Universidade Federal Rural de Pernambuco
Departamento de Estatística e Informática

Aplicação de Recuperação de Informação e Análise de Sentimentos para Suporte à Pesquisas de Mercado

Silas José da Silva Junior

Recife

Julho de 2015

Silas José da Silva Junior

Aplicação de Recuperação de Informação e Análise de Sentimentos para Suporte à Pesquisas de Mercado

Orientadora: Teresa M. Maciel

Coorientador: Giordano R. E. Cabral

Monografia apresentada ao Curso de Bacharelado em Sistemas de Informação da Universidade Federal Rural de Pernambuco, como requisito parcial para obtenção do título de Bacharel em Sistemas de Informação.

Recife

Julho de 2015

A minha querida namorada, Eliane.

Ao meu pai, Silas.

A minha mãe, Mirian.

A minha irmã, Suelen.

Agradecimentos

Agradeço primeiramente a Deus, que me deu a oportunidade de hoje estar vivendo este momento e por ter colocado as pessoas certas em minha vida.

À minha família que tem me apoiado em qualquer circunstância e que me incentivou a ganhar mais conhecimento e sempre procurar me aprimorar naquilo que faço.

Agradeço também a minha namorada pela paciência e por todo apoio dado a mim, você é demais.

Aos meus orientadores Teresa Maciel e Giordano Cabral pelos ensinamentos e pelo papel fundamental no desenvolvimento deste trabalho.

Ao Gleibson, que nesta reta final teve um papel fundamental no desenvolvimento deste trabalho.

Não poderia esquecer de duas pessoas que tiveram um papel essencial para que hoje eu estivesse entregando este trabalho, meus amigos Heitor Fagner e Gilberto Junior, agradeço a Deus pelas suas vidas e por todo investimento de tempo que vocês tiveram comigo. Esse trabalho também é conquista de vocês.

Obrigado a todos.

Resumo

Em virtude do grande aumento da quantidade de dados alimentados na internet e a necessidade de acesso a esses dados de forma precisa, mecanismos de recuperação de informação e análise de sentimentos têm ganhado forças, pois os usuários da rede se sentem mais confortáveis em expressar suas opiniões através de meios eletrônicos, despertando assim um interesse por empresas e empreendedores em busca de informações que remetam a opinião de usuários da rede em relação a marcas, produtos, serviços, dentre outros. Com o objetivo de classificar comentários ou documentos surge a análise de sentimentos, que também é conhecida como mineração de opinião, em busca de opiniões expressas sobre um determinado tema. Neste contexto foi realizado um trabalho de utilização de técnicas tanto de recuperação de informação, quanto de análise de sentimentos, para dar suporte à empresas e empreendedores a fazerem pesquisas de mercado mais precisas, levando em consideração as opiniões dos usuários da rede. Utilizando-se como base de pesquisa o Twitter, onde são postados diariamente uma quantidade considerável de opiniões, foi necessário a utilização de um conjunto de APIs e técnicas para classificar cada opinião expressa nas postagens.

Palavras-chave: recuperação de informação, análise de sentimentos, PLN, processamento de linguagem natural, twitter, sentiment140, microsoft translator api, treetagger, tweetinvi, adjetivos, hashtags, expressão regular

Abstract

Given the great increase in the amount of data fed through the Internet, and the need to access those data in an accurate manner, mechanisms to information retrieval and sentiment analysis have gained much popularity lately. That is due to the fact that network users feel more comfortable expressing their opinions through electronic means. That in turn has sparked the interest of companies and entrepreneurs who are in search of information that reflects the opinion of network users concerning brands, products, services, and other variables.

The goal of classifying comments or documents gives rise to the sentiment analysis, which is also known as “opinion mining”, a process used to research the opinions expressed about a given theme. Within that context, a study of the use of information retrieval and sentiment analysis techniques was done in order to give companies and entrepreneurs the support necessary to make a more accurate market research that takes into account the opinion of the network users. Since Twitter was used as the data source due to the considerable amount of opinions it broadcasts on a daily basis, it was necessary to use a group of APIs and techniques to classify each opinion expressed on Twitter posts.

Keywords: information retrieval, sentiment analysis, PLN, processing of natural language, twitter, sentiment140, microsoft translator api, treetagger, tweetinvi, adjectives, hashtags, regular expression.

Lista de Tabelas

Lista de Figuras

Lista de gráficos

Capítulo 1

Introdução

Pesquisas de mercado tem sido bastante importante para empreendedores que desejam disponibilizar um novo produto ou serviço no mercado, pois o ajuda a ter uma noção se aquele serviço ou produto terá uma boa aceitação para uma determinada região, determinada faixa etária, entre outros. Segundo o SEBRAE [1], todo tipo de decisão relacionada a novos empreendimentos ou a novos produtos tem um grau de incerteza consideravelmente alto, tanto no que diz respeito à informação que as decisões estão baseadas, como nas suas consequências.

Grande parte das pesquisas de mercado são feitas de forma trabalhosa, o que muitas vezes maquiam os resultados finais. Um dos mecanismos mais usados para se fazer uma pesquisa de mercado é o formulário de pesquisa, que nem sempre é eficaz, pois estes podem se tornar cansativos, fazendo com que as pessoas que o estão respondendo, o façam de qualquer forma, sem dar a devida atenção.

Em busca de pesquisas que tragam retornos mais rápidos e fidedignos em pesquisas de mercado, uma alternativa que pode trazer bons resultados seria o uso de informações que estão sendo alimentadas diariamente por usuários da internet, através de redes sociais, blogs e micro blogs, uma vez que, estes usuários estão constantemente expressando suas opiniões sobre um determinado tema, serviços e produtos de forma livre, criando assim uma base de informação gigantesca para ser explorada.

Este trabalho está inserido neste contexto, aplicando áreas de conhecimentos tais como recuperação de informação e análise de sentimentos para propor alternativas de soluções para este problema.

1.1 Apresentação do Trabalho

Este trabalho tem como objetivo principal criar uma aplicação de software, utilizando a tecnologia de recuperação de informação e análise de sentimentos, visando dar suporte a empreendedores a fazerem pesquisas de mercado com o mínimo de esforço possível, de forma rápida, cômoda e automatizada. A aplicação usa como base de pesquisa a rede social Twitter, que tem gerado diariamente uma grande

quantidade de informação na internet, tanto por empresas especializadas em notícias, serviços e produtos, quanto de usuários comuns.

No mês de janeiro de 2010, a quantidade de tweets foi de 1,2 bilhões no mundo inteiro, o equivalente a 40 milhões de tweets por dia [20]. No site da *onesecond* em [13], é atualizada a cada segundo a quantidade de informações geradas por cada uma das principais redes sociais na internet. Dentre elas está o Twitter, que contabiliza aproximadamente 4000 tweets por segundos, o que equivale a cerca de 345.600.000 tweets por dia. Levando em consideração que o mês tenha 30 dias chegamos a uma quantia de 10.368.000.000 tweet por mês.

Como o foco deste trabalho é fazer análises a partir dos posts gerados pelos usuários no Twitter, são aplicadas técnicas de recuperação de informação para extrair dados relevantes para pesquisas de mercado. Complementarmente técnicas de análise de sentimentos são aplicadas para refinar esta extração de informações.

1.1.1 Objetivo Geral

Desenvolver um software com acesso a plataforma web, que possibilite fazer consultas e extração automática no Twitter, tendo como objeto de pesquisa uma palavra-chave, com o foco em tornar mais fácil a obtenção e análise de dados para uma pesquisa de mercado.

Para isto este software contará com uma série de técnicas que ajudarão a fazer esta análise como: técnicas de análise de sentimento, técnicas de análise de relevância e uma API do Twitter que possibilitará extrair o conteúdo das postagens relacionadas ao tema da pesquisa, verificando assim padrões na composição dos tweets obtidos, identificando assim um grau de relevância e um real interesse sobre determinado tema, ou um não interesse sobre o mesmo.

1.1.2 Objetivos Específicos

Os objetivos específicos deste trabalho são:

- Gerar conhecimento sobre a aplicabilidade das áreas de recuperação da informação e análise de sentimento em problemas reais.
- Aplicar técnicas de análise de sentimentos, para identificar opiniões expressadas dentro de um determinado contexto.

- Desenvolver um software que utilize um algoritmo de classificação estrutural para cada tweet, a fim de identificar relevância e ranquear cada tweet conforme seu grau de relevância.
- Utilizar uma API do Twitter para extração de informações relacionadas ao tema da pesquisa.

1.2 Resultados Gerados

O desenvolvimento deste trabalho de conclusão de curso gerou para a comunidade acadêmica e científica um conhecimento sobre aplicação de áreas da computação para solução de problemas reais, particularmente para agilização de pesquisas de mercado. Com este foco, foram desenvolvidos os seguintes produtos:

- Criação de um software que permite ao usuário fazer buscas no Twitter, através de uma palavra-chave.
- Análise automatizada de cada tweet de forma estrutural e identificação se determinado tweet é relevante ou não.
- Criação de código para limpeza de caracteres ou termos contraídos, substituindo-os por palavras que possuam o mesmo significado.
- Criação de código que se conecte com a API de Tradução automática, com o objetivo de traduzir cada tweet para o inglês para que possa ser feita a classificação de opinião de forma mais eficiente.
- Utilização automática de ferramenta de análise de sentimento (Sentiment140), para classificação de cada tweet, extraindo assim de seu conteúdo a opinião expressa por cada usuário que o postou na rede.
- Avaliação automática em busca de hashtags padrão em cada tweet que foi classificado com êxito, com o objetivo de identificar hashtags padrões, para poder classificar os tweets que não conseguiram ser classificados anteriormente.
- Avaliação automática de forma gramatical cada tweet que foi classificado com êxito, com o objetivo de identificar adjetivos que caracterizam o sentido da frase, para poder classificar os tweets que não conseguiram ser classificados anteriormente.

1.3 Organização do documento

Este documento está organizado da seguinte forma: este capítulo introdutório compreendeu uma apresentação do trabalho e os resultados gerados por ele. No capítulo 2 é apresentada a metodologia utilizada para o desenvolvimento deste trabalho. No capítulo 3 são apresentados os fundamentos necessários para um melhor entendimento do problema e melhor entendimento das técnicas existentes para se chegar a solução proposta neste trabalho. No capítulo 4 é apresentada a forma de desenvolvimento, a lógica contida dentro da aplicação resultante deste trabalho e tecnologias utilizadas. No capítulo 5 são apresentados os resultados obtidos neste trabalho. E por fim no capítulo 6 é apresentada a conclusão deste trabalho e algumas sugestões do que se pode ser feito para dar continuidade a este trabalho no futuro.

Capítulo 2

Metodologia de desenvolvimento

Este capítulo descreve a metodologia utilizada para criação da aplicação de software relatada na sessão introdutória deste documento.

A solução proposta por este trabalho consiste em se certificar de que é possível utilizar recuperação de informação e análise de sentimentos para extrair opiniões expressas em postagens feitas por usuários através do Twitter, a fim de agilizar o processo de pesquisas de mercado para empresas ou empreendedores, tornando este processo mais preciso e seguro. O processo seguido no intuito de alcançar este objetivo encontra-se descrito na Figura 2.1.

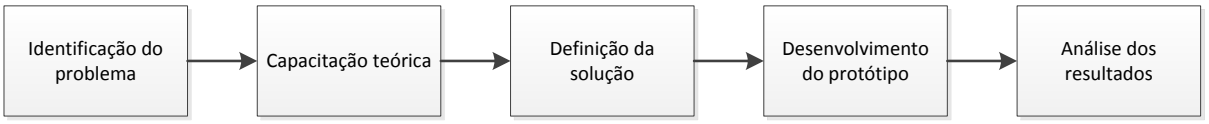

Figura 2.1 - Metodologia adotada

Inicialmente a identificação do problema surgiu de sugestão de um dos orientadores que havia identificado o problema. E como existia um interesse pessoal por redes sociais e pesquisas, a sugestão dada se tornou agradável para ser solucionada.

Já com o problema identificado, foram feitas pesquisas exploratórias com o intuito de tornar cada vez mais sólido o conhecimento sobre as áreas de estudo que possibilitariam a resolução do problema identificado, através de pesquisas em livros, artigos, monografias, dentre outros. Em seguida, com uma base teórica fundamentada, foram propostas soluções para o problema identificado, chegando a ser cogitado o uso de classificadores baseados em aprendizado de máquina, como o KNN (*k*-Nearest Neighbors), análise de sentimentos, recuperação de informação e processamento de linguagem natural.

Inicialmente foi feita uma implementação da solução utilizando o algoritmo KNN, onde foram encontradas dificuldades com a classificação dos tweets, levando a uma taxa de erros e acertos muito parecidas. Sendo necessário voltar a fase de definição de solução, com o objetivo de estruturar uma nova solução para o problema.

Com a nova solução definida entrou-se novamente na etapa de desenvolvimento onde foi preciso definir quais ferramentas, APIS e técnicas seriam utilizadas na construção da aplicação de software resultante deste trabalho, dando início ao desenvolvimento da solução atual.

Ao término do desenvolvimento, foi criado um protótipo responsável por classificar um conjunto de tweets em busca de opiniões sobre um determinado tema, utilizando-se do conjunto de ferramentas e APIs estudadas em etapas anteriores, a fim de classificar tweets como “positivos” ou “negativos”.

Com os resultados retornados pela aplicação foi iniciada a etapa de análise dos resultados, a fim de identificar os pontos considerados como eficientes e deficientes da aplicação.

Capítulo 3

Fundamentação teórica

Esta seção apresenta um conjunto de informações que serviram de base de conhecimento para o tema desta proposta de trabalho.

As áreas de conhecimento, assim como trabalhos relacionados estão descritos nas seções a seguir.

3.1 Expressão regular

Expressões regulares são padrões de caracteres que associam sequencias de caracteres no texto. Podem ser usados com o objetivo de pesquisar, validar, substituir e extrair determinadas porções de textos, como: endereços ou links de imagens em uma página HTML, modificar formato de texto ou remover caracteres inválidos.

