CENTRO UNIVERSITÁRIO LUTERANO DE PALMAS
ULBRA
Recredenciado pela Portaria Ministerial nº 3.607, de 17/10/05, D.O.U. nº 202, de 20/10/2005
ASSOCIAÇÃO EDUCACIONAL LUTERANA DO BRASIL

Felipe Eduardo Bechert Schmitz

APLICAÇÃO DA TÉCNICA DE TEXT MINING PARA COMENTÁRIOS RELACIONADOS AO CONTEXTO DO TURISMO

Palmas - TO

2015

Felipe Eduardo Bechert Schmitz

APLICAÇÃO DA TÉCNICA DE TEXT MINING PARA COMENTÁRIOS RELACIONADOS AO CONTEXTO DO TURISMO

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Sistemas de Informação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientadora: Prof. M.Sc. Parcilene Fernandes de Brito.

Palmas - TO

2015

Felipe Eduardo Bechert Schmitz

APLICAÇÃO DA TÉCNICA DE TEXT MINING PARA COMENTÁRIOS RELACIONADOS AO CONTEXTO DO TURISMO

Trabalho de Conclusão de Curso (TCC) elaborado e apresentado como requisito parcial para obtenção do título de bacharel em Ciência da Computação pelo Centro Universitário Luterano de Palmas (CEULP/ULBRA).

Orientadora: Prof. M.Sc. Parcilene Fernandes de Brito.

Aprovada em: 7 de dezembro de 2015.

BANCA EXAMINADORA

_____

Prof. M.Sc. Parcilene Fernandes de Brito

Centro Universitário Luterano de Palmas

_____

Prof. M.Sc. Fernando Luiz de Oliveira

Centro Universitário Luterano de Palmas

_____

Prof. M.Sc. Madianita Bogo

Centro Universitário Luterano de Palmas

Palmas - TO

2015

Aos meus queridos amigos, família e professores.

AGRADECIMENTOS

Aos meus pais, por terem me dado forças e me apoiado por toda minha vida.

Ao meu irmão, por sempre ter uma visão positiva e me incentivar a continuar.

Ao CEULP/ULBRA e ao corpo docente que oportunizaram este momento, pela confiança, mérito e ética.

A minha orientadora Parcilene Fernandes de Brito por ter acreditado em mim e dedicado seu precioso tempo a me orientar, mesmo depois de eu desaparecer por semanas sem dar notícia.

Agradeço ao meu grande amigo Lucas Moreno, que não me deixou desistir em nenhum momento e me mandava estudar todos os dias.

Aos amigos de infância e aos que conheci na universidade, que mesmo sem manter contato constante, estavam sempre do meu lado.

A todos que participaram direta ou indiretamente da minha formação, o meu muito obrigado.

RESUMO

Este trabalho faz parte da pesquisa realizada por Brito (2105) e tem como objetivo utilizar técnicas de *Text Mining* para extrair regras de associação de aspectos de comentários relacionados à temática turismo. Com o crescimento da colaboração *online*, o volume de dados contendo opiniões na forma de comentários se tornou muito grande para ser analisado sem o auxílio de ferramentas especializadas. Esta coleta de grandes quantidades de texto e a preparação desses dados para a descoberta de conhecimento é chamada de *Text Mining*. A partir da mineração, diversas técnicas podem ser aplicadas sobre os dados extraídos para se obter diversos tipos de informação. Sobre o banco de dados minerado de comentários de um site de turismo e aspectos extraídos através da técnica de análise de sentimentos por Christhie (2015), aplicou-se a técnica de associação com o algoritmo Apriori para se extrair as regras associativas entre os aspectos positivos mais frequentes nos comentários. A partir destas regras, foi possível identificar conexões, analisar como estes aspectos interagem e tirar conclusões sobre a importância destes aspectos para o domínio estudado.

PALAVRAS-CHAVE: Text Mining, Associação, Aspectos, Apriori, Turismo.

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

1 INTRODUÇÃO

A criação de ferramentas de interação e colaboração na internet acarreta no crescimento contínuo da quantidade de informações disponíveis sobre as mais diversas temáticas. Uma forma de colaboração que se tornou comum na internet foi o compartilhamento de opiniões de usuários sobre todo tipo de assunto. A grande quantidade de críticas fornecidas sobre um produto ou serviço se mostra valiosa, tanto para clientes que querem tomar a decisão informada sobre o que será adquirido, quanto para os fornecedores, inclusive concorrentes, que querem conhecer os pontos positivos e negativos avaliados para se adaptar e conquistar esses clientes.

Mas, o acompanhamento dessa grande quantidade de informação se tornou impraticável sem o auxílio de ferramentas capazes de fazer agregações e fornecer dados sucintos e relevantes.

Este trabalho faz parte da pesquisa realizada por Brito (2105), que tem como foco a análise de avaliações de usuários sobre destinos turísticos. Em Christhie (2015), os comentários e dados relacionados foram extraídos do site TripAdvisor, passaram por pré-processamento e análise de sentimentos a nível de aspecto para extrair opiniões positivas e negativas. Depois de extraídos e processados, os dados obtidos servem de entrada para diversos tipos de análises, entre eles a classificação, agrupamento e associação. O banco de dados resultante foi utilizado como base para o presente estudo.

O objetivo deste trabalho foi utilizar técnicas de Text Mining para extrair regras de associação dos comentários relacionados à temática turismo, e para isso objetivou-se especificamente:

- Organizar as informações do domínio;
- Extrair regras de associação de aspectos dos comentários utilizando o algoritmo Apriori;
- Analisar os resultados para se verificar as informações extraídas são pertinentes ao objeto de estudo.

Neste trabalho foi utilizada a técnica de associação, com o algoritmo Apriori, para encontrar regras associativas entre aspectos encontrados via análise de sentimentos em uma base de comentários minerados de um website de turismo. O algoritmo foi aplicado utilizando-se a ferramenta Weka, que usa como entrada um

formato de arquivo específico e permite alterar parâmetros da execução do algoritmo via interface gráfica.

Esta monografia foi dividida com a seguinte estrutura: a seção 2 contém uma revisão sobre *Text Mining*, suas etapas e técnicas, a seção 3 dispõe os materiais e métodos utilizados para o desenvolvimento do trabalho, o item 4 apresenta os desafios e discussões sobre as análises realizadas e, por fim, as considerações finais sobre o desenvolvimento do trabalho e possíveis trabalhos futuros.

2 REFERENCIAL TEÓRICO

Neste referencial será feita uma introdução ao conceito de Text Mining, as etapas necessárias para o processo e algumas de suas aplicações. Também serão expostos alguns exemplos da utilização das técnicas envolvidas, assim como os principais algoritmos para sua implementação.

2.1. Text Mining

O Text Mining é uma etapa intermediária do processo KDT, sigla em inglês para Descoberta de Conhecimento em Textos. Dentro deste processo maior, a mineração de textos é uma etapa de recuperação da informação, que acontece após a coleta de documentos e resulta em uma base de informações das quais será extraído o conhecimento.

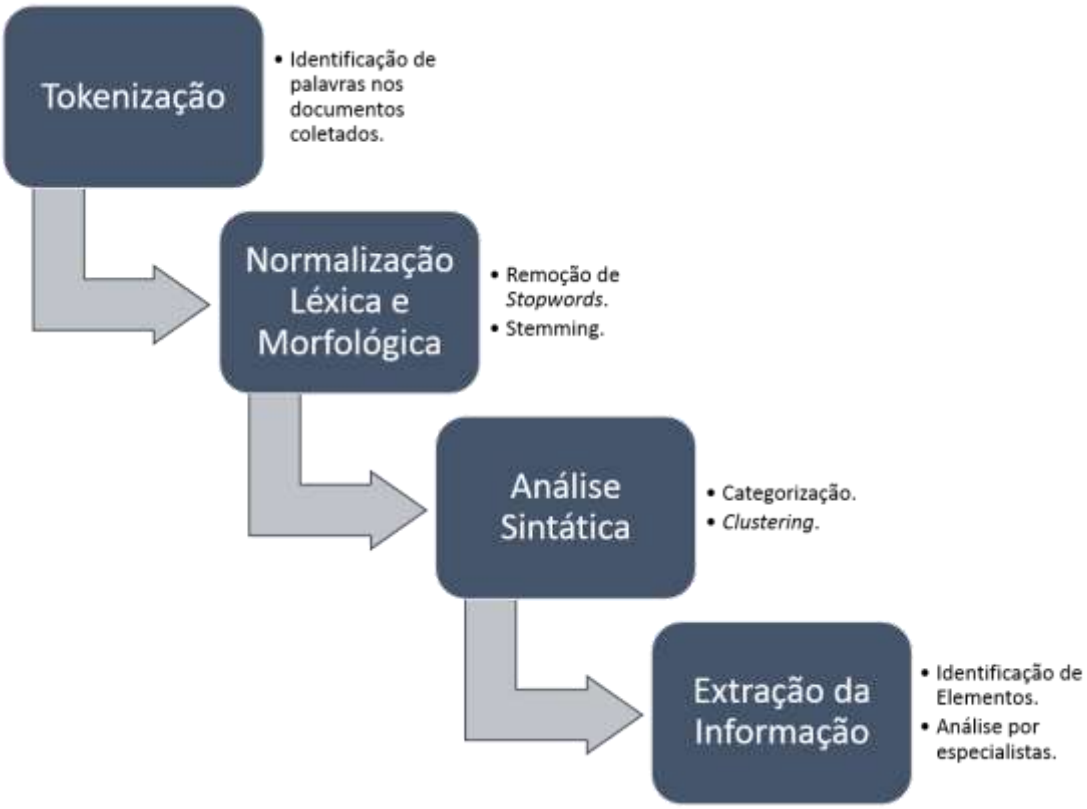
Segundo Feldman (2007, p. 1, tradução nossa), a mineração de texto ou Text Mining pode ser definida como um processo “análogo ao Data Mining, no qual se extrai informações úteis de documentos textuais através da identificação e exploração de padrões interessantes”. Text Mining é uma técnica de automação de análise de grandes quantidades de documentos textuais para se obter conhecimento sobre um determinado domínio.

Chen (2001, p. 18) estimava que pelo menos 80% das informações online eram baseadas em texto não estruturado, ou seja, documentos em linguagem natural. Isso torna o Text Mining ainda mais importante na descoberta de conhecimento dentro de um domínio.

Estimativas mais recentes (HOLZINGER et al., 2013) acrescentam que até 2020 os dados online poderão totalizar algo em torno de 40 zettabytes (40 bilhões de gigabytes) e ainda será mantida a proporção de 80% de textos não estruturados.

