesquisadores analisaram comentários sobre dois aplicativos para dispositivos móveis, extraindo 600 resenhas em texto para cada um dos aplicativos da *Google Play*. Sua abordagem, entretanto, foi aplicar regras baseadas em processamento da linguagem natural para classificar o sentimento expresso pelos comentários.

No que concerne a palavras erradas e gírias, Islam[14], em seu trabalho, aplicou uma etapa específica na fase de pré-processamento para lidar com essa situação em comentários extraídos da *App Store*. Em seu pré-processamento ele substitui as gírias encontradas por sua respectiva expressão equivalente encontrada no *Urban Dictionaty*[2]. Ele afirma, ainda, que o ranking adotado pela *App Store*, baseado em estrelas e semelhante ao que a *Google Play* utiliza, nem sempre reflete o real sentimento do comentário, e apresenta uma escala numérica para demonstrar o sentimento de cada resenha.

Com uma abordagem mais geral sobre as características dos comentários em lojas de aplicativos, Hoon et. al. [10] conduziram uma análise estatística sobre 8 milhões de comentários extraídos da loja de aplicativos para dispositivos *Apple*, a *App Store*. No estudo os autores levaram em conta aspectos como tamanho do comentário e as diferenças na atribuição de estrelas no decorrer do tempo de vida de cada aplicativo. Dentre seus achados, afirmam que os aplicativos passam a receber resenhas mais curtas à medida que envelhecem, e que cerca de metade dos aplicativos analisados tiveram uma queda no seu ranking em estrelas no decorrer do tempo.

2.2 Mineração de Dados

O crescimento explosivo na quantidade da dados produzidos diariamente fez necessária a criação de novas técnicas e ferramentas para a análise automática de tais bancos de dados. Tais ferramentas e técnicas são estudadas no campo de descoberta de conhecimento em bancos de dados (do inglês *Knowledge Discovery in Databases* - KDD). No processo de KDD, a etapa responsável em determinar métodos para definir padrões nos dados e em

[2] http://www.urbandictionary.com/

seguida buscar padrões de interesse é denominada mineração de dados [23]. A mineração de dados utiliza técnicas de vários outros campos para conseguir atingir seus objetivos, notadamente das áreas de inteligencia artificial, banco de dados e estatística.

2.2.1 Aprendizagem de máquina

Algoritmos que identificam padrões são classificados de acordo com seu tipo de aprendizagem. Nas aplicações de aprendizagem de máquina para classificação se destacam dois tipos de aprendizagem: o aprendizado supervisionado e o não-supervisionado. Simon [24] define em seu trabalho que aprendizado de máquina é qualquer mudança em um sistema que melhore o seu desempenho de forma automática em uma posterior repetição da mesma tarefa ou em outra tarefa utilizando a mesma base. O campo que estuda as formas que as máquinas aprendem a identificar padrões é denominado aprendizagem de máquina (do inglês *machine learning*)

Aprendizado supervisionado

No aprendizado supervisionado utiliza-se de uma massa de dados com a finalidade de 'ensinar' a máquina quais são os padrões que ela deve conseguir identificar. Se os padrões possuírem valores discretos (como classes de sentimento positivo, negativo ou neutro) o problema se torna uma classificação. Quando a base possui valores contínuos (por exemplo escalas variando de 0 a 10), o problema é categorizado como uma regressão.

Algumas situações acerca das características desse conjunto de aprendizado devem ser levadas em conta, como seu tamanho em relação ao todo, a confiança da sua classificação e o balanceamento das diferentes classes presentes no conjunto. Essas questões serão discutidas na sessão que trata de classificadores.

Aprendizado não-supervisionado

No tipo de aprendizado não-supervisionado não existe a figura da base de aprendizado. Neste tipo de situação tenta-se aprender baseando-se unicamente nas relações presentes no conjunto estudado. Um tipo de aplicação que geralmente utiliza este tipo de aprendizado é a clusterização. Na clusterização a meta do estudo é agrupar os dados que são similares entre si. Pra tal, não necessariamente existe um conjunto prévio de *clusters*, mas o algoritmo deve identificar a quantidade de grupos similares e apresentar como saída a qual destes grupos cada uma das entradas pertence.

2.2.2 Classificadores

Classificadores são algoritmos e ferramentas que classificam entradas de dados em classes discretas. Neste trabalho apresentamos e utilizamos os classificadores denominados *Support Vector Machine* e *Naive Bayes*.

Support Vector Machine

Um *Support Vector Machine* é um classificador baseado em aprendizagem supervisionada: aquele que aprende a classificar os dados de acordo com um conjunto previamente etiquetado. Seu funcionamento se baseia na construção de um hiperplano com a maior distância possível separando duas classes no espaço vetorial estudado e foi proposto inicialmente por Cortes e Vapnik [8]. Os elementos, para serem classificados utilizando este modelo, devem ser representados em forma de vetores. No caso da mineração de texto os textos são representados como *bags-of-words*.

Para ilustrar o funcionamento dos *Support Vector Machines* suponhamos que em uma base de treino, com vetores de duas dimensões, existam círculos e quadrados, conforme imagem 2.1. O classificador irá analisar os dados e traçar uma reta que melhor separa as duas classes de dados. O exemplo é uma forma simplificada de mostrar tal separação, pois em casos reais é comum a quantidade de dimensões ultrapassarem as centenas de milhares, impossibilitando um entendimento gráfico mas mantendo a relação vetorial.

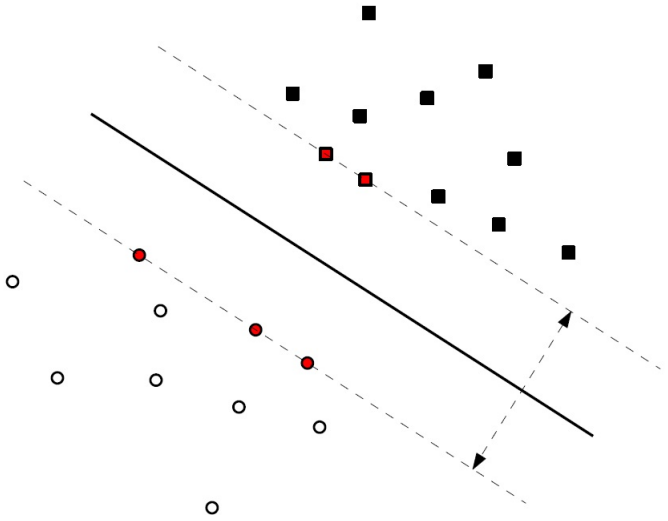

Figura 2.1: Separação em um hiperplano no SVM.

É compreensível que em alguns casos não seja possível separar linearmente os dados do grupo de treino. Para tal utilizamos uma função ϕ para mapear os vetores para uma

dimensão de ordem maior, tornando-os separáveis. À essa função ϕ damos o nome de kernel. A figura 2.2 ilustra o funcionamento do kernel de um *Support Vector Machine*.

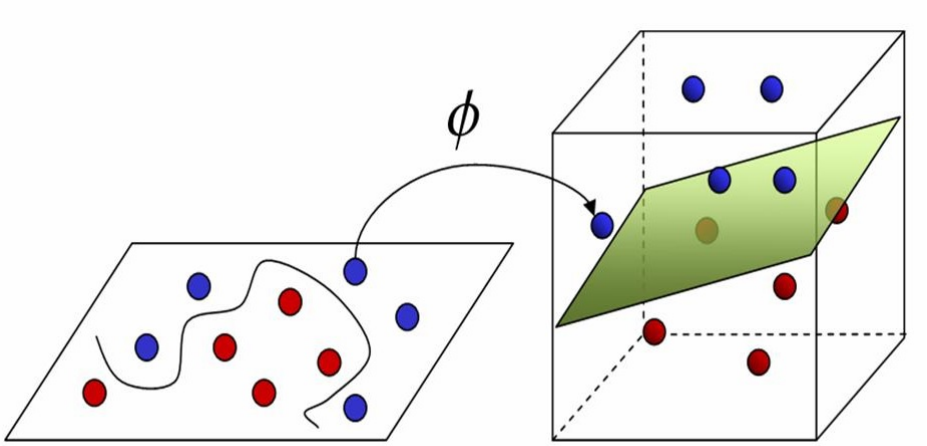

Figura 2.2: Função ϕ mapeando os vetores para uma dimensão de ordem superior.

Entretanto, mesmo com a utilização do artifício de se aumentar a dimensão estudada, alguns pontos se destacam por estarem muito distantes dos demais da sua mesma classe. Esses pontos são chamados *outliers*. Quando definimos o modelo a ser utilizado no *Support Vector Machine* devemos levar em conta a generalização deste modelo: ele deve funcionar bem não apenas na base de aprendizado, mas também na validação e situações reais.

Podemos, de forma geral, descrever um modelo que identifique 100% dos membros de uma classe, inclusive os *outliers*, mas que não será um modelo com generalização suficiente para classificar novas entradas da mesma base de dados. Dá-se o nome de *overfitting* para a situação onde o modelo está implementado de tal forma que funciona muito bem para a base de treino mas não generaliza a resolução do problema. Para evitar tal situação existe um parâmetro na implementação do SVM que define pesos para pontos classificados de maneira errada, geralmente descrito como C. Para um C de valor alto, o valor de penalização por erro será bastante alto, aumentando os resultados de precisão na base de teste mas levando o classificador a se aproximar do *overfitting*.

Naive Bayes

O *Naive Bayes* é um classificador que, apesar de razoavelmente simples, têm mostrado um desempenho notável nas tarefas de mineração de texto e análise de sentimento [25]. É um algoritmo probabilístico que se baseia no teorema de Bayes, apresentado na equação

2.1, e é chamado ingênuo por assumir que as variáveis são independentes entre si, o que a rigor seria uma hipótese errada na análise de textos.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{2.1}$$

Em aplicações de mineração de texto, quando desejamos classificar um documento em determinada classe, a equação de Bayes pode se traduzir na probabilidade de uma classe c para um documento d, que se traduz na equação 2.2.

$$P(c|d) = \frac{P(d|c)P(c)}{P(d)} \tag{2.2}$$

A tarefa de um classificador *Naive Bayes* consiste em calcular e identificar a classe $c_{NB} \in C$ que maximiza o resultado da equação 2.2, onde C representa o conjunto de possíveis classes em que o documento d pode ser classificado. Tal relação é explicitada utilizando a notação presente na equação 2.3.

$$c_{NB} = \arg\max_{c \in C} \frac{P(d|c)P(c)}{P(d)} \tag{2.3}$$

Dado que $P(d)$ é constante sobre todas as possíveis classes, pode ser eliminado da equação, resultando na equação 2.4.

$$c_{NB} = \arg\max_{c \in C} P(d|c)P(c) \tag{2.4}$$

A probabilidade $P(d|c)$ de um documento d pertencer à classe c pode ser expressa como a probabilidade de cada um de seus atributos dada a classe c: $P(x_1, x_2, \ldots, x_n|c)$. Portanto podemos reescrever a equação 2.4 da seguinte forma:

$$c_{NB} = \arg\max_{c \in C} P(x_1, x_2, \ldots, x_n|c)P(c) \tag{2.5}$$

Entretanto, como discorrido anteriormente, o classificador *Naive Bayes* assume que as variáveis do documento são condicionalmente independentes dado c. De posse dessa informação, podemos concluir a relação presente na equação 2.6.

$$P(x_1, x_2, ..., x_n|c) = P(x_1|c) * P(x_2|c) * ... * P(x_n|c) \tag{2.6}$$

Finalmente podemos chegar à equação que baliza o classificador *Naive Bayes* na escolha da classe a qual um documento d pertence, apresentada na equação .

$$c_{NB} = \arg\max_{c \in C} P(c_j) \prod_{x \in X} P(x|c) \tag{2.7}$$

2.2.3 Representação e seleção de atributos

Classificadores utilizam atributos (ou características) dos dados analisados para identificar padrões que os distinguam dentre as classes propostas. Ambos os classificadores apresentados nesta sessão utilizam vetores como formato dos dados de entrada. Veremos a seguir as definições de *bag-of-words*, n-grama, TF-IDF e da representação matriz termo x documento.

N-grama

Ao tentar representar as informações contidas em um texto podemos dividir o texto em sequências de palavras que serão utilizadas como atributos na etapa de classificação. Um N-grama é uma sequência de N palavras que aparece em um texto. Na literatura os valores de N usualmente utilizados são 1 (unigrama), 2 (bigrama) e 3 (trigrama). Essa variação se justifica por que alguns sintagmas necessitam de mais de uma palavra para terem seu significado completo, como 'pé de cabra', e algumas construções mudam completamente seu significado quando próximas a outras palavas, como 'gostei' e 'não gostei'.

A Figura 2.3 mostra um exemplo de como as diferentes representações de N-gramas se comportam sobre a mesma frase. As diferentes representações terão sua eficiência testadas durante este trabalho.

Matriz Termo x Documento

Uma estrutura frequentemente utilizada para representar os atributos em uma mineração de texto é a matriz termo x documento. A matriz é uma representação matemática dos atributos (termos) presentes em um documento e seus valores. Na representação em matriz cada elemento $m_{i,j}$ significa o valor agregado ao atributo que representa o termo de índice j no documento de índice i, como é mostrado a seguir:

$$M = \begin{bmatrix} m_{documento1,termo1} & m_{documento1,termo2} & m_{documento1,termo3} & \cdots \\ m_{documento2,termo1} & m_{documento2,termo2} & m_{documento2,termo3} & \cdots \\ m_{documento3,termo1} & m_{documento3,termo2} & m_{documento3,termo3} & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{bmatrix}$$

Sendo assim, a classificação ocorrerá sobre uma estrutura com ixj valores. Dá-se o nome de *bag-of-words* para a representação do documento nesta forma de vetor, correspondente a cada linha na matriz apresentada.

Uma vez que a quantidade de documentos analisados i chega a 4 milhões e que a quantidade de atributos j pode chegar a 1,5 milhão, o seu armazenamento inicialmente pode ser considerado um problema. Se fosse necessário guardar o valor para cada atributo,

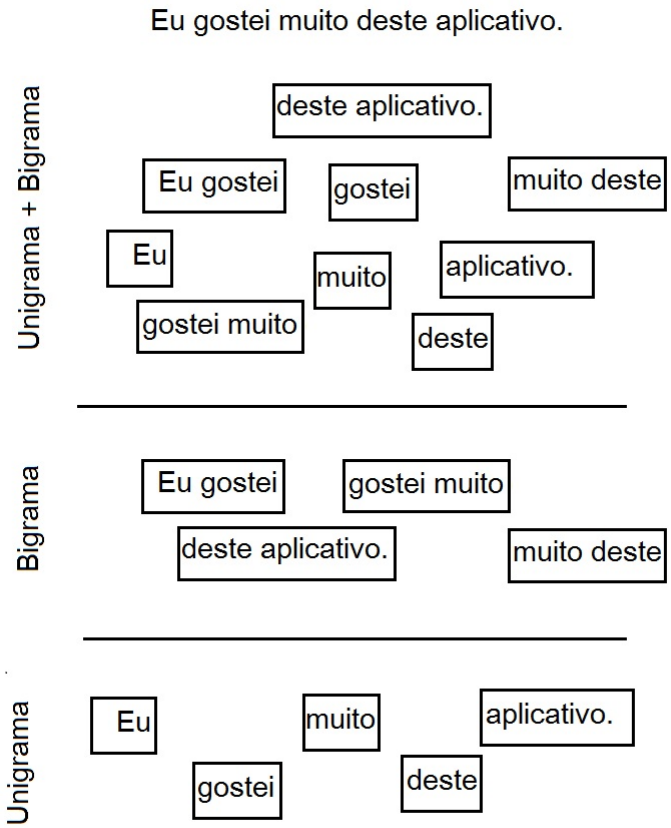

Figura 2.3: Extração de n-gramas.

assumindo um valor inteiro de 4 bytes, teríamos que 4.000.000 * 1.500.000 * 4 bytes = 24 000 gigabytes necessários para armazenar esta estrutura.

Entretanto a matriz termo x documento é extremamente esparsa, pois cada documento (no caso desta análise, um comentário) possui um numero pequeno de atributos (palavras) se considerarmos o total de 1,5 milhão presente na matriz. Utilizamos esta característica para representar a estrutura na forma de matriz esparsa. O suporte para esse tipo de matriz é provido pelo módulo Scipy[3] no Python.

Os valores dos atributos do texto variam de acordo com a modelagem da classificação selecionada. Três abordagens que podemos destacar são as que seguem:

- **Presença do termo -TP (*Term presence*)**: Recebe um marcador binário que representa a presença do termo j no documento i, com o valor 1 caso esteja presente e 0 caso não esteja presente.

[3] http://docs.scipy.org/doc/scipy/reference/sparse.html

- **Frequência do termo - TF (*Term frequency*)**: O valor $m_{i,j}$ recebe a frequência com que o termo j aparece no mesmo documento i.
- **Frequência ponderada - TF-IDF (*Term Frequency - Inverse Document Frequency*)**: No TF-IDF o elemento $m_{i,j}$ recebe o valor da frequência de j em i depreciada pelo inverso da frequência da palavra em todos os documentos, de acordo com a fórmula a seguir:

$$TF - IDF = TF * Log(N/DF)$$

onde N representa o número total de documentos.

Esta medida leva em conta a capacidade de descrição do termo em relação ao documento. Um termo que é muito frequente em todos os documentos do corpus não é um bom descritor, e recebe portanto um valor baixo ao ser dividido por sua frequência alta. Um termo que aparece em poucos documentos tende a ser um descritor mais eficaz e recebe um valor mais alto por ser divido por sua frequência baixa.

2.2.4 *K-fold cross validation*

O tamanho da base de dados utilizada na fase de treino impacta diretamente a análise dos resultados e o desempenho de uma tarefa de mineração de dados.

Quando a base não tem um tamanho suficiente, a divisão em coleção de teste e coleção de treino pode ser desfavorável à aprendizagem do classificador, por tornar os conjuntos ainda menores. Em situações análogas a esta, recomenda-se utilizar a técnica de *K-fold cross validation*[22]. A técnica, chamada também de validação cruzada, consiste na divisão da base completa de dados em K conjuntos (*folds*). Feita a divisão, executam-se K turnos de treinamento e validação, onde em cada turno um conjunto diferente é eleito para se tornar conjunto de validação e as outras $K - 1$ partições são utilizadas na etapa de aprendizagem. Ao fim, um resultado mais realista da capacidade de classificação do algoritmo é a média do desempenho de cada um dos turnos executados. A Figura 2.4 exemplifica um caso onde utiliza-se $K = 3$.

Durante o trabalho, esta noção de precisão que as várias execuções no *K-fold cross validation* nos dá será utilizada para gerar margens de erro para os valores de *F-Measure* retornados pelos classificadores.

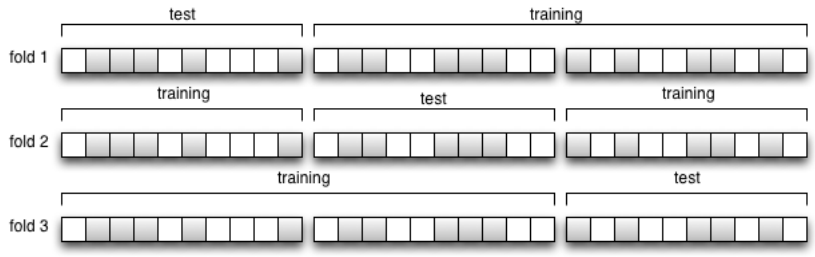

Figura 2.4: Aplicação da técnica de k-fold cross validation com k=3.

2.3 Análise de sentimento

A análise de sentimento é um campo multidisciplinar que envolve áreas como ciência da computação, linguística, estatística e até psicologia. De maneira geral tem por objetivo analisar o sentimento expresso pelo autor de um texto em seu conteúdo. Quando a intenção é identificar o sentimento expresso em uma classe específica, como positivo ou negativo, é tida como uma tarefa de classificação[17].