Aurélio Marinho Vargas afirma em [19] que expressões regulares são uma composição de símbolos, caracteres com funções especiais, que, agrupados entre si e com caracteres literais, formam uma sequência, uma expressão. Essa expressão é interpretada como uma regra que indicará sucesso se uma entrada de dados qualquer casar com essa regra, ou seja, obedecer exatamente a todas as suas condições.

Para construir uma expressão regular é preciso entender estruturalmente como organizar a sequência de metacaracteres para se obter o resultado desejado.

Na Tabela 3.1 abaixo é mostrado alguns exemplos de metacaracteres usados em expressões regulares com suas funções:

Metacaractere	Nome	Função
.	Ponto	Um caractere qualquer
[...]	Lista	Lista de caracteres permitidos
[^...]	Lista negada	Lista de caracteres proibidos
?	Opcional	Zero ou um

*	Asterisco	Zero, um ou mais
+	Mais	Um ou mais
{n, m}	Chaves	De *n* até *m*
^	Circunflexo	Início da linha
$	Cifrão	Fim da linha
\b	Borda	Início ou fim de palavra
\c	Escape	Torna literal o caractere *c*
|	Ou	Ou um ou outro
(...)	Grupo	Delimita um grupo
\1...\9	Retrovisor	Texto casado nos grupos 1...9
\d	Sequência	Sequência de caracteres de 0 a 9.

Tabela 3.1 - Metacaracteres usados em expressões regulares

Como forma de pesquisar, validar, substituir e extrair determinadas porções de textos, é preciso estruturar sua expressão de forma que ela retorne o resultado desejado.

Logo abaixo na Tabela 3.2 são apresentados alguns exemplos de uso de expressões regulares.

Tipo	Objeto	Expressão regular	Detalhes
Placa de automóvel	PGH-4562	[A-Z]{3}-[0-9]{4}	**[A-Z]{3}** sequência de 3 dígitos de A à Z, seguido de um "-" (hífen), seguido de **[0-9]{4}** uma sequência de 4 dígitos de 0 a 9.

CPF	777.777.777-77	\d{3}.\d{3}.\d{3}-\d{2}	Onde **\d{3}.** significa uma sequência de 3 dígitos de 0 a 9 seguida de um "." (ponto). **\d{3}-** significa uma sequência de 3 dígitos de 0 a 9 seguida de um "-" (hífen). **\d{2}** significa uma sequência de 2 dígitos de 0 a 9.
URL	http://www.google.com	(http|https)://(www).[a-z]+.[a-z]+.[a-z]*	**(http|https)://** sequência exata de "http" ou "https" seguida de "**://**". Seguida de **(www).** sequência exata de "www" seguido de um "**.**" (ponto). **[a-z]+.** uma ou mais ocorrência de sequência de "a" a "z" seguida de "**.**" (ponto). **[a-z]*** zero, um ou várias ocorrências de "a" a "z'.

Tabela 3.2 - Uso de expressões regulares

3.2 Recuperação de informação

Recuperação de informação (RI) é uma subárea da ciência da computação que estuda o armazenamento e recuperação automática de documentos, que são objetos de dados, geralmente em forma de textos, tais como documentos diversos, páginas Web e livros, embora possa conter também outros tipos de conteúdo, como: imagens, áudios e gráficos.

Com a rápida expansão da Web, gerou também um aumento no número de desafios para as técnicas de recuperação de informação disponíveis. Fazendo com que fosse preciso dar mais importância aos estudos de novas técnicas para recuperação de informação.

3.2.1 Modelos Clássicos

Na recuperação de informação existem três modelos clássicos utilizados no processo de RI (booleano, vetorial e probabilístico). Todos estes modelos apresentam uma estratégia de busca de documentos relevantes para uma consulta.

Estes modelos consideram que cada documento é composto por um grupo de palavras chaves, chamadas de termos de indexação [4]. A cada termo de indexação no documento é atribuído um peso maior que zero, que quantifica a correlação entre os termos e o documento.

Além destes modelos, também existem outros modelos bastante avançados que foram propostos ao longo dos anos, para resolver de forma eficaz o problema de recuperação de informação, dentre estes, destacam-se modelos baseados em bases de conhecimento, lógica fuzzi e redes neurais. Na figura 3.1 são mostrados os modelos clássicos da recuperação de informação.

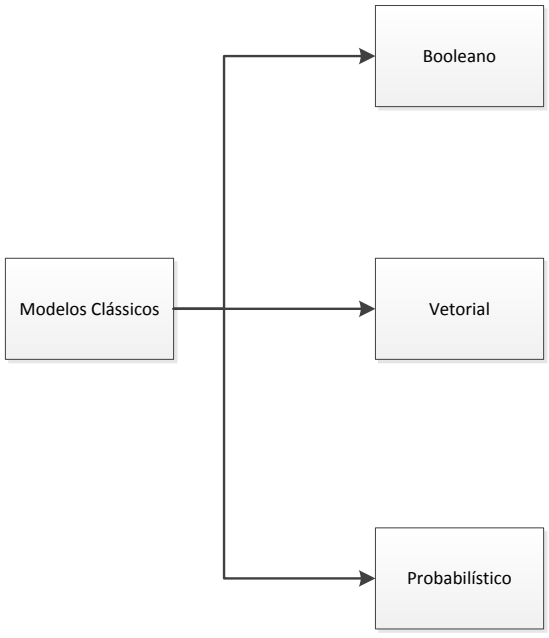

Figura 3.1 - Modelos clássicos da recuperação de informação

3.2.1.1 Modelo Booleano

O modelo booleano trata-se de um modelo bem simples, baseado na teoria dos conjuntos. As consultas são definidas por meio de expressões booleanas compostas por termos de índices e conectores *not, and* e *or*.

Neste modelo os termos de índices assumem valores binários, pois neste modelo, os termos de índice existem ou não existem em um documento.

Baeza-Yates e Ribeiro-Neto [5] dizem que no modelo Booleano, as variáveis do peso associado a um termo de índice assumem valores binários, isto é, $w_{ij} \in \{0,1\}$. Uma consulta q é uma expressão Booleana convencional. Seja $\vec{q}_{dnf}$ a forma normal disjuntiva da consulta q. Além disso, seja $\vec{q}_{cc}$ qualquer componente conjuntivo de $\vec{q}_{dnf}$. A similaridade do documento d_j à consulta q é definida por:

$$sim(d_j, q) = \begin{cases} 1, & se\ \exists\ \vec{q}_{cc} \mid (\vec{q}_{cc} \in \vec{q}_{dnf}) \wedge (\forall k_i, g_i(\vec{d}_i) = g_i(\vec{q}_{cc})) \\ 0, & caso\ contrário \end{cases}$$

Uma das maiores desvantagens deste modelo trata-se da sua habilidade de ordenar os documentos resultantes de uma busca [31]. Por este motivo, geralmente este modelo é usado em conjunto com algum outro, pois muitos dos sistemas de busca da Web, onde o ordenamento dos documentos tem um grau de importância bastante alto, partiram do uso deste modelo. Além do uso em recuperação de informação e nos sistemas de buscas na Web, seu poder pode ser expressado no SQL.

3.2.1.2 Modelo Vetorial

O modelo de espaço vetorial, ou simplesmente modelo vetorial, representa documentos e consultas como vetores de termos. Termos são ocorrências únicas nos documentos. Associado a cada um dos termos, atribui-se um peso, um valor não binário (diferente do modelo booleano).

Baeza-Yates e Ribeiro-Neto [5] dizem que no modelo vetorial, o peso w_{ij} associado ao par (k_i, d_j) é positivo e não binário, onde k_i é um termo de índice e d_j um documento. Além disso, os termos de índices na consulta também são ponderados. Seja w_{iq} um peso associado ao par $[k_i, q]$, onde $w_{iq} \geq 0$. Então, o vetor da consulta $\vec{q}$ é definido como $\vec{q} = (w_{1,q}, w_{2,q}, \ldots, w_{t,q})$ onde *t* é o número total de termos de índices no sistema. O vetor do documento d_j é representado por $\overrightarrow{d_j} = (w_{1,j}, w_{2,j}, \ldots, w_{t,j})$.

Assim, dada uma consulta, o modelo vetorial avalia de forma ponderada o quanto o documento é relevante.

3.2.1.3 Modelo Probabilístico

O modelo probabilístico descreve documentos levando em consideração pesos binários que representam a presença ou ausência dos termos no documento. O vetor resultante gerado pelo modelo tem como base o cálculo da probabilidade de que um documento seja relevante para uma consulta. A principal ferramenta matemática do modelo probabilístico é o teorema de Bayes [4].

Em [25] é dito que este modelo diferentemente do vetorial não trabalha com pesos pré-definidos para os termos da consulta e dos documentos. Ao invés disso, são utilizadas técnicas de estatística, para o cálculo dinâmico dos termos da consulta em relação aos documentos para a posterior obtenção dos documentos mais relevantes. O modelo probabilístico baseia-se no Princípio da Ordenação Probabilística, que objetiva saber se um dado documento *D* é ou não relevante para uma consulta Q_a.

Embora o modelo probabilístico seja um modelo com um forte embasamento teórico, as formas assumidas para realizar simplificações nos cálculos probabilísticos podem deixar dúvidas sobre a sua precisão. Como um dos exemplos temos o uso de termos de indexação como binários, sem levar em consideração a frequência com que os termos ocorrem no corpo do documento [31].

3.3 Processamento de Linguagem Natural

Em [7] é dito que o Processamento de Linguagem Natural (PLN) consiste na aplicação de métodos e técnicas que possibilitem ao computador extrair a semântica da linguagem humana expressa em textos e voz. Trata-se de uma área da computação que estuda formas de desenvolver modelos computacionais para realização de tarefas como: tradução automática, sumarização automática de textos, ferramentas de auxílio à escrita, perguntas e respostas, categorização textual, recuperação e extração de informação.

Em [6] é afirmado que, hoje os estudos de PLN estão voltados para três aspectos da comunicação em linguagem natural:

1. Som: Fonologia

2. Estrutura: Morfologia e sintaxe
3. Significado: Semântica e pragmática

O primeiro ponto trata da fonologia, aspecto que está relacionado a identificação dos sons que compõem uma determinada língua. O ponto dois refere-se a parte morfológica e sintática das frases, identificando assim unidades primitivas que compõem cada frase (morfologia) e a definição da estrutura, com base na forma com que as palavras se relacionam dentro da frase (sintaxe). O terceiro ponto é voltado para duas formas de análise textual que definem o significado de uma determinada frase, onde primeiramente é feita uma análise semântica para identificar um significado para a frase, tendo como base sua estrutura sintática; e a pragmática que verifica se o significado associado a uma estrutura sintática é realmente o significado mais apropriado no contexto considerado.

3.3.1 Análise de Sentimentos

Análise de sentimento é uma área dentro de processamento de linguagem natural que estuda formas de identificar o humor ou opiniões de elementos subjetivos dentro de um texto. Basicamente, análise de sentimento tem a tarefa de identificar a opinião expressa dentro de um texto.

Existem 3 tipos de resultados possíveis retornados pela análise de sentimentos que são a análise positiva, onde foi identificado uma opinião positiva em relação a algum tema; a neutra, onde não foi possível identificar a opinião expressa no texto; a negativa, onde foi identificada uma opinião negativa a respeito de algum tema específico. Na figura 3.3 são ilustrados os tipos de respostas retornadas pela análise de sentimentos.

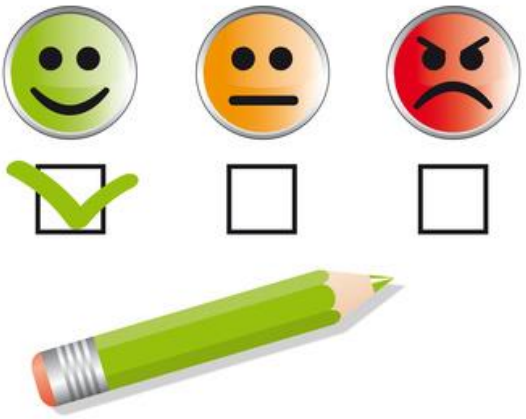

Figura 3.2 - Ilustração sobre análise de sentimento[27]

Segundo Pang e Lee [16] afirmam que, descobrir o que outras pessoas pensam sempre foi um objeto de interesse. Tal interesse pode ser notado nos usuários da internet, quando se é preciso adquirir algum serviço ou produto, muitas vezes procuram na internet em busca de opiniões de outros usuários sobre o tema que ele procura, como também se atualizam sobre política ou tendências de mercado.

Conforme é dito em [17], com a popularização das plataformas que fornecem acesso à grande quantidade de dados subjetivos, a tarefa de identificar a polaridade e tentar classificar qual emoção um texto possui, passou a ser o foco de diversas pesquisas, possibilitando assim para a análise de sentimentos uma grande quantidade de informação que venha a contribuir com seu desenvolvimento.

3.4 Pesquisas de mercado

A pesquisa de mercado trata-se de um estudo feito para determinar as perspectivas de venda de um determinado produto no mercado, indicando assim a maneira de se obter os melhores resultados [2]. Esse tipo de pesquisa busca identificar a aceitação de um determinado produto, e se este produto poderá ser vendido a um preço e em quantidades satisfatórias. Também permite analisar os mercados que oferecem melhores perspectivas, os padrões de qualidade exigidos pelo mercado importador e o tempo necessário para se alcançar o nível ideal de vendas.

A pesquisa de mercado é uma ferramenta muito importante que pode ajudar a economizar dinheiro e fornecer elementos que propiciem um melhor alinhamento com o mercado consumidor.

O Instituto PHD [18] afirma que, assim como muitas pessoas pesquisam preços de produtos ou serviços antes de decidirem qual mercadoria comprar, para fazer o dinheiro render e economizar um pouco, também os empreendedores devem pesquisar onde devem investir, para aplicar o dinheiro onde provavelmente irão ter maior retorno e maximizar seu investimento.

Este tipo de pesquisa trata-se de uma técnica que deve ser aplicada somente quando os resultados contribuírem para reduzir as incertezas ou influenciar decisões.