Na Figura 1 é apresentada a arquitetura simplificada de um sistema de Text Mining e suas etapas principais.

Figura 1: Arquitetura básica de um sistema de Text Mining.



De acordo com a figura 1, Tokenização se refere à separação do texto nos documentos a serem analisados, através da remoção de símbolos e marcações. Normalização Léxica e Morfológica se refere à preparação desse texto para ser processado e, de certa forma, compreendido por um computador. Análise Sintática é a parte do processo em que o texto é processado e organizado por algum algoritmo preparado para encontrar padrões e auxiliar no processo de Extração da Informação, que é de fato o momento em que um especialista analisa o que foi encontrado para extrair conhecimento sobre o domínio. Todas estas etapas são explicadas com mais detalhes nas seções subsequentes.

2.1.1. Captação de Dados Textuais

O elemento chave do Text Mining é a coleção de documentos, que pode ser definida como qualquer grupo de elementos textuais a serem processados. Os elementos textuais podem variar de pequenos trechos de poucos caracteres até livros completos, e a quantidade entre algumas centenas até dezenas de milhões (e contando). A coleção de documentos pode ser estática, contendo apenas os elementos coletados

inicialmente, ou dinâmica, caracterizada pela inclusão de novos elementos ao longo do tempo.

Os documentos podem ser coletados a partir de uma base (e.g. PubMed, IEEExplore, ACM Digital Library, Domínio Público), fornecidos manualmente (arquivos ou textos alimentados por um ser humano) ou obtidos com o uso de um *crawler*, que descobre elementos textuais a partir de uma página web e links subsequentes.

"O objetivo do crawling é coletar a maior quantidade possível de páginas web, juntamente com os links que as interconectam, de maneira rápida e eficiente" (MANNING, 2009, online, tradução nossa). É possível também projetar o *crawler* para extrair partes das páginas coletadas a partir da estrutura de elementos do código HTML automatizando a separação do que é relevante para a mineração, como artigos, listas e comentários, do resto do conteúdo da página.

Porém, os processos de Text Mining geralmente não executam seus algoritmos de descoberta de conhecimento em elementos textuais não previamente tratados e não estruturados. Para isso, antes da extração de conhecimento, os elementos textuais devem passar por um processo de limpeza, tratamento de normalização, conhecidos como Operações de Processamento. Esse processo possui algumas etapas que serão abordadas na próxima seção.

2.1.2. Pré-Processamento

Esta etapa da mineração de texto consiste em preparar a coleção de documentos para a extração do conhecimento. Ela pode ser dividida em alguns processos gerais, mas podem variar em quantidade de acordo com a necessidade de preparação do domínio. A ordem mais comum desses processos pode ser observada na Figura 2.

Figura 2: Etapas comuns do pré-processamento.

A Análise Léxica (Tokenização) consiste na conversão de uma cadeia de caracteres de entrada em um vetor de palavras ou *tokens*, isto é, a separação de palavras e eliminação de marcações, símbolos, caracteres especiais e espaços. Nesta parte se pode utilizar um dicionário de termos para eliminar a ocorrência de erros ortográficos e um dicionário de sinônimos para se trabalhar com um vocabulário

controlado (MORAIS, 2007, p. 13). Para auxiliar no entendimento, a Figura 3 oferece um pequeno exemplo do resultado deste processo.

Figura 3: Exemplo de Tokenização.

Texto de entrada	Texto após a Tokenizaão
A ligeira raposa marrom salta sobre o cão preguiçoso.	"A", "ligeira", "raposa", "marrom", "salta", "sobre", "o", "cão", "preguiçoso"

A Identificação de Termos Compostos (*Word-phrase formation*) é a detecção de conceitos que só podem ser expressos através da utilização de duas ou mais palavras adjacentes, as quais perdem o sentido quando separadas (e.g. programa social, programa de computador). Para isso o algoritmo pode, ao identificar uma grande frequência de correlação de termos, solicitar ao usuário a seleção correta ou recorrer a um dicionário de expressões, quando disponível.

A Remoção de Termos Irregulares (*stopwords*) é a etapa em que ocorre a remoção de palavras que não fazem diferença na indexação, como artigos, conjunções e preposições. Essas palavras podem apresentar uma frequência muito alta e não tem nenhum valor para a busca dentro do contexto do documento, por isso devem ser removidas antes da extração do conhecimento. Muitos estudos oferecem listas de *stopwords* (*stoplists* ou dicionários negativos) que podem ser utilizadas livremente (WIVES, 2002, p. 53). É utilizado o mesmo exemplo para demonstrar a remoção de *stopwords* na Figura 4.

Figura 4: Exemplo de remoção de Stopwords.

Texto de entrada	Texto após a Remoção de Stopwords
"A", "ligeira", "raposa", "marrom", "salta", "sobre", "o", "cão", "preguiçoso"	"ligeira", "raposa", "marrom", "salta", "sobre", "cão", "preguiçoso"

A Normalização Morfológica (*stemming*) não é uma parte obrigatória e pode ser dispensada dependendo do tipo de análise que se deseja realizar. Consiste na redução de palavras ao seu radical pela eliminação de sufixos, plurais e inflexões de gênero. Com esse processo é possível diminuir o tamanho de um documento em até 50% (MORAIS, 2007, p. 15). Existem diversos algoritmos para essa etapa, mas a grande maioria deles são específicos para uma determinada língua. Um algoritmo eficiente de *stemming* para a língua portuguesa pode ser observado em Orengo (2001). A seguir, na Figura 5, está o exemplo após o processo de *stemming*.

Figura 5: Exemplo de Stemming

Texto de entrada	Texto após o Stemming
"ligeira", "raposa", "marrom", "salta", "sobre", "cão", "preguiçoso"	"ligeir", "rapos", "marrom", "salt", "sobr", "cao", "preguic"

O Cálculo de Relevância visa identificar em um texto quais palavras representam maior peso ou relevância em relação às outras palavras. A definição desse peso pode ser calculada, dependendo do caso, analisando-se a frequência, posição sintática, palavras adjacentes, estrutura do documento e até a classe gramatical de um termo. Isso vai variar dependendo do tipo de análise que se pretende fazer.

A partir destes passos se pode partir para o processo de classificação ou agrupamento dos documentos textuais tratados. Esses dois processos têm finalidades diferentes e serão abordados nas próximas seções.

2.1.3. Classificação ou Categorização

Esta etapa consiste na organização de documentos em categorias ou tópicos a partir do seu conteúdo. "Na categorização, coleções de documentos são processadas e agrupadas em categorias predeterminadas baseado em uma taxonomia fornecida pelo usuário" (DÖRRE, 1999, p. 3, tradução nossa). Provavelmente, é a maneira mais utilizada para a separação de textos de acordo com seus temas. Feito a partir de engenharia do conhecimento, programado no algoritmo ou por aprendizado de máquina, o processo separa os textos ou documentos em categorias pré-determinadas. A categorização tem três aplicações mais comuns para organizar os documentos em contextos diferentes, como é disposto a seguir (ARANHA, 2009; FELDMAN, 2007; MANNING, 2009).

Na Indexação de textos, utiliza-se um vocabulário controlado para auxiliar a recuperação da informação em algoritmos de busca. Por exemplo, a classificação visa atribuir de 1 a *k* (sendo *k* o máximo previsto pelo algoritmo) palavras-chave para o texto. Essas palavras-chave são parte de um vocabulário ou lista de *tags* que vai organizar os textos em um thesaurus (enciclopédia) onde poderá ser facilmente recuperado com uma busca simples.

Para armazenamento e filtragem de textos, a classificação atribui cada texto a uma única "caixa". Por exemplo: uma revista deve dividir seus artigos entre "carros",

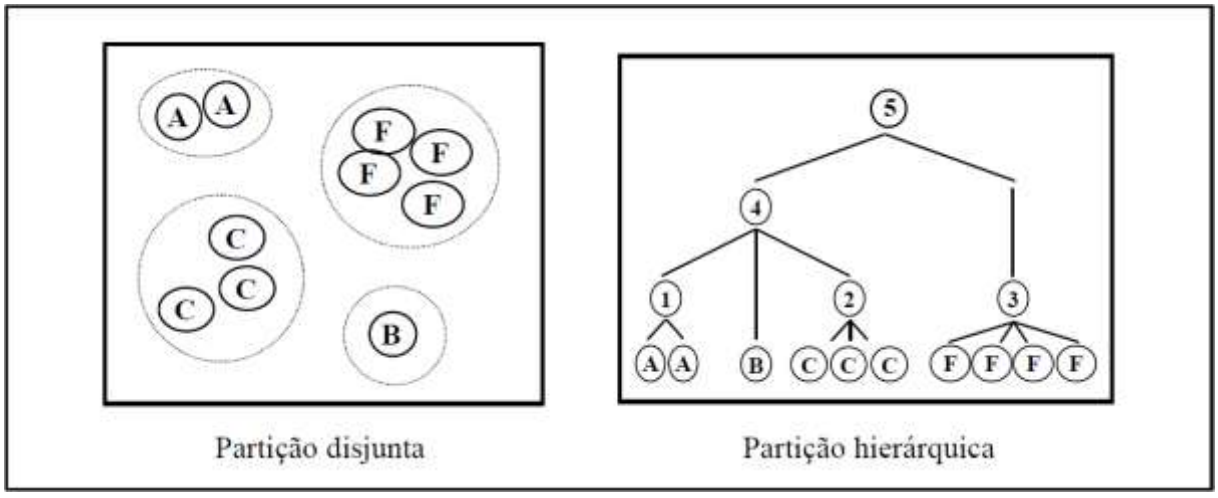
“economia”, “mundo”, “politica” entre outros. O maior problema desta aplicação é a precisão necessária para a classificação, que pode ser melhorada a partir do aprendizado de máquina assistido.

Outra aplicação é a categorização hierárquica de páginas web, semelhante à anterior, mas devido à natureza e volatilidade dos documentos, cada “caixa” é dividida em subseções para facilitar a navegação e evitar que uma seção fique excessivamente grande. O processo deve suportar a inserção e remoção de tais subseções.

2.1.4. Agrupamento ou Aglomeração (Clustering)

Clustering é um processo não supervisionado de agrupamento a partir de análise de correlação. Diferente da classificação, o algoritmo não recebe um conjunto pré-determinado de categorias ou grupos para separação, mas é encarregado de agrupar textos por correlação e semelhança. Este agrupamento também pode ser dividido (ou especializado) em duas categorias: Plana e Hierárquica. A Figura 6 é uma representação da diferença entre os dois tipos. À esquerda é apresentado o Agrupamento Plano (partição disjunta) e à direita o Agrupamento Hierárquico (partição hierárquica).