A análise pode ser feita em diversos níveis, destacando-se na literatura os níveis de sentença, documento ou aspecto (características)[6]. Na análise por sentença busca-se identificar a orientação do sentimento que a sentença expressa, frequentemente antes classificando a sentença como opinativa ou apenas descritiva. Sentenças que não carreguem uma opinião não são levadas em consideração na análise. Na análise a nível de documento busca-se classificar todo um documento de acordo com o sentimento que por ele é expresso. O documento em geral pode conter diversas opiniões e sentimentos no decorrer do seu conteúdo, mas busca-se atribuir um sentimento geral a ele. Por fim, de maneira bem mais específica, existe a análise de sentimento a nível de aspectos. Nesse nível a análise busca identificar a opinião do autor sobre determinadas características de um assunto, e não de forma geral. Por exemplo, na análise baseada em aspectos não deseja-se identificar se falam bem ou mal de um carro, mas sim qual o sentimento expresso acerca dos seus pneus, da sua economia de combustível e do conforto de seu interior.

Existem, em geral, três abordagens para se efetuar a análise de sentimento em textos: abordagens baseadas em aprendizagem de máquina, em análises léxicas ou em análises linguísticas. Nas abordagens baseadas em aprendizagem de máquina treina-se um algoritmo com exemplos previamente classificados, buscando atributos que melhor classifiquem cada classe. Em uma abordagem baseada em análise léxica se utiliza um grupo de palavras cujo sentimento foi previamente anotado. Um algoritmo classifica então um documento baseado na presença de tais palavras no seu texto. Um exemplo de ferramenta que é útil nesse tipo de abordagem é o SentiWordNet[7]. A ferramenta disponibiliza, para cada palavra, 3 níveis de sentimento, variando de 0 a 1, que representam sua positividade, negatividade

e objetividade (neutralidade). Caso a abordagem seja voltada a uma análise linguística, as estruturas das sentenças e suas funções sintáticas são levadas em consideração durante a classificação, para tentar identificar padrões que melhor descrevam textos positivos e textos negativos. Nesse caso muitos estudos utilizam de ferramentas de processamento da linguagem natural, como *POSTaggers*.

2.4 Processamento da linguagem natural

Para se comunicar umas com as outras, as pessoas utilizam a linguagem escrita, a linguagem falada e até mesmo linguagens de sinais. Diferentemente da comunicação entre computadores, que utilizam uma linguagem precisa e concebida especificamente para este fim, tais linguagens apresentam ambiguidades e estruturas com significados implícitos como anáforas e elipses. Quando o interesse é estabelecer uma comunicação entre homem e computador, precisamos utilizar técnicas para que a máquina possa trabalhar com a linguagem utilizada pelas pessoas. A área de estudo que visa estabelecer uma comunicação entre a linguagem do homem e a linguagem da máquina denomina-se processamento de linguagem natural (ou linguística computacional) e é um campo que converge conceitos da ciência da computação, da inteligência artificial e da linguística.

Para efetuar tal processamento, utiliza-se de diversas ferramentas que extraem informações de texto baseadas em regras dos idiomas no quais estão escritos. Dentre essas ferramentas destacam-se os tokenizadores, os etiquetadores sintáticos e os lematizadores (*stemmers*), que serão descritos nas sessões seguintes.

2.4.1 Tokenizador

Um tokenizador é uma ferramenta que separa o texto de acordo com o objeto que será estudado. Tais objetos, em análise de sentimento e mineração de texto, geralmente são as palavras de um texto, mas uma ferramenta que separe letras ou até frases também pode ser considerada um tokenizador, se esse for o objeto de estudo desejado. A a ferramenta utilizada foi implementada em Python no *Natural Language Tooklit* - NLTK[2].

2.4.2 Lematizador (*Stemmer*)

A técnica de lematização, mais referenciada como *stemming*, se baseia na redução das palavras em seu morfema. Um morfema (*stem*, em inglês), ou radical, é a menor parte com significado de uma palavra, portanto, no processo de *stemming*, palavras como casa, casas, casinhas, casebre e casarão resultam no mesmo morfema: cas.

2.5 Avaliação dos resultados

Precisamos definir métricas para avaliar a performance das classificações que serão efetuadas. As medidas de Acurácia, Precisão, Recall e *F-measure* são as mais comumente utilizadas na literatura e terão seu funcionamento descrito a seguir.

As avaliações utilizam informações de uma estrutura conhecida como tabela de contingência. A tabela traça uma relação entre o valor efetivamente calculado pelos classificadores e o valor real da classificação. Um exemplo de tabela de contingência é mostrado na Tabela 2.1.Compreendemos da tabela que os valores que o classificador efetivamente acerta são os verdadeiro-positivos e verdadeiro-negativos.

Tabela 2.1: Exemplo de tabela de contingência

Classificação\Valor Real	**Positivo**	**Negativo**
Positivo	Verdadeiro-Positivo(VP)	Falso-Positivo(FP)
Negativo	Falso-Negativo(FN)	Verdadeiro-Negativo(VN)

2.5.1 Acurácia

A medida de acurácia serve para quantificar os acertos efetuados pelo classificador de forma geral. A medida de acurácia é calculada pela Fórmula 2.8 e assume um custo igual para todos os tipos de erros. Um valor de acurácia, sozinho, não pode ser balizador de uma análise. Tomemos por exemplo uma base de dados que contenha 95% de membros da classe 1 e 5% de membros da classe 2. Um classificador que classifique todos os membros como classe 1 terá 95% de acurácia. Entretanto este classificador não terá classificado nenhum elemento da classe 2 corretamente.

$$\text{Acurácia} = \frac{VP + VN}{VP + FP + FN + VN} \tag{2.8}$$

2.5.2 Precisão

A medida de precisão calcula um valor para a quantidade de documentos corretos dentre os documentos classificados como corretos. A fórmula para o cálculo é apresentada na Equação 2.9

$$\text{Precisão} = \frac{VP}{VP + FP} \tag{2.9}$$

2.5.3 *Recall*

A medida de *Recall* (também chamada de sensibilidade) calcula a quantidade de documentos que deveriam ter sido classificados como positivos e efetivamente o foram. A fórmula para o cálculo é dado de acordo com a Equação 2.10.

$$\text{Recall} = \frac{VP}{VP + FN} \tag{2.10}$$

2.5.4 *F-Measure*

A medida mais comumente utilizada para a comparação de resultados entre diferentes classificadores é a medida de *F-measure*, também denominada na literatura como *F-score* ou *F1-score*. Por utilizar em sua fórmula as medidas de Precisão e *Recall* torna a análise mais abrangente. Quando aplicada a classificações binárias pode ser entendida como a média ponderada dos valores de *Recall* e Precisão. A fórmula é apresentada na Equação 2.11.

$$F_\beta = \frac{(1 + \beta^2) * \text{Precisão} * \text{Recall}}{\beta^2 * \text{Precisão} + \text{Recall}} \tag{2.11}$$

Percebemos que o parâmetro β, na fórmula, baliza a importância dos pesos para o *recall* e para a precisão. Um valor de $\beta > 1$ retornará uma média que atribui mais peso ao valor de *Recall*, e um valor de β tal que $0 < \beta < 1$ dá mais peso à medida de Precisão. Esta liberdade permite a adaptação do indicador para diferentes tarefas de classificação, de acordo com suas especificidades.

O valor do parâmetro β mais utilizado é β=1, quando a equação é então denominada *F1-score*. Percebemos, pela Equação 2.12 que neste caso ambas as medidas de *Recall* e Precisão têm o mesmo peso sobre o valor final do *F1-score*.

$$F_1 = \frac{2 * \text{Precisão} * \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{2.12}$$

Capítulo 3

Metodologia

Nesta sessão serão apresentados os passos percorridos durante a pesquisa e que são necessários para sua reprodução. O trabalho consiste em uma tarefa de mineração de dados e, portanto, balizou-se de forma geral nas etapas apresentadas pelo modelo de referência *Cross-Industry Standard Process for DataMining* - CRISP-DM, apresentada a seguir.

3.1 CRISP-DM

A metodologia CRISP-DM[26] sugere um ciclo de fases que podem ser seguidos na resolução de problemas, de forma independente da indústria cujos dados são relacionados. A metodologia é apresentada na forma de um modelo hierárquico de processos, apresentando 4 níveis de abstração (do mais genérico para o mais específico): fases, tarefas genéricas, tarefas especializadas e instâncias de processos. A figura 3.1[1] apresenta a sequência e relação das fases da metodologia, que em tradução livre são: entendimento do negócio, entendimento dos dados, preparação dos dados, modelagem, avaliação e utilização. As fases e suas aplicações são explicadas a seguir.

O entendimento do negócio (*Business Understanding*) se refere à etapa inicial da mineração de dados. Essa fase foca no entendimento do problema sob uma ótica negocial e então na conversão desse conhecimento em uma definição de um problema de mineração de dados. É elaborado um projeto focado em atingir determinados objetivos, também elencados nessa etapa, e são desenvolvidos cronogramas e listas de requisitos que poderão ser necessários durante toda a sequência do projeto. Durante esta etapa foram estudados os comentários dos aplicativos, a loja de aplicativos, a organização em categorias e o tipo de ranking utilizado para classificar as resenhas.

[1] www.blue-granite.com/blog/bid/281766/Advanced-Analytics-Introduction-to-Data-Mining

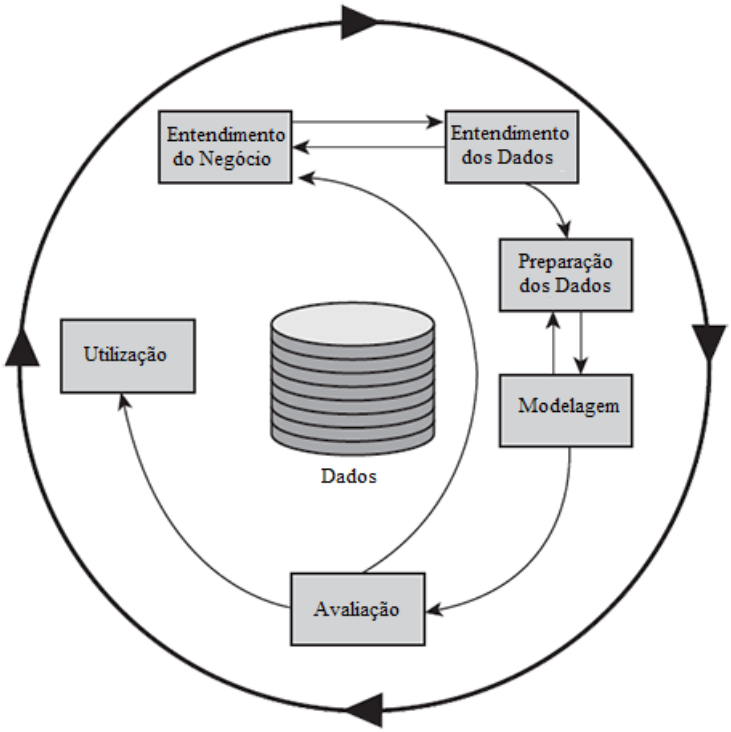

Figura 3.1: Descrição das fases do CRISP-DM.

A fase de entendimento dos dados (*Data Understanding*) se inicia com uma coleta inicial dos dados que serão utilizados e sua análise preliminar, de forma a gerar uma familiarização da equipe com o seu objeto de estudo. Como os dados são a peça primordial de toda a análise, esta etapa pode ser feita em paralelo ao entendimento do negócio, visto que os objetivos estão intrinsecamente ligados à disponibilidade e qualidade dos dados que são coletados. Durante essa fase deve ser determinado o tamanho da coleta dos dados e feito um relatório contendo a descrição e qualidade das informações disponíveis para estudo. Nesta fase foi definida uma quantidade considerada suficiente à nossa análise e foi realizado um estudo sobre uma coleta inicial, consideravelmente menor que o *corpus* final, para que pudéssemos identificar as características dos textos informais que seriam estudados.

A etapa de preparação dos dados (*Data Preparation*) cobre todas as etapas até a finalização da base de dados que alimentará a análise. A etapa é muitas vezes a mais demorada do processo e requer múltiplas alterações nos dados originais. Em mineração de texto é esta a etapa responsável pelo pré-processamento do texto para que o mesmo esteja disponível da melhor maneira possível para as ferramentas de classificação. Durante esta etapa todas as ferramentas de processamento de linguagem natural apresentadas foram aplicadas, gerando 96 diferentes *corpora* de cada idioma para serem analisados.

A modelagem (*Modelling*) é a fase em que as técnicas e ferramentas de mineração são

efetivamente aplicadas sobre os dados. Nesta etapa são escolhidos os melhores valores para os parâmetros das ferramentas, para que executem de forma otimizada. São experimentadas diversas técnicas aqui, e muitas vezes é necessário retornar à fase de preparação dos dados para ajustes na base para adaptação a técnicas que estão sendo testadas. Durante a pesquisa, nesta etapa foi efetivamente realizada a análise de sentimento utilizando classificadores baseados em *Naive Bayes* e em *Support Vector Machines*. Os parâmetros foram testados e foram identificados valores para serem utilizados em todas as análises posteriores. Foram testados todos os *corpora* obtidos com os diversos pré-processamentos e testados diferentes tamanhos de base de treino.

Avaliação (*Evaluation*) é a etapa onde todo o processo até aqui deve ser reavaliado, juntamente com os seus resultados parciais obtidos, para que tenha-se a certeza de que nenhum aspecto do negócio foi deixado para trás ou não se está distanciando dos objetivos inicialmente propostos que devem ser atingidos.

Por fim, a fase de utilização (*Deployment*), é onde o cientista de dados organiza e apresenta as suas conclusões e dados para um analista do negócio estudado. É uma fase que depende exclusivamente do escopo do projeto, mas pode incluir a aplicação do modelo proposto a outras bases de dado ou apenas a apresentação dos resultados e conclusões ao solicitante de forma a auxiliar em sua tomada de decisões. Aqui os dados foram organizados em forma de gráficos e planilhas e analisados, para que as conclusões aqui expostas pudessem ser atingidas. Nas sessões seguintes as etapas seguidas no decorrer da pesquisa são descritas em maiores detalhes.

3.2 Análise preliminar

A primeira etapa do projeto consistiu no entendimento do problema e dos dados a serem colhidos. Foi feito um levantamento preliminar do estado-da-arte acerca da área de análise de sentimento. Após definido o escopo e objetivos, procedeu-se com a coleta dos dados, extraídos da loja de aplicativos do *Google*.

3.3 Coleta

Tendo sido definido o problema e o escopo, procedeu-se com a coleta dos dados. Foram colhidos comentários sobre aplicativos de dispositivos móveis, utilizando a adaptação de um *crawler*, o GoogleMarketAPI[2]. A coleta utilizou mais de uma máquina, para que a coleta em ambos idiomas pudesse ocorrer de forma paralela, e ocorreu entre março e maio de 2015, sendo a etapa que mais consumiu tempo durante o estudo.

[2]https://code.google.com/p/android-market-api/

Ao fim da coleta foram definidos dois *corpora* distintos, o Corpus-PT e o Corpus-EN, respectivamente com comentários em português e inglês.

3.4 Pré-processamento

Uma vez definidos os *corpora* que estudaríamos, procedemos com a aplicação das diferentes técnicas de pré-processamento nos mesmos. As técnicas foram aplicadas de maneira gradual, de um modo que fosse possível identificar possíveis contribuições da técnica ao desempenho dos classificadores.

Nesse momento do trabalho os seguintes processamentos foram utilizadas:

- Remoção de repetições de letras.
- Correção de palavras e gírias.
- *Stemming*
- Remoção de *Stop-words*

3.4.1 Remoção de repetições

Autores de textos nas redes sociais muitas vezes tentam expressar seus sentimentos se valendo de repetições de letras em palavras de seus textos[15]. A Figura 3.2 mostra um exemplo de como essa situação aparece nos comentários aqui estudados.

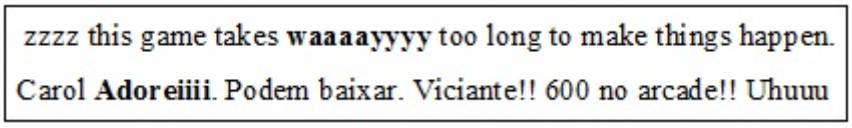

Figura 3.2: Exemplo de repetição de letras em comentários.

A eliminação de tais repetições contribui para a redução das dimensões da *bag-of-words*, pois, por exemplo, após o processamento as palavras "Nooooo"e "Nooo"resultarão ambas em "Noo". Entretanto, assumindo que essa repetição carrega algum tipo de sentimento, eliminar as repetições não contribuiria para identificar melhor a posição do autor do texto. Como em nossa análise faremos testes com e sem esta etapa, teremos uma visão real da aplicação dessa remoção no nosso escopo.

3.4.2 Correção de palavras e gírias

Textos em redes sociais possuem muitas palavras escritas de maneira errada e muitas vezes uma linguagem específica repleta de gírias e neologismos não abordados pelas fer-

ramentas de processamento da linguagem natural, que em geral são desenvolvidas para trabalhar com textos assumidamente escritos em forma correta. Hoon et. al. [11] argumenta que, em seu estudo, apenas 10% das palavras presentes em seus 8.7 milhões de comentários sobre aplicativos extraídos da *App Store* estavam escritos de acordo com as regras gramaticais da língua inglesa.

Para contornar tal situação, foi incluído no pré-processamento uma etapa de correção automática dessas palavras escritas de maneira incorreta. Foi, ainda, construído um dicionário de gírias e abreviações, contendo cerca de 1000 itens para cada idioma, que foi utilizado para tratar casos de gírias e palavras não reconhecidas pelo dicionário.

3.4.3 Lematização (*Stemming*)

A etapa de *stemming* consiste na aplicação de uma ferramenta de lematização em todas as palavras do corpus. A técnica consiste na divisão da palavra em radical e terminação, e na eliminação da terminação.

Um exemplo de funcionamento de uma ferramenta de *stemming* é apresentado na Figura 3.3. Tal técnica é valiosa para a redução das dimensões da *bag-of-words* e reduz várias variações de um mesmo morfema para uma única representação na *BoW* aumentando, assim, seu peso nas classificações. Entretanto é uma etapa custosa computacionalmente e suprime o significado real das palavras para futuras análises.

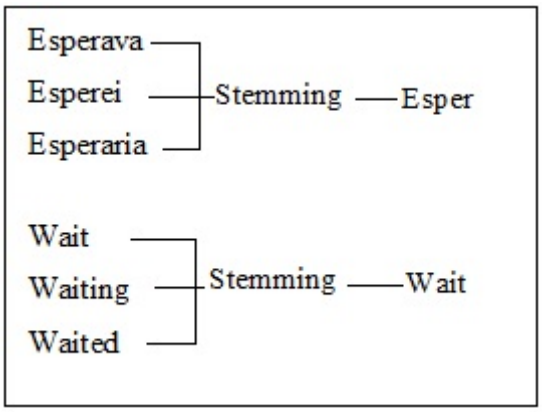

Figura 3.3: Exemplo de Stemming, em Portugês e Inglês.

3.4.4 Remoção de *Stop-words*

Na literatura sobre processamento da linguagem natural é comum a retirada de *stop-words* quando se utiliza ferramentas com alguma base em estatística. As *stop-words* são palavras cuja sua frequência no idioma é tão alta que, em teoria, a sua presença na análise pode ser descartada pois representaria valores parecidos independentemente da classe em

que determinado texto está inserido. Um exemplo para *stop-word* no idioma inglês é o artigo "The". Em português, os artigos definidos o, a, os e as são exemplo de *stop-words* que podem ser removidas da análise.

3.5 Aplicação de mineração de dados

Os 96 *corpora* criados a partir da aplicação das técnicas de pré-processamento foram então submetidos à classificação utilizando *Support Vector Machines* e *Naive Bayes*. Cada etapa gerou também uma margem de erro para seus valores de *F-measure* retornados, baseada nas iterações do *Kfold cross validation*, para que os valores pudessem ser comparados com alguma margem de confiança.

Os comentários são apresentados com uma nota em estrelas que varia de 1 a 5. Entretanto a análise que desejamos fazer se baseia em 3 classes: positivo, negativo ou neutro.