Existem 4 tipos de métodos utilizados pela pesquisa de mercado, que são: pesquisa de mercado qualitativa, pesquisa de mercado quantitativa, técnicas de observação e técnicas experimentais [30].

- Pesquisa de mercado qualitativa: geralmente utilizada para pequenos grupos de pesquisados. Neste tipo de pesquisa, se busca mais conteúdo informativo, visando assim ter uma análise das

informações com mais qualidade. Exemplos para este tipo de método são o *focus groups* (grupo focal), entrevistas em profundidade e técnicas de projeção.

- Pesquisa de mercado quantitativa: geralmente utilizada para tirar conclusões, testando assim uma hipótese específica. Este tipo de pesquisa em geral, busca um resultado estatístico. Para tal é utilizado um questionário estruturado.
- Técnicas de observação: o pesquisador observa o fenômeno natural no seu ambiente natural. Como exemplo temos a análise de uso de produtos e a utilização de cookies para observar o comportamento de usuários na internet.
- Técnicas experimentais: o pesquisador cria um ambiente quase-artificial para tentar controlar fatores espúrios e depois manipula pelo menos uma das variáveis. Exemplos são laboratórios de compra e testes de mercado e também mercearias.

Os objetivos da pesquisa de mercado são [2]:

- Selecionar mercados para a venda do produto;
- Identificar tendências e expectativas;
- Reconhecer a concorrência;
- Conhecer e avaliar oportunidades e ameaças.

3.5 Trabalhos relacionados

Nesta sessão são mostradas algumas ferramentas que buscam o mesmo objetivo ou que se assemelham aos objetivos deste trabalho.

3.5.1 Twitter advanced search

Ferramenta de busca avançada do próprio twitter, que disponibiliza ao usuário uma forma de fazer pesquisas mais elaboradas sobre um determinado tema. Com esta ferramenta o usuário é capaz de criar filtros de pesquisas, possibilitando ao mesmo construir uma pesquisa levando em consideração opções de pesquisas por palavras, pessoas, localização e datas como mostrado na Figura 3.4.

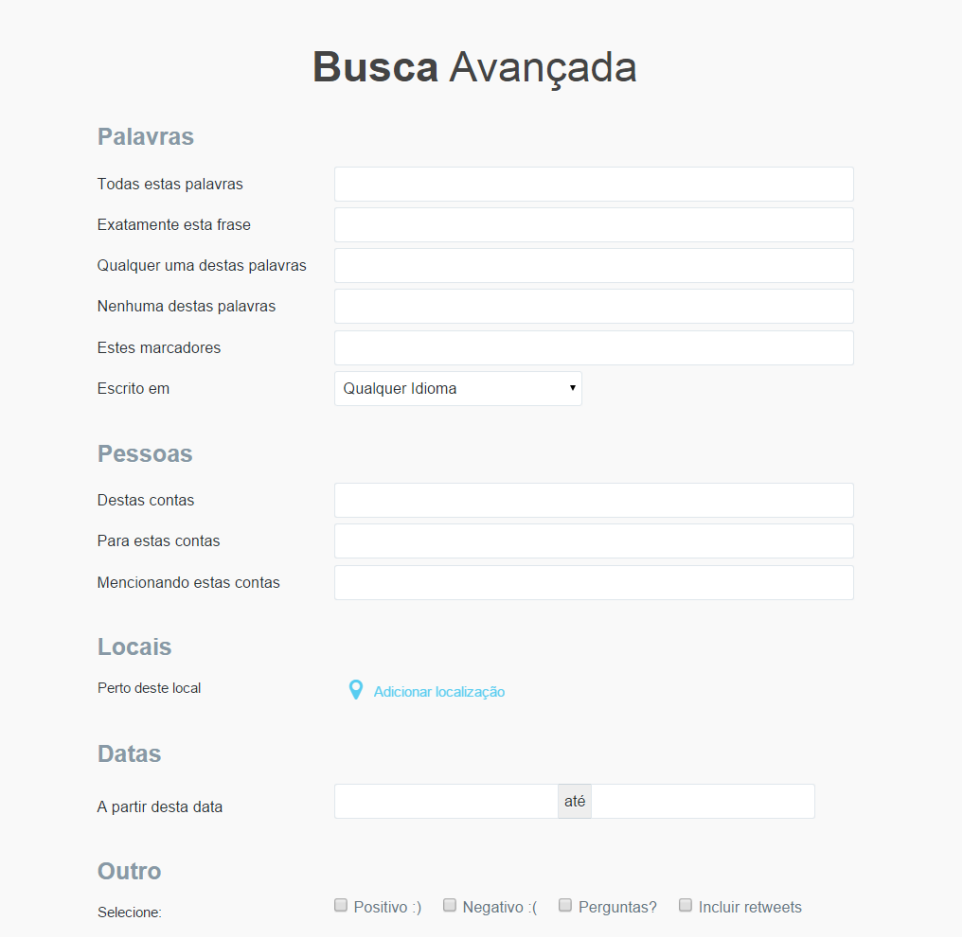

Figura 3.3 - Tela principal do Twitter Advanced Search [28]

Abaixo encontram-se alguns exemplos encontrados no próprio site do Twitter [14]:

Palavras

- Tweets que contêm todas as palavras em qualquer posição (“Twitter” e “busca”).
- Tweets que contêm frases exatas (“busca no Twitter”).
- Tweets que contêm qualquer uma das palavras (“Twitter” ou “busca”).
- Tweets que excluem palavras específicas (“Twitter”, mas não “busca”).
- Tweets com um marcador específico (#twitter).
- Tweets em um idioma específico (escritos em inglês).

Pessoas

- Tweets de uma conta específica (Tweetado por “@TwitterComms”).

- Tweets enviados como respostas a uma conta específica (em resposta a "@TwitterComms").
- Tweets que mencionam uma conta específica (o Tweet inclui "@TwitterComms").

Locais

- Tweets enviados de uma localização geográfica, como determinada cidade, estado, país.
- Use a lista suspensa de locais para selecionar a localização geográfica (selecionando a localização na opção "Adicionar localização").

Datas

- Tweets enviados antes de uma data específica, após uma data específica ou dentro de um intervalo de datas
- Use o calendário suspenso para selecionar a data inicial, a data final ou ambas (selecionando a data no calendário disponibilizado)
- Busca por Tweets de qualquer data desde o primeiro Tweet público

Outros

- Busca por tweets positivos ☺, negativos ☹, perguntas (?) e compostos por retweets.

Combinando os campos disponíveis na busca avançada do Twitter, é possível realizar pesquisas por tweets classificados como positivo ou negativo de forma inteligente, o que se assemelha ao tema deste trabalho. Ficando de forma livre para que o usuário pesquise conforme seus interesses.

3.5.2 Google Consumer Survey

O Google Consumer Surveys trata-se de uma ferramenta de negócio da Google, que serve para criação de enquetes e pesquisas de mercado customizadas.

Esta ferramenta possui três versões disponibilizadas, onde uma delas é uma versão "free", que possibilita o uso de perguntas padrão disponibilizadas pela ferramenta, e em caso de serem feitas algumas personalizações nas perguntas, será cobrado um valor de 1 centavo por pergunta. Depois de criada, as perguntas da pesquisa são disponibilizadas em sites de notícias e entretenimento, onde ficam incorporadas ao conteúdo e também através de um aplicativo para celular. Como forma de atrair o público pesquisado,

esta ferramenta troca as respostas fornecidas por acesso a conteúdo, e no caso dos celulares, as pessoas respondem as perguntas em troca de créditos para livros, músicas e apps [14].

Como forma de visualizar os resultados das pesquisas, fica disponível uma interface online, onde é possível analisá-los através de gráficos e segmentação demográfica clicável. Na Figura 3.5 é exemplificado o tipo de resultado gerado pela ferramenta.

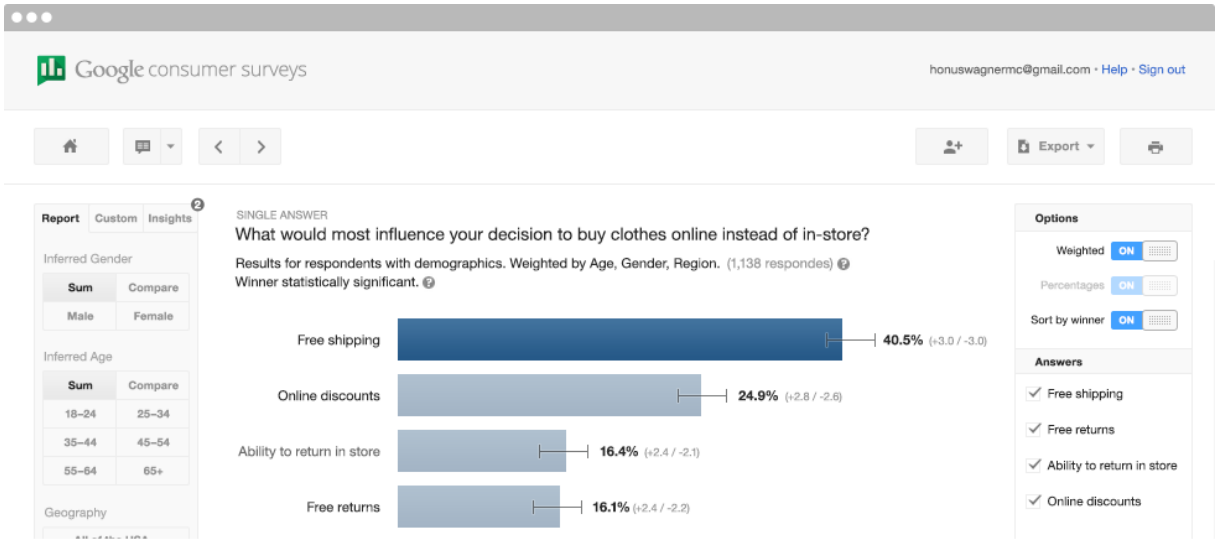

Figura 3.4 - Resultado de pesquisa pelo Google Consumer Survey [32].

Esta ferramenta da Google se assemelha ao tema deste trabalho, pois ela disponibiliza à empresas e empreendedores uma forma inteligente de se fazer pesquisas de mercado.

3.5.3 Improving Twitter Retrieval by Exploiting Structural Information

Este artigo científico desenvolvido por Zhunchen Luo, Miles Osborne, Sasa Petrovic e Ting Wang, na Universidade Nacional de Tecnologia de defesa na China [33], tem como objetivo a criação de uma aplicação de recuperação de informação voltada para o Twitter. Neste artigo os tweets embora possuam um tamanho curto, são estudados de forma estrutural para se obter tweets com conteúdo de melhor qualidade.

As estruturas estudadas neste artigo encontram-se detalhadas na Tabela 3.3 abaixo.

Estrutura	Descrição
TAG	Combinação de hashtag (#) e palavra-chave. Ex.: #iphone
MET	Menção a outro usuário do Twitter, caracterizando o direcionamento do twitter para o mesmo.
RWT	Para indicar a retransmissão do tweet original. Ex.: RT @ladygaga
URL	Links para conteúdos externos. Ex.: www.google.com.br
COM	Comentários descrevendo o sentimento e avaliação em relação a outra estrutura.
MSG	Conteúdo da mensagem do tweet.

Tabela 3.3 - Estruturas encontradas no tweet

Como contribuição deste trabalho foi desenvolvida uma forma de identificar as estruturas contidas no tweet e suas sequências, obtendo assim uma recuperação de informação mais confiável e de qualidade.

3.6 Ferramentas/API Utilizadas

Nesta seção é apresentado mais detalhes sobre o conjunto de APIs e ferramentas utilizadas para desenvolvimento deste trabalho.

3.6.1 TweetInvi API

O TweetInvi é uma biblioteca intuitiva do C#.Net que provê um rápido e intuitivo acesso para REST API que fornece acesso programático a leitura e escrita de dados no Twitter e Stream API 1.1 que dá ao desenvolvedor acesso a latência dos fluxos de dados globais do Twitter [12]. Esta API fornece ao usuário acesso a três classes diferentes para Twitter, que são a de usuário, tweets e mensagens.

Com o TweetInvi é possível estabelecer uma conexão com o Twitter, e disponibilizar para o usuário acesso ao conteúdo que é atualizado diariamente na rede. TweetInvi foi projetado para ser usado de forma simples, pois uma vez informada suas credenciais, a API disponibiliza para uso todos os métodos disponíveis na biblioteca.

Antes de utilizar qualquer método de aplicação terá de especificar suas credenciais. Na Figura 7 abaixo é mostrado um exemplo de como é feita a configuração das credenciais.

```
// Configure o seu credentials
TwitterCredentials.SetCredentials( "Access_Token" , "Access_Token_Secret" ,
"Consumer_Key" , "Consumer_Secret" );
```

Figura 3.5 - Configuração de credenciais do TweetInvi [22]

O sistema de credenciais foi projetado para ser bastante simples, precisando assim configurar as credenciais uma única vez. Após a configuração de credenciais de acesso a API, os métodos ficam disponíveis para que o usuário possa executar pesquisas, postagens e mensagens na rede.

Abaixo na Figura 3.7 é mostrado um exemplo de como é estruturada sua pesquisa.

```
// Search the tweets containing tweetinvi
var tweets = Search.SearchTweets("tweetinvi");

// Complex search
var searchParameter = Search.GenerateSearchTweetParameter("tweetinvi");
searchParameter.SetGeoCode(-122.398720, 37.781157, 1, DistanceMeasure.Miles);
searchParameter.Lang = Language.English;
searchParameter.SearchType = SearchResultType.Popular;
searchParameter.MaximumNumberOfResults = 100;
searchParameter.Until = new DateTime(2013, 12, 1);
searchParameter.SinceId = 399616835892781056;
searchParameter.MaxId = 405001488843284480;
var tweets = Search.SearchTweets(searchParameter);
```

Figura 3.6 - Exemplo de código de pesquisa do TweetInvi [22]

Segundo a equipe do TweetInvi em [22], com esta API é possível utilizar as seguintes classes estáticas: Tweet, User, Timeline, Message, Search, Stream, TwitterCredentials, CredentialsCreator, ExceptionHandler, TweetinviConfig, RateLimit, TwitterAccessor,

TweetinviEvents, Account, Friendship, Trends, TweetList, SavedSearch, Geo, Help, Sync, TweetinviContainer. Tornando assim a sua manipulação mais simples.