Figura 6: Representação dos dois tipos de agrupamento



Fonte: Wives (2002, p. 93)

No Agrupamento Plano (*Flat Clustering*) os documentos são separados de forma que sejam coerentes entre si e claramente diferentes ou isolados dos outros *clusters* (MANNING, 2009, p. 349). O método é eficiente e conceitualmente simples,

com o objetivo de criar grupos completamente autocontidos e dissimilares, ou seja, sem qualquer estrutura aparente entre eles.

O Agrupamento Hierárquico (*Hierarchical Clustering*) pode ser implementado de maneira ascendente (*bottom-up*) ou descendente (*top-down*). Na abordagem *bottom-up* cada documento é considerado um *cluster* isolado e então é sucessivamente aglomerado em pares próximos até resultar em um único cluster que engloba todos os documentos (ver implementação HAC). Já na *top-down* é necessário um método de cálculo para separar os documentos relacionados em *sub-clusters*. Inicialmente todo o conjunto de documentos é considerado um único *cluster*, que é subdividido até que se chegue no *cluster* formado por apenas um documento.

Existem diversos algoritmos que podem ser utilizados para se chegar ao agrupamento de termos, aqui estão destacados os três mais citados na literatura (FELDMAN, 2007; SOARES, 2008):

O algoritmo *K-means* particiona uma coleção de vetores entre um conjunto de grupos (clusters). É o mais conhecido, por ser simples e eficiente. O lado ruim desse algoritmo é a alta dependência da escolha inicial das *seeds* (sementes aleatórias ou sequências de caracteres que são usadas pelo computador para gerar números mais aleatórios), mas ainda é viável pois necessita de poucas iterações se comparado com os outros métodos. O algoritmo procede como a seguir:

- Inicialização: k-seeds (sementes-k) são fornecidas ou selecionadas aleatoriamente. Cada documento, considerado um “vetor”, é atribuído ao cluster com a semente mais próxima;
- Iteração: Os centroides (documentos centrais de cada *cluster*) são computados. Cada vetor é atribuído ao centroide mais próximo;
- Condição de parada: Convergência, quando não ocorrem mais mudanças.

O *EM-based Probabilistic Clustering Algorithm* (em inglês, Algoritmo de agrupamento probabilístico baseado em Maximização de Expectativa) é um *framework* de propósito geral para estimar parâmetros de distribuição na presença de variáveis ocultas em dados visíveis. Pode ser adaptado para o agrupamento de termos como exposto:

- Inicialização: Os parâmetros de distribuição *k* são selecionados, seja manualmente ou aleatoriamente;
- Iteração:
    - Passo E - Computa a probabilidade para os objetos usando os parâmetros atuais de distribuição. Rotula todos os objetos de acordo com a probabilidade;
    - Passo M - Estima novamente os parâmetros de distribuição para maximizar a probabilidade de o objeto assumir seu rótulo atual;
- Condição de parada: Convergência, quando a probabilidade após a iteração é muito pequena.

No *Hierarchical Agglomerative Clustering* (HAC) (do inglês, Agrupamento Aglomerativo Hierárquico), cada objeto é considerado um *cluster*, e a cada iteração eles são agrupados em pares mais próximos em uma espécie de árvore binária, até resultar em um único *cluster*:

- Inicialização: Cada objeto é considerado um *cluster*;
- Iteração: Encontra um par de clusters mais próximos e os funde em um só;
- Condição de parada: Quando tudo está agrupado em um único *cluster*.

Existem vários outros algoritmos de agrupamento baseados em grafos, mas são pouco comuns, sendo aplicados a casos específicos. Alguns exemplos são a Árvore de Mínima Abrangência (MST), o Agrupamento de Vizinho Mais Próximo e o algoritmo Buckshot.

2.1.5. Técnica de Associação (Apriori)

Regras de associação tem o objetivo de determinar os elementos que implicam a presença de outros elementos em um conjunto, ou seja, identificar padrões frequentes de dados.

Existem diversos algoritmos que buscam encontrar regras de associação em um conjunto de dados, dentre eles o Apriori e FP-Growth. O algoritmo FP-Growth utiliza a estrutura de dados FP-Tree (do inglês, Árvore de Padrões Frequentes) e um método simplificado de busca para encontrar regras de associação em um conjunto grande de variáveis. Como neste trabalho foi decidido por um número reduzido de aspectos/variáveis, utilizou-se o algoritmo Apriori que é descrito a seguir.

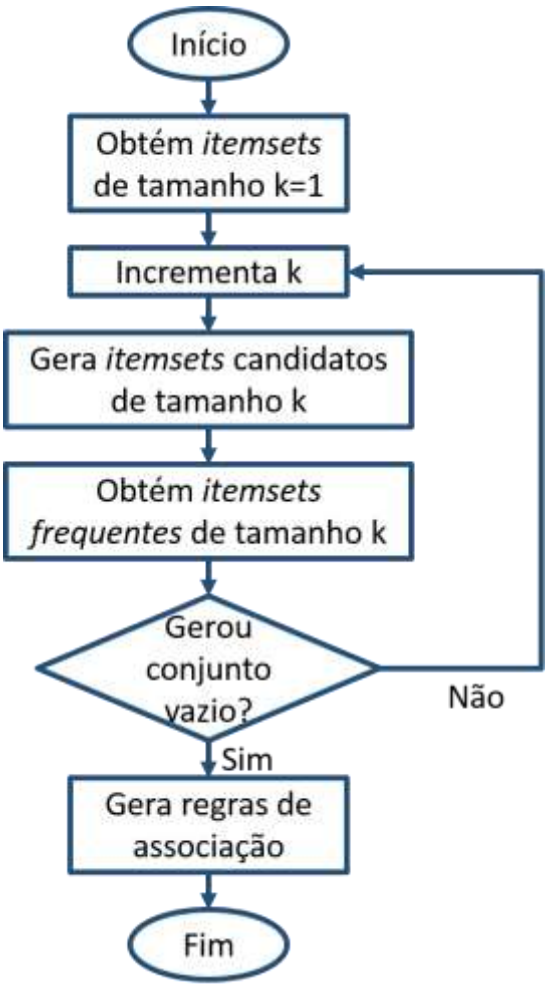
O algoritmo Apriori, introduzido inicialmente por Agrawal (1993) parte do princípio de que, se um conjunto é frequente, todos seus subconjuntos também devem ser frequentes. A análise consiste nas duas etapas a seguir. (1) identificar conjuntos frequentes e (2) gerar regras de associação a partir dos conjuntos frequentes. A Figura 7 apresenta o pseudocódigo desse algoritmo (KRISHNA, 2010).

Figura 7: Pseudocódigo do algoritmo Apriori.

C_k: $Conjunto\ candidato\ de\ tamanko$ k
I_k: $Conjunto\ frequente\ de\ tamanho$ k
$para\ (k = 2;\ I_{k-1} \neq 0; k++)\ faça$:
    $C_k = apriori - gen(I_{k-1})$ // Novos candidatos
    $para\ todas\ transações\ T \in D\ faça$:
        $C_T = subset(C_k, T)$ // Candidatos contidos em T
        $para\ todos\ candidatos\ c \in C_t\ faça$:
            $c.cont++$
        fim
    fim
    $I_k = \{c \in C_k | c.cont \geq min\ sup\}$
fim
$Resposta = \bigcup_k I_k$

O pseudocódigo apresenta o funcionamento da primeira etapa do algoritmo em nível de programação. Para um melhor entendimento do processo, considere o fluxograma apresentado na Figura 8.

Figura 8: Fluxograma do algoritmo Apriori.



Como primeiro passo, o algoritmo identifica os conjuntos possíveis com a quantidade de itens (*k*) mínima, ou seja, os itens únicos existentes no conjunto. Em cada iteração o *k* é incrementado para a geração de conjuntos (*itemsets*) candidatos.

Conjuntos candidatos são todas as possibilidades de combinações de tamanho *k* entre os itens do conjunto anterior. *Itemsets* frequentes são os conjuntos, dentre os candidatos, que tem o suporte mínimo, ou seja, que ocorrem pelo menos um número determinado de vezes. O suporte pode ser uma quantidade fixa no algoritmo ou uma porcentagem referente ao número total de dados.

Estes passos do algoritmo se repetem até que seja gerado um conjunto vazio de *itemsets* frequentes, i.e., quando não há ocorrências com o suporte mínimo para nenhum conjunto candidato. Neste caso, as listas de conjuntos frequentes das iterações anteriores são utilizadas para a geração das regras de associação.

Para se gerar as regras de associação, cada *Itemset* frequente $I_k \mid k > 1$ passa por duas etapas. Na primeira etapa os conjuntos são divididos em todos seus subconjuntos S_{k-1} possíveis. Na segunda etapa são filtrados os conjuntos com a confiança mínima. A confiança *c* é calculada a partir do suporte (frequência) do conjunto I em relação ao conjunto S ($c = suporte(I)/suporte(S)$). As regras com a confiança maior ou igual à mínima são, então, sugeridas no formato $S \rightarrow (I - S)$.

Um exemplo prático deste algoritmo pode ser observado a seguir, utilizando dados fictícios relacionados ao tema do trabalho. A Figura 9 apresenta 6 aspectos comuns a avaliações e a presença de cada um no comentário da avaliação (representado pelo valor 1).

Figura 9: Exemplo de dados de entrada para o algoritmo Apriori.

	Atendimento	Comida	Lugar	Ambiente	Hotel	Restaurante
1	1	0	1	0	1	0
2	1	0	0	0	0	1
3	0	1	0	0	0	0
4	1	0	0	1	0	1
5	1	0	0	0	0	0
6	0	0	0	0	0	0
7	0	0	0	1	0	1
8	0	0	0	1	0	0
9	1	0	1	0	1	0
10	0	1	0	0	1	0
11	0	1	0	0	0	0
12	0	0	1	1	0	0

Para estas entradas, o algoritmo Apriori foi executado com os valores padrão de confiança mínima (a medida da frequência com a qual essa regra é validada) e encontrou os seguintes padrões:

1. lugar=1 hotel=1 2 ==> atendimento=1 2 conf:(1)
2. atendimento=1 hotel=1 2 ==> lugar=1 2 conf:(1)

3. atendimento=1 lugar=1 2 ==> hotel=1 2 conf:(1)
4. atendimento=1 ambiente=1 1 ==> restaurante=1 1 conf:(1)

O formato de saída do algoritmo apresenta uma regra por linha, dividida em duas partes:

1. Premissa (lugar hotel 2): o padrão de variáveis associadas que apresenta uma consequência frequente e a quantidade de ocorrências desse padrão.
2. Consequência (atendimento 2 conf:(1)): o resultado frequente do padrão, a quantidade de ocorrências e o nível de confiança da regra, que é a frequência da consequência dividida pela frequência do padrão.