Com o intuito de traçar uma relação entre estrelas e sentimento foi executada uma tarefa de classificação manual de uma porção considerável de comentários e com isso um intervalo de médias para cada classe de sentimento foi definido. Com base nessas informações foram definidos classes de sentimento para cada nota em estrela.

Após as comparações a classificação do melhor resultado para cada idioma foi refeita variando o tamanho da base de treinamento do corpus, a fim de identificar qual seria a quantidade comentários necessária para estabilizar o valor do *F-measure* para cada idioma.

Capítulo 4

Experimentação e resultados

Este capítulo descreve a experimentação que foi realizada para atingir os objetivos propostos no início do documento. A experimentação se baseou na classificação de diversos *corpora*, compostos pelo *corpus* original seguido de diferentes pré-processamentos. Todos os dados utilizados para a construção dos gráficos aqui apresentados estão disponíveis no apêndice A desde documento.

4.1 Entendimento e descrição dos dados

Os dados utilizados neste trabalho são compostos de comentários sobre aplicativos para dispositivos móveis com sistema operacional Android. Tais resenhas foram retiradas da loja oficial de aplicativos do Goog le, a *Google Play*[1]. A loja de aplicativos foi lançada em 2008, inicialmente chamada de *Android Market*, e em março de 2012 foi batizada de *Google Play*. Os dados referentes aos anos pretéritos a 2012 continuaram na plataforma. Dados[3] apontam que a *Google Play* já possui mais de 1.500.000 aplicativos disponíveis e mais de 50 bilhões de downloads.

A loja é a via oficial para compra e download de aplicativos para os dispositivos, e permite que os usuários escrevam comentários sobre os aplicativos que adquiriram, bem como classificá-los em um ranking que varia de 1 a 5 estrelas, onde 1 representa a pior nota e 5 a mais alta. Os aplicativos coletados para o trabalho contemplam resenhas escritas no período de 2008 a 2015.

Os aplicativos na loja são classificados em categorias como Produtividade, Educação, Música e Jogos, entre outros. Para analisar apenas aplicativos de um mesmo domínio, onde os temas discutidos tratam dos mesmos assuntos e características, foi decidido focar o escopo do trabalho à análise de comentários de aplicativos na categoria de Jogos. Marcello Lins, em seu trabalho[16], disponibiliza um arquivo com informações de 1.1 milhões de

[1] https://play.google.com/store

aplicativos, incluindo nome, categoria e desenvolvedor, mas não os comentários. Tais informações foram utilizadas para criar uma lista contendo 27.198 nomes de aplicativos na categoria Jogos, e os comentários dessa lista foram extraídos, juntamente com suas notas em estrelas.

A *Google Play* permite que os usuários escrevam seus comentários no idioma que desejarem. Nesse trabalho, durante a etapa de coleta, optamos por restringir nossos comentários aos idiomas português e inglês. O português é a língua materna dos envolvidos no trabalho e a crescente quantidade de internautas e conteúdo escritos em português justifica a escolha desse idioma. Ademais, as ferramentas e estado-da-arte de análise de sentimento utilizam dados em inglês, e a escolha de também coletar dados nesse idioma se justifica para que fosse possível comparar os resultados com outros estudos na área. O filtro de idioma foi uma das adições feitas ao *crawler* utilizado, o *Android Market API*[2].

4.1.1 *Corpora*

A etapa de coleta dos dados, ocorrida entre março e abril de 2015, resultou em dois *corpora* contendo comentários em português e em inglês sobre aplicativos de Jogos da *Google Play*. Tais corpora são descritos e discutidos a seguir:

Corpus em português

A etapa da coleta de dados se iniciou focando em comentários escritos em português sobre os aplicativos presentes na lista com 27.198 nomes descrita anteriormente. Esta etapa nos levou ao *corpus*, doravante denominado Corpus-PT, contendo 2.031.480 comentários sobre aplicativos da categoria Jogos. A distribuição das estrelas nos comentários é apresentada na Tabela 4.1.

Tabela 4.1: Distribuição das estrelas em Corpus-PT

Estrelas	**Número total de comentários**	**%**
1	365675	18.01
2	92458	4.55
3	192195	9.46
4	225517	11.10
5	1155635	56.88

A Figura 4.1 nos mostra a quantidade de comentários em cada uma das classificações em estrelas. Observamos uma predominância nas notas mais altas pros aplicativos.

[2] https://code.google.com/p/android-market-api/

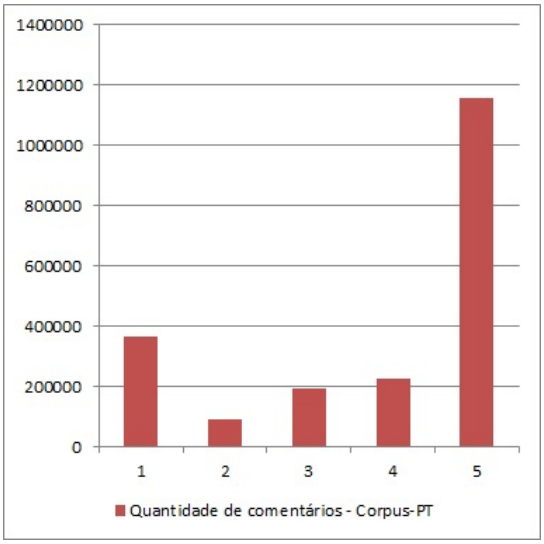

Figura 4.1: Comentários do Corpus-PT de acordo com estrelas.

Corpus em inglês

A mesma lista com nomes aplicativos foi utilizada para coletar comentários no idioma inglês, resultando no *corpus* que é denominado Corpus-EN. Como esperado, a quantidade de conteúdo disponível nesse idioma foi maior que a encontrada em português e possibilitou a coleta de 4.843.410 comentários. A Tabela 4.2 nos dá uma visão quantitativa da distribuição dos comentários entre as 5 classificações de estrelas.

Tabela 4.2: Distribuição das estrelas em Corpus-EN

Estrelas	**Número total de comentários**	**%**
1	812338	16.78
2	233933	4.81
3	423868	8.76
4	762086	15.74
5	2610885	53.91

Na Figura 4.1 podemos observar que a quantidade de comentários com mais estrelas é maior que as quantidades com menos estrelas, em consonância com o que foi identificado no Corpus-PT.

Como podemos observar nas Figuras 4.1 e 4.2 as curvas das quantidades de comentários em relação às estrelas tem um formato semelhante, acumulando mais comentários na classe com 5 estrelas e menos comentários nas classes intermediárias, de 2 e 3 estrelas. Em um estudo focado em resenhas de produtos físicos comprados no site Amazon[3] os pesquisadores Hu, Pavlou and Zhang[12] propõem que o formato dessa curva, *J-shaped*,

[3] http://www.amazon.com/

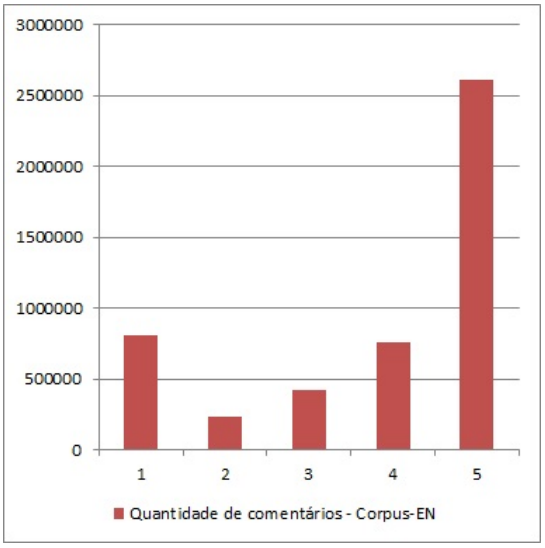

Figura 4.2: Comentários do Corpus-EN de acordo com estrelas.

é recorrente neste tipo de análise e se justifica pelo fato de consumidores que compram um produto têm uma maior tendência a opinar positivamente sobre o que adquiriram, e que consumidores com sentimentos moderados, entre positivo e negativo, tendem a não manifestar suas opiniões.

Uma análise adicional que pode ser feita é acerca da associação entre a distribuição das classes dos comentários em português e em inglês. Ao realizarmos um teste de χ^2 nos valores referentes às classes de estrelas dos comentários, expresso na Figura 4.3, verificamos que o valor de p-value $< 0,05$ nos indica que há associação entre as distribuições em estrelas das duas coleções de comentários.

```
> data.table
      [,1]    [,2]
r1  365675  812338
r2   92458  233933
r3  192195  423868
r4  225517  762086
r5 1155635 2610885
> chisq.test(data.table)

        Pearson's Chi-squared test

data:  data.table
X-squared = 26006, df = 4, p-value < 2.2e-16
```

Figura 4.3: Teste de relação entre Corpus-PT e Corpus-EN .

4.2 Preparação dos Dados

Uma vez coletados todos os dados que serviriam como base para este trabalho, procedeu-se com a etapa de preparação dos dados. Nesta etapa foram utilizadas técnicas e ferramentas de processamento da linguagem natural para criar diversos corpora diferentes de modo que o resultado da classificação em cada um deles pudesse ser analisado e comparado com os demais, assim identificando melhores estratégias para futuras pesquisas.

Nessa sessão os seguintes processamentos, discutidos em seguida, foram utilizadas:

- Remoção de repetições de letras.
- Correção de palavras e gírias.
- Stemming
- Remoção de Stop-words

4.2.1 Remoção de repetições

Foi identificado, na etapa de análise dos dados, que muitas palavras nos comentários apresentavam repetições de letras na sua escrita. Esta etapa consistiu na remoção destas repetições.

A quantidade máxima de repetições para cada letra foi fixada em 2 repetições, pois ambos os idiomas possuem palavras com esse tipo de dígrafo. Nesta etapa a repetições de pontuação, como "!!!"ou "???"também foi suprimida.

4.2.2 Aplicação dos dicionários e corretores ortográficos

Outra característica recorrente dos textos coloquiais é a presença de palavras escritas de maneira errada. Com o intuito de corrigir tais erros ortográficos e deixar o *corpus* o mais próximo possível de um texto escrito de acordo com as regras gramaticais, um script percorreu cada palavra dos comentários e fez uma consulta no dicionário disponibilizado pelo módulo PyEnchant [4] para verificar sua ortografia.

O dicionário, que dá suporte a ambos os idiomas estudados, identifica e informa se a palavra está escrita de maneira correta ou incorreta e, caso incorreta, sugere uma lista de correções. Para a correção das palavras, por padrão, escolheu-se a primeira sugestão de correção apresentada pelo PyEnchant. Entretanto algumas palavras não são reconhecidas pelo módulo, sendo identificadas como incorretas mas não possuindo uma sugestão de correção.

[4] http://pythonhosted.org/pyenchant/

Uma análise dessas palavras sem sugestão de correção mostrou que uma considerável parte delas era composta por gírias e linguagem utilizada pelos internautas. Essas palavras, apesar de não estarem presentes em dicionários formais dos idiomas estudados, possuem um significado e podem ser importantes na classificação. Foi então construída uma lista com cerca de 1000 palavras erradas, gírias e abreviações mais frequentes, para cada idioma, que não foram corrigidas pelo dicionário. Esse conjunto de palavras foi corrigido uma a uma, manualmente, e assim foi criado um dicionário de gírias e abreviações. Após o dicionário PyEnchant ter sido aplicado nos corpora o dicionário de gírias foi aplicado logo em seguida. Uma das contribuições deste trabalho é a disponibilização desta lista de palavras para futura utilização em outras pesquisas.

4.2.3 Aplicação do lematizador (*stemmer*)

Nesta etapa os *corpora* são submetidos às ferramentas de *Stemming*. Neste trabalho as ferramentas utilizadas foram, em ambos os idiomas, as disponíveis no NLTK.

4.2.4 Remoção de *stop-words*

A etapa de remoção de *stop-words* se deu no momento da extração dos atributos, utilizando a função TfidfVectorizer[5] do Scikit-learn. Um dos parâmetros da função é uma lista contendo as *stop-words* que se deseja excluir da análise. Não existe uma lista fechada contendo todas as *stop-words* para um idioma. Para esta tarefa utilizamos a lista disponível no NLTK. Em português a lista contém 203 palavras e em inglês é composta por um total de 127 palavras.

4.3 Criação dos *corpora*

Com o objetivo de verificar quais etapas e técnicas são efetivamente vantajosas e necessárias à classificação dos comentários, aplicamos as técnicas de forma sequencial nos Corpus-PT e Corpus-EN. Foram, finalmente, criados 192 *corpora* para cada idioma, para ser analisado pelos classificadors. Os aspectos que foram testados para gerar os diferentes *corpora* são descritos na Tabela 4.3

Para uma melhor disposição dos resultados obtidos, seguiremos utilizando abreviações para descrever os *corpora* gerados. Uma lista com todas as abreviações é mostrada na Tabela 4.4. Por exemplo, o *corpus* que foi gerado do Corpus-PT, utilizando Unigrama + Bigramas para representar os atributos, mantendo suas *stop-words*, que teve, nessa

[5] http://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html

Tabela 4.3: Técnicas e etapas testadas.

Etapas e Técnicas	**Casos testados**
Atributos	Unigrama, Bigrama ou Unigrama + Bigrama
Pré-processamento	Remoção de Stopwords, Stemming, Remoção de repetições e Correção de Palavras e Gírias
Classificadores	Support Vector Machine e Naive Bayes

sequência, retiradas as repetições de letras, aplicada correção de palavras e gírias e então aplicado o stemming será identificado como: unibi-sw-ptrw-nr-cr-st.

Tabela 4.4: Tabela com as abreviações das técnicas aplicadas para a criação dos corpora.

Etapas e Técnicas	**Casos testados**
uni	Unigrama
bi	Bigrama
unibi	Unigrama + Bigrama
ptrw	Português 'raw'. O Corpus-PT sem alterações
enrw	Inglês 'raw'. Corpus-EN sem alterações.
cr	Correção de palavras erradas e gírias.
st	Stemming
nr	Retirar repetições de letras.
sw	Corpus com Stop-words
nosw	Corpus sem Stop-words

4.4 Modelagem e classificação

Para classificar os comentários estudados neste trabalho foram definidas 3 classes de sentimento: positivo, negativo e neutro. Dois modelos de classificadores, *Naive Bayes* e *Support Vector Machine*, foram eleitos para analisar tal classificação, de acordo com sua relevância na literatura[25], para que os resultados e conclusões pudessem ser comparados com outros trabalhos. Ambos os classificadores utilizam uma representação do texto em *bag-of-words* e foi utilizado, na matrix termo-documento, o esquema de valores TF-IDF.

Os comentários, em sua representação original, são classificados em um ranking variando de 1 a 5 estrelas por seus autores. Hoon et. al.[11] afirma que, em seu estudo em comentários da *App Store*, os comentários que foram classificados entre 1 e 3 estrelas possuem uma predominância de sentimentos negativos, enquanto os comentários classificados entre 3 e 5 estrelas tendem a uma opinião mais positiva.

Para determinar melhor a relação entre cada classe de estrelas e o sentimento que seus comentários refletiriam conduzimos uma análise escolhendo, de forma aleatória e estrati-

ficada de acordo com distribuição de estrelas, uma quantidade relevante de comentários em ambos os corpus e classificando-os manualmente.

Um total de 4168 comentários em inglês e 4001 comentários em português foi classificado de forma cega, sem se saber a classificação em estrelas, entre positivo, negativo ou neutro, a fim de identificar quais as médias do ranking em estrelas presente em cada classe de sentimento

Com essas informações foi possível calcular um intervalo de média das estrelas para cada uma das classes, com um intervalo de confiança de 95%. Os intervalos das médias podem ser verificados na Tabela 4.5. De posse dessa informação, determinamos que uma boa classificação para relacionar a quantidade de estrelas com o sentimento expresso é considerar comentários com 1 ou 2 estrelas como expressando sentimentos negativos, comentários com 3 estrelas como representando neutralidade e comentários com 4 ou 5 estrelas como representando comentários positivos.

Tabela 4.5: Intervalo de médias de estrelas para cada classe.

Classificação	**Corpus-PT**	**Corpus-EN**
Positivo	[4,35, 4.44]	[4.26,4.36]
Negativo	[2.72, 2.83]	[3.00, 3.09]
Neutro	[1.41, 1.49]	[1.94, 2.02]

4.4.1 Comparação dos resultados

É esperado que o valor do *F1-measure* para cada classificação varie levemente de acordo com qual massa de dados é escolhida como massa de teste e massa de treino. Para tentar amenizar qualquer erro induzido pela escolha da massa de treino (que, apesar de ser escolhida aleatoriamente, impacta na performance do classificador) decidimos por utilizar a técnica de *K-fold cross validation*, que usualmente é utilizada em classificações onde a massa de dados é pequena, para gerar um intervalo de valores do *F1-measure* atingidos.

Cada classificação foi feita utilizando-se um $K = 5$, conforme o *cross validation*, variando-se os conjuntos de treino e teste. Os resultados para cada rodada foram anotados e foi então calculado um intervalo com 95% de confiança da média dos valores de *F1-score* retornados pelos classificadores. Esta abordagem foi tomada por, inicialmente, algumas classificações retornarem valores extremamente próximos, com diferenças de 0.001% entre suas performances. Julgou-se acertado que tomar decisões e fazer análises em variações tão pequenas não era uma boa estratégia, então utilizamos o intervalo de confiança adquirido na etapa do *cross validation* para ter dados mais confiáveis e trabalhar em cima de margens de erro.

4.4.2 Implementação dos classificadores

Os classificadores baseados em *Support Vector Machines* e *Naive Bayes* são amplamente utilizados na literatura de análise de sentimento e classificação de texto[25]. Para conduzir nossa analise utilizamos a ferramenta Scikit-learn [21]. O kit oferece implentações de diversos classificadores, incluindo SVM e *Naive Bayes* e ferramentas de extração de atributos.

Para a classificação baseada em *Support Vector Machines* utilizamos a implementação do classificador LinearSVC[6]. É uma implementação de classificador com kernel linear e utiliza a biblioteca liblinear[9] para efetuar a classificação. A classificação baseada em *Naive Bayes* foi feita utilizando-se o MultinomialNB[7].

4.5 Impactos das técnicas

Esta sessão apresentará os resultados e discutirá o impacto das diversas técnicas de pré-processamento sobre os valores de *F-Measure* retornados pelos classificadores. Os resultados serão apresentados de acordo com o idioma dos corpora utilizados.

4.5.1 Corpus-PT

O Corpus-PT é composto por 2.031.480 comentários. A seguir os resultados para as técnicas aplicadas sobre seu texto, sob diversos pré-processamentos, são apresentados.

Suppport Vector Machine sobre Corpus-PT

A Figura 4.4 nos dá uma visão geral dos resultados de classificação dos corpora utilizando o classificador *Support Vector Machine*. Os resultados foram apresentados de forma inversamente ordenada para que pudéssemos averiguar de forma empírica que de fato existe uma diferença no desempenho de acordo com os tipos de pré-processamento utilizados.