A utilização desta API neste trabalho se deu com o intuito de fazer uma conexão entre a aplicação e o Twitter, a fim de tornar possível a extração dos tweets que contenham a palavra-chave da pesquisa e ter acesso as classes estáticas citadas acima.

3.6.2 Microsoft Translator API

O Microsoft Translator API é um serviço de tradução automática baseada em nuvem. Suportando vários idiomas, o tradutor pode ser utilizado para construir aplicativos, sites e ferramentas, ou qualquer solução que exige suporte a vários idiomas [8].

Usada internamente pela Microsoft desde 2006, e disponível como uma API para os desenvolvedores desde 2011, Microsoft Translator é a solução de tradução automática da Microsoft.

Na Figura 3.8 abaixo está sendo ilustrado de forma macro, o processo de tradução da ferramenta de tradução da Microsoft.

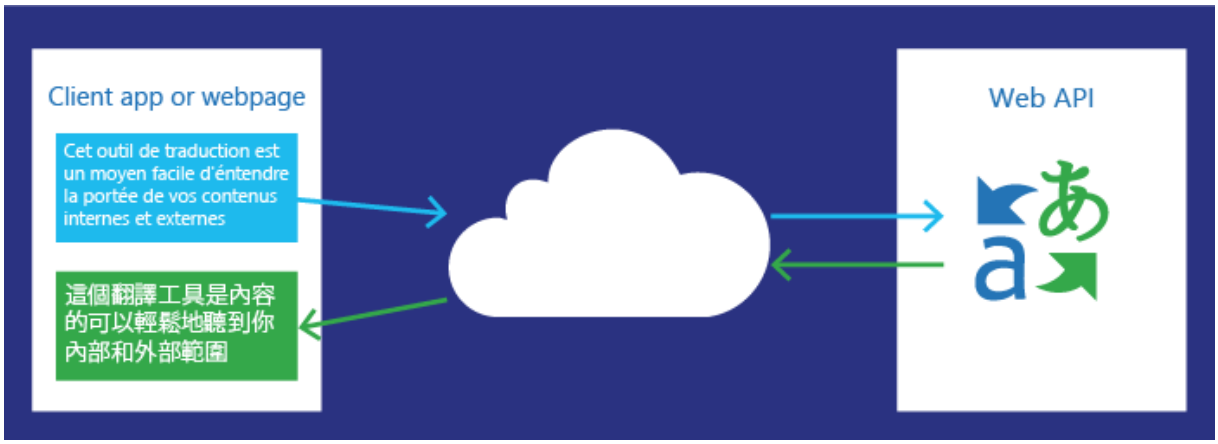

Figura 3.7 - Processo do Microsoft Translator [9]

Sendo amplamente utilizado dentro da Microsoft, o Microsoft Translator API é incorporado entre as equipes de localização de produtos, equipes de apoio e equipes de comunicação online. Este serviço também é acessível, sem nenhum custo adicional, dentro de produtos da Microsoft, como o Office, SharePoint, Yammer, Lync, Internet Explorer, Bing e Skype [9] como mostrado na Figura 3.9.

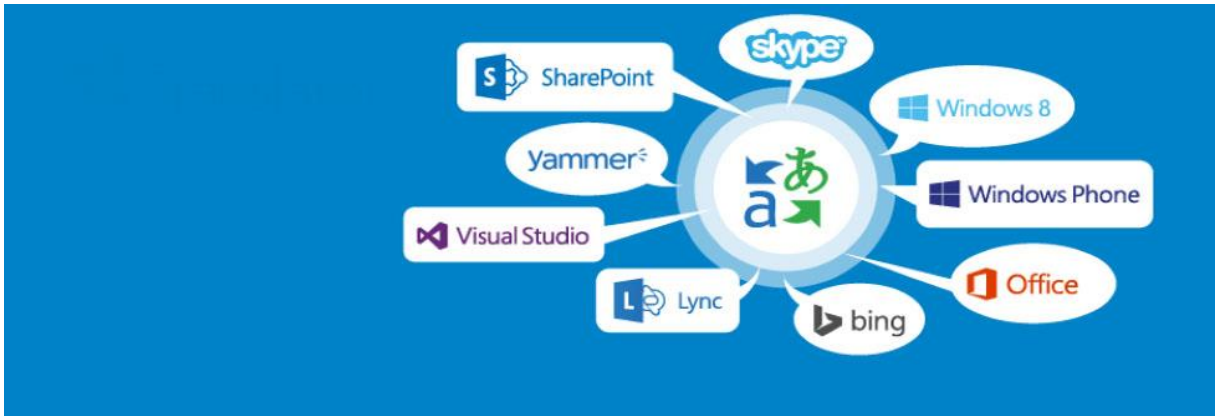

Figura 3.8 - Produtos que utilizam o Microsof Translator API [29]

Esta ferramenta foi incorporada ao trabalho por causa da dificuldade de se encontrar uma ferramenta ou API que classificasse os tweets em português, sendo necessária a tradução de cada tweet do português para o inglês para análises posteriores.

3.6.3 TreeTagger

O TreeTagger foi desenvolvido por Helmut Schmid no projeto TC, no Instituto de Linguística Computacional da Universidade de Stuttgart [11]. O TreeTagger é uma ferramenta de análise textual utilizada para identificar estruturas gramaticais dentro de um texto. Esta ferramenta atribui a cada palavra uma tag que representa uma classe gramatical específica.

Na Tabela 3.4 é mostrado um exemplo de resultado de saída do TreeTagger, para a frase "The TreeTagger is easy to use".

Word	Pos	Lemma
The	DT	The
TreeTagger	NP	TreeTagger
Is	VBZ	Be
Easy	JJ	Easy
To	TO	To
Use	VV	Use

Tabela 3.4 - Resultado de saída do TreeTagger

Na Tabela 3.4 pode ser notado na coluna com o nome “Pos” as Tags retornadas pelo TreeTagger. Para melhor entendimento do que significa cada tag, na Tabela 3.5 encontram-se os significados de cada tag que é retornada pelo TreeTagger.

Tag	Descrição
NN	Substantivo
NNS	Substantivo Plural
NP	Nome Próprio
NPS	Nome Próprio Plural
JJ	Adjetivo
JJR	Adjetivo Comparativo
JJS	Adjetivo Superlativo
RB	Advérbio
RBR	Advérbio Comparativo
RBS	Advérbio Superlativo
TO	Preposição
DT	Artigo
VBZ	Verbo
VV	Verbo

Tabela 3.5 - Tags utilizadas pelo TreeTagger

O TreeTagger foi utilizado com sucesso para etiquetar palavras do Alemão, Inglês, Francês, Italiano, Holandês, Espanhol, Búlgaro, Russo, Português, Galego, Chinês, Swahili, Eslovaco, Latina, Estoniano e Polonês [11].

Com o intuito de identificar as estruturas gramaticais presentes em cada tweet, esta ferramenta foi incorporada a este trabalho.

3.6.4 Sentiment140

Esta API de classificação de sentimentos para o Twitter, permite que sejam identificados no corpo de um tweet o sentimento expresso pelo usuário dono da postagem. Em [23] é dito que, Sentiment140 (antes conhecido como “Twitter Sentiment”) permite ao usuário descobrir o sentimento em uma marca, produto ou tópico no twitter.

O Sentiment140 foi projetado com o objetivo de usar uma abordagem baseada em aprendizado de máquina, utilizando alguns classificadores conhecidos como os de Naive Bayes, Maximum Entropy (MaxEnt), e Support Vector Machines (SVM).

Segundo os desenvolvedores a abordagem de treinamento usada no Sentiment140 foi feita automaticamente, sem precisar da intervenção humana, onde para treinar seu algoritmo foram analisados tweets que continham emoticons, onde, todos tweets com emoticons positivos :), foram considerados como positivos, e todos os que continham emoticons negativos :(, foram considerados como negativos.

Para utilizar esta API é preciso antes registrar a aplicação no site [24], pois é preciso validar o acesso a API durante a solicitação HTTP. A solicitação da análise é feita através de requisições HTTP GET como exemplo retirado do próprio site abaixo.

Exemplo:

http://www.sentiment140.com/api/classify?text=new+moon+is+awesome&query=new+moon&callback=myJsFunction

Segue abaixo a Tabela 3.6 com os parâmetros que são passados na solicitação.

Parâmetros	Descrição
text	O texto que deve ser classificado
query	A palavra-chave da pesquisa (opcional)
callback	Função de retorno da chamada
language	Idioma do texto (opcional) • en (Inglês - padrão) • es (Espanhol) • auto (auto detectar o idioma)

Tabela 3.6 - Tabela de parâmetros do Sentiment140

Na Tabela 3.7 abaixo são detalhados os dados de resposta desta API.

Dado de resposta	Descrição
text	Texto original apresentado
query	Consulta original apresentada
polarity	Valor da polaridade. • 0: negativo

	• 2: neutro • 4: positivo

Tabela 3.7 - Tabela de dados de resposta do Sentiment140

Após solicitação feita por método HTTP GET, é retornado um resultado em formato JSON, como no exemplo abaixo.

Exemplo:

myJsFunction({"results":{"text":"new moon is awesome","polarity":4,"query":"new moon"}})

A API desta ferramenta foi incorporada ao trabalho com o objetivo de se fazer uma primeira classificação das opiniões contidas em cada tweet, servindo assim como base para ser possível a classificação dos demais tweets através de hashtags e adjetivos.

Capítulo 4

Desenvolvimento do Produto

Para desenvolvimento deste produto foram utilizadas a linguagem C#.Net, juntamente com o banco de dados SQL Server 2014, dentre outras tecnologias como: Tweetinvi API, Microsoft Translator API e a ferramenta Sentiment140.

Na Figura 4.1 abaixo é mostrado o fluxo do processo completo da aplicação de software desenvolvida neste projeto:

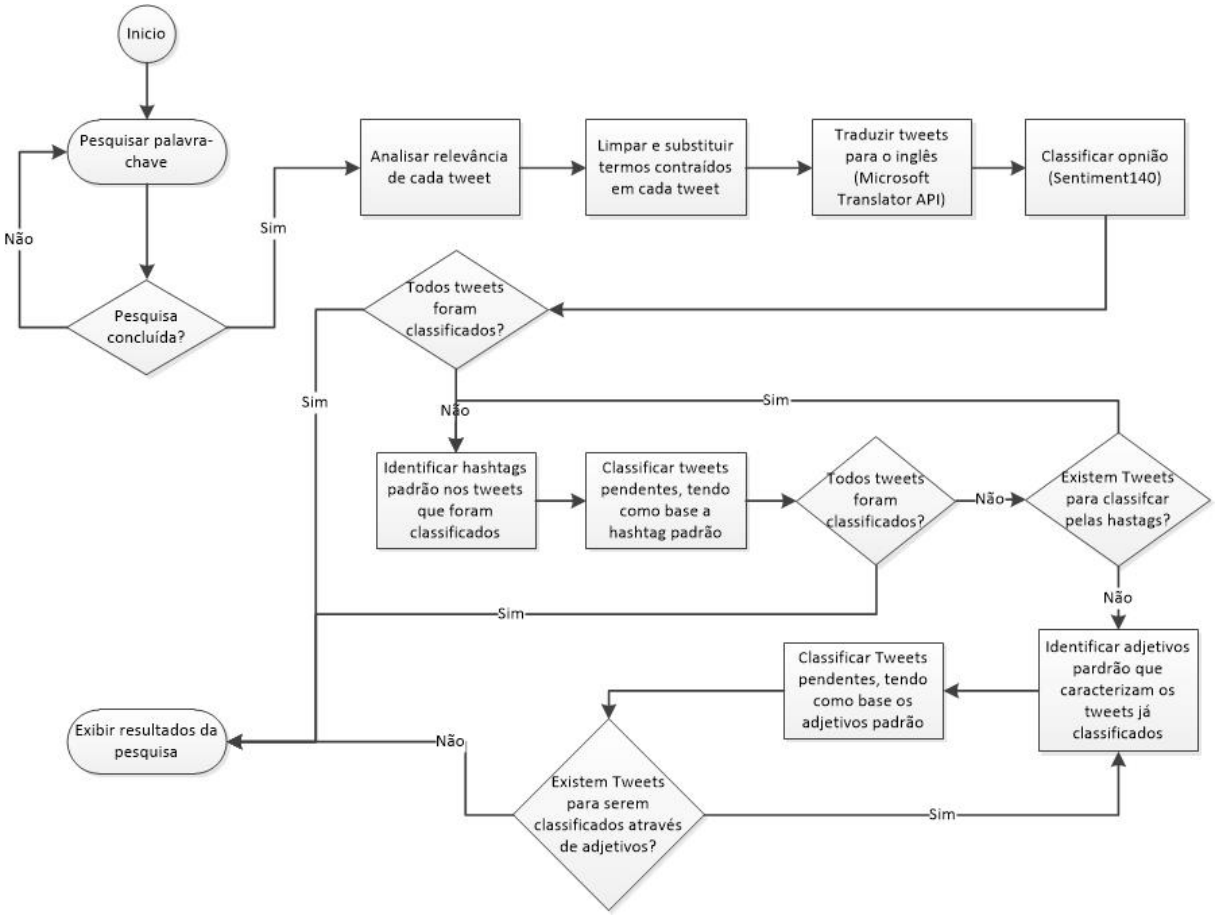

Figura 4.1- Processo completo da aplicação de software

4.1 Pesquisar palavra-chave

Seguindo o fluxo que foi detalhado na figura acima, vemos que o processo foi inicializado no momento da pesquisa pela palavra-chave (tema da pesquisa).