Os padrões encontrados são organizados por confiança, a medida da frequência com a qual essa regra é validada. O primeiro padrão foi encontrado nas entradas 1 e 9, indicando que se os aspectos lugar e hotel são citados, logo o aspecto atendimento poderá ser citado com um grau de confiança 1 (100%). O último padrão foi encontrado na entrada 4, e afirma que se os aspectos atendimento e ambiente são citados, o aspecto restaurante pode ser citado.

Com um conjunto de dados maior, com mais entradas e aspectos para comparação, o algoritmo pode encontrar padrões com uma confiança inferior a 1, demonstrando uma probabilidade de o aspecto ocorrer quando seus predecessores ocorrem.

2.1.6. Extração da Informação

Processo assistido por especialistas (humanos) que ajustam a precisão do sistema e corrigem possíveis erros. No processo de Extração da Informação (EI) são buscados alguns padrões ou *templates* textuais dentro dos documentos preparados para identificar um elemento a ser extraído.

Em geral sistemas de Extração de Informação são eficientes em alguns casos (FELDMAN, 2007, p. 94, tradução nossa):

- A informação a ser extraída é explícita e não necessita maior inferência;
- Um número pequeno de *templates* é necessário para resumir as partes relevantes do documento;
- A informação necessária é expressa relativamente localmente no texto.

Também segundo Feldman (2007), existem quatro tipos básicos de elementos e formas compostas desses elementos que podem ser extraídos de um texto utilizando padrões especializados, assim como qualquer combinação desses elementos que pode ser formada. Esses elementos são apresentados a seguir:

- Entidades: são os sujeitos básicos de um texto, como pessoas, locais e empresas;
- Atributos: são características das entidades extraídas, como idade da pessoa e tipo de uma organização;
- Fatos: são relações entre entidades, como o cargo de uma pessoa em uma companhia;
- Eventos: são atividades das quais as entidades participam, como a venda de uma empresa ou a contratação de um funcionário.

Um dos padrões de texto comumente utilizados é o de data que, dependendo da região e língua utilizada, pode ser encontrado, por exemplo, no formato “99/99/9999”, onde cada “9” representa um caractere numérico e as barras (“/”) separam dia, mês e ano respectivamente. Uma extração mais precisa também buscaria com uma expressão regular mais flexível e consideraria como válidas apenas datas onde o dia está em um intervalo ente 1 e 31 e mês entre 1 e 12, por exemplo.

Outro padrão, comum em textos médicos, é um número seguido de consoantes como “mg”, “ml” e “cc”, que informa a quantidade de uma determinada substância em uma fórmula ou em um tratamento. Formas melhoradas desse padrão consideram palavras que precedem e/ou seguem essa quantidade, como em “ácido ascórbico 1000mg” e “5cc de soro fisiológico”.

2.1.7. Visualização da Informação Extraída

Após todo o processo de extração da informação, um usuário especialista precisa de alguma forma de visualizar e interagir com o que foi obtido pelo processo de Text Mining. Para isso é necessário a utilização de uma ferramenta que permita pelo menos fazer pesquisas e filtragens nesses resultados.

Mesmo com formas de pesquisar e interagir com as informações resultantes da mineração, uma grande quantidade de texto pode ser retornada para ser analisada por um ser humano. Esta abundancia de dados levou desenvolvedores de

ferramentas de Text Mining a criarem também métodos criativos de visualização desses dados (FELDMAN, 2007, p. 189). Diferentes tipos de gráficos podem responder diferentes perguntas sobre o mesmo conjunto de informações. Por exemplo: Quantos padrões foram encontrados sobre o assunto X? Qual é a concentração do termo Y em diferentes agrupamentos? Cada pergunta leva a um tipo específico de pesquisa que pode ser representada por tipos diferentes de gráficos.

Existem várias ferramentas no mercado para essa finalidade, cada uma oferecendo diferentes métodos para organizar e visualizar dados, de listas filtráveis a panoramas topográficos tridimensionais. Yang et al. (2008) apresenta um apanhado das ferramentas disponíveis até então, comparando vários aspectos entre elas, como quão estruturado os dados de entrada precisam ser, que tipo de visualização eles permitem e a quantidade de opções de visualização que oferecem.

3 MATERIAIS E MÉTODOS

Esta seção apresenta a metodologia utilizada para a realização deste projeto, desde a finalidade do trabalho até os materiais e etapas necessárias para sua conclusão.

3.1. Objeto de estudo

Este trabalho teve como propósito a aplicação de técnicas de Text Mining sobre comentários de um website voltado à temática do turismo para a identificação das associações que podem ser geradas a partir das características presentes nesses comentários. Os comentários foram extraídos do site TripAdvisor a partir de um trabalho realizado em Christhie (2015), e faz parte da pesquisa sobre Análise de Sentimento e Comportamental realizada em Brito et al (2015). Após a filtragem pelos aspectos mais frequentes, foram analisados 905.039 comentários com aspectos positivos.

3.2. Materiais

Após a definição do tema foi feito o levantamento de materiais bibliográficos relevantes, incluindo livros, artigos e outros conteúdos digitais.

Com o material bibliográfico coletado e estudado criou-se o referencial teórico sobre Text Mining com uma passagem do assunto e ênfase nas partes mais relevantes para o entendimento do trabalho.

Para a realização da análise dos dados foi escolhida a ferramenta Weka, desenvolvida na Universidade de Waikato, que consiste em uma coleção de algoritmos de análise de dados e aprendizado de máquina e fornece uma interface gráfica que facilita a configuração e utilização desses algoritmos. O Weka trabalha com um formato de arquivo próprio, ARFF (Attribute-Relation File Format), que descreve instâncias que compartilham os mesmos atributos. A descrição desse formato de arquivo é aprofundada na seção 4.2.

3.3. Procedimentos

O trabalho consistiu na análise de uma base de dados contendo avaliações de usuários de um site de turismo e aspectos previamente extraídos utilizando a técnica de análise de sentimentos.

A partir da análise do domínio, decidiu-se pela extração de regras de associação de aspectos identificados nos comentários. Um aspecto é representado por um substantivo, simples ou composto, que identifica uma característica de um produto turístico citada em um comentário.

Para a extração das regras de associação, foi escolhido o algoritmo Apriori, por ser mais eficiente em conjuntos menores de variáveis, já que cada comentário geralmente não apresenta muitos aspectos.

O algoritmo Apriori foi aplicado utilizando a ferramenta Weka, que fornece uma interface gráfica para facilitar a alteração de parâmetros da análise, como grau mínimo de confiança para criação da regra.

Após a aplicação do processo de descoberta de regras de associação, foram realizadas mais reuniões com a orientadora para analisar os conjuntos encontrados e discutir modificações nos dados que seriam extraídos do banco para melhorar os dados e analisar os resultados.

4 RESULTADOS E DISCUSSÃO

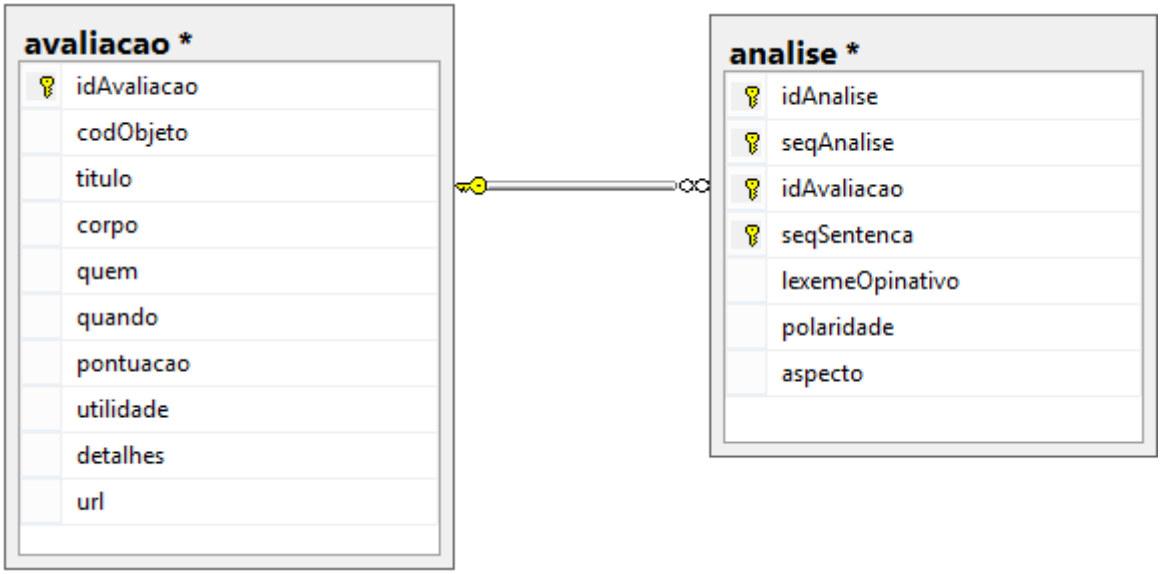
Para a definição dos resultados apresentados nessa seção, foi utilizado um banco de dados que consiste no resultado da mineração, pré-processamento e análise de sentimentos em avaliações dos usuários de um site de turismo. O banco já possui os dados referentes à análise de sentimentos, onde foram levantados os aspectos citados nos comentários das avaliações e sua polaridade (positiva ou negativa).

As seções a seguir apresentam detalhamentos dos passos do trabalho, desde a extração dos dados do banco até a análise dos resultados do algoritmo Apriori em relação ao domínio.

4.1. Criação de consultas no Banco de Dados para Realizar a Análise

Os dados utilizados nas análises foram extraídos de duas tabelas do banco de dados. A estrutura destas tabelas é exibida na Figura 10.

Figura 10: Estrutura das tabelas utilizadas para extração dos dados.



A tabela *avaliação* contém todas as informações de um comentário, como texto completo e pontuação dada pelo usuário. Os resultados da análise de sentimentos realizada por Christhie (2015) foram salvas na tabela *analise*, que tem referências para outras tabelas relacionadas, o aspecto analisado e a polaridade do aspecto.