Entretanto, quando analisamos os 10 maiores valores do *F-measure* neste caso, apresentados na Figura 4.5, percebemos que, ao levarmos em conta as margens de erro apresentadas, não é possível eleger uma sequência de pré-processamento que se destaque das demais

Ao analisarmos o impacto da representação em diferentes N-gramas, representada na Figura 4.6, observamos que, neste caso, é possível identificar uma representação que se

[6] http://scikit-learn.org/stable/modules/generated/sklearn.svm.LinearSVC.html

[7] http://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.MultinomialNB.html

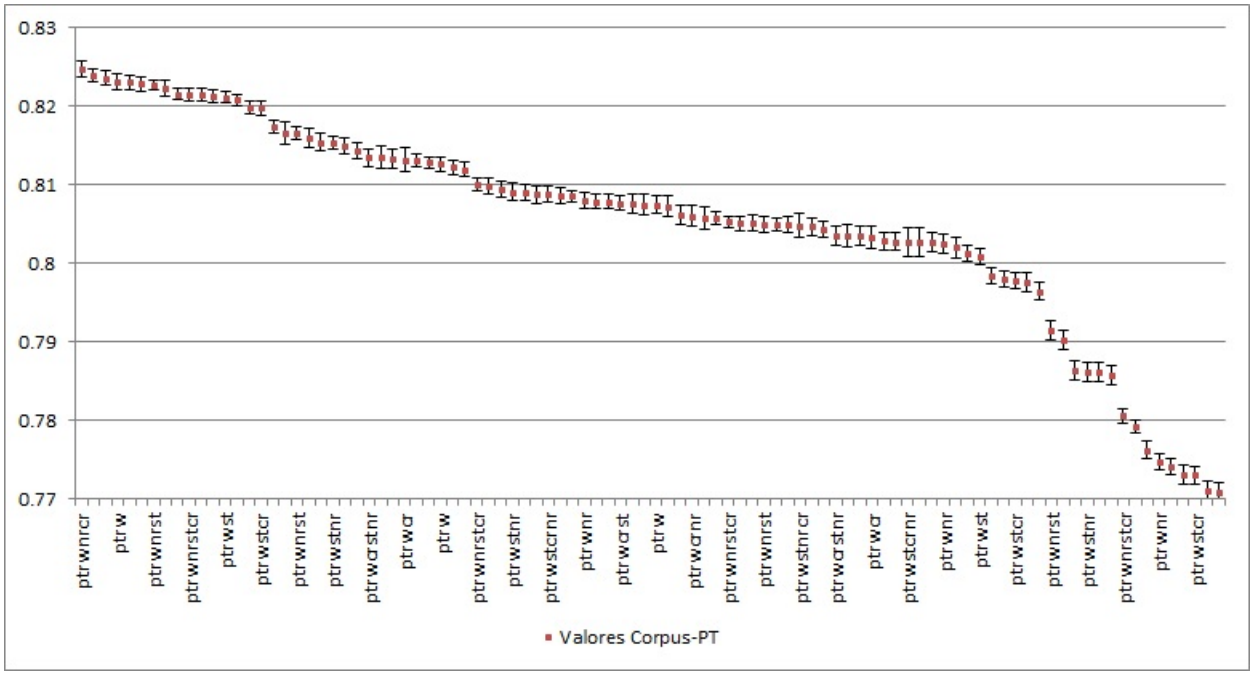

Figura 4.4: Valores do F-measure para Corpus-PT utilizando TF-IDF e Support Vector Machine.

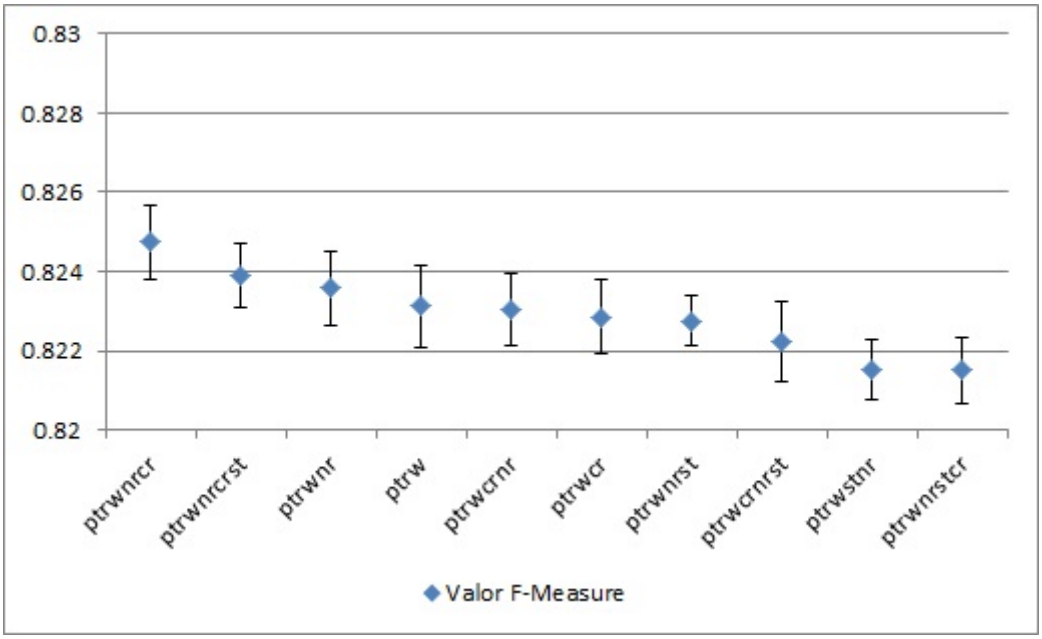

Figura 4.5: 10 maiores valores do F-measure para Corpus-PT utilizando TF-IDF e SVM.

destaca das demais. Em todos os casos mostrados na figura a representação utilizando Unigrama + Bigrama apresentou melhores resultados, inclusive levando em conta as margens de erro apresentadas. É possível verificar ainda que a representação utilizando Bigramas tem resultados piores do que a representação em Unigramas.

A Figura 4.7 nos mostra o impacto da remoção de *stop-words* na análise. Podemos

ver que, para os corpora analisados, retirar as *stop-words* não contribui para a melhora do desempenho do *Support Vector Machine*.

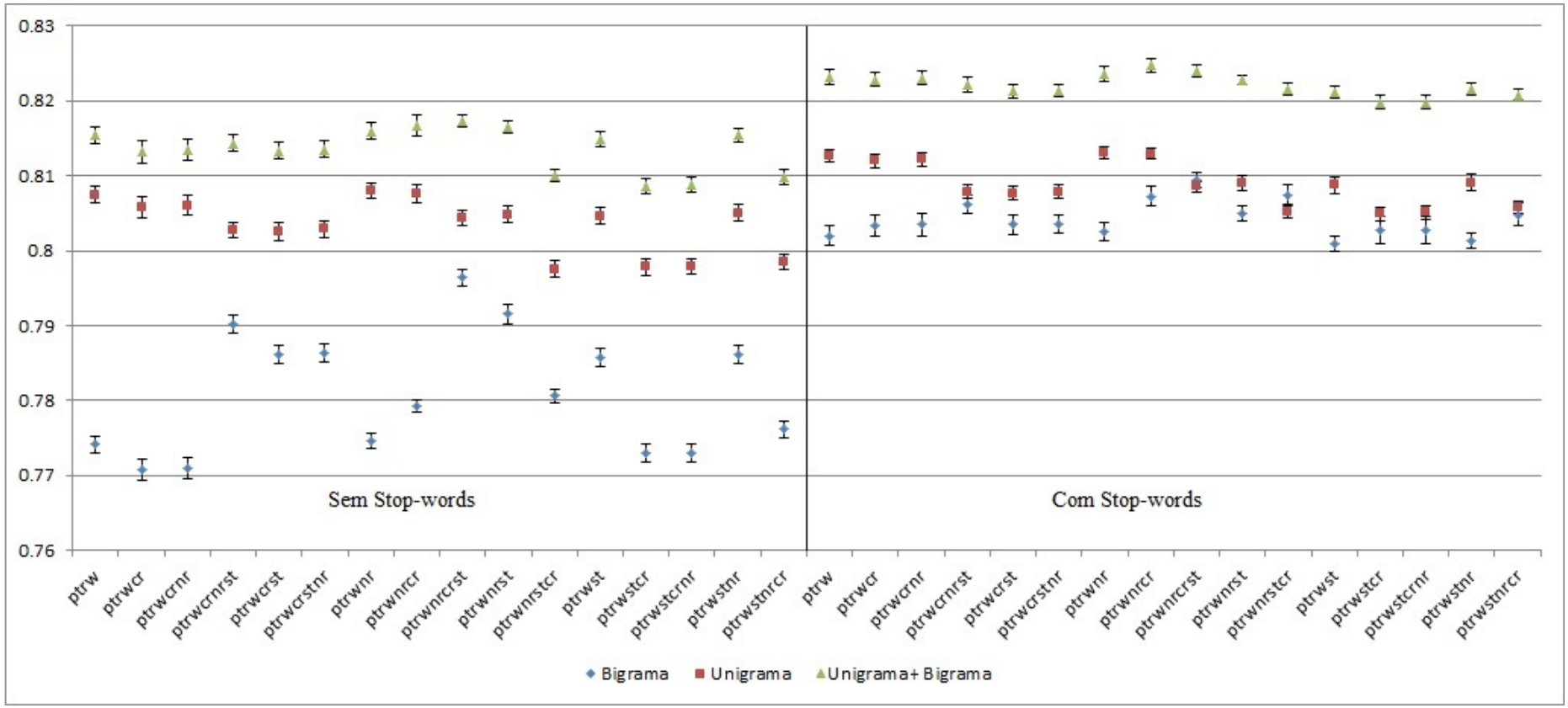

Figura 4.6: Impacto da representação de n-gramas para Corpus-PT utilizando TF-IDF e SVM.

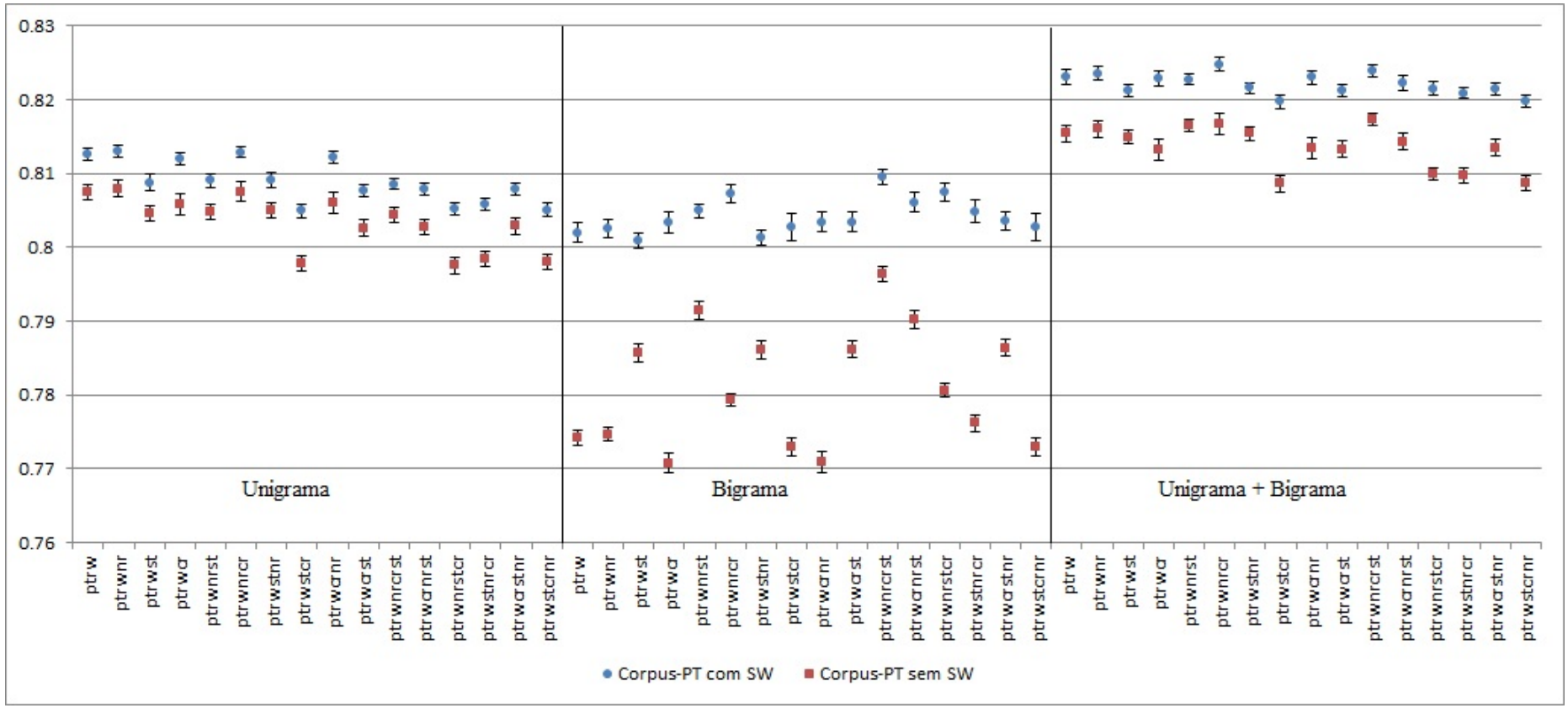

Figura 4.7: Impacto da retirada de stop-words para Corpus-PT utilizando TF-IDF e SVM.

Naive Bayes sobre Corpus-PT

Analogamente aos resultados obtidos utilizando o classificador *Support Vector Machine*, é visível a diferença nos resultados de uma forma macroscópica, apresentados na Figura 4.8, mas ao analisarmos os maiores valores retornados, apresentados na Figura 4.9 observamos que apesar de um ponto se destacar dos demais, sua proximidade é grande demais para o eleger com confiança como uma sequência de pré-processamento ideal de forma geral.

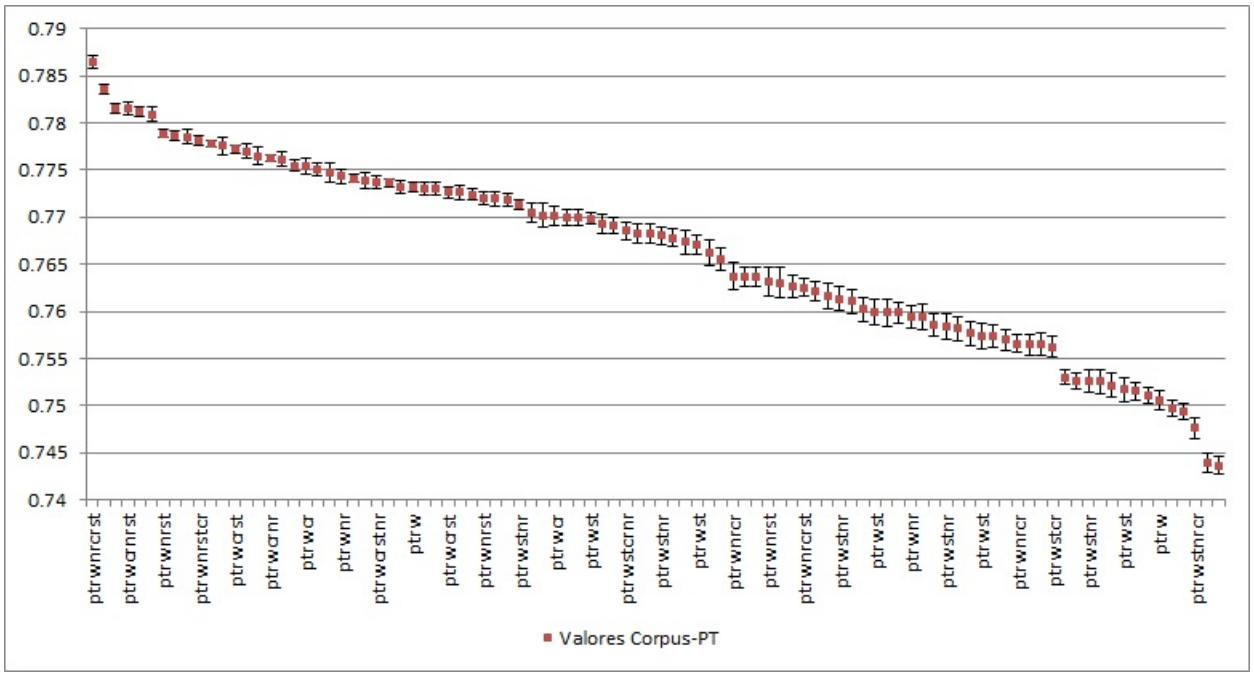

Figura 4.8: Valores do F-measure para Corpus-PT utilizando Naive Bayes.

Entretanto, ao analisarmos os dados levando em conta a representação em N-gramas, verificamos que, diferentemente da classificação utilizando *Support Vector Machines*, não é possível traçar uma relação entre o tipo de representação utilizada e um melhor resultado do classificador.

Quando voltamos à análise do impacto da remoção de *stop-words* do texto verificamos que, assim como no *Support Vector Machine*, não há evidência de que essa etapa contribui para a melhoria dos resultados dos classificadores, já que podemos ver, na Figura 4.11 que em nenhum caso esta etapa mostrou um resultado melhor que permanecer com as *stop-words* no texto durante a análise.

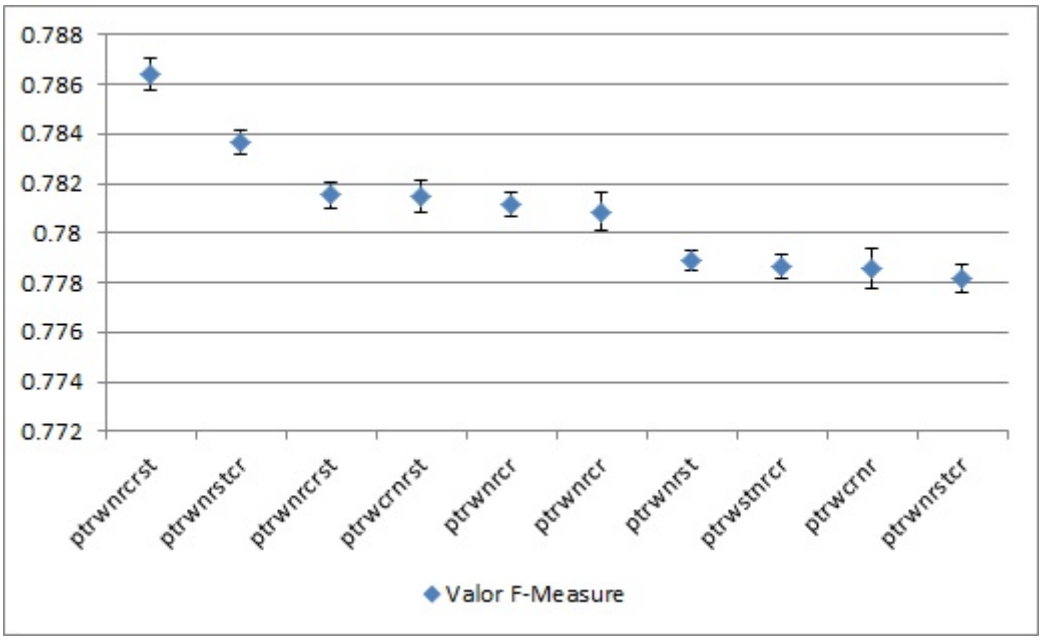

Figura 4.9: 10 maiores valores do F-measure para Corpus-PT utilizando Nayve Bayes.