Para conseguir extrair do Twitter as informações necessárias, para utilizá-la como base de pesquisa desta aplicação de software, foram feitas pesquisas em sites e fóruns em busca de uma biblioteca capaz de fazer esta comunicação entre a linguagem C#.Net e a base de dados do Twitter, com o objetivo de obter as postagens mais recentes armazenadas. Depois de serem feitas as pesquisas, foi selecionada a biblioteca Tweetinvi que está implementada para C#.Net, biblioteca esta que é disponibilizada em [21] pelo próprio Twitter como uma maneira de se conectar e extrair informações, tais como postagens (tweets), número de retweets, usuário dono da postagem, data da postagem, idioma, localização geográfica, dentre outras.

Na Figura 4.2 abaixo é mostrado um trecho do código utilizado para se conectar com o Twitter, através da biblioteca Tweetinvi.

```
ClassControle controle = new ClassControle();
// Credenciais de acesso ao Twitter
var credentials = TwitterCredentials.CreateCredentials(
    "[redacted]", "[redacted]",
    "[redacted]", "[redacted]"
);
TwitterCredentials.ExecuteOperationWithCredentials(credentials, () =>
{
    int qtdTweetsPesquisados = 5000;

    var searchParameter = Search.GenerateSearchTweetParameter(searchAPI.Text);
    searchParameter.Lang = Language.Portuguese; // Filtro de tweets com idioma português
    searchParameter.MaximumNumberOfResults = qtdTweetsPesquisados; // Quantidade de retornos da pesquisa

    var tweets = Search.SearchTweets(searchParameter);

    try
    {
        foreach (var tweet in tweets)
        {
            int QtRetweets = 0; // Quantidade de retweets

            if (String.IsNullOrEmpty(tweet.RetweetCount.ToString()) || tweet.RetweetCount.ToString().Equals("0"))
            {
                QtRetweets = 0;
            }
            else
            {
                QtRetweets = Convert.ToInt32(tweet.RetweetCount.ToString());
            }
            // Inserindo na base de dados os tweets
            controle.controlePesquisaInserir(tweet.IdStr, tweet.Text, QtRetweets);
        }
    }
    catch (Exception) {
        throw;
    }
});
Response.Redirect("View/About.aspx");
```

Figura 4.2 - Código de busca e inserção dos dados da pesquisa

Após a extração das informações, os tweets são armazenados em uma base de dados, para posteriormente serem analisados em busca de obter uma pesquisa de mercado mais eficiente.

4.2 Análise de relevância de cada tweet

Nesta etapa do processo foram utilizados conceitos e técnicas da área de recuperação de informação, para se fazer uma análise de relevância sobre os tweets. Alguns conceitos foram explicados no tópico 3.2, onde são detalhados os modelos booleanos, vetorial e probabilísticos.

Esta aplicação de software foi baseada em dois modelos, os modelos booleano e vetorial. Como contribuição para o desenvolvimento desta aplicação, o modelo booleano entra com a parte de identificação de estruturas (urls, mensões a usuários, hashtags e retweets) presentes dentro de cada tweet, obtendo de forma estrutural como cada tweet está sendo construído por cada usuário participante da pesquisa.

No que se refere a identificar as estruturas presentes no tweet, foram utilizadas expressões regulares para cumprir esta missão. Estas sequências de caracteres foram montadas com este objetivo, a fim de obter evidências de que tal estrutura existe ou não dentro de um determinado tweet.

Na Tabela 4.1 abaixo encontram-se as expressões regulares utilizadas neste momento do processo.

Objeto	Expressão regular
URL(http://www.google.com)	(http|https|ftp)?://([\\w-]+\\.)+[\\w-]+(/[\\w- ./?%&=][a-zA-Z0-9]*)?
Hashtags (#partiu, #comida)	(#)+((?:[A-Za-z0-9-_]*))
Menção a usuário (@usuario)	(@)+((?:[A-Za-z0-9-_]*))
Retweets (RT @usuario)	(RT @)+((?:[A-Za-z0-9-_]*))

Tabela 4.1 - Expressões regulares para identificação das estruturas contidas no tweet

A Figura 4.3 corresponde ao código utilizado para identificação das estruturas que compõem um tweet. Foi declarado um método “pesquisaBlocos”, que recebe três parâmetros, que são: “padrao”, “tweet”, “typeURL”.

- O parâmetro “padrao”: refere-se ao modelo de expressão regular utilizado para identificar a estrutura.
- O parâmetro “tweet”: trata-se do corpo do tweet que irá ser analisado.
- O parâmetro “typeURL”: identifica o modelo de expressão regular que está sendo passado, pois em caso de um modelo referente a URL, a aplicação precisará verificar se esta URL trata-se de uma URL válida ou não.

```
public int pesquisaBlocos(string padrao, string tweet, bool typeURL) {
    string text = tweet; //Tweet

    Regex regex = new Regex(padrao, RegexOptions.IgnoreCase);

    MatchCollection matches = regex.Matches(text); //Criando coleção com os resultados que correspondem a pesquisa

    int cont = 0;
    foreach (Match match in matches) { // Percorrendo a coleção com os resultados que correspondem a pesquisa
        bool valida = false;

        if (typeURL == true) // Caso o padrão passado seja de URL
        {
            valida = this.URLEhValida(match.ToString());

            if (valida.Equals(true))
            {
                cont++;
            }
        }
        else
        {
            cont++;
        }
    }
    return cont;
}
```

Figura 4.3 - Código de identificação de estruturas do tweet

Após feita a identificação das estruturas presentes no tweet, a base de dados é atualizada com a quantidade exata de ocorrências de cada estrutura no tweet.

De posse destas informações a aplicação leva em consideração os conceitos do modelo vetorial detalhado no tópico 3.2, atribui para cada estrutura um peso não binário, que identifica o tipo de estrutura mais relevante conforme ilustra Figura 4.4. O peso atribuído para cada estrutura foi pensado levando em consideração o volume de informação que uma determinada estrutura adiciona ao tweet.

```
int pesoURL = 5,
    pesoRetweet = 3,
    pesoHashTag = 2,
    pesoMensUser = 1;
```

Figura 4.4 - Atribuição de pesos

Em seguida é feito um cálculo para se obter o grau de relevância de cada tweet. Este cálculo é feito da seguinte forma: a quantidade total da estrutura no tweet, divido pela quantidade total da estrutura em toda pesquisa, multiplicado pelo peso. Como forma de facilitar a manipulação dos valores, o resultado deste cálculo foi multiplicado pelo valor 1000, como mostra a Figura 4.5 abaixo.

```
vlPesoURL = 1000 * (qtdUrlTweet / totalURL) * pesoURL;
vlPesoHashTag = 1000 * (qtdHashtagTweet / totalHashTag) * pesoHashTag;
vlPesoMensUsuario = 1000 * (qtdMenUsuario / totalMensUser) * pesoMensUser;
vlPesoRetweet = 1000 * (qtdRetweets / totalRetweets) * pesoRetweet;

vlTotalRanking = vlPesoURL + vlPesoHashTag + vlPesoMensUsuario + vlPesoRetweet;
```

Figura 4.5 - Cálculo de relevância

Ao final é somado os valores de relevância de cada estrutura, obtendo assim o grau de relevância do tweet e em seguida armazena-se os valores na base de dados.

4.3 Limpeza e substituição de termos contraídos em cada tweet

Esta etapa é considerada uma etapa muito importante para o processo, pois é nela onde é feita toda a substituição de termos contraídos que não fazem nenhum sentido na hora de fazer a análise de sentimentos. Termos como: “mt’ são substituídos por “muito”, “obg” por “obrigado”, “p/” por “para”, dentre outros. Na Tabela 4.2 abaixo são exemplificados mais alguns exemplos de substituições feitas pela aplicação.

Termo	Significado
vc	você
tbm	também
mt	muito
agr	agora
bjs	beijos
qro	quero
dps	depois

Tabela 4.2 - Termos contraídos com seus respectivos significados

Para se ter o resultado esperado nesta etapa, foi preciso criar uma base de termos contraídos com seus respectivos significados. Sendo assim possível substituir cada termo por um outro mais apropriado para análises posteriores. Na Figura 4.6 encontra-se um trecho do código responsável por fazer estas substituições. Após substituições, os dados são armazenados na base de dados.

```
string tweet = reader["DS_Post_Tweet"].ToString(); // tweet

// transformando o tweet em um array
string[] separaTweet = tweet.Split(' ', '.', ',', '!', ':', '?', '(', ')', '\'', '"');

List<string> lista = new List<string>(separaTweet);

for (int i = 0; i < lista.Count; i++) // percorrendo a coleção
{
    if (lista[i].Length <= 5 && lista[i].Length >= 1)
    {
        // Verificando a existência do termo abreviado na base
        SqlDataReader Abrev = bdTweet2.selectTermosAbreviados(lista[i].ToUpper());

        if (Abrev.Read())
        {
            //substituindo o termo contraído pelo seu significado
            lista[i] = Abrev["DS_TermoAbrev"].ToString();
        }
        Abrev.Dispose();
    }
}
// juntando novamente as peças. Formando o tweet novamente.
string textTweet = String.Join(" ", lista);
```

Figura 4.6 - Código de limpeza e substituição de termos contraídos

4.4 Tradução dos tweets

A tradução de cada tweet foi feita com o objetivo de superar um problema encontrado, pois foi difícil encontrar ferramentas de análise de sentimentos eficientes para língua portuguesa. Diante deste problema foi decidido utilizar uma API de tradução automática chamada Microsoft Translator, que por sua vez teve como fator principal para sua escolha por se tratar de uma API bastante compatível com a linguagem C#.Net.

Para poder usar esta API, foi preciso obter um token de acesso para o Microsoft Translator API. Este token de acesso tem o objetivo de autenticar o acesso a API de tradução automática.

Tornando assim o acesso a API mais seguro, para obtê-lo foi necessário seguir os seguintes passos:

- Fazer inscrição para o Microsoft Translator API na Azure Marketplace.
- Registrar o aplicativo com Azure DataMarket.
- Fazer solicitações http post para o serviço de token.

O acesso a esta API foi feito via requisições http, retornando o texto traduzido.

4.5 Classificação de opinião

Esta etapa é de grande importância para o projeto por se tratar do ponto onde é feita a análise de sentimento em cada tweet. Para que isto fosse possível, esta etapa teve como pré-requisito a tradução dos tweets para o inglês como detalhado em tópico anterior, pois esta ferramenta de análise de sentimentos mostrou-se um melhor desempenho analisando textos que estavam em inglês.

Esta API permite classificar os tweets de forma individual via solicitações HTTP GET. Como exemplo de solicitação, vamos analisar a frase “I like japanese food”.

http://www.sentiment140.com/api/classify?text=I%20like%20japanese%20food

Podemos identificar visivelmente a presença de um parâmetro “text”, onde é informada a frase que precisa ser analisada. Como resultado desta solicitação a API retorna um formato JSON com as respostas desta solicitação.

Segue abaixo o retorno gerado por esta solicitação:

{"results":{"text":"I like japanese food","polarity":4,"query":"NO_QUERY"}}

Em outro exemplo vamos analisar a frase “Japanese food is horrible”, a fim de verificar a situação oposta ao primeiro exemplo.

http://www.sentiment140.com/api/classify?text=Japanese%20food%20is%20horrible

Segue abaixo retorno da API para esta solicitação:

{"results":{"text":"Japanese food is horrible","polarity":0,"query":"NO_QUERY"}}

Pode-se notar uma diferença retornada no resultado, onde no primeiro exemplo na tag polarity foi retornado o resultado 4 que se refere a “positivo”, caracterizando assim um interesse sobre o tema da frase que é “comida japonesa”, já no segundo exemplo foi retornado o valor 0 que se refere a “negativo”, caracterizando assim um não interesse sobre o tema da frase.

Na Figura 4.7 abaixo segue um trecho do código responsável por fazer a solicitação.

```
tweet = tweet.Replace("#", "");
tweet = tweet.Replace("`", "'");
// Montando url com o tweet para ser analisado
string url = "http://www.sentiment140.com/api/classify?text=" + tweet + "&appid=[redacted]";

//solicitação http
HttpWebRequest request = (HttpWebRequest)WebRequest.Create(url);
try
{
    WebResponse response = request.GetResponse();
    using (Stream responseStream = response.GetResponseStream())
    {
        StreamReader reader = new StreamReader(responseStream, Encoding.UTF8);
        var text = reader.ReadToEnd(); // retorno gerado pela API

        text = text.ToString().Replace("'", "`");
        text = text.ToString().Replace("\"", "'");
        text = text.ToString().Replace("\n", "");

        dynamic stuff = Newtonsoft.Json.Linq.JObject.Parse(text);

        int polarity = stuff.results.polarity;

        return polarity;
    }
}
catch (WebException ex)
{
    WebResponse errorResponse = ex.Response;
    using (Stream responseStream = errorResponse.GetResponseStream())
    {
        StreamReader reader = new StreamReader(responseStream, Encoding.GetEncoding("utf-8"));
        String errorText = reader.ReadToEnd();
        // log errorText
    }
    throw;
}
```

Figura 4.7 - Código de classificação de opinião

Após ser feita a análise de sentimentos de todos os tweets presentes na base de dados, é verificado se todos os tweets foram analisados ou não. Caso possua ainda tweets com o status 2 que se refere a “neutro”, caracterizando que não foi possível identificar o sentimento presente

na frase, será preciso fazer uma varredura em todos os tweets que conseguiram ser analisados pela ferramenta Sentiment140, em busca de hashtags padrão que possam caracterizar a opinião contida nos tweets. Mais detalhes sobre a identificação de hashtags padrão no próximo tópico.

4.6 Identificação de hashtags padrão

Com o intuito de classificar os tweets que não foram possíveis de ser classificados pelo Sentiment140, foram feitas pesquisas sobre como identificar a opinião contida nos demais tweets. Durante as pesquisas ficou perceptível que dos tweets que foram classificados como "positivo" ou "negativo", em grande parte eles continham hashtags que apareciam nos demais tweets, e analisando os tweets que não foram classificados, também foram encontradas essas hashtags, que por sua vez fazendo uma análise do conteúdo dos tweets foram identificados os mesmos tipos de opiniões (positivo ou negativo), encontrando assim um padrão na escrita dos tweets que estão relacionados ao mesmo tema.