Para a execução do algoritmo Apriori, foram extraídos do banco de dados os aspectos de polaridade positiva relacionados a cada comentário. Os dados contidos no banco consistem na análise de 1.4 milhão de avaliações (comentários e notas referentes a um destino turístico). Sobre essas avaliações foram levantados mais de 16 milhões de aspectos, sendo eles positivos, negativos ou neutros. Entre eles estão praticamente 120 mil aspectos únicos, muitos encontrados em apenas uma, ou em um número muito pequeno de avaliações para ser considerado frequente em uma base de dados deste tamanho.

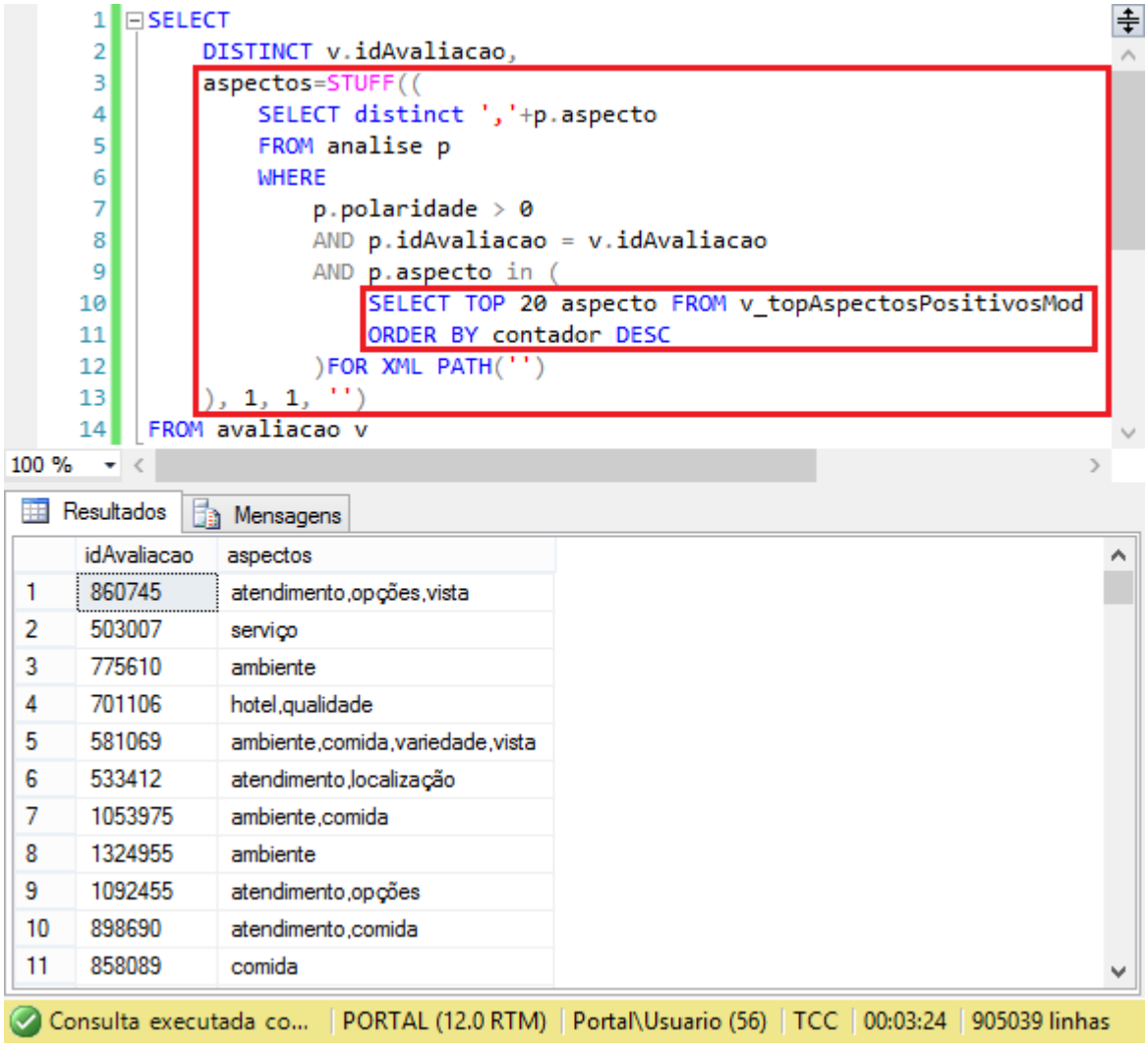
Pela elevada quantidade de dados, a consulta estava levando muito tempo para ser concluída pelo banco de dados. Para otimizar esse tempo, foram criados alguns índices na tabela ‘Analise’ que contem, em cada entrada, um aspecto de uma avaliação e sua polaridade, as principais informações a serem extraídas para a análise.

Para acelerar a recuperação do aspecto, foi indexada a coluna ‘aspecto’, permitindo a otimização da contagem e classificação dos aspectos para utilização dos que forem mais frequentes na análise. Criou-se também um índice associando as colunas ‘aspecto’ e ‘polaridade’ para acelerar a filtragem por polaridade.

Na recuperação dos aspectos de cada avaliação foi necessário criar um índice que associa as colunas ‘idAvaliacao’ e ‘aspecto’ para acelerar a agregação de resultados.

Para diminuir o grau de complexidade da consulta na parte que seleciona apenas os aspectos relevantes à análise, foram criadas duas *views* (visualizações) que tem como propósito agregar e contar previamente as incidências dos aspectos de cada polaridade, definindo assim o *ranking* dos aspectos mais citados e tornando a consulta mais flexível. Isso foi feito tanto para aspectos positivos quanto para negativos. A seguir, a Figura 11 apresenta uma captura de tela do código da consulta e alguns resultados.

Figura 11: Captura de tela da consulta realizada no banco de dados.



A Figura 11 está dividida em duas seções: a parte superior contém a consulta que foi executada e a parte inferior mostra os resultados trazidos do banco de dados. Na consulta, a seção mais externa traz a lista de todas as avaliações para que sejam agregados os aspectos de cada uma. A segunda parte é uma função do SQL Server que agrega todos os resultados dessa sub consulta em uma única linha. Essa sub consulta busca apenas os aspectos positivos da avaliação que está sendo recuperada (v) no nível superior da consulta. A terceira parte da consulta, a mais interna, utiliza uma das *views* criadas (`v_topAspectosPositivosMod`) para trazer apenas os aspectos relevantes para a análise, no caso as 20 mais citadas (`TOP 20`), exceto ‘Local’ e ‘Lugar’, ordenadas por frequência (`contador`) para filtrar o que será retornado. Na parte inferior são exibidos os resultados no formato que serão utilizados

para criar o arquivo de entrada para o algoritmo Apriori do Weka. O processo de transformação desses dados é descrito na próxima seção.

4.2. Conversão dos dados para o formato de arquivo do Weka

A biblioteca Weka trabalha com um formato de entrada chamado ARFF (*Attribute-Relation File Format*). O conteúdo do arquivo é texto plano dividido em duas seções: *Header* (Cabeçalho) e *Data* (dados). A Figura 12 apresenta um trecho de um arquivo ARFF para exemplificar as seções.

Figura 12: Exemplo de estrutura de arquivo ARFF.

```
@RELATION iris

@ATTRIBUTE sepallength  NUMERIC
@ATTRIBUTE sepalwidth   NUMERIC
@ATTRIBUTE petallength  NUMERIC
@ATTRIBUTE petalwidth   NUMERIC
@ATTRIBUTE class        {Iris-setosa,Iris-versicolor,Iris-virginica}


@DATA
5.1,3.5,1.4,0.2,Iris-setosa
4.9,3.0,1.4,0.2,Iris-setosa
4.7,3.2,1.3,0.2,Iris-setosa
4.6,3.1,1.5,0.2,Iris-setosa
5.0,3.6,1.4,0.2,Iris-setosa
5.4,3.9,1.7,0.4,Iris-setosa
4.6,3.4,1.4,0.3,Iris-setosa
5.0,3.4,1.5,0.2,Iris-setosa
4.4,2.9,1.4,0.2,Iris-setosa
4.9,3.1,1.5,0.1,Iris-setosa
```

No início do arquivo apresentado na Figura 12, a seção *Header* contém o nome da relação (`@RELATION`), a lista de atributos (`@ATTRIBUTE`) e os tipos de dados de cada atributo. Os tipos de dados de um atributo podem ser `NUMERIC` (sinônimos: `INTEGER` e `REAL`) para valores numéricos, `STRING` para valores textuais, “`DATE [<formato>]`” para data e hora ou nominal, um conjunto de valores possíveis, como é o caso do atributo ‘class’ no exemplo da figura.

A segunda parte do arquivo contém os dados da relação, a partir da declaração `@DATA`. Cada linha apresenta uma instancia de dados, separados por vírgula, para cada atributo. Os valores textuais, data e hora ou nominal composto por mais de uma

palavra contendo espaço, devem ser isolados entre aspas. Os valores faltantes ou nulos são representados pelo caractere ‘?’.

Para converter os dados retornados pela consulta apresentada na Figura 11 em um arquivo ARFF, o formato recomendado para utilização na ferramenta Weka, foi implementado um código Java que usa a biblioteca do Weka para gerar o arquivo. Abaixo é apresentado o pseudocódigo (Figura 13) para gerar o conjunto de dados a ser utilizado pelo algoritmo.

Figura 13: Pseudocódigo da conversão dos dados.

$A_k = conjunto\ de$ k $aspectos\ mais\ frequentes$
$S_n = conjunto\ de\ strings\ com\ os\ aspectos\ de$ n $avaliações$
$E_n = conjunto\ de\ entradas\ de\ tamanho$ n
$para\ (v = 0; v < n; v + +)$: // itera sobre S_n
    $e = nova\ entrada\ vazia\ para$ k $valores$
    $para\ (p = 0; p < k; p + +)$: // itera sobre A_k
        $se\ A^p \ni S^v$:
            $e^p = 1$
        $senão$:
            $e^p = 0$
        fim
    fim
    $E \leftarrow e$ // insere a entrada no conjunto
fim
$Gera\ o\ arquivo\ com\ a\ lista\ de\ atributos\ A\ e\ de\ entradas\ E$

Para gerar o arquivo a partir dos dados coletados é utilizado o seguinte trecho de código em Java (Figura 14) que utiliza a funcionalidade de salvar arquivo ARFF da biblioteca Weka.

Figura 14: Trecho de código para salvar arquivo em formato ARFF com a biblioteca Weka.

```
Instances dataSet;
dataSet = new Instances("Aspectos-" + polaridade, aspectos, instances);
ArffSaver arffSaverInstance = new ArffSaver();
arffSaverInstance.setInstances(dataSet);
String filename = "Aspectos-" + polaridade + ".arff";
arffSaverInstance.setFile(new File(filename));
arffSaverInstance.writeBatch();
```

No código apresentado na Figura 14, `dataSet` é o conjunto de entradas da relação 'Aspectos-polaridade', com atributos `aspectos`, a lista dos '*k*' aspectos mais frequentes, e dados `instances`, a lista de entradas (E_n) gerada a partir da execução do código.