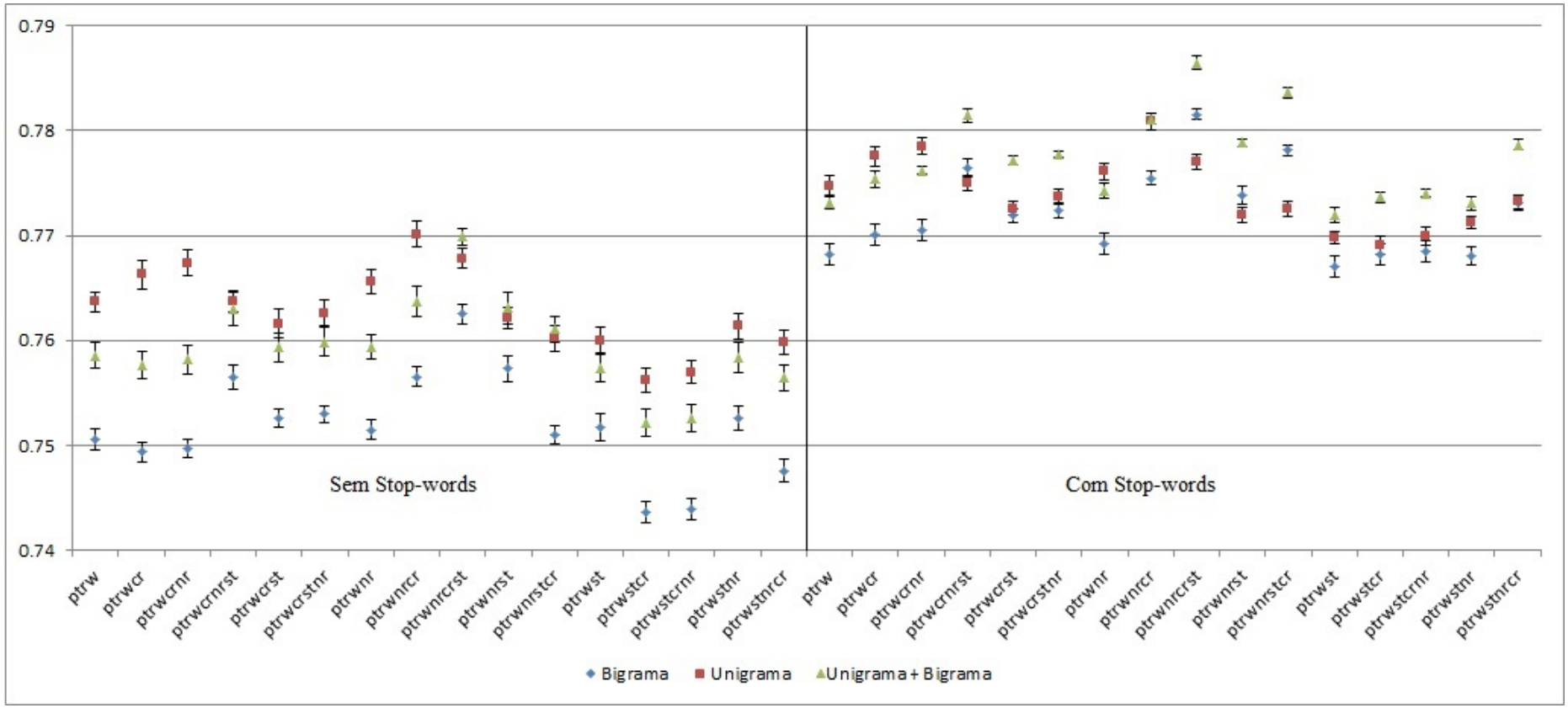

Figura 4.10: Impacto da representação de n-gramas para Corpus-PT utilizando Naive Bayes.

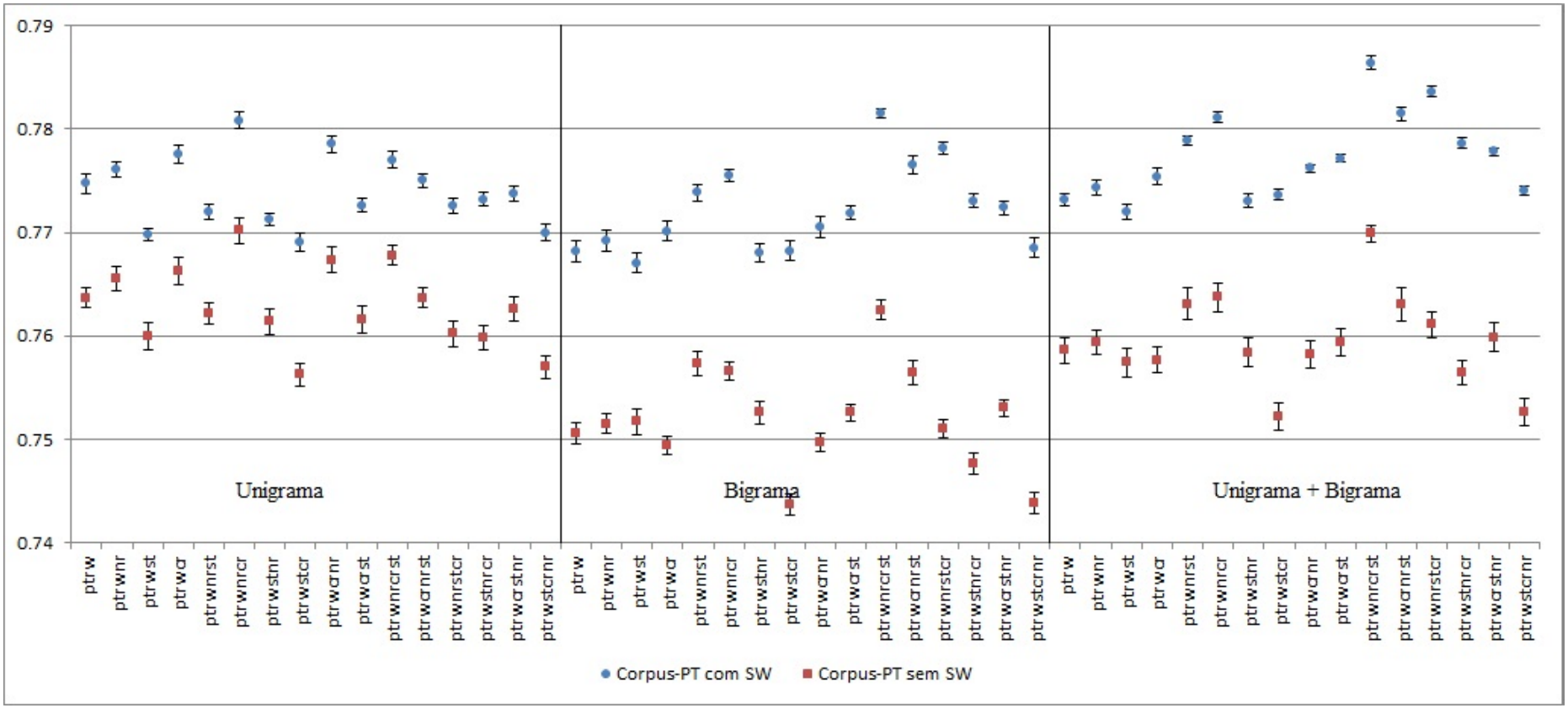

Figura 4.11: Impacto da retirada de stop-words para Corpus-PT utilizando Naive Bayes.

4.5.2 Corpus-EN

O Corpus-EN é composto por 4.843.410 comentários. A seguir os resultados para as técnicas aplicadas sobre seus textos são apresentados.

Support Vector Machine sobre Corpus-EN

Os resultados obtidos utilizando *Support Vector Machines* no Corpus-EN foram similares aos retornados para comentários escritos em português. A Figura 4.12 mostra os resultados para todas as diferentes distribuições de etapa de pré-processamento que foram aplicadas nos *corpora*.

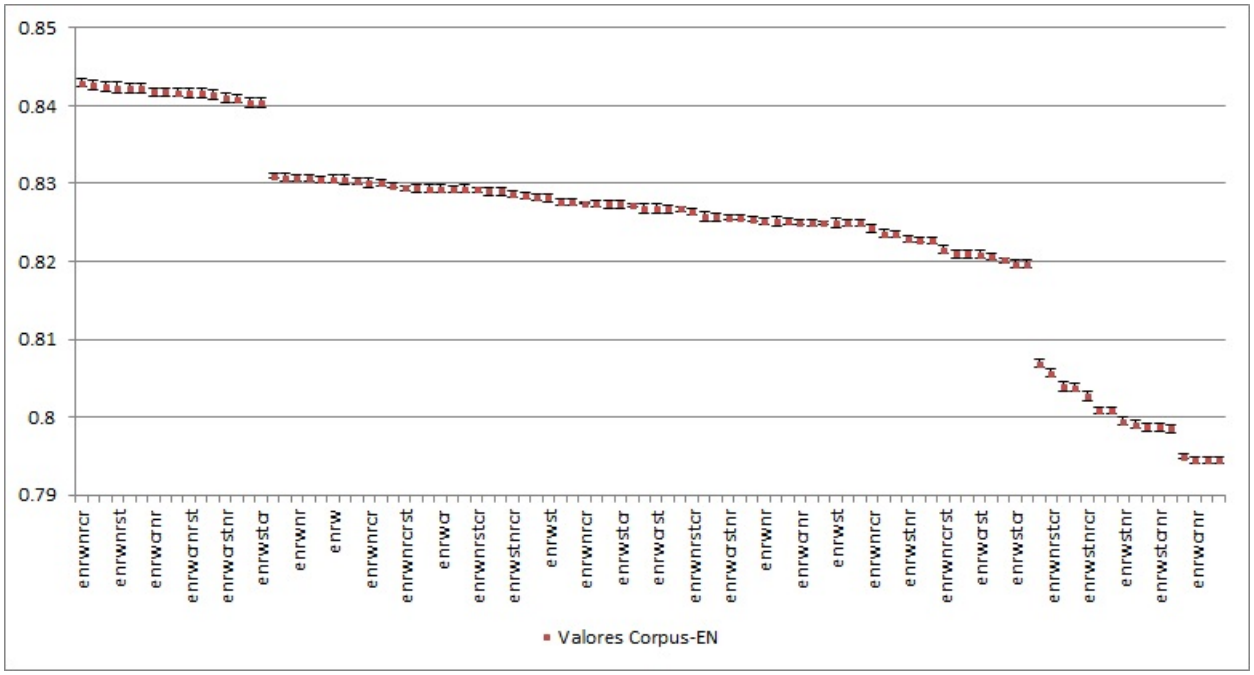

Figura 4.12: Valores do F-measure para Corpus-EN utilizando TF-IDF e SVM.

Como mostra a Figura 4.13, não é possível identificar uma sequência ideal de pré-processamento para o classificador *Support Vector Machine* sobre comentários em inglês.

Quando voltamos nossa análise à forma de representação dos atributos observamos que, analogamente à análise feita sobre comentários em português, é possível verificar experimentalmente analisando os dados demonstrados na Figura 4.14 que a configuração Unigrama + Bigrama se mostra melhor para todos os casos estudados neste etapa. A representação em Bigramas continuou se mostrando inferior à representação em Unigramas, novamente como na análise do Corpus-PT.

A análise da remoção das *stop-words* manteve as mesmas características de quando foi performada para comentários em português. A Figura 4.15 nos mostra que para

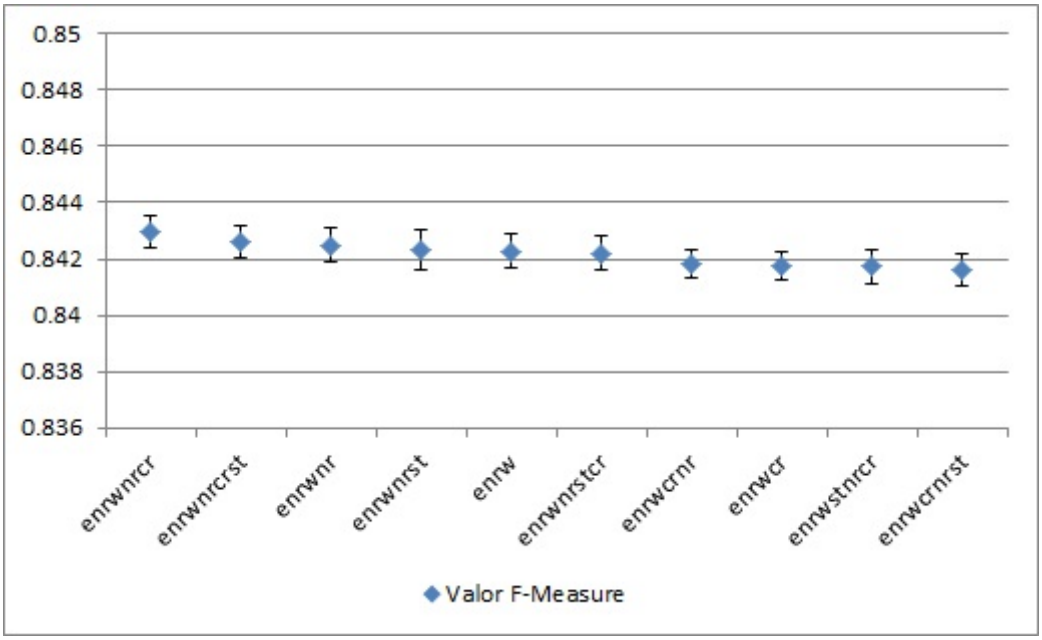

Figura 4.13: 10 maiores valores do F-measure para Corpus-EN utilizando TF-IDF e SVM.

nenhum caso analisado a remoção das *stop-words* foi vantajosa em termos de melhoria do *F-measure* em uma classificação utilizando SVM.

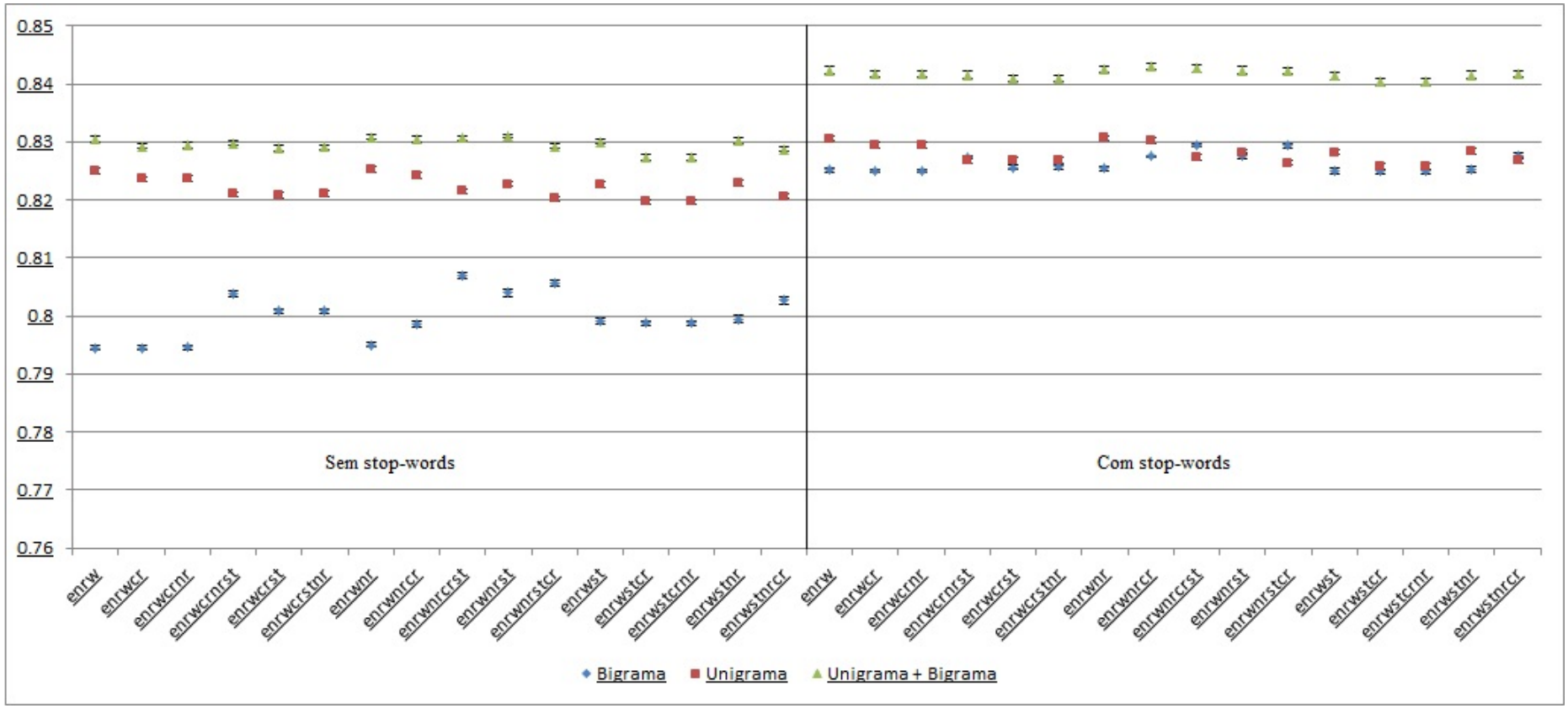

Figura 4.14: Impacto da representação de n-gramas para Corpus-EN utilizando SVM.

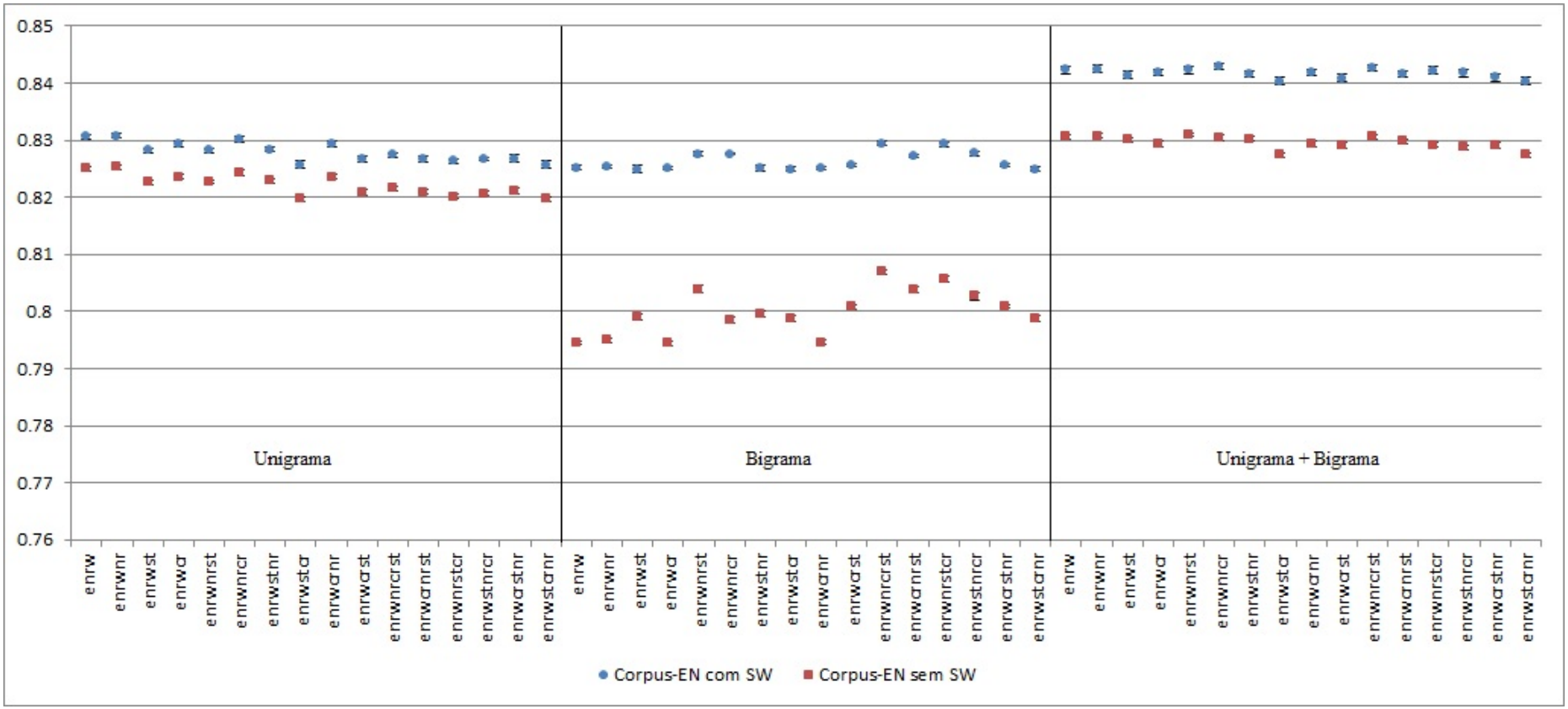

Figura 4.15: Retirada de stop-words para Corpus-EN utilizando SVM.

Naive Bayes sobre Corpus-EN

Assim como pudemos verificar nos resultasos obtidos para o *Support Vector Machine* aplicado ao Corpus-EN e para ambos os classificadores aplicados ao Corpus-PT, os resultados mostram uma melhoria com algumas sequências mas não demonstram uma diferença considerável em seus valores, muitas vezes se sobrepondo quando as margens de erro são levadas em consideração.

A Figura 4.16 mostra os resultados obtidos nesta etapa da análise.