Com base nisso foi desenvolvido um código com o objetivo de identificar estes padrões, a fim de automatizar este processo, fazendo uma varredura nos tweets já classificados em busca de hashtags e para identificar a presença delas foi utilizada expressões regulares que se mostraram bastante eficaz no cumprimento desta missão.

Abaixo é mostrado na Tabela 4.3 a sequência utilizada para identificação das hastags.

HashTag	Detalhes
(#)+((?:[A-Za-z0-9-_]*))	Verifica todas as sequências começando com "#", seguida de uma ou mais sequencias de letras e números.

Tabela 4.3 - Expressão regular utilizada para identificar hashtags no tweet

Na Figura 4.8 abaixo é mostrado o código desenvolvido para identificar as hashtags.

```
string text = tweet; //Tweet
Regex regex = new Regex("(#)+((?:[A-Za-z0-9-_]*))", RegexOptions.IgnoreCase);

MatchCollection matches = regex.Matches(text); //Criando coleção com os resultados que correspondem a pesquisa

string hashTags = "";
foreach (Match match in matches)
{ // Percorrendo a coleção com os resultados que correspondem a pesquisa
    hashTags = hashTags + match + " ";
}

return hashTags;
```

Figura 4.8 - Código de identificação de hastags padrão

Despois de capturadas pelo código, as hashtags são gravadas na base de dados, juntamente com o identificador do tweet e sua polaridade (4 - positivo ou 0 - negativo).

4.6.1 Classificação por hashtags padrão

Primeiramente o código da aplicação executa uma varredura em todos os tweets que não foram classificados como “4 - positivo” ou “0 - negativo”, em busca de identificar hastags no corpo do tweet. Quando encontrada uma hashtag o código verifica se ela encontra-se contida na lista de hastags padrão identificadas nos tweets que foram classificados anteriormente, e caso encontre a aplicação contabiliza a quantidade de hashtags relacionada a tweets “positivos” e “negativos”, sendo somado 1 ao contador se encontrar um “positivo” e decrementado 1 ao contador caso encontre um “negativo”, ao final de toda verificação a aplicação verifica se o valor do contador é maior ou menor que 0. Sendo *contador* > *0*, “positivo” e *contador* < *0*, “negativo”.

Logo em seguida foram feitas análises de tweets que continham hashtags compostas pelo tema da pesquisa completo ou por parte dele, classificando-os como “positivos”. Como por exemplo tweets com o tema de pesquisa “Comida japonesa”, caso fossem encontradas hashtags contendo “#comidajaponesa”, “#comida” ou “#japonesa” serão classificados como positivo, pois identificam em sua composição de hashtags o tema da pesquisa.

Na Figura 4.9 abaixo é exemplificado algumas postagens contendo a palavra-chave como hashtags.

Figura 4.9 - Exemplo de uso de palavra-chave como hashtag

4.7 Identificação de adjetivos padrão

Utilizando-se da mesma ideia de classificação através das hashtags vista no tópico anterior, em caso de não conseguirmos classificar todos os tweets até esse momento, foi utilizada a ferramenta TreeTagger, que se encontra detalhada no tópico 3.7.3, que contribuiu na verificação e identificação das estruturas gramaticais que compõem o tweet. Por não possuir uma API compatível com a linguagem C#.Net, foram encontrados alguns problemas para extrair o resultado desta ferramenta, porém depois de estudar a ferramenta mais profundamente, foi percebido que era possível acessá-la via linha de comando como mostrado na Figura 4.10 logo abaixo.

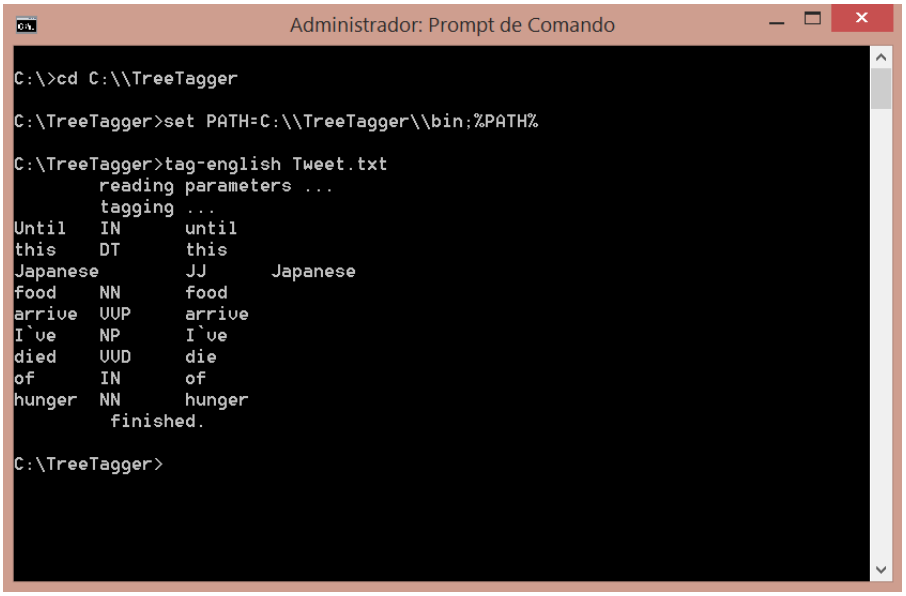

Figura 4.10 - Acesso TreeTagger via cmd

Sendo possível acessar o resultado da ferramenta, ficou possível extrair os adjetivos que se encontram nos tweets. A utilização de adjetivos foi proposta pelo fato de em sua maioria eles classificarem o sentido da frase.

Com o resultado retornado pela ferramenta, foi preciso armazená-los em um arquivo txt para que fosse possível acessar seu conteúdo linha a linha e capturar os adjetivos (JJ) presentes dentro da análise feita pelo TreeTagger, capturando e armazenando todos os adjetivos na base de dados, juntamente com o identificador do tweet ao qual ele pertence.

Na Figura 4.11 abaixo é mostrado o código deste processo.

```
// ler arquivo com o retorno do TreeTagger
System.IO.StreamReader file = new System.IO.StreamReader(@"C:\TreeTagger\TweetAnalisado.txt");

ClassControle classControle = new ClassControle();

// Acesso linha a linha do arquivo
while ((linha = file.ReadLine()) != null)
{
    int pos1 = linha.IndexOf("\t");
    int pos2 = linha.IndexOf("\t", pos1 + 1);
    reg = linha.Substring((pos1 + 1), 2);

    // Verificando se é um adjetivo ou não
    if (reg == "JJ")
    {
        adj = linha.Substring(0, pos1);
        adj = adj.Replace("-(", "");
        string padrao = "(" + adj + ")";

        // verifica se o adjetivo não pertence a palavra-chave da pesquisa
        if (classControle.pesquisaBlocos(padrao, palavraChave, false) == 0)
        {
            if (cont == 0)
            {
                bdTweet.insertAdjTweets(Convert.ToInt32(idTweet), adj, idClass);
                cont++;
            }
            else
            {
                bdTweet.updateAdjTweet(Convert.ToInt32(idTweet), adj);
                cont++;
            }
        }
    }
}
```

Figura 4.11 - Identificação de adjetivos

4.7.1 Classificação por adjetivos padrão

Com a base de dados preenchida com os adjetivos identificados nos tweets já classificados, o próximo passo é fazer uma varredura nos tweets que continuam pendentes de classificação, em busca da ocorrência de algum adjetivo encontrado anteriormente pelo TreeTagger. Tendo como base os adjetivos de tweets que já foram classificados, a aplicação polariza o novo tweet, levando em consideração a maior quantidade de ocorrência de uma determinada classe (0 – negativo ou 4 - positivo) em relação ao adjetivo na base de dados. Caso a maior quantidade de uma classe de um adjetivo seja “0 - negativo”, é subtraído 1 do contador, e caso seja “4 - positivo”, é somado 1 ao contador. Depois de ser feita toda análise com os adjetivos presentes no tweet, é verificado se o contador é maior ou menor que 0 (zero), classificando assim a polaridade do tweet. Para *contador* < *0*, a polaridade é “0 - negativa” e para *contador* > *0*, a polaridade é “4 - positiva”.

Capítulo 5

Análise do Resultados Alcançados

Nesta sessão serão exibidos todos os resultados gerados por este trabalho, a fim de comprovar as técnicas utilizadas para desenvolvimento desta aplicação de software.

5.1 Análise de resultados

Tendo como experimento uma pesquisa de mercado em relação à área alimentícia, onde foram feitas buscas pelos tweets que contivessem a palavra-chave “Comida Japonesa”, foi retornado pela aplicação de software uma quantidade de 5000 tweets para análise.

Após essa fase de extração, os tweets obtidos possuíam bastantes palavras contraídas, que não poderiam ser analisadas pela aplicação, sendo necessário substituí-las por palavras que possuíssem o mesmo significado.

Já com todos os tweets limpos para análise, foi feita a tradução de todos para o idioma inglês que é o padrão da API de análise de sentimentos utilizada neste trabalho.

Após estas fases iniciais todos tweets foram colocados para serem analisado pela API do Sentiment140, o qual retornou os seguintes resultados conforme o Gráfico 5.1 abaixo.

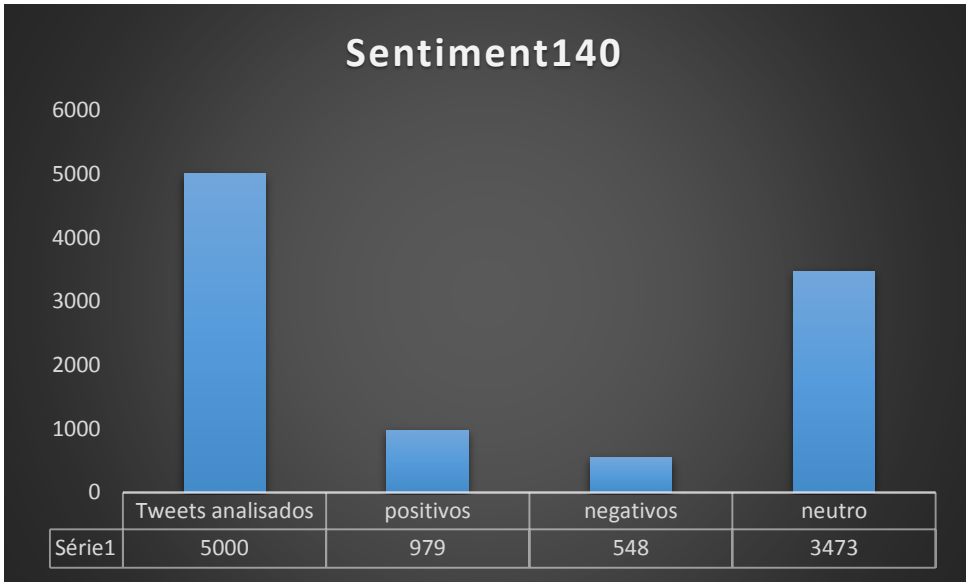

Gráfico 5.1 - Análise de resultado do Sentiment140

Analisando o gráfico é visto que dos 5000 tweets que foram analisados 979 deles foram classificados como positivo, 548 foram classificados como negativos e 3473 não conseguiram ser classificados pelo Sentiment140.

Após a análise feita pelo Sentiment140, a aplicação passou a verificar dentre os tweets já classificados as hashtags contidas em seu corpo, a fim de fazer uma análise em cima dos tweets pendentes. No Gráfico 5.2 são mostradas evidências dos resultados retornados após identificação das hashtags e nova análise a partir das hashtags encontradas.

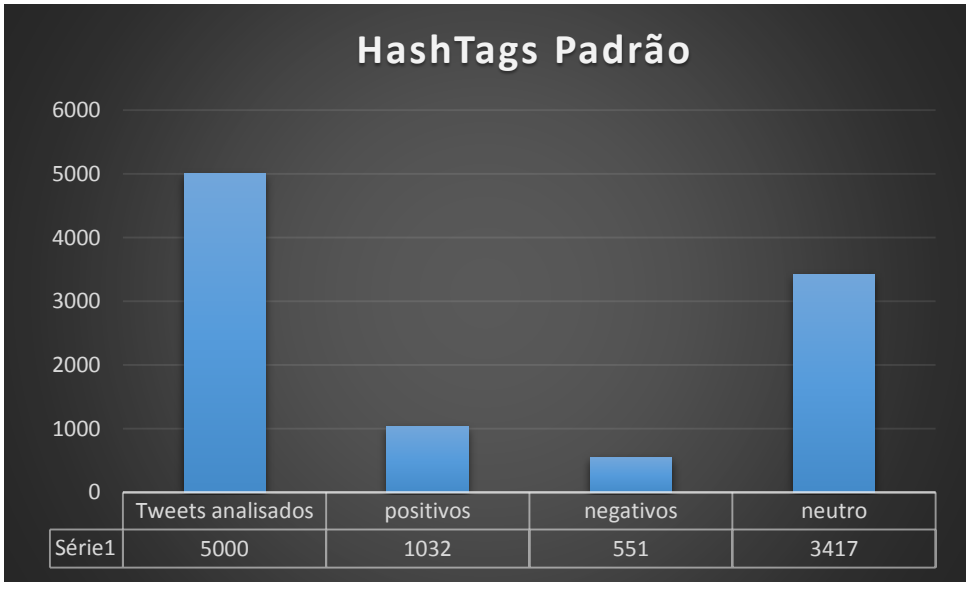

Gráfico 5.2 - Análise de resultado de hashtags padrão

Tendo como base os resultados retornados, verificamos que neste momento foi possível classificar mais alguns tweets como positivos e negativos, ficando assim, dos 5000 tweets analisados, 1032 foram classificados como positivos, 551 foram classificados como negativos e 3417 como neutros.

Com base nos resultados gerados até a última fase, foi feita uma nova verificação nos tweets já classificados, em busca de adjetivos padrões para tentar classificar os demais tweets, e após verificação e análise feita, foi visto que o desempenho do algoritmo usando agora como base de classificação os adjetivos conseguiu classificar a maior parte dos tweets como mostrado no Gráfico 5.3 abaixo.