Devido ao formato de dados implementado na biblioteca Weka, os valores de cada entrada precisam ser numéricos para a geração do arquivo, por isso a utilização do valor 1 para a presença de um aspecto e 0 para a ausência. Mas para a execução ao algoritmo Apriori o valor zero não é considerado ausência, mas um valor qualquer, logo, as entradas com valor '0' são consideradas para a geração das regras de associação, o que implicou em outra alteração no arquivo.

O formato ARFF não suporta valores booleanos para os atributos, mas isso foi resolvido com a utilização de classes, um conjunto de valores possíveis para um atributo, com apenas uma possibilidade. Isso resultou na alteração manual do arquivo, substituindo todas as ocorrências de '0' pelo valor nulo do Weka, o caractere '?' e todos os valores '1' pela possibilidade da classe, no caso, o caractere 't'. Estas modificações fizeram com que o arquivo fosse interpretado corretamente pelo programa e reconhecido pelo algoritmo. Com o arquivo exportado e pronto, foram realizadas as análises descritas na seção seguinte.

4.3. Aplicação do Algoritmo Apriori

Para realizar as próximas análises foram extraídos os 20 aspectos positivos com maior frequência entre as avaliações e os dados foram exportados considerando a presença ou ausência de tal aspecto em cada comentário. A Figura 15 apresenta, em ordem, os aspectos positivos mais frequentes encontrados no banco de dados.

Figura 15: Top 20 aspectos positivos citados nas avaliações.

1	Atendimento	6	Restaurante	11	Praia	16	Pratos
2	Comida	7	Localização	12	Café da manhã	17	Passeio
3	Lugar	8	Preço	13	Vista	18	Funcionários
4	Ambiente	9	Local	14	Qualidade	19	Variedade
5	Hotel	10	Opção	15	Serviço	20	Quartos

As seções a seguir apresentam os resultados das análises feitas sobre os dados obtidos do banco.

4.3.1. Primeiro experimento com o algoritmo Apriori no Weka

A partir dos aspectos apresentados na Tabela 2 foi feita a análise inicial, onde foram selecionadas 10.000 entradas aleatórias, que após remover entradas vazias resultou em 7.468 conjuntos para se extrair as regras de associação.

O algoritmo foi aplicado e encontrou 13 regras, com confiança (frequência da consequência em relação à premissa) maior que 88%. As regras encontradas são exibidas abaixo na Tabela 1.

Tabela 1: Resultado do primeiro experimento.

Premissa	Ocorrência Premissa	Ocorrência Local	Confiança
localização	1027	1027	1
atendimento localização	276	276	1
hotel localização	263	263	1
localização “café da manhã”	255	255	1
localização funcionários	152	152	1
localização quartos	130	130	1
hotel localização “café da manhã”	83	83	1
localização preço	76	76	1
Premissa	**Ocorrência Premissa**	**Ocorrência Localização**	**Confiança**
hotel local “café da manhã”	86	83	0.97
local quartos	135	130	0.96
local “café da manhã”	265	255	0.96
hotel local	280	263	0.94
local funcionários	172	152	0.88

A tabela apresenta os dados da seguinte maneira: A primeira coluna apresenta a Premissa, um conjunto frequente de aspectos. Em seguia a quantidade de ocorrências (suporte) desta premissa. Para resumir a tabela, ela foi separada por consequências, tendo na primeira parte as ocorrências do aspecto Local e na segunda as ocorrências de Localização. A última coluna apresenta a confiança da regra, baseada nas duas colunas anteriores.

Em uma análise preliminar pela especialista do domínio, os aspectos Local e Localização foram muito predominantes, aparecendo em todas as regras extraídas tanto como premissa, quanto como consequência. Além disso, o nível de confiança das consequências foi muito alto para a quantidade de dados analisada. Mas o conjunto de dados foi muito pequeno, portanto pouco representativo para se tirar conclusões. O objetivo desta amostragem foi testar o algoritmo com os dados reais e verificar se extrairia as regras corretamente para, então, se extrair os aspectos de todas as avaliações do banco de dados.

4.3.2. Extração de regras de associação para todas as avaliações

Nesta etapa todas as avaliações que continham pelo menos um dos aspectos da lista foram consideradas, um total de 969.266. Para esta análise assumiu-se uma confiança mínima de 0.8 (80%) baseado nos resultados da análise anterior. Foram encontradas 10 regras de associação, dispostas na Tabela 2 com estrutura semelhante à tabela da seção 4.3.1.

Tabela 2: Resultado da análise de todas as avaliações

Premissa	Ocorrência Premissa	Ocorrência Local	Confiança
Localização	97258	97258	1
atendimento localizacao	28887	28887	1
hotel localização	22035	22035	1
localizacao “cafe da manha”	17510	17510	1
localizacao quartos	10784	10784	1
localizacao funcionarios	10008	10008	1
Localização	97258	97258	1
Premissa	**Ocorrência Premissa**	**Ocorrência Localização**	**Confiança**
local quartos	11791	10784	0.91
hotel local	24141	22035	0.91
local “cafe da manha”	19224	17510	0.91

local funcionarios	11777	10008	0.85

Para um conjunto de dados tão grande, regras com 100% de confiança deveriam ser raras, e o padrão que apresentaram ficou muito próximo ao da análise anterior, com os aspectos local e localização prevalecendo. Isso não era o esperado, considerando que os aspectos com mais ocorrência total são atendimento e comida. A análise dos dados da Tabela 2 não foi satisfatória, tanto pela alta ocorrência dos aspectos local e localização, quanto pelo alto índice de confiança das regras encontradas. A próxima seção apresenta como foi feita a correção dos dados para possibilitar uma análise mais representativa do domínio.

4.3.3. Análise com top aspectos modificados

Para a próxima análise foi decidido pela remoção dos aspectos local e lugar, que eram as expressões mais genéricas segundo análise inicial da especialista do domínio (a orientadora desse trabalho), da lista de top aspectos, mas mantendo a quantidade. A Figura 16 mostra a nova lista de top aspectos positivos após a remoção dos valores redundantes:

Figura 16: Top 20 aspectos positivos citados nas avaliações, exceto local e lugar.

1	Atendimento	6	Localização	11	Vista	16	Funcionários
2	Comida	7	Preço	12	Qualidade	17	Variedade
3	Ambiente	8	Opção	13	Serviço	18	Quartos
4	Hotel	9	Praia	14	Pratos	19	Opções
5	Restaurante	10	Café da manhã	15	Passeio	20	Pizza

Com a nova lista de aspectos, a consulta no banco de dados foi repetida e, após a remoção de valores vazios, retornou 894.310 resultados. Para esta análise, a confiança mínima foi modificada algumas vezes para retornar uma quantidade satisfatória de regras sobre a número elevado de conjuntos analisados, e foi reduzida para 40%. O algoritmo encontrou 82 regras de associação com confiança máxima de 51%.

Segundo análise da especialista do domínio, a orientadora Parcilene Fernandes de Brito, essas relações, ainda que com grau de confiança baixo, já

indicam algumas relações interessantes para análise de produtos turísticos. Por exemplo, a força do aspecto “hotel” para as pessoas que escreveram os comentários. Hotel é um dos três objetos gerais do estudo (Hotel, Restaurante e Atrações) mas é entendido pelos autores dos comentários tanto como um objeto como uma característica (aspecto).

Com os resultados formatados em tabelas com a frequência de ocorrências foi possível fazer análises mais precisas e chegar a conclusões direcionadas sobre as regras de associação.

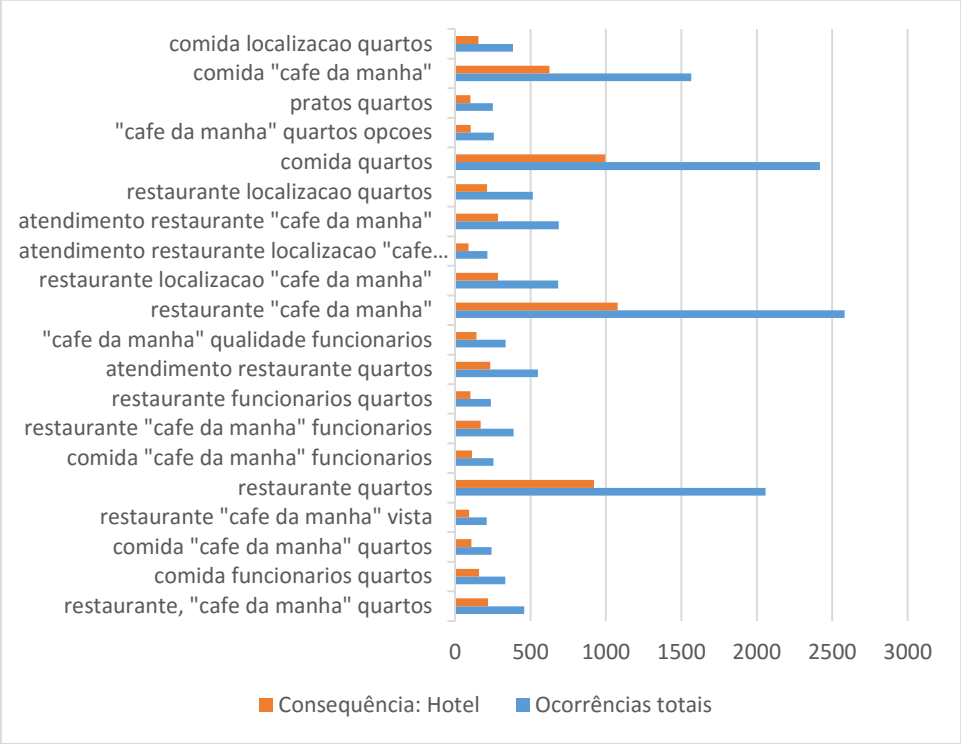
As tabelas a seguir têm os dados dispostos de maneira semelhante às tabelas Tabela 1 e Tabela 2, porém sem a divisão entre consequências. Cada tabela apresenta as premissas e ocorrências de apenas um aspecto, seguidas de um gráfico para melhor analisar as frequências e o índice de confiança.