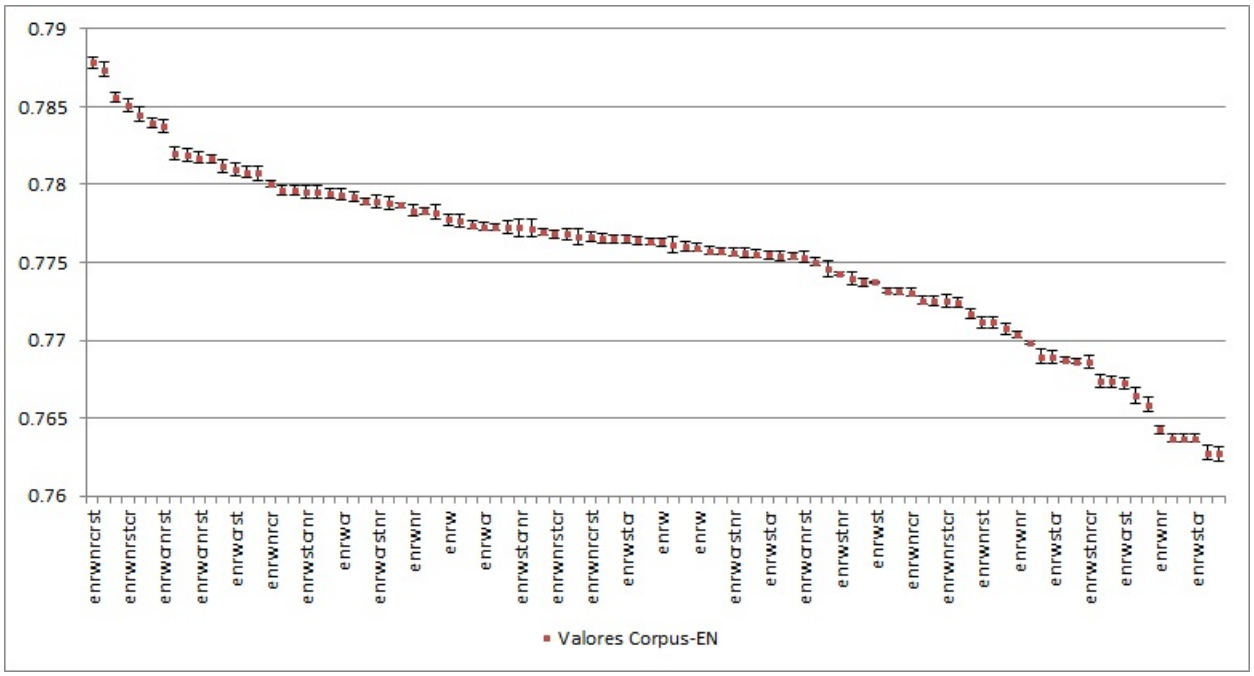

Figura 4.16: Valores do F-measure para Corpus-EN utilizando Naive Bayes.

Verificando os 10 resultados mais expressivos que foram encontrados na análise, apresentados na Figura 4.17, podemos identificar 2 pontos que se destacam levemente dos demais. Entretanto as diferenças entre os resultados dessas classificações é extremamente baixa, o que não nos permite tomar conclusões definitivas acerca de qual a melhor sequência de pré-processamento, como em todas as etapas anteriores.

Não foi possível, assim como na análise feita em comentários em português, definir uma representação de atributos ideal para a classificação utilizando o *Naive Bayes*. Da Figura 4.18, que mostra os resultados para representações em Unigrama, Bigrama e Unigrama + Bigrama, percebemos que o valor máximo dos resultados varia de acordo com o *corpus*, o que não nos dá nenhuma evidência de que existe uma representação que se destaque como ideal para este tipo de classificação.

Finalmente, ao analisarmos o papel da remoção das *stop-words* na tarefa de classificação, fica evidente que esta etapa é dispensável para a análise de comentários da *Google*

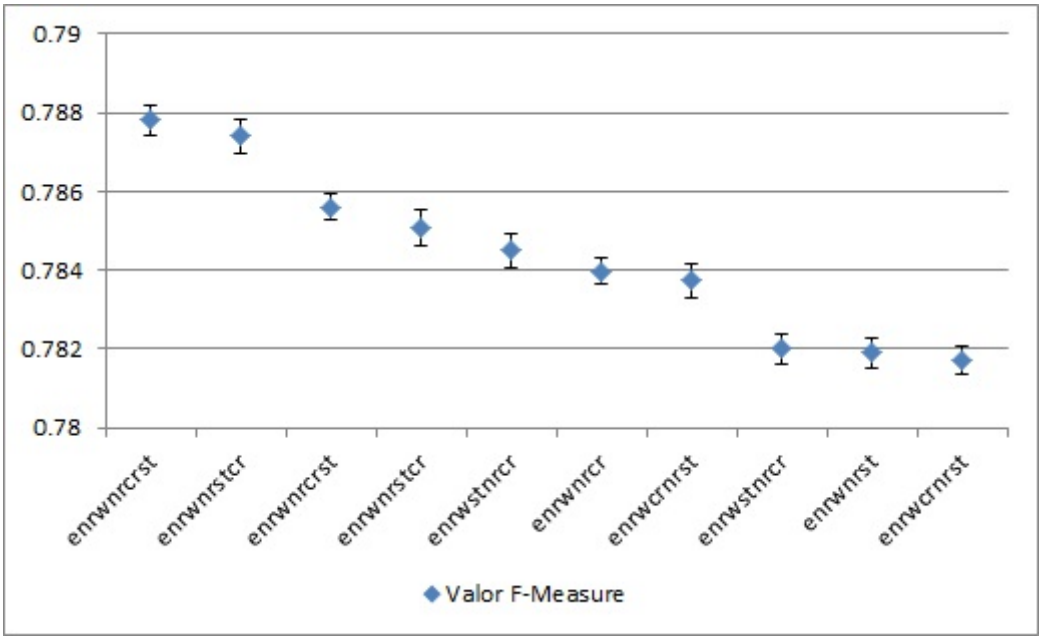

Figura 4.17: 10 maiores valores do F-measure para Corpus-EN utilizando Naive Bayes.

Play tanto em português quanto em inglês. A Figura 4.19 demonstra essa relação ao evidenciar que em nenhum caso na classificação utilizando *Naive Bayes* ou SVM o valor do *F-measure* foi melhorado ao se retirar as *stop-words*.

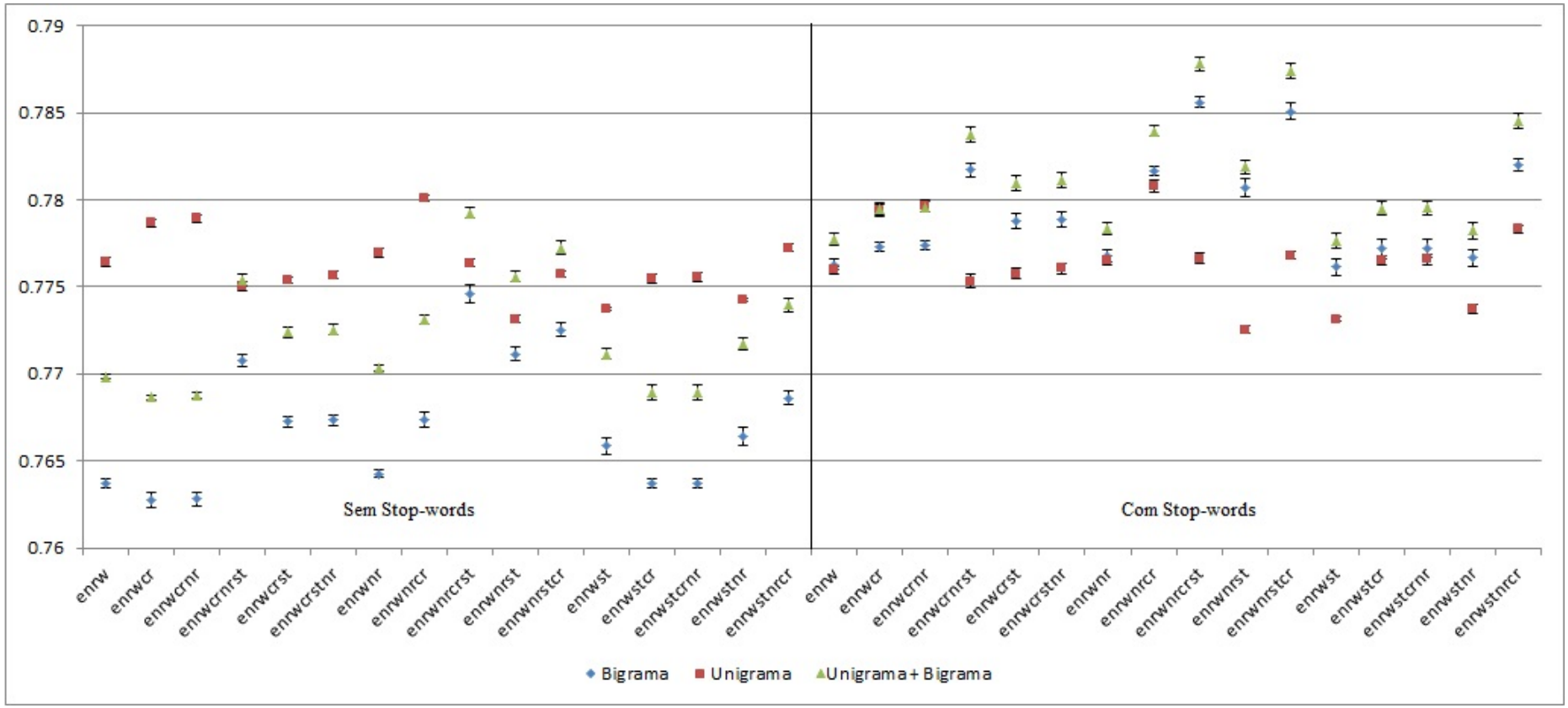

Figura 4.18: Impacto da representação de n-gramas para Corpus-EN utilizando Naive Bayes.

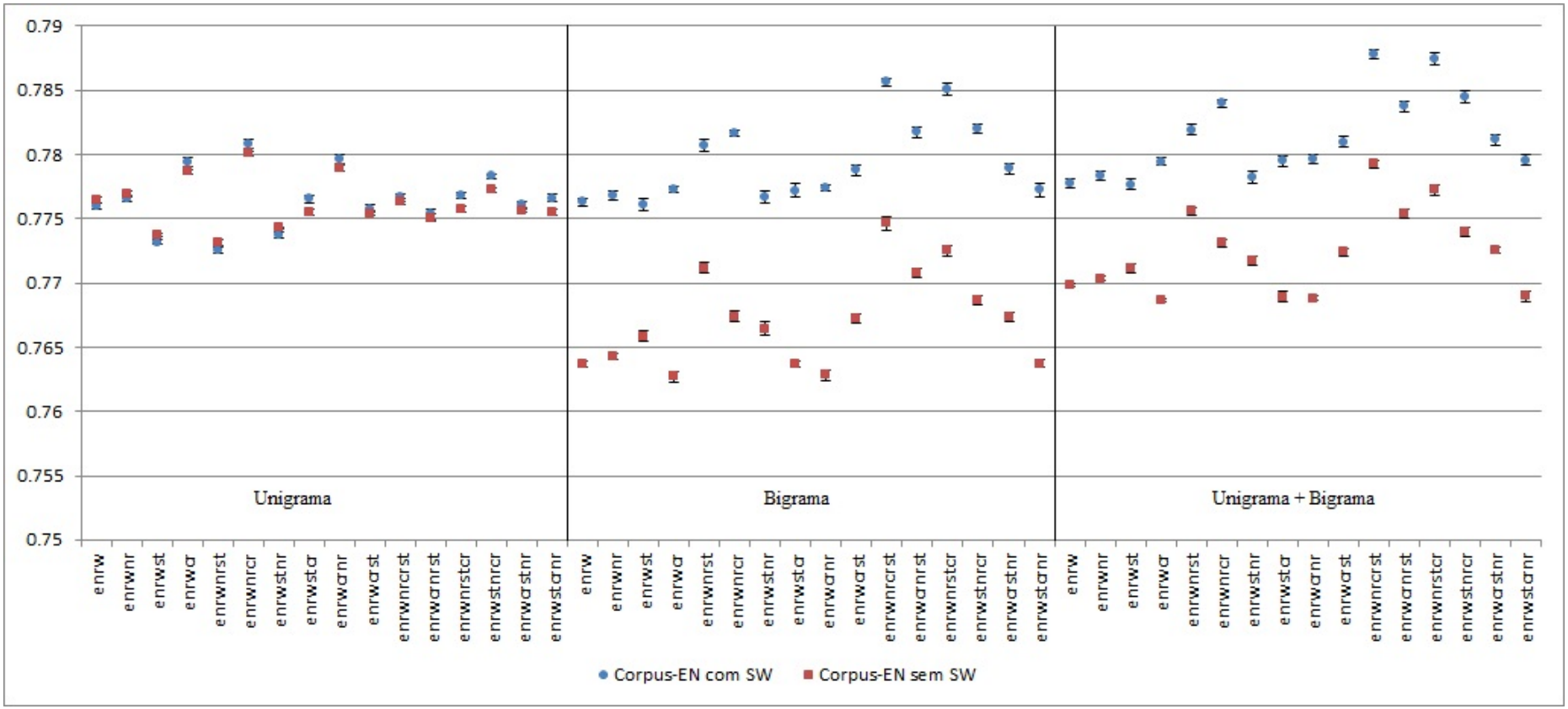

Figura 4.19: Impacto da retirada de stop-words para Corpus-EN utilizando Naive Bayes.

4.6 A curva de aprendizagem

Uma outra análise que é feita neste trabalho é acerca da curva de aprendizagem para cada idioma e cada classificador, e por fim tentar identificar uma quantidade aceitável de comentários que devem ser coletados para se chegar a resultados aceitáveis de classificação.

Para tal, efetuamos uma sequência de classificações gradualmente aumentando o tamanho dos corpora, para verificar como os valores do *F-measure* se comportavam nessas situações. Diante dos dados expostos anteriormente, que sugeriram que não existe uma sequência de classificação ideal para a tarefa, executamos os testes apenas sobre os corpus não pré-processados (ptrw e enrw) e sobre os melhores resultados geralmente apresentados (ptrw-nrcrst e enrw-nrcrst), rodando os testes sem a remoção de *stop-words* e utilizando Unigrama + Bigrama e valores *TF-IDF*.

4.6.1 Corpus-PT

Os resultados para a curva de aprendizagem dos classificadores aplicados ao Corpus-PT são mostrados na Figura 4.20. Podemos verificar que os classificadores mostraram diferentes velocidades de aprendizado, demonstradas pela porcentagem do *corpus* que foi necessária para que a curva estabilizasse a sua inclinação.

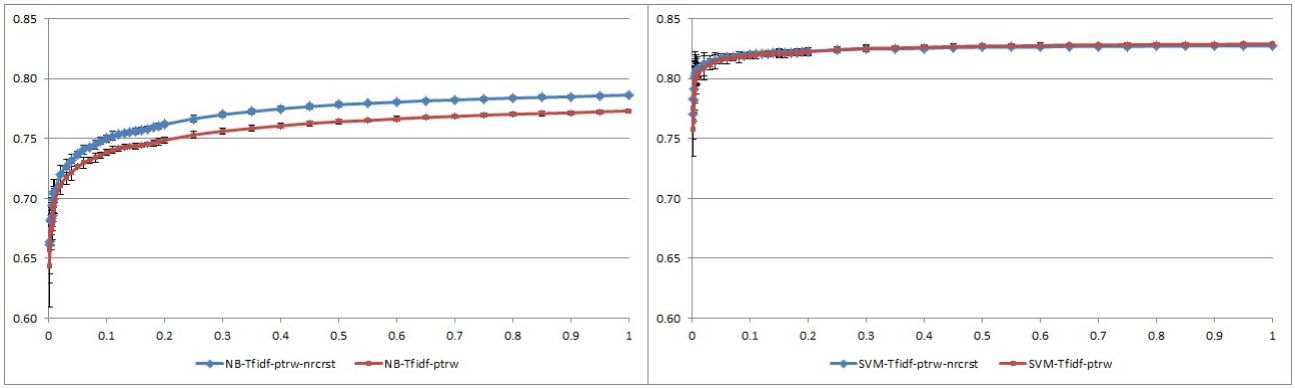

Figura 4.20: Curva de aprendizagem para classificação utilizando SVM e Naive bayes do Corpus-PT.

O *Support Vector Machine* apresentou a necessidade de um conjunto menor para chegar a resultados aceitáveis. Podemos verificar que a partir de 9% da quantidade originalmente analisada de comentários o ganho a cada aumento de 1% no tamanho do *corpus* passa a ser inferior a 0,001% no valor do *F-Measure*. Tal quantitativo, 9%, representa cerca de 182.833 comentários.

A análise da curva de aprendizagem do *Naive Bayes* mostrou um platô menos evidente e sugere que o classificador é mais dependente da quantidade de comentários coletados.

Uma quantidade de comentários que retorna um valor aceitável, seguindo o mesmo critério aplicado à análise do *Support Vector Machine*, é de 25% do tamanho do *corpus* original, cerca de 507.870 comentários.

4.6.2 Corpus-EN

A Figura 4.21. Nos mostra a curva de aprendizagem dos classificadores estudados sobre o Corpus-EN.

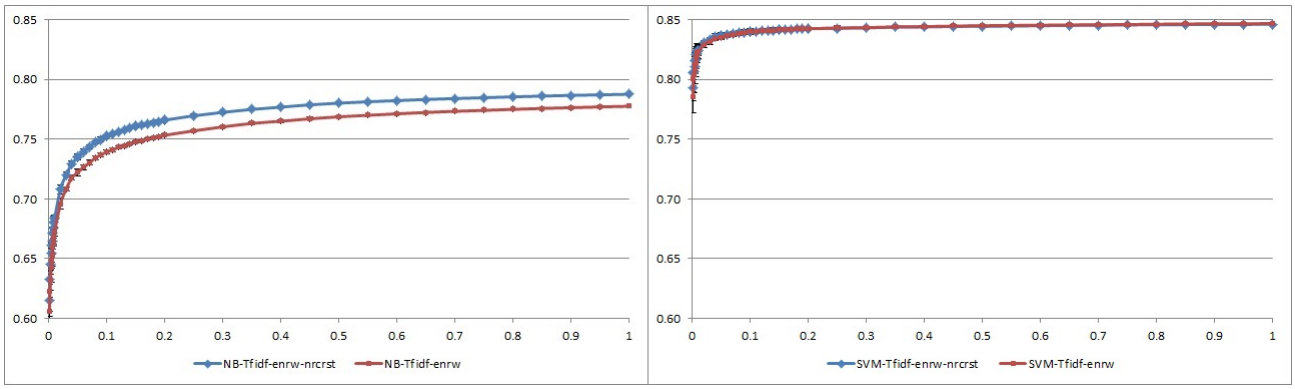

Figura 4.21: Curva de aprendizagem para classificação utilizando SVM e Naive bayes do Corpus-EN.

Na classificação utilizando SVM verificamos que ao atingir 5% do tamanho do *corpus* original, equivalente a 242.170 comentários, o ganho no *F-Measure* começou a ser menor que 0,001% para cada acréscimo de 1% na quantidade de comentários.

Por fim, ao analisarmos a performance do classificador *Naive Bayes* sobre diferentes tamanhos do Corpus-EN verificamos que uma quantidade de cerca de 871.813 comentários, 18% da quantidade original, se mostrou suficiente pro classificador atingir um platô de ganho inferior a 0,001% a cada acréscimo de 1% no tamanho do Corpus-EN.

Capítulo 5

Conclusões

Foi coletado um total de 2.031.480 comentários em português e 4.843.110 comentários em inglês, referentes a aplicativos da categoria "Jogos"da loja de aplicativos da Google. Os comentários são referentes à uma mesma lista de aplicativos, contendo 27.198 jogos.

A distribuição dos sentimentos expressos nos comentários por meio das estrelas não apresenta diferença estatisticamente significativa entre os idiomas português e inglês.

Apesar da hipótese de que o pré-procesamento utilizado na literatura contribui pra tarefa de análise de sentimento, não foi possível determinar uma sequência de aplicação das etapas de pré-processamento utilizadas cujo impacto nas classificações se destacasse de uma maneira conclusiva em relação às outras.