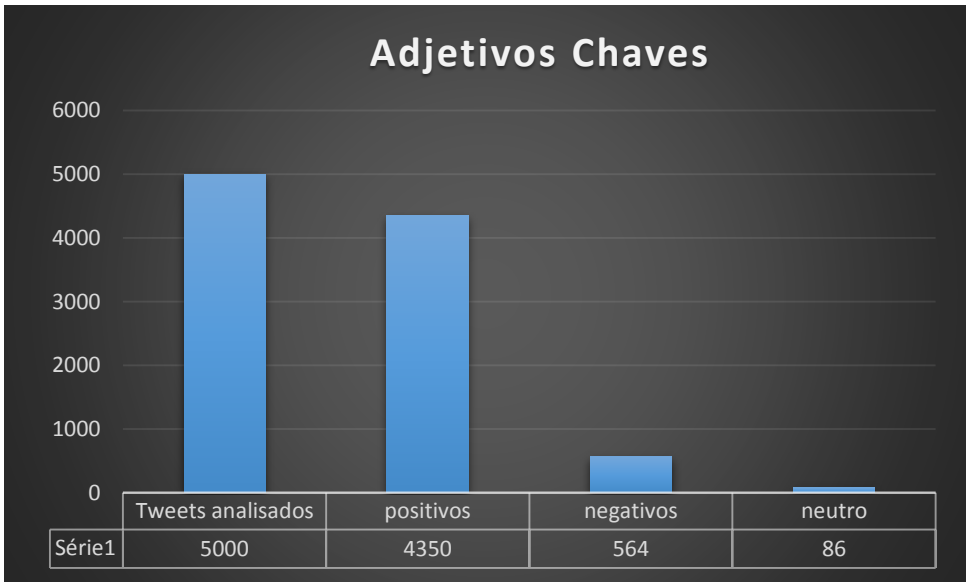

Gráfico 5.3 - Análise de adjetivos chave

Após conclusão do processo foi verificado que dos 5000 tweets, 4914 tweets conseguiram ser classificados pela aplicação de software desenvolvida neste trabalho e apenas 86 não foram possíveis de ser classificadas. Tendo como resultado da classificação 4350 tweets classificados como positivos o que equivale a 87% dos tweets pesquisados, 564 tweets classificados como negativos o que equivale a 11,28% dos tweets pesquisados.

Logo em seguida foi visto também que é possível determinar um ranking dos tweets classificados levando em consideração seu grau de relevância, exibindo assim uma lista dos tweets separados por categoria (“positivo” ou “negativo”), como mostrado nas tabelas 5.1 e 5.2 abaixo.

Id. do Tweet	Tweet	Opinião	Grau de relevância
68479	E você, já conhece o melhor delivery de comida japonesa de BH #bh #sushi #sushiday #japa http://t.co/2f9ohijqvQ http://t.co/mVpI8nSbb7	Positivo	41,71131484
72656	#Comida #japonesa bem leve - 345 gramas de #biodiversidade e #degustação! - no #almoço de quinta… https://t.co/2OO2EuxXhN	Positivo	41,12592769
68571	Gosta de comida japonesa? Veja esta matéria http://t.co/26NbPuNakg Via Guia !Yoba #Sushi #ComidaJaponesa #Restaurante http://t.co/5P0axQ8d66	Positivo	34,95455808
71427	Social de sábado, nada melhor do que comida japonesa ?????? #sushibox #amigas #social @ SushiBox… https://t.co/cifyOWNFgo	Positivo	28,12261826
70878	Comida japonesa de ontem ????... #javas #sushi #amigos https://t.co/JCvSEE6BDU	Positivo	27,61241418

Tabela 5.1 - Grau de relevância de tweets positivos

Id. do Tweet	Tweet	Opinião	Grau de relevância
71382	RT @FALLTAEXEMPLO: eu odeio comida japonesa chinesa sei la n gosto #askbelieber	Negativo	9,599775768
71867	eu odeio comida japonesa https://t.co/Fv2eiJh3lu	Negativo	7,342143906
71435	Não acredito quando alguém fala que acha comida japonesa gostosa...não consigo entender !!!! Uma coisa é achar... http://t.co/9WPQHx8qQN	Negativo	7,342143906
72513	RT @FloressMarcella: Nossa, eu odeio comida japonesa	Negativo	2,843019012
72033	RT @diamndream: sobre comida japonesa:odeio	Negativo	2,843019012

Tabela 5.2 - Grau de relevância de tweets negativos

5.2 Taxas de Erros e Acertos

Após retorno dado pela aplicação, foi necessária uma análise mais detalhada dos resultados, com o objetivo de validar até que ponto a aplicação se mostrou eficiente na análise feita em cima dos tweets. Foi proposto então a escolha de 100 tweets sequenciais e aleatórios para serem verificados manualmente, a fim de identificar os pontos onde a aplicação obtém um melhor desempenho.

Para validar os resultados obtidos, a taxa de erros e acertos foi medida de duas formas, onde a primeira trata-se da verificação dos tweets de forma sequencial e a segunda de forma aleatória.

5.2.1 Positivos e Negativos

Primeiramente foram verificados manualmente os tweets classificados, obtendo um resultado satisfatório, onde dos 100 tweets sequenciais separados para análise, foi validada uma taxa de acertos de 84 tweets, o que equivale a 84% dos analisados, e uma taxa de erro de 16 tweets, equivalente a 16% do total como mostrado no Gráfico 5.4 abaixo.

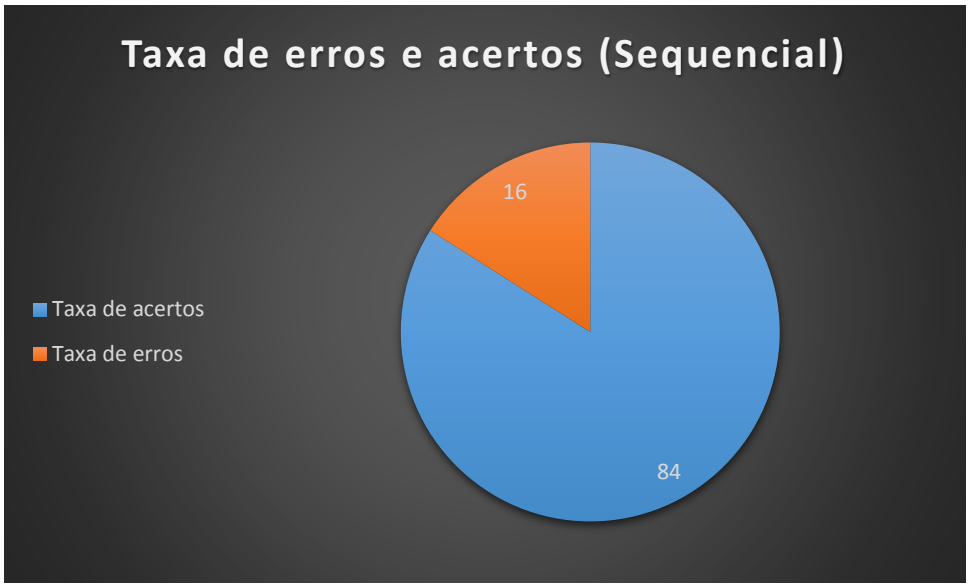

Gráfico 5.4 - Taxa de erros da aplicação (Tweets sequenciais)

Em seguida, tendo como base os tweets coletados de forma aleatória, primeiramente foram verificados os tweets manualmente, a fim de verificar se a análise feita pela aplicação está de forma satisfatória, onde dos 100 tweets utilizados para análise, foi observada uma taxa de acertos de 83 tweets, o que equivale a 83% dos analisados, e uma taxa de erro de 17 tweets, o que equivale a 17% dos analisados, como mostra o Gráfico 5.5. Sendo assim um resultado satisfatório.

Gráfico 5.5 - Taxa de erros da aplicação (Tweets aleatórios)

5.2.2 Sequencial

Utilizando da mesma estratégia usada para verificar os resultados retornados pela aplicação, também foi feita uma análise dos resultados específicos em relação a classificação de tweets como positivos.

Como mostrado no Gráfico 5.6, dos 100 tweets sequenciais classificados como positivos, 91 foram classificados corretamente e 9 foram classificados de forma equivocada.

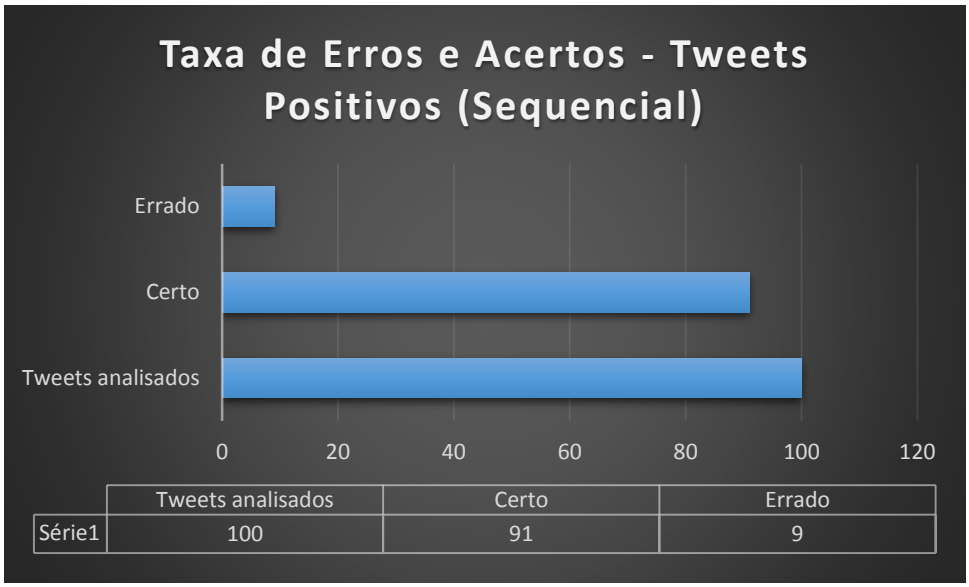

Gráfico5.6 - Taxa de erros e acertos em tweets positivos (Sequenciais)

Em seguida, foram analisados os resultados dos tweets classificados como negativos, como mostra no Gráfico 5.7, dos 100 tweets coletados de forma sequencial que foram classificados como negativos, houveram 53 tweets classificados corretamente e 47 tweets classificados errados. Resultado este que pode estar sendo influenciado pela proximidade dos tweets, sendo necessária a verificação dos resultados gerados pela verificação aleatória.

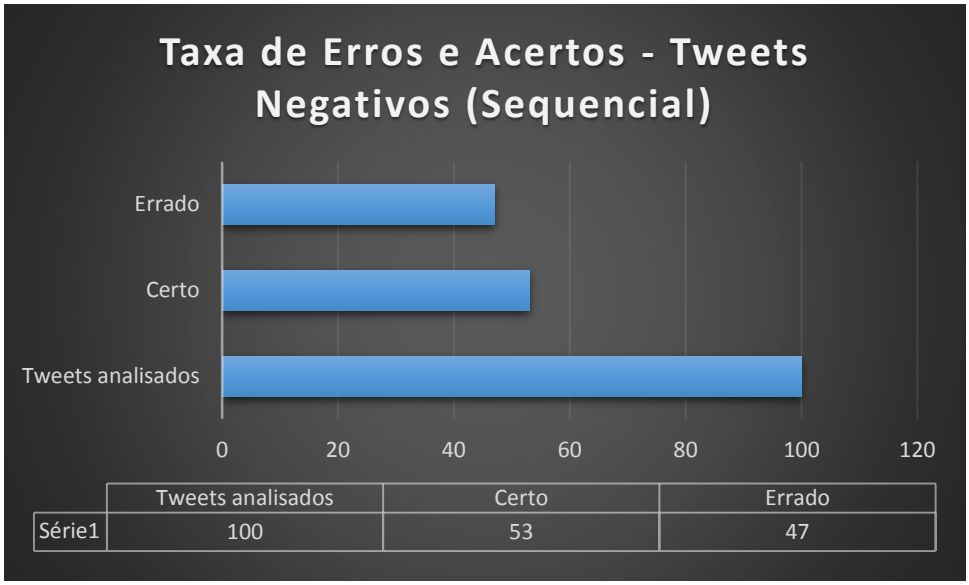

Gráfico 5.7 - Taxa de erros e acertos em tweets negativos (Sequenciais)

Analisando os resultados mostrados na Tabela 5.3 , na primeira linha vemos que 91% dos tweets analisados são chamados de positivos verdadeiros, pois suas classificações ficaram corretas, e 9% são conhecidos como falsos positivos, pois foram classificados como positivos, quando deveriam ter sido classificados como negativos. Em relação aos tweets negativos temos 53% classificados como negativos verdadeiros, pois foram classificados como negativos de forma correta e 47% classificados como falsos negativos, pois suas classificações não foram concluídas com êxito, visto que foram classificados como negativos, quando deveriam ter sido classificados como positivos.

	Positivos	Negativo
Positivo	91	9
Negativo	47	53

Tabela 5.3 - Relação entre positivos, falsos positivos, negativos e falsos negativos (Sequenciais)

5.2.3 Aleatório

No caso dos Tweets positivos escolhidos de forma aleatória, foi verificado que, dos 100 tweets classificados como positivos, 92 deles foram classificados de forma correta e 8 foram classificados erroneamente como mostra o Gráfico 5.8.