Tabela 3: Regras de associação com consequência Hotel

Premissa	Ocorrência	Ocorrência Hotel	Confiança
restaurante "cafe da manha" quartos	459	219	0.48
comida funcionarios quartos	334	159	0.48
comida "cafe da manha" quartos	242	109	0.45
restaurante "cafe da manha" vista	210	94	0.45
restaurante quartos	2059	921	0.45
comida "cafe da manha" funcionarios	254	113	0.44
restaurante "cafe da manha" funcionarios	389	169	0.43
restaurante funcionarios quartos	238	102	0.43
atendimento restaurante quartos	549	234	0.43
"cafe da manha" qualidade funcionarios	335	142	0.42
restaurante "cafe da manha"	2583	1078	0.42
restaurante localizacao "cafe da manha"	684	285	0.42
atendimento restaurante localizacao "cafe da manha"	214	89	0.42
atendimento restaurante "cafe da manha"	687	284	0.41
restaurante localizacao quartos	516	213	0.41
comida quartos	2418	996	0.41
"cafe da manha" quartos opcoes	256	104	0.41
pratos quartos	251	101	0.4
comida "cafe da manha"	1565	627	0.4
comida localizacao quartos	385	154	0.4

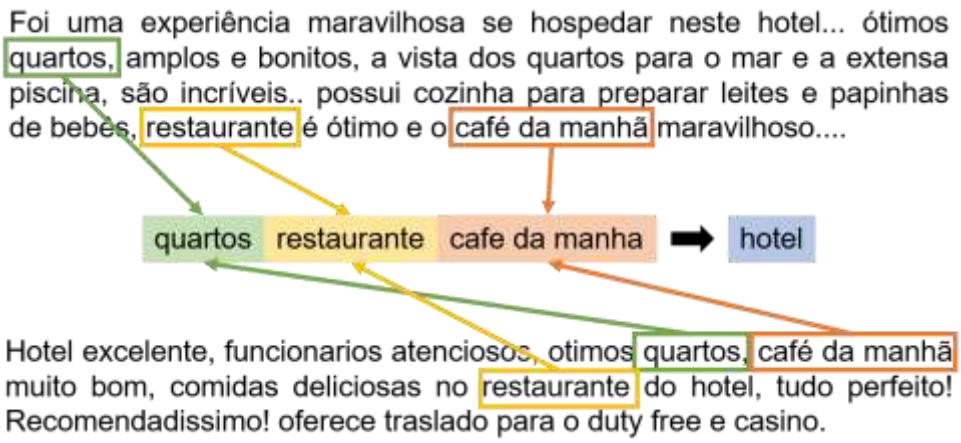
A Tabela 3, acima, apresenta as premissas que tiveram como consequência o aspecto hotel, e as ocorrências foram comparadas na Figura 17: Gráfico de ocorrências de premissas com consequência Hotel.

Figura 17: Gráfico de ocorrências de premissas com consequência Hotel



A Tabela 3 mostra que, quando um usuário escreve um comentário indicando que o hotel é bom, ele pode a elencar características boas do hotel, como café da manhã, quartos e comida, por isso as regras associativas que tem aspectos relacionados a hotel tem como consequência, em grande parte, o hotel. Pode ser extrapolado que, caso mais aspectos relacionados a hotel fizessem parte da lista, eles provavelmente apareceriam como parte das premissas de consequência hotel. Para corroborar com a análise, foram coletados exemplos de comentários que podem ser observados na Figura 18.

Figura 18: Exemplos de comentários referentes a Hotel.



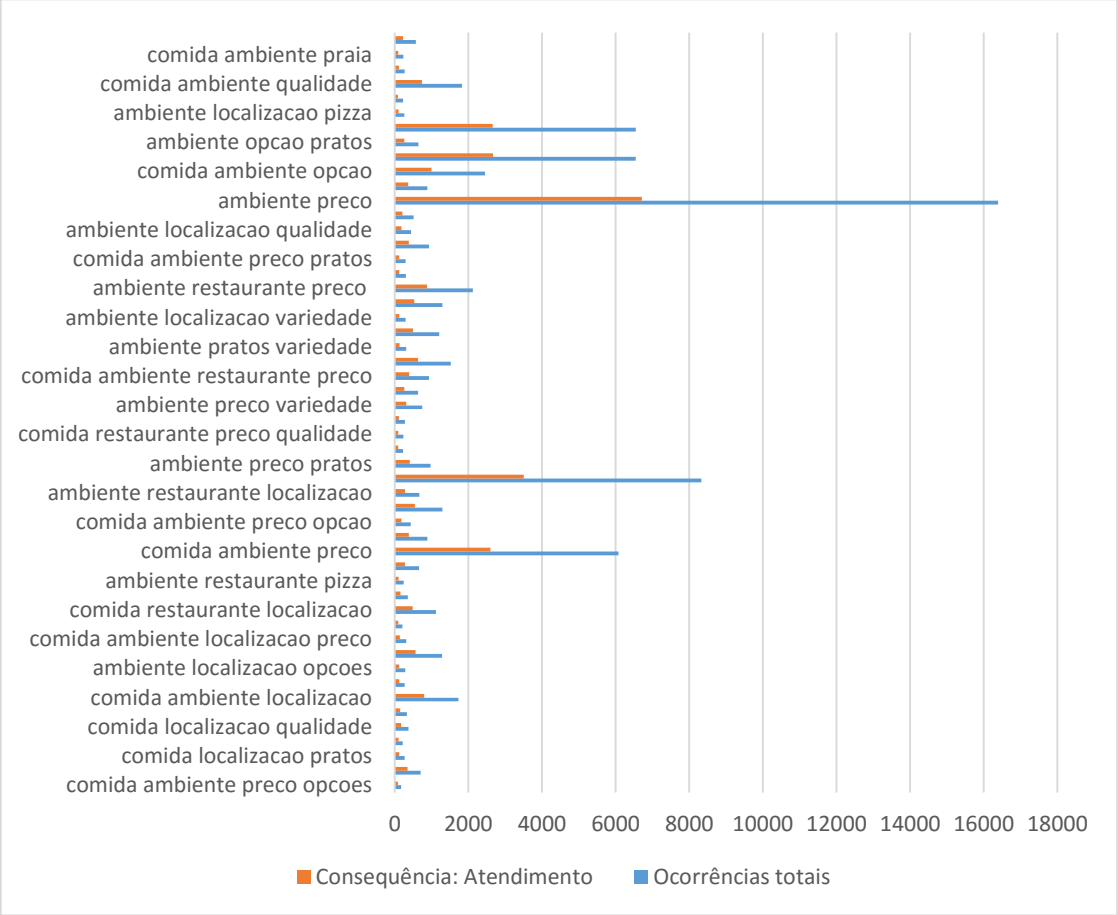
Para facilitar o entendimento da relação, os aspectos foram destacados e ligados à regra de associação. A próxima tabela apresenta as regras de associação com consequência ‘atendimento’.

Tabela 4: Regras de associação com consequência Atendimento

Premissa	Ocorrência Premissa	Ocorrência Atendimento	Confiança
comida ambiente preco opcoes	176	90	0.51
ambiente preco pizza	702	351	0.5
comida localizacao pratos	266	125	0.47
comida ambiente restaurante localizacao	218	101	0.46
comida localizacao qualidade	373	172	0.46
ambiente localizacao pratos	328	151	0.46
comida ambiente localizacao	1732	796	0.46
comida localizacao variedade	272	122	0.45
ambiente localizacao opcoes	286	126	0.44
comida localizacao preco	1287	566	0.44
comida ambiente localizacao preco	316	138	0.44
ambiente variedade pizza	212	92	0.43
comida restaurante localizacao	1123	487	0.43
comida ambiente preco qualidade	360	156	0.43
ambiente restaurante pizza	241	104	0.43
comida preco vista	658	282	0.43
comida ambiente preco	6078	2600	0.43
localizacao pizza	890	380	0.43
comida ambiente preco opcao	434	185	0.43

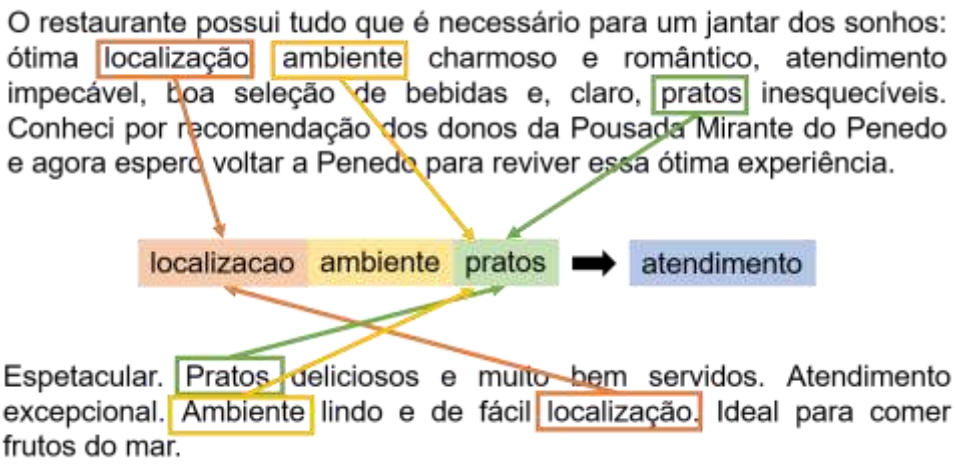
A Figura 19 apresenta o gráfico de ocorrências das premissas que tiveram como consequência o aspecto atendimento, assim como a frequência dessa consequência, de acordo com os dados dispostos na Tabela 4.

Figura 19: Gráfico de ocorrências de premissas com consequência Atendimento



O aspecto mais citado é atendimento, tanto em avaliações positivas (com 827 mil ocorrências enquanto o segundo mais citado tem com 635 mil), como em negativas com (85 mil contra 38 mil) e de acordo com a Tabela 4, esse fato tem muita relação com restaurantes e comida, o que faz sentido se considerarmos que é nos restaurantes que acontece boa parte do contato com funcionários, o que reforça este aspecto no comentário. Assim como na análise anterior, a Figura 20 apresenta exemplos contextualizados de comentários relacionados a esta análise.

Figura 20: Exemplo de comentários referentes a Atendimento.