Apesar de amplamente utilizada em análise de sentimento e classificação de textos, a remoção das *stop-words* não contribuiu para a melhoria no desempenho dos classificadores, independentemente do idioma. Para tentar identificar uma causa para esta relação executamos alguns testes utilizando *Total Frequency* ao invés de *TF-IDF*, mostrados na Figura 5.1. Observamos que, mesmo utilizando a frequência total das palavras, não houve benefício em se retirar as *stop-words*. Um estudo mais aprofundado sobre o tema é necessário para averiguar o motivo desta característica neste *corpus*, bem como averiguar se tal conclusão pode ser generalizada para outras bases.

A representação dos atributos como unigrama + bigrama retornou melhores resultados para todos os casos em que foi utilizado *Support Vector Machine*. Entretanto não foi possível identificar uma representação que se mostrasse predominantemente melhor para a classificação utilizando *Naive Bayes*.

Os classificadores *Support Vector Machine* apresentaram resultados aceitáveis a partir de uma quantidade de cerca de 185 mil comentários para o Corpus-PT e 245 mil comentários para o Corpus-EN. As quantidades de comentários necessárias ao aprendizado do *Naive Bayes* foram consideravelmente maiores, necessitando de cerca de 510 mil comentários para o Corpus-PT e 870 mil comentários para o Corpus-EN atingirem resultados

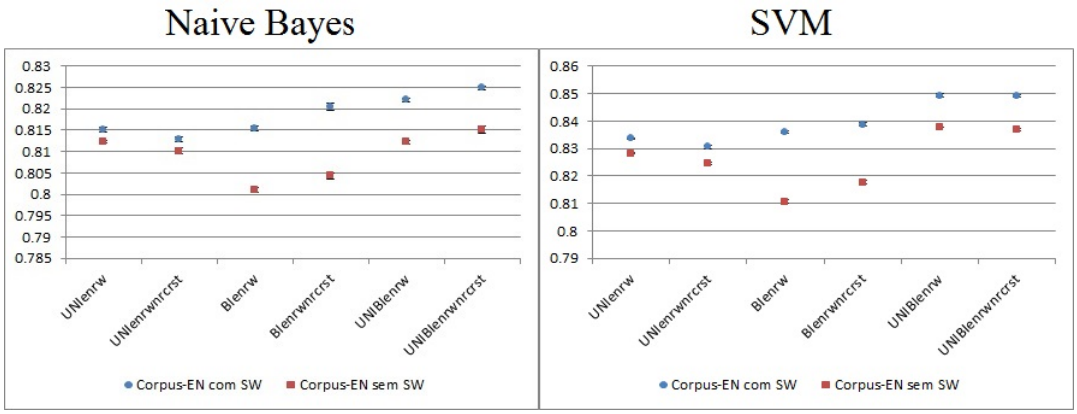

Figura 5.1: Retirada de stop-words utilizando contagem bruta(TF).

suficientes. Verificamos que, de maneira geral, a análise de textos em inglês necessitou de mais amostras em seu conjunto de treino.

Capítulo 6

Trabalhos Futuros

No decorrer deste trabalho foi possível identificar algumas questões e dificuldades que podem ser investigadas em trabalhos futuros, bem como a falta de algumas ferramentas que facilitariam a tarefa caso fossem implementadas.

- Apesar de ter sido identificado que não existe uma sequência de pré-processamento ideal para ser performada na análise de sentimento dos comentários, não foi possível executar uma verificação aprofundada levando em consideração a complexidade dos algoritmos de pré-processamento. Tal estudo pode contribuir para as escolhas de quais etapas o cientista de dados aplicará em seu estudo.
- Não foi possível utilizar uma medida de incerteza acerca dos valores de F-Measure que os classificadores apresentaram apesar de, como foi mostrado ao utilizar o K-fold cross validation, o valor de F-Measure ser dependente dos conjuntos de treino e validação, o que pode levar à análises equivocadas de valores muito próximos. Um estudo que identifique uma forma de identificar uma margem de valores para o F-measure se mostra interessante.
- Não foi possível identificar a causa de a remoção das stop-words não causar melhorias no desempenho dos algoritmos de classificação, contrariando outros estudos. Um estudo mais voltado para a linguística e utilizando diferentes *corpora* poderia validar melhor esta hipótese.
- A implementação de um crawler específico, ou a disponibilização dos dados por parte do Google, tornaria a análise mais ágil, pois uma grande parte do tempo necessário a esta pesquisa foi despendido na fase de coleta dos dados.
- Apesar de alguns exemplos de classificação utilizando como atributo a contagem geral da frequência das palavras no documento terem sido comparados com os resultados apresentados na pesquisa, julga-se necessário um estudo mais aprofundado

sobre a relação da utilização de TF-IDF, da frequência das palavras e da remoção ou não de *stop-words* dos comentários analisados.

Referências

[1] Internet users by country (2014) - internet live stats. http://www.internetlivestats.com/internet-users-by-country/. Accessed: 2015-05-15. 1

[2] Natural language toolkit. http://www.nltk.org/. Accessed: 2015-06-13. 14

[3] Number of android applications. http://www.appbrain.com/stats/number-of-android-apps. Accessed: 2015-06-13. 23

[4] Usage of content languages for websites. http://w3techs.com/technologies/overview/content_language/all. Accessed: 2015-06-13. 1

[5] Patrick Paroubek Alexander Pak. Twitter as a corpus for sentiment analysis and opinion mining. *LREC*, 2010. 4, 5

[6] E Haddi amd X Liu and Y Shi. The role of text pre-processing in sentiment analysis. *Procedia Computer Science*, 2013. 13

[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. *in Proc. of LREC*, 2010. 13

[8] CORINNA CORTES and VLADIMIR VAPNIK. Support-vector networks. *Machine learning*, 1995. 7

[9] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin. Liblinear: A library for large linear classification. *Journal of Machine Learning Research*, 2008. 31

[10] Leonard Hoon, Rajesh Vasa, Jean-Guy Schneider, and John Grundy. An analysis of the mobile app review landscape: Trends and implications. *Swinburne University of Technology. Faculty of Information and Communication Technologies*, 2013. 5

[11] Leonard Hoon, Rajesh Vasa, Jean-Guy Schneider, and Kon Mouzakis. A preliminary analysis of vocabulary in mobile app user reviews. *24th Australian Computer-Human Interaction Conference.*, 2012. 21, 29

[12] Nan Hu, Jie Zhang, and Paul A. Pavlou. Overcoming the j-shaped distribution of product reviews. *Communications of the ACM 52.10*, 2009. 25

[13] A. Govardhan I. Hemalatha, G.P. Saradhi Varma. Preprocessing the informal text for efficient sentiment analysis. *International Journal of Emerging Trends and Technology in Computer Science*, 2012. 4, 5

[14] Mir Riyanul Islam. Numeric rating of apps on google play store by sentiment analysis on user reviews. *Electrical Engineering and Information and Communication Technology (ICEEICT)*, 2014. 5

[15] Johanna Moore Kouloumpis Efthymios, Theresa Wilson. Twitter sentiment analysis: The good the bad and the omg!. *ICWSM 11*, 2011. 4, 20

[16] Marcello Lins. Google play apps crawler, 2015. Accessed: 2015-06-13. 23

[17] Bing Liu. Sentiment analysis and opinion mining. *Morgan & Claypool Publishers*, 2012. 13

[18] Jiawen Liu, Mantosh Kumar Sarkar, and Goutam Chakraborty. Feature-based sentiment analysis on android app reviews using sas® text miner and sas® sentiment analysis studio. *Proceedings of the SAS Global Forum 2013 Conference.*, 2013. 5

[19] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, pages 1–135, 2008. 1

[20] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. *Proceedings of the ACL-02 conference on Empirical methods in natural language processing*, 2002. 4

[21] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011. 31

[22] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of Database Systems*, 2009. 12

[23] Marcelino Silva. Mineração de dados - conceitos, aplicações e experimentos com weka. *Sociedade Brasileira de Computação*, 2004. 6

[24] Herbert A. Simon. Why should machines learn? *Machine Learning*, 1983. 6

[25] Sida Wang and Christopher D. Manning. Baselines and bigrams: Simple, good sentiment and topic classification. *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics*, 2012. 2, 8, 29, 31

[26] Rüdiger Wirth and Jochen Hipp. Crisp-dm: Towards a standard process model for data mining. *4th International Conference on the Practical Applications of Knowledge Discovery and Data Mining*, 2000. 17

APÊNDICE A

Tabela 1: Valores para a classificação do Corpus-PT utilizando SVM.

Stop-words	N-grama	Pré-processamento	F-Measure	Erro (+/-)
SW	UNI	ptrw	0.8126300	0.0008800
SW	UNI	ptrwnr	0.8130400	0.0008200
SW	UNI	ptrwst	0.8087400	0.0011000
SW	UNI	ptrwcr	0.8119800	0.0009200
SW	UNI	ptrwnrst	0.8090100	0.0010000
SW	UNI	ptrwnrcr	0.8128700	0.0007000
SW	UNI	ptrwstnr	0.8091100	0.0011000
SW	UNI	ptrwstcr	0.8049300	0.0009000
SW	UNI	ptrwcrnr	0.8122000	0.0009100
SW	UNI	ptrwcrst	0.8076700	0.0009200
SW	UNI	ptrwnrcrst	0.8085500	0.0006700
SW	UNI	ptrwcrnrst	0.8078600	0.0009000
SW	UNI	ptrwnrstcr	0.8052400	0.0008000
SW	UNI	ptrwstnrcr	0.8057200	0.0008200
SW	UNI	ptrwcrstnr	0.8078900	0.0008400
SW	UNI	ptrwstcrnr	0.8050700	0.0008900
SW	BI	ptrw	0.8020100	0.0012900
SW	BI	ptrwnr	0.8025200	0.0012600
SW	BI	ptrwst	0.8009000	0.0010300
SW	BI	ptrwcr	0.8033000	0.0014200
SW	BI	ptrwnrst	0.8049500	0.0009600
SW	BI	ptrwnrcr	0.8072800	0.0012800
SW	BI	ptrwstnr	0.8012900	0.0009800
SW	BI	ptrwstcr	0.8027300	0.0018100
SW	BI	ptrwcrnr	0.8034500	0.0014300
SW	BI	ptrwcrst	0.8034500	0.0012400
SW	BI	ptrwnrcrst	0.8094300	0.0010300
SW	BI	ptrwcrnrst	0.8061200	0.0012500
SW	BI	ptrwnrstcr	0.8074600	0.0013300
SW	BI	ptrwstnrcr	0.8048300	0.0015600

SW	BI	ptrwcrstnr	0.8035600	0.0012200
SW	BI	ptrwstcrnr	0.8027700	0.0018300
SW	UNIBI	ptrw	0.8231400	0.0010400
SW	UNIBI	ptrwnr	0.8235800	0.0009400
SW	UNIBI	ptrwst	0.8211700	0.0007800
SW	UNIBI	ptrwcr	0.8228600	0.0009400
SW	UNIBI	ptrwnrst	0.8227600	0.0006300
SW	UNIBI	ptrwnrcr	0.8247400	0.0009300
SW	UNIBI	ptrwstnr	0.8215400	0.0007700
SW	UNIBI	ptrwstcr	0.8197600	0.0009000
SW	UNIBI	ptrwcrnr	0.8230400	0.0009100
SW	UNIBI	ptrwcrst	0.8212700	0.0008300
SW	UNIBI	ptrwnrcrst	0.8239100	0.0008200
SW	UNIBI	ptrwcrnrst	0.8222500	0.0010000
SW	UNIBI	ptrwnrstcr	0.8215100	0.0008500
SW	UNIBI	ptrwstnrcr	0.8208400	0.0007000
SW	UNIBI	ptrwcrstnr	0.8214200	0.0007900
SW	UNIBI	ptrwstcrnr	0.8198300	0.0008800
NOSW	UNI	ptrw	0.8074400	0.0010900
NOSW	UNI	ptrwnr	0.8079600	0.0010700
NOSW	UNI	ptrwst	0.8046300	0.0010500
NOSW	UNI	ptrwcr	0.8057900	0.0014100
NOSW	UNI	ptrwnrst	0.8048500	0.0010400
NOSW	UNI	ptrwnrcr	0.8075600	0.0012800
NOSW	UNI	ptrwstnr	0.8050500	0.0010300
NOSW	UNI	ptrwstcr	0.7978100	0.0010600
NOSW	UNI	ptrwcrnr	0.8060300	0.0013600
NOSW	UNI	ptrwcrst	0.8026100	0.0012000
NOSW	UNI	ptrwnrcrst	0.8043100	0.0010000
NOSW	UNI	ptrwcrnrst	0.8027800	0.0010500
NOSW	UNI	ptrwnrstcr	0.7975400	0.0011700
NOSW	UNI	ptrwstnrcr	0.7984600	0.0010200
NOSW	UNI	ptrwcrstnr	0.8028600	0.0011400
NOSW	UNI	ptrwstcrnr	0.7979600	0.0010000
NOSW	BI	ptrw	0.7741000	0.0010300

NOSW	BI	ptrwnr	0.7746700	0.0009900
NOSW	BI	ptrwst	0.7856800	0.0012400
NOSW	BI	ptrwcr	0.7707600	0.0013800
NOSW	BI	ptrwnrst	0.7915100	0.0012300
NOSW	BI	ptrwnrcr	0.7792600	0.0008500
NOSW	BI	ptrwstnr	0.7861700	0.0012500
NOSW	BI	ptrwstcr	0.7729600	0.0012000
NOSW	BI	ptrwcrnr	0.7709300	0.0014000
NOSW	BI	ptrwcrst	0.7861600	0.0011500
NOSW	BI	ptrwnrcrst	0.7963600	0.0011200
NOSW	BI	ptrwcrnrst	0.7901600	0.0012300
NOSW	BI	ptrwnrstcr	0.7805800	0.0009600
NOSW	BI	ptrwstnrcr	0.7761600	0.0011300
NOSW	BI	ptrwcrstnr	0.7863400	0.0011300
NOSW	BI	ptrwstcrnr	0.7730000	0.0012300
NOSW	UNIBI	ptrw	0.8154200	0.0011800
NOSW	UNIBI	ptrwnr	0.8159900	0.0011800
NOSW	UNIBI	ptrwst	0.8149400	0.0009800
NOSW	UNIBI	ptrwcr	0.8131900	0.0014800
NOSW	UNIBI	ptrwnrst	0.8165500	0.0008200
NOSW	UNIBI	ptrwnrcr	0.8166500	0.0014200
NOSW	UNIBI	ptrwstnr	0.8153900	0.0008600
NOSW	UNIBI	ptrwstcr	0.8086200	0.0010600
NOSW	UNIBI	ptrwcrnr	0.8134600	0.0014100
NOSW	UNIBI	ptrwcrst	0.8133100	0.0011300
NOSW	UNIBI	ptrwnrcrst	0.8174000	0.0008000
NOSW	UNIBI	ptrwcrnrst	0.8143100	0.0010900
NOSW	UNIBI	ptrwnrstcr	0.8099900	0.0008400
NOSW	UNIBI	ptrwstnrcr	0.8098100	0.0010400
NOSW	UNIBI	ptrwcrstnr	0.8135000	0.0011300
NOSW	UNIBI	ptrwstcrnr	0.8087400	0.0010000

Tabela 2: Valores para a classificação do Corpus-PT utilizando Naive Bayes.

Stop-words	N-grama	Pré-processamento	F-Measure	Erro (+/-)

SW	UNI	ptrw	0.77473	0.00095
SW	UNI	ptrwnr	0.77612	0.00077
SW	UNI	ptrwst	0.76980	0.00059
SW	UNI	ptrwcr	0.77756	0.00090
SW	UNI	ptrwnrst	0.77201	0.00071
SW	UNI	ptrwnrcr	0.78087	0.00076
SW	UNI	ptrwstnr	0.77130	0.00056
SW	UNI	ptrwstcr	0.76909	0.00084
SW	UNI	ptrwcrnr	0.77853	0.00081
SW	UNI	ptrwcrst	0.77262	0.00063
SW	UNI	ptrwnrcrst	0.77702	0.00077
SW	UNI	ptrwcrnrst	0.77502	0.00069
SW	UNI	ptrwnrstcr	0.77261	0.00073
SW	UNI	ptrwstnrcr	0.77322	0.00069
SW	UNI	ptrwcrstnr	0.77375	0.00068
SW	UNI	ptrwstcrnr	0.76996	0.00081
SW	BI	ptrw	0.76820	0.00101
SW	BI	ptrwnr	0.76921	0.00102
SW	BI	ptrwst	0.76707	0.00096
SW	BI	ptrwcr	0.77012	0.00098
SW	BI	ptrwnrst	0.77386	0.00084
SW	BI	ptrwnrcr	0.77548	0.00062
SW	BI	ptrwstnr	0.76804	0.00090
SW	BI	ptrwstcr	0.76825	0.00101
SW	BI	ptrwcrnr	0.77050	0.00098
SW	BI	ptrwcrst	0.77190	0.00067
SW	BI	ptrwnrcrst	0.78153	0.00050
SW	BI	ptrwcrnrst	0.77649	0.00088
SW	BI	ptrwnrstcr	0.77813	0.00056
SW	BI	ptrwstnrcr	0.77307	0.00071
SW	BI	ptrwcrstnr	0.77237	0.00062
SW	BI	ptrwstcrnr	0.76852	0.00098
SW	UNIBI	ptrw	0.77315	0.00054
SW	UNIBI	ptrwnr	0.77432	0.00070
SW	UNIBI	ptrwst	0.77197	0.00076

SW	UNIBI	ptrwcr	0.77539	0.00082
SW	UNIBI	ptrwnrst	0.77886	0.00041
SW	UNIBI	ptrwnrcr	0.78115	0.00049
SW	UNIBI	ptrwstnr	0.77306	0.00065
SW	UNIBI	ptrwstcr	0.77366	0.00047
SW	UNIBI	ptrwcrnr	0.77619	0.00034
SW	UNIBI	ptrwcrst	0.77720	0.00043
SW	UNIBI	ptrwnrcrst	0.78644	0.00064
SW	UNIBI	ptrwcrnrst	0.78148	0.00066
SW	UNIBI	ptrwnrstcr	0.78363	0.00048
SW	UNIBI	ptrwstnrcr	0.77867	0.00049
SW	UNIBI	ptrwcrstnr	0.77779	0.00031
SW	UNIBI	ptrwstcrnr	0.77406	0.00042
NOSW	UNI	ptrw	0.76366	0.00097
NOSW	UNI	ptrwnr	0.76559	0.00117
NOSW	UNI	ptrwst	0.75991	0.00130
NOSW	UNI	ptrwcr	0.76626	0.00135
NOSW	UNI	ptrwnrst	0.76215	0.00104
NOSW	UNI	ptrwnrcr	0.77016	0.00126
NOSW	UNI	ptrwstnr	0.76136	0.00128
NOSW	UNI	ptrwstcr	0.75624	0.00112
NOSW	UNI	ptrwcrnr	0.76735	0.00123
NOSW	UNI	ptrwcrst	0.76161	0.00135
NOSW	UNI	ptrwnrcrst	0.76779	0.00095
NOSW	UNI	ptrwcrnrst	0.76370	0.00100
NOSW	UNI	ptrwnrstcr	0.76019	0.00128
NOSW	UNI	ptrwstnrcr	0.75986	0.00115
NOSW	UNI	ptrwcrstnr	0.76262	0.00121
NOSW	UNI	ptrwstcrnr	0.75698	0.00111
NOSW	BI	ptrw	0.75057	0.00106
NOSW	BI	ptrwnr	0.75149	0.00098
NOSW	BI	ptrwst	0.75170	0.00127
NOSW	BI	ptrwcr	0.74935	0.00088
NOSW	BI	ptrwnrst	0.75734	0.00122
NOSW	BI	ptrwnrcr	0.75656	0.00091

NOSW	BI	ptrwstnr	0.75256	0.00115
NOSW	BI	ptrwstcr	0.74364	0.00100
NOSW	BI	ptrwcrnr	0.74967	0.00083
NOSW	BI	ptrwcrst	0.75260	0.00080
NOSW	BI	ptrwnrcrst	0.76251	0.00097
NOSW	BI	ptrwcrnrst	0.75646	0.00115
NOSW	BI	ptrwnrstcr	0.75102	0.00088
NOSW	BI	ptrwstnrcr	0.74759	0.00106
NOSW	BI	ptrwcrstnr	0.75299	0.00078
NOSW	BI	ptrwstcrnr	0.74387	0.00103
NOSW	UNIBI	ptrw	0.75858	0.00124
NOSW	UNIBI	ptrwnr	0.75940	0.00119
NOSW	UNIBI	ptrwst	0.75743	0.00137
NOSW	UNIBI	ptrwcr	0.75765	0.00128
NOSW	UNIBI	ptrwnrst	0.76310	0.00156
NOSW	UNIBI	ptrwnrcr	0.76373	0.00140
NOSW	UNIBI	ptrwstnr	0.75838	0.00141
NOSW	UNIBI	ptrwstcr	0.75217	0.00128
NOSW	UNIBI	ptrwcrnr	0.75817	0.00131
NOSW	UNIBI	ptrwcrst	0.75938	0.00138
NOSW	UNIBI	ptrwnrcrst	0.76993	0.00080
NOSW	UNIBI	ptrwcrnrst	0.76304	0.00161
NOSW	UNIBI	ptrwnrstcr	0.76106	0.00127
NOSW	UNIBI	ptrwstnrcr	0.75646	0.00121
NOSW	UNIBI	ptrwcrstnr	0.75988	0.00141
NOSW	UNIBI	ptrwstcrnr	0.75255	0.00132

Tabela 3: Valores para a classificação do Corpus-EN utilizando Naive Bayes.