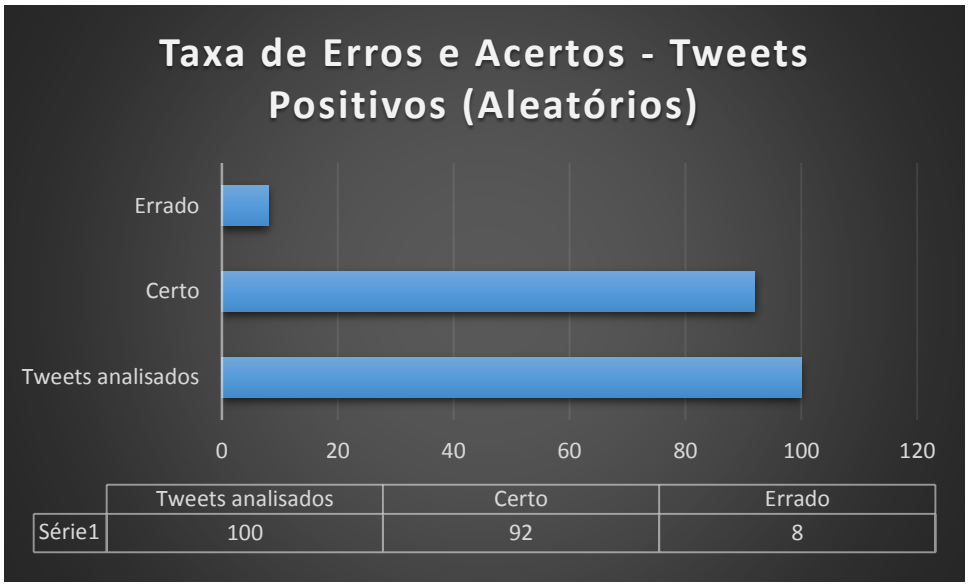

Gráfico 5.8 - Taxa de erros e acertos em tweets positivos (Aleatórios)

Em seguida, fazendo coletas de tweets negativos de forma aleatória, foi verificado que, dos 100 tweets classificados como negativos, 63 foram classificados corretamente e 37 classificados de forma errada. Gerando assim uma melhoria nos resultados em relação ao sequencial, pois dos 100 tweets, 63 conseguiram ser classificados de forma correta, o equivalente a 63% do total da pesquisa como mostrado no Gráfico 5.9 abaixo.

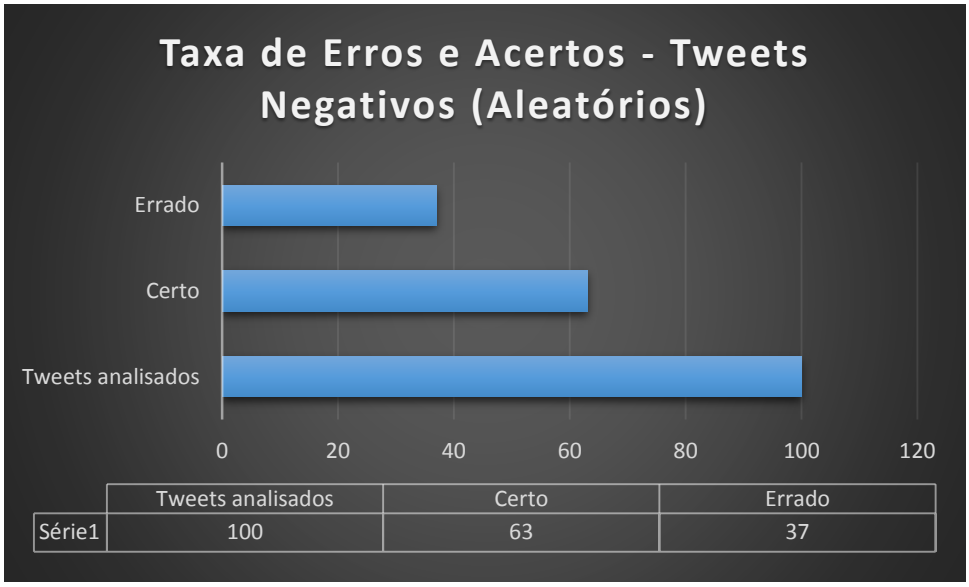

Gráfico 5.9 - Taxa de erros e acertos em tweets negativos (Aleatórios)

Analisando os resultados mostrados na Tabela 5.4, na primeira linha vemos que 92% dos tweets analisados são chamados de positivos verdadeiros, pois suas classificações ficaram corretas, e 8% são conhecidos como falsos positivos, pois foram classificados como positivos de forma errônea. Em relação aos tweets negativos temos 63% classificados como negativos verdadeiros, pois foram classificados como negativos de forma correta e 37% classificados como falsos negativos, pois suas classificações não foram concluídas com êxito, visto que foram classificados como negativos, quando deveriam ter sido classificados como positivos.

	Positivos	Negativo
Positivo	92	8
Negativo	37	63

Tabela 5.4 - Relação entre positivos, falsos positivos, negativos, falsos negativos (Aleatórios)

Capítulo 6

Conclusão

A recuperação de informação e a análise de sentimento são duas áreas que estão em constante desenvolvimento, pois cada vez mais o volume de informações geradas na internet vem crescendo, e para ter acesso a este volume de informações é preciso desenvolver mecanismos precisos que deem acesso seguro a tais informações.

Durante o desenvolvimento deste trabalho, foi notado que a classificação de tweets através de análise de sentimentos, é uma tarefa complexa, não trivial, pois pela quantidade limitada de palavras e erros grosseiros de escrita pelos usuários do Twitter, foram identificadas algumas dificuldades que foram superadas durante o desenvolvimento deste trabalho. Foi identificado também que a ocorrência de tweets com ambiguidade tem se mostrado um grande problema de classificação através da análise de sentimento.

Apesar destas dificuldades encontradas, pode-se concluir que é possível utilizar o Twitter como fonte de pesquisa, para classificar tweets utilizando análise de sentimentos e suportar empreendedores em pesquisas de mercado, lembrando que, pesquisas de mercado servem exclusivamente para ajudar ao empreendedor a tomar decisões. Portanto, cabe ao empreendedor fazer bom uso da aplicação.

6.1 Trabalhos futuros

De posse dos conhecimentos adquiridos durante o desenvolvimento deste trabalho, surgiu o interesse de estudar cada vez mais as áreas de recuperação de informação e análise de sentimentos em busca de cada vez mais melhorar o desempenho do produto resultante deste trabalho. Alguns pontos relevantes foram levantados com o objetivo de complementar o trabalho aqui realizado. Dentre estas propostas, pode-se citar:

- Polarização de smiles (emoticons) como positivos e negativos.
- Analisar a classificação feita sem ser preciso traduzir os tweets para o inglês, fazendo um comparativo de desempenho entre as duas formas.
- Classificação de positivos e negativos, levando em consideração toda a estrutura gramatical que compõe o tweet.

- Utilização de um corretor ortográfico, a fim de melhorar o conteúdo de cada tweet, eliminando ao máximo a quantidade de erros ortográficos encontrados nos tweets.

Referências Bibliográficas

[1] SEBRAE – SERVIÇO BRASILEIRO DE APOIO ÀS MICRO E PEQUENAS EMPRESAS (Brasil) (Ed.). **Pesquisa de mercado: o que é e para que serve.** 2015. Disponível em: <http://www.sebrae.com.br/sites/PortalSebrae/artigos/Pesquisa-de-mercado:-o-que-é-e-para-que-serve>. Acesso em: 10 abr. 2015.

[2] A EXPORTAR, Aprendendo. **Planejando a Exportação.** Disponível em: <http://www.aprendendoaexportar.gov.br/flores/planejando/pesquisa.asp>. Acesso em: 10 abr. 2015.

[3] DEVMEDIA. **Iniciando expressões regulares.** 2015. Disponível em: <http://www.devmedia.com.br/iniciando-expressoes-regulares/6557>. Acesso em: 05 maio 2015.

[4] CARDOSO, Olinda Nogueira Paes. **Recuperação de Informação.** Disponível em: <http://www.dcc.ufla.br/infocomp/artigos/v2.1/art07.pdf>. Acesso em: 10 abr. 2015.

[5] BAEZA-YATES, Ricardo; RIBEIRO-NETO, Berthier. **Modern Information Retrieval.** Harlow: Addison Wesley, 1999.

[6] PEREIRA, Silvio do Lago. **Processamento de Linguagem Natural.** Disponível em: <http://www.ime.usp.br/~slago/IA-pln.pdf>. Acesso em: 20 jun. 2015.

[7] CORREA, Renato Fernandes. **Processamento de Linguagem Natural.** Disponível em: <https://sites.google.com/site/renatocorrea/temas-de-interesse/processamento-de-linguagem-natural>. Acesso em: 20 jun. 2015.

[8] TRANSLATOR, Microsoft. **Microsoft Translator API.** Disponível em: <https://www.microsoft.com/translator/api.aspx>. Acesso em: 20 jun. 2015.

[9] TRANSLATOR, Microsoft. **Automatic translation and Microsoft Translator.** Disponível em: <https://www.microsoft.com/translator/at.aspx>. Acesso em: 20 jun. 2015.

[10] LIMA, Diego Carlos Lucena de Albuquerque. **PairExtractor: Extração de Pares Livre de Domínio para Análise de Sentimentos.** 2011. 48 f. TCC (Graduação) - Curso de Ciência da Computação, Centro de Informática, Universidade Federal de Pernambuco, Recife, 2011. Disponível em: <http://www.cin.ufpe.br/~tg/2011-2/dclal.pdf>. Acesso em: 20 jun. 2015.

[11] TREETAGGER. **TreeTagger - a language independent part-of-speech tagger.** Disponível em: <http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/>. Acesso em: 20 jun. 2015.

[12] CODEPLAX. **Project Hosting for Open Source Software:** Tweetinvi uma biblioteca C# amigável Twitter. Disponível em: <https://tweetinvi.codeplex.com/>. Acesso em: 20 jun. 2015.

[13] ONESECOND. **One Second on the Internet.** Disponível em: <http://onesecond.designly.com/>. Acesso em: 27 jun. 2015.

[14] SURVEY, Google Consumer. **Google Consumer Survey.** Disponível em: <http://www.google.com/insights/consumersurveys/how>. Acesso em: 27 jun. 2015.

[15] TWITTER; ADVANCED, Twitter Search. **Twitter Search Advanced.** Disponível em: <https://support.twitter.com/articles/359432-usando-a-busca-avancada>. Acesso em: 28 jun. 2015.

[16] PANG, Bo; LEE, Lillian. **Opinion mining and sentiment analysis,** vol. 2 (1-2)**.** New York, 2008. 135 p.

[17] NASCIMENTO, Paula; Et al. **Análise de Sentimentos de Tweets com Foco em Notícias**. Universidade Federal do Rio de Janeiro - UFRJ. Disponível em: < http://www.lbd.dcc.ufmg.br/colecoes/brasnam/2012/007.pdf>. Acesso em: 28 de junho de 2015.

[18] PHD, Instituto. **Pesquisas de Mercado:** conceitos, técnicas e análises. Disponível em: <http://www.institutophd.com.br/blog/como-funcionam-as-pesquisas-de-mercado-conceitos-tecnicas-e-analises/>. Acesso em: 29 jun. 2015.

[19] JARGAS, Aurélio Marinho. **Expressões Regulares: Uma Abordagem Divertida**. 4ª Edição – Revisada e Ampliada - Novatec. Disponível em: <http://www.novatec.com.br/livros/expressoesregulares4/capitulo9788575223376.pdf >. Acesso em: 29 de junho de 2015.

[20] GALILEU, Redação. **Tweets passam a marca de 1 bilhão por mês**. Disponível em: < http://revistagalileu.globo.com/Revista/Common/0,,EMI121452-17770,00-TWEETS+PASSAM+A+MARCA+DE+BILHAO+POR+MES.html >. Acesso em: 1 de julho de 2015.

[21] Twitter. **Twitter Libraries**. Disponível em: < https://dev.twitter.com/overview/api/twitter-libraries>. Acesso em: 01 de julho de 2015.

[22] CODEPLEX. **Tweetinvi a friendly Twitter C# library**. Disponível em: < https://tweetinvi.codeplex.com/documentation?version=49>. Acesso em: 05 de julho de 2015.

[23] SENTIMENT140. **General Information**. Disponível em: < http://help.sentiment140.com/>. Acesso em: 05 de julho de 2015.

[24] SENTIMENT140. **API Registration**. Disponível em: < http://help.sentiment140.com/api/registration>. Acesso em: 05 de julho de 2015.

[25] BLUMETTI, Bruno; Et al. **Seleção de Informações Usando Text Mining com RI**. Disponível em: <http://textmining.xpg.uol.com.br/TrabBD.pdf>. Acesso em: 05 de julho de 2015.

[26] LAND, Marketing. **Google Launches Consumer Surveys**. Disponível em:< http://marketingland.com/google-consumer-surveys-9008>. Acesso em: 06 de julho de 2015.

[27] CLARABRIDGE. **Sentiment Analysis and Business Sense**. Disponível em:< http://www.clarabridge.com/sentiment-analysis-and-business-sense/>. Acesso em: 06 de julho de 2015.

[28] TWITTER. **Twitter Advanced Search**. Disponível em:< https://twitter.com/search-advanced>. Acesso em: 06 de julho de 2015.

[29] LINKEDIN. **Microsoft Translator**. Disponível em:< https://www.linkedin.com/company/microsoft-translator?trk=biz-brand-tree-co-name>. Acesso em: 06 de julho de 2015.

[30] ARTIGOS. **Tipos de métodos de pesquisa de mercado**. Disponível em:< http://www.administradores.com.br/artigos/negocios/tipos-de-metodos-de-pesquisa-de-mercado/12920/>. Acesso em: 06 de julho de 2015.

[31] FERNEDA, Edberto. **Recuperação da informação: Análise sobre a contribuição da Ciência da Computação para a Ciência da Informação**. Disponível em:< file:///C:/Users/Silas%20Silva/Downloads/Tese%20(4).pdf>. Acesso em: 07 de julho de 2015.

[32] SURVEY, Google Consumer. **Google Consumer Survey.** Disponível em: < http://www.google.com/insights/consumersurveys/home>. Acesso em: 27 jun. 2015.

[33] LUO, Zhunchen; OSBORN, Miles; Et al. **Improving Twitter Retrieval by Exploiting Structural Information.** Disponível em: < https://www.aaai.org/ocs/index.php/AAAI/AAAI12/paper/download/4913/5252>. Acesso em: 01 Ago. 2015.