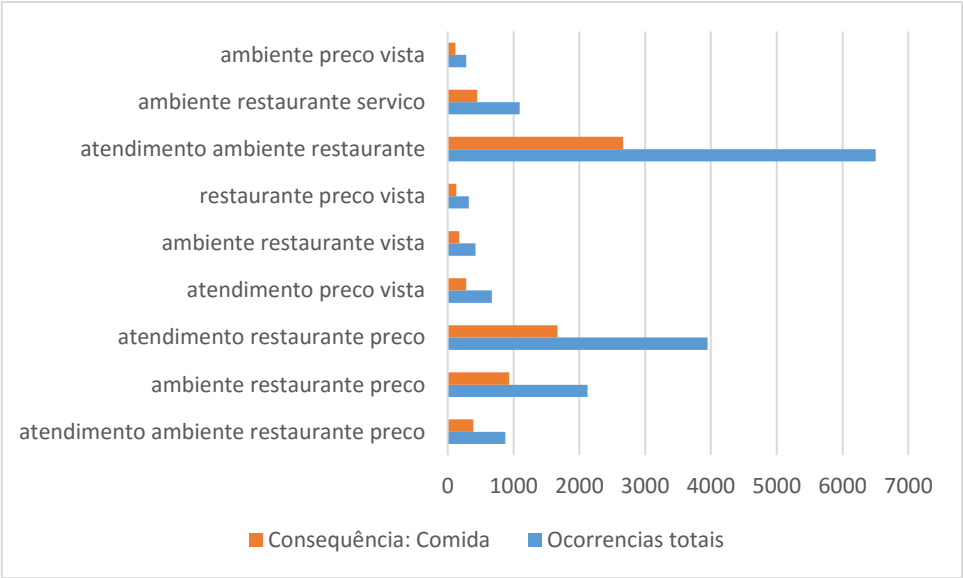
Para uma última análise, a Tabela 5 mostra o conjunto de regras de associação que tiveram como consequência o aspecto Comida.

Tabela 5: Regras de associação com consequência Comida

Premissa	Ocorrência Premissa	Ocorrência Comida	Confiança
ambiente restaurante preco	2124	932	0.44
atendimento restaurante preco	3947	1669	0.42
atendimento preco vista	669	282	0.42
ambiente restaurante vista	424	176	0.42
restaurante preco vista	319	131	0.41
atendimento ambiente restaurante	6504	2663	0.41
ambiente restaurante servico	1092	446	0.41
ambiente preco vista	283	115	0.41

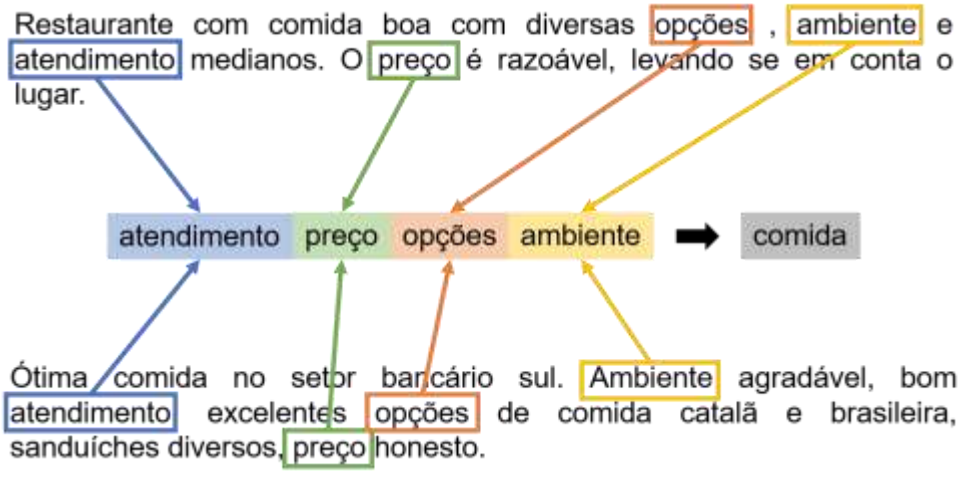
Assim como para as tabelas anteriores, o gráfico apresentado na Figura 21 apresenta a relação entre ocorrências de premissas e de consequência para regras encontradas que em como consequência o aspecto comida, baseado na Tabela 5.

Figura 21: Gráfico de ocorrências de premissas com consequência Comida.



Um ponto em que se considerou comida como aspecto consequência positivo na Tabela 5 foi o aspecto preço, presente em grande parte das premissas. Para os usuários, não basta a comida ter boa qualidade, ela deve ser compatível com o poder aquisitivo da pessoa que está visitando este destino turístico. Exemplos de comentários relacionados são exibidos na Figura 22.

Figura 22: Exemplo de comentários referentes a Comida.



Estas foram as análises sobre as regras de associação obtidas dos 20 aspectos positivos mais frequentes encontrados no banco de dados de avaliações. A análise de regras de associação pode fornecer novas introspecções relacionadas ao domínio. Se for desejado, é possível selecionar outros conjuntos de aspectos para geração de novas regras de associação com intuito de se obter um entendimento mais completo acerca do objeto, para que seja possível chegar a conclusões mais direcionadas.

5 CONSIDERAÇÕES FINAIS

O crescimento da colaboração na internet tem gerado um grande volume de dados. A maioria destes dados está na forma de texto não estruturado, linguagem natural, e grande parte expressa opiniões sobre diversos produtos e serviços. A extração de informação desses textos se faz por meio da utilização de técnicas de *Text Mining* para separar, processar e analisar esses dados.

Este trabalho utilizou um banco de dados de comentários de um site de turismo, que já tinham sido extraídos e pré-processados. O banco de dados também já tinha sido analisado em um processo de Análise de Sentimentos em nível de aspectos.

A organização das informações do domínio para uma análise de associação foi feita criando uma consulta no banco de dados. Esta consulta exigiu a criação de visualizações (*views*) para diminuir a sua complexidade, e novos índices nas tabelas para agilizar a recuperação dos dados. Estes dados passaram, posteriormente, por um processamento para organizá-los em um arquivo no formato ARFF, que serviu de entrada para a análise. A criação do arquivo de entrada, porém, implicou em dificuldades por conta da restrição ao tipo de estrutura de dados utilizado pela biblioteca do Weka. Os arquivos gerados ainda precisaram passar por substituições manuais de caracteres. Os dados retornados também precisaram ser calibrados pois, após o teste inicial detectou-se que alguns aspectos redundantes acabavam sendo predominantes, atrapalhando a análise dos dados.

Para extrair as regras de associação dos aspectos foi utilizado o algoritmo Apriori, que testa cada conjunto de aspectos para determinar sua frequência e separar a premissa e a consequência dos conjuntos mais frequentes. O algoritmo foi executado através da interface gráfica da ferramenta Weka, escolhida pela facilidade de alterar parâmetros da análise, como representatividade mínima de dados e grau mínimo de confiança para a criação de uma regra de associação.

Os dados de saída do algoritmo foram transformados em tabelas analisados pela especialista do domínio, a orientadora Parcilene Fernandes de Brito, que chegou a conclusões direcionadas sobre as regras de associação e como as premissas se relacionam às consequências das regras de associação encontradas.

As análises realizadas foram satisfatórias em relação ao domínio, e trabalhos futuros poderão inferir mais relações trabalhando com mais aspectos ou utilizando uma lista discreta de aspectos direcionada a um objeto geral de estudo (Hotel, Restaurante ou Atração).

Como mencionado em Christhie (2015), apesar de os dados coletados serem referentes ao turismo, os mesmos procedimentos podem ser aplicados a outras áreas, com os devidos ajustes de parâmetros, resultando em regras de associação de similar importância.

6 REFERÊNCIAS BIBLIOGRÁFICAS

ARANHA, C.; PASSOS, E. A tecnologia de mineração de textos. Revista Eletrônica de Sistemas de Informação ISSN 1677-3071 doi: 10.5329/RESI, v. 5, n. 2, 2006.

ARANHA, C. N. Uma abordagem de pré-processamento automático para mineração de textos em português: sob o enfoque da inteligência computacional, Tese de Doutorado, Departamento de Engenharia Elétrica, PUC-Rio. 2007. 144 p.

AGRAWAL, R; IMIELIŃSKI, T; SWAMI, A. Mining association rules between sets of items in large databases. In: ACM SIGMOD Record. ACM, 1993. p. 207-216.

BRITO, P. F.; CHRISTHIE, W.; SOUZA, J. G.; SILVA, E.M. (2015). SentimentALL. Fábrica de Software: CEULP/ULBRA.

CHEN, H. Knowledge management systems: a text mining perspective. University of Arizona (Knowledge Computing Corporation), Tucson, Arizona. 2001. 50 p.

CHRISTHIE, W. SentimentALL: Módulo para Análise de Sentimentos em Português. 86 f. Trabalho de Conclusão de Curso (Bacharelado em Sistemas de Informação) – Centro Universitário Luterano de Palmas, Palmas, 2015. [Orientadora: Parcilene Fernandes de Brito].

DÖRRE, J; GERSTL, P; SEIFFERT, R. Text mining: finding nuggets in mountains of textual data. In: Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining. ACM, 1999. p. 398-401.

FELDMAN, R.; SANGER, J. The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data. Cambridge University Press, 2007. 410 p.

GUPTA, V; LEHAL, G. S. A survey of text mining techniques and applications. Journal of emerging technologies in web intelligence, v. 1, n. 1, p. 60-76, 2009.

HOLZINGER, A., STOCKER, C., OFNER, B., PROHASKA, G., BRABENETZ, A., & HOFMANN-WELLENHOF, R. Combining HCI, Natural Language Processing, and Knowledge Discovery-Potential of IBM Content Analytics as an assistive technology in the biomedical field. In: Human-Computer Interaction and Knowledge Discovery in Complex, Unstructured, Big Data. Springer Berlin Heidelberg, 2013. p. 13-24.

KRISHNA, S. Murali; BHAVANI, S. Durga. An efficient approach for text clustering based on frequent itemsets. European Journal of Scientific Research, v. 42, n. 3, p. 399-410, 2010.

MANNING C. D.; RAGHAVAN, P.; SCHÜTZE, H. Introduction to Information Retrieval. Cambridge University Press, 2009. Disponível em <http://nlp.stanford.edu/IR-book>

ORENGO, V. M; HUYCK, C. A stemming algorithm for the Portuguese language. In: EIGHTH INTERNATIONAL SYMPOSIUM ON STRING PROCESSING AND INFORMATION RETRIEVAL (SPIRE 2001), p. 186–193, 2001.

SOARES, F. A Mineração de Textos na Coleta Inteligente de Dados na Web. Dissertação (Mestrado em Engenharia Elétrica), PUC-Rio, 2008. 120 p.

WIVES, L. K. Tecnologias de descoberta de conhecimento em textos aplicadas à inteligência competitiva. Exame de Qualificação EQ-069, PPGC-UFRGS, 2002.

YANG, Y., AKERS, L., KLOSE, T., YANG, C. B. Text mining and visualization tools–impressions of emerging capabilities. World Patent Information, v. 30, n. 4, p. 280-293, 2008.