Stop-words	N-grama	Pré-processamento	F-Measure	Erro (+/-)
SW	UNI	enrw	0.77596	0.00023
SW	UNI	enrwnr	0.77654	0.00025
SW	UNI	enrwst	0.77316	0.00016
SW	UNI	enrwcr	0.77938	0.00035
SW	UNI	enrwnrst	0.77255	0.00025

SW	UNI	enrwnrcr	0.78078	0.00035
SW	UNI	enrwstnr	0.77373	0.00025
SW	UNI	enrwstcr	0.77652	0.00028
SW	UNI	enrwcrnr	0.77964	0.00034
SW	UNI	enrwcrst	0.77577	0.00030
SW	UNI	enrwnrcrst	0.77665	0.00028
SW	UNI	enrwcrnrst	0.77533	0.00037
SW	UNI	enrwnrstcr	0.77682	0.00023
SW	UNI	enrwstnrcr	0.77831	0.00020
SW	UNI	enrwcrstnr	0.77606	0.00032
SW	UNI	enrwstcrnr	0.77659	0.00032
SW	BI	enrw	0.77630	0.00029
SW	BI	enrwnr	0.77682	0.00035
SW	BI	enrwst	0.77613	0.00049
SW	BI	enrwcr	0.77729	0.00025
SW	BI	enrwnrst	0.78072	0.00051
SW	BI	enrwnrcr	0.78165	0.00029
SW	BI	enrwstnr	0.77667	0.00049
SW	BI	enrwstcr	0.77720	0.00055
SW	BI	enrwcrnr	0.77739	0.00025
SW	BI	enrwcrst	0.77880	0.00041
SW	BI	enrwnrcrst	0.78561	0.00033
SW	BI	enrwcrnrst	0.78171	0.00037
SW	BI	enrwnrstcr	0.78509	0.00045
SW	BI	enrwstnrcr	0.78200	0.00038
SW	BI	enrwcrstnr	0.77889	0.00041
SW	BI	enrwstcrnr	0.77722	0.00056
SW	UNIBI	enrw	0.77774	0.00034
SW	UNIBI	enrwnr	0.77832	0.00035
SW	UNIBI	enrwst	0.77764	0.00041
SW	UNIBI	enrwcr	0.77946	0.00033
SW	UNIBI	enrwnrst	0.78190	0.00040
SW	UNIBI	enrwnrcr	0.78397	0.00033
SW	UNIBI	enrwstnr	0.77823	0.00047
SW	UNIBI	enrwstcr	0.77951	0.00041

SW	UNIBI	enrwcrnr	0.77961	0.00033
SW	UNIBI	enrwcrst	0.78099	0.00042
SW	UNIBI	enrwnrcrst	0.78782	0.00038
SW	UNIBI	enrwcrnrst	0.78374	0.00044
SW	UNIBI	enrwnrstcr	0.78740	0.00045
SW	UNIBI	enrwstnrcr	0.78450	0.00044
SW	UNIBI	enrwcrstnr	0.78114	0.00042
SW	UNIBI	enrwstcrnr	0.77954	0.00041
NOSW	UNI	enrw	0.77643	0.00025
NOSW	UNI	enrwnr	0.77695	0.00023
NOSW	UNI	enrwst	0.77372	0.00010
NOSW	UNI	enrwcr	0.77867	0.00019
NOSW	UNI	enrwnrst	0.77316	0.00022
NOSW	UNI	enrwnrcr	0.78008	0.00019
NOSW	UNI	enrwstnr	0.77427	0.00009
NOSW	UNI	enrwstcr	0.77549	0.00025
NOSW	UNI	enrwcrnr	0.77895	0.00021
NOSW	UNI	enrwcrst	0.77540	0.00019
NOSW	UNI	enrwnrcrst	0.77637	0.00023
NOSW	UNI	enrwcrnrst	0.77504	0.00029
NOSW	UNI	enrwnrstcr	0.77575	0.00020
NOSW	UNI	enrwstnrcr	0.77726	0.00018
NOSW	UNI	enrwcrstnr	0.77567	0.00022
NOSW	UNI	enrwstcrnr	0.77555	0.00024
NOSW	BI	enrw	0.76371	0.00025
NOSW	BI	enrwnr	0.76427	0.00023
NOSW	BI	enrwst	0.76587	0.00047
NOSW	BI	enrwcr	0.76272	0.00043
NOSW	BI	enrwnrst	0.77115	0.00040
NOSW	BI	enrwnrcr	0.76739	0.00045
NOSW	BI	enrwstnr	0.76644	0.00052
NOSW	BI	enrwstcr	0.76371	0.00027
NOSW	BI	enrwcrnr	0.76281	0.00042
NOSW	BI	enrwcrst	0.76724	0.00033
NOSW	BI	enrwnrcrst	0.77462	0.00051

NOSW	BI	enrwcrnrst	0.77077	0.00037
NOSW	BI	enrwnrstcr	0.77253	0.00041
NOSW	BI	enrwstnrcr	0.76863	0.00039
NOSW	BI	enrwcrstnr	0.76734	0.00033
NOSW	BI	enrwstcrnr	0.76373	0.00027
NOSW	UNIBI	enrw	0.76985	0.00014
NOSW	UNIBI	enrwnr	0.77035	0.00018
NOSW	UNIBI	enrwst	0.77115	0.00032
NOSW	UNIBI	enrwcr	0.76864	0.00016
NOSW	UNIBI	enrwnrst	0.77559	0.00031
NOSW	UNIBI	enrwnrcr	0.77309	0.00026
NOSW	UNIBI	enrwstnr	0.77171	0.00033
NOSW	UNIBI	enrwstcr	0.76894	0.00042
NOSW	UNIBI	enrwcrnr	0.76878	0.00018
NOSW	UNIBI	enrwcrst	0.77240	0.00032
NOSW	UNIBI	enrwnrcrst	0.77923	0.00030
NOSW	UNIBI	enrwcrnrst	0.77541	0.00035
NOSW	UNIBI	enrwnrstcr	0.77725	0.00041
NOSW	UNIBI	enrwstnrcr	0.77397	0.00040
NOSW	UNIBI	enrwcrstnr	0.77255	0.00028
NOSW	UNIBI	enrwstcrnr	0.76897	0.00043

Tabela 4: Valores para a classificação do Corpus-EN utilizando SVM.

Stop-words	N-grama	Pré-processamento	F-Measure	Erro(+/-)
SW	UNI	enrw	0.83055	0.00039
SW	UNI	enrwnr	0.83073	0.00039
SW	UNI	enrwst	0.82819	0.00046
SW	UNI	enrwcr	0.82934	0.00055
SW	UNI	enrwnrst	0.82825	0.00039
SW	UNI	enrwnrcr	0.83013	0.00057
SW	UNI	enrwstnr	0.82838	0.00043
SW	UNI	enrwstcr	0.82574	0.00055
SW	UNI	enrwcrnr	0.8294	0.00058
SW	UNI	enrwcrst	0.82673	0.00059

SW	UNI	enrwnrcrst	0.82738	0.00045
SW	UNI	enrwcrnrst	0.82673	0.00048
SW	UNI	enrwnrstcr	0.82636	0.0004
SW	UNI	enrwstnrcr	0.82671	0.00036
SW	UNI	enrwcrstnr	0.82679	0.00059
SW	UNI	enrwstcrnr	0.82575	0.00055
SW	BI	enrw	0.82516	0.00035
SW	BI	enrwnr	0.82541	0.00032
SW	BI	enrwst	0.82495	0.00056
SW	BI	enrwcr	0.82497	0.00027
SW	BI	enrwnrst	0.82757	0.00045
SW	BI	enrwnrcr	0.82752	0.00017
SW	BI	enrwstnr	0.82517	0.00054
SW	BI	enrwstcr	0.82489	0.00041
SW	BI	enrwcrnr	0.82501	0.00026
SW	BI	enrwcrst	0.82563	0.00032
SW	BI	enrwnrcrst	0.82943	0.00035
SW	BI	enrwcrnrst	0.82723	0.00027
SW	BI	enrwnrstcr	0.82931	0.00038
SW	BI	enrwstnrcr	0.82762	0.00041
SW	BI	enrwcrstnr	0.82566	0.00033
SW	BI	enrwstcrnr	0.82489	0.00041
SW	UNIBI	enrw	0.84228	0.00062
SW	UNIBI	enrwnr	0.84249	0.00061
SW	UNIBI	enrwst	0.84136	0.00062
SW	UNIBI	enrwcr	0.84177	0.0005
SW	UNIBI	enrwnrst	0.84231	0.00071
SW	UNIBI	enrwnrcr	0.84296	0.00055
SW	UNIBI	enrwstnr	0.84154	0.0006
SW	UNIBI	enrwstcr	0.84036	0.00068
SW	UNIBI	enrwcrnr	0.84181	0.0005
SW	UNIBI	enrwcrst	0.8409	0.00058
SW	UNIBI	enrwnrcrst	0.84263	0.00058
SW	UNIBI	enrwcrnrst	0.84158	0.00057
SW	UNIBI	enrwnrstcr	0.84219	0.0006

SW	UNIBI	enrwstnrcr	0.84172	0.0006
SW	UNIBI	enrwcrstnr	0.84095	0.00058
SW	UNIBI	enrwstcrnr	0.84037	0.00068
NOSW	UNI	enrw	0.82501	0.00043
NOSW	UNI	enrwnr	0.82523	0.00044
NOSW	UNI	enrwst	0.82265	0.00044
NOSW	UNI	enrwcr	0.82357	0.00044
NOSW	UNI	enrwnrst	0.82271	0.00033
NOSW	UNI	enrwnrcr	0.82427	0.0005
NOSW	UNI	enrwstnr	0.82285	0.00044
NOSW	UNI	enrwstcr	0.81964	0.00048
NOSW	UNI	enrwcrnr	0.82361	0.00047
NOSW	UNI	enrwcrst	0.82092	0.00052
NOSW	UNI	enrwnrcrst	0.82152	0.00045
NOSW	UNI	enrwcrnrst	0.82094	0.00048
NOSW	UNI	enrwnrstcr	0.82016	0.00035
NOSW	UNI	enrwstnrcr	0.82066	0.00036
NOSW	UNI	enrwcrstnr	0.82098	0.00053
NOSW	UNI	enrwstcrnr	0.81964	0.00049
NOSW	BI	enrw	0.7945	0.00034
NOSW	BI	enrwnr	0.79497	0.00036
NOSW	BI	enrwst	0.79908	0.00051
NOSW	BI	enrwcr	0.79448	0.00039
NOSW	BI	enrwnrst	0.80394	0.00055
NOSW	BI	enrwnrcr	0.79851	0.00047
NOSW	BI	enrwstnr	0.79948	0.00054
NOSW	BI	enrwstcr	0.79873	0.0005
NOSW	BI	enrwcrnr	0.79456	0.0004
NOSW	BI	enrwcrst	0.80083	0.00038
NOSW	BI	enrwnrcrst	0.80693	0.00046
NOSW	BI	enrwcrnrst	0.80385	0.00049
NOSW	BI	enrwnrstcr	0.80564	0.0005
NOSW	BI	enrwstnrcr	0.80266	0.00061
NOSW	BI	enrwcrstnr	0.80089	0.00038
NOSW	BI	enrwstcrnr	0.79873	0.0005

NOSW	UNIBI	enrw	0.83055	0.00048
NOSW	UNIBI	enrwnr	0.83079	0.00044
NOSW	UNIBI	enrwst	0.83002	0.00043
NOSW	UNIBI	enrwcr	0.82929	0.00051
NOSW	UNIBI	enrwnrst	0.83101	0.00032
NOSW	UNIBI	enrwnrcr	0.83044	0.00059
NOSW	UNIBI	enrwstnr	0.83025	0.00044
NOSW	UNIBI	enrwstcr	0.82735	0.00048
NOSW	UNIBI	enrwcrnr	0.82935	0.00052
NOSW	UNIBI	enrwcrst	0.82901	0.0005
NOSW	UNIBI	enrwnrcrst	0.83066	0.0004
NOSW	UNIBI	enrwcrnrst	0.82975	0.00035
NOSW	UNIBI	enrwnrstcr	0.82919	0.00037
NOSW	UNIBI	enrwstnrcr	0.82871	0.00041
NOSW	UNIBI	enrwcrstnr	0.82904	0.00049
NOSW	UNIBI	enrwstcrnr	0.82736	0.00047

Tabela 5: Valores para a classificação do Corpus-EN variando o seu tamanho.

	Naive Bayes		SVM	
Porcentagem	F-Measure	Erro	F-Measure	Erro
0.001	0.61508	0.01381	0.79287	0.01071
0.002	0.63239	0.01462	0.80610	0.01125
0.003	0.64559	0.00848	0.80684	0.00998
0.004	0.65495	0.00838	0.81118	0.00664
0.005	0.66091	0.00249	0.81568	0.00694
0.006	0.6649	0.00658	0.81656	0.00746
0.007	0.67098	0.00512	0.82111	0.00627
0.008	0.6753	0.00463	0.82132	0.00341
0.009	0.68035	0.0052	0.82421	0.00615
0.01	0.68379	0.00267	0.82443	0.00452
0.01	0.68379	0.00267	0.82443	0.00452
0.02	0.708	0.00368	0.83083	0.00191
0.03	0.72042	0.0022	0.83346	0.00204
0.04	0.72964	0.00214	0.83564	0.00246

0.05	0.73537	0.00238	0.83667	0.00242
0.05	0.73537	0.00238	0.83667	0.00242
0.06	0.73956	0.00226	0.83745	0.00155
0.07	0.74362	0.00155	0.83810	0.00185
0.08	0.74751	0.00144	0.83897	0.00242
0.09	0.74985	0.0018	0.83975	0.00199
0.1	0.75244	0.00039	0.84016	0.00225
0.1	0.75244	0.00039	0.84016	0.00225
0.11	0.75409	0.00077	0.84044	0.00181
0.12	0.75629	0.00097	0.84076	0.00168
0.13	0.75768	0.00041	0.84099	0.00206
0.14	0.75929	0.00092	0.84143	0.00194
0.15	0.76086	0.00091	0.84179	0.00157
0.15	0.76086	0.00091	0.84179	0.00157
0.16	0.7619	0.00126	0.84220	0.00146
0.17	0.763	0.00085	0.84221	0.00102
0.18	0.76413	0.00099	0.84239	0.00086
0.19	0.76491	0.00093	0.84255	0.00127
0.2	0.76611	0.00077	0.84272	0.00034
0.2	0.76612	0.00077	0.84272	0.00033
0.25	0.76976	0.00125	0.84307	0.00170
0.3	0.77262	0.0012	0.84335	0.00142
0.35	0.77511	0.0008	0.84398	0.00146
0.4	0.77697	0.00087	0.84415	0.00104
0.45	0.77896	0.00088	0.84455	0.00124
0.5	0.78033	0.00081	0.84474	0.00113
0.55	0.78143	0.0005	0.84489	0.00097
0.6	0.78245	0.00055	0.84514	0.00125
0.65	0.78338	0.00041	0.84545	0.00107
0.7	0.78416	0.00051	0.84563	0.00101
0.75	0.78485	0.00073	0.84582	0.00109
0.8	0.7856	0.00072	0.84595	0.00096
0.85	0.78621	0.00061	0.84610	0.00092
0.9	0.78681	0.00078	0.84627	0.00083
0.95	0.78729	0.00052	0.84625	0.00073

1	0.7878	0.00067	0.84647	0.00094

Tabela 6: Valores para a classificação do Corpus-PT variando o seu tamanho.

	Naive Bayes		SVM	
Porcentagem	F-Measure	Erro	F-Measure	Erro
0.001	0.66088	0.03168	0.77029	0.02054
0.002	0.66338	0.02622	0.78301	0.01667
0.003	0.68172	0.00514	0.79126	0.01934
0.004	0.68055	0.02	0.80108	0.02150
0.005	0.68315	0.01356	0.80260	0.01227
0.006	0.68527	0.01262	0.80373	0.01626
0.007	0.69416	0.01122	0.80479	0.00792
0.008	0.69905	0.01119	0.80710	0.01125
0.009	0.70428	0.01197	0.80790	0.00544
0.01	0.70613	0.00989	0.80718	0.00576
0.01	0.70613	0.00989	0.80718	0.00576
0.02	0.71999	0.00765	0.81216	0.00938
0.03	0.72716	0.00521	0.81447	0.00482
0.04	0.73215	0.00504	0.81517	0.00656
0.05	0.73664	0.00248	0.81702	0.00423
0.05	0.73664	0.00248	0.81702	0.00423
0.06	0.74076	0.00408	0.81811	0.00305
0.07	0.74285	0.00176	0.81861	0.00250
0.08	0.74509	0.00373	0.81932	0.00306
0.09	0.74817	0.00272	0.81945	0.00230
0.1	0.75006	0.0029	0.82010	0.00242
0.1	0.75006	0.0029	0.82010	0.00242
0.11	0.75222	0.00358	0.82063	0.00116
0.12	0.75365	0.00099	0.82106	0.00126
0.13	0.7547	0.00178	0.82087	0.00246
0.14	0.7553	0.00149	0.82132	0.00242
0.15	0.75611	0.00273	0.82173	0.00362
0.15	0.75611	0.00273	0.82173	0.00362
0.16	0.75716	0.00208	0.82202	0.00249

0.17	0.75813	0.00198	0.82156	0.00245
0.18	0.75923	0.00176	0.82206	0.00266
0.19	0.76043	0.00235	0.82231	0.00232
0.2	0.76192	0.00171	0.82287	0.00329
0.2	0.76192	0.00171	0.82287	0.00329
0.25	0.76665	0.00281	0.82398	0.00248
0.3	0.77014	0.00231	0.82492	0.00240
0.35	0.77259	0.00207	0.82503	0.00185
0.4	0.77501	0.00239	0.82544	0.00198
0.45	0.77679	0.00198	0.82617	0.00153
0.5	0.77844	0.00174	0.82637	0.00149
0.55	0.7794	0.0014	0.82644	0.00171
0.6	0.78049	0.00162	0.82657	0.00171
0.65	0.78167	0.00119	0.82707	0.00069
0.7	0.78236	0.00125	0.82714	0.00092
0.75	0.7832	0.00181	0.82703	0.00163
0.8	0.7839	0.00162	0.82726	0.00132
0.85	0.78443	0.0017	0.82741	0.00158
0.9	0.785	0.00143	0.82741	0.00122
0.95	0.7858	0.00157	0.82737	0.00108
1	0.78646	0.00136	0.82758	0.00087