3
Metodologia de Mineração de Textos

Neste capítulo são analisadas e discutidas as etapas de uma metodologia para Mineração de Textos. Embora Mineração de Textos possa ser empregada para a realização de diversas tarefas, como por exemplo, para a classificação automática de textos (KUDO & MATSUMOTO, 2004), segundo (ARANHA C. N., 2007), todo processo de Mineração de Textos consiste de cinco etapas, encadeadas nesta ordem: coleta de documentos, pré-processamento, indexação, mineração e análise. Na Figura 10 são exibidos o encadeamento destas etapas e principais atividades realizadas em cada uma delas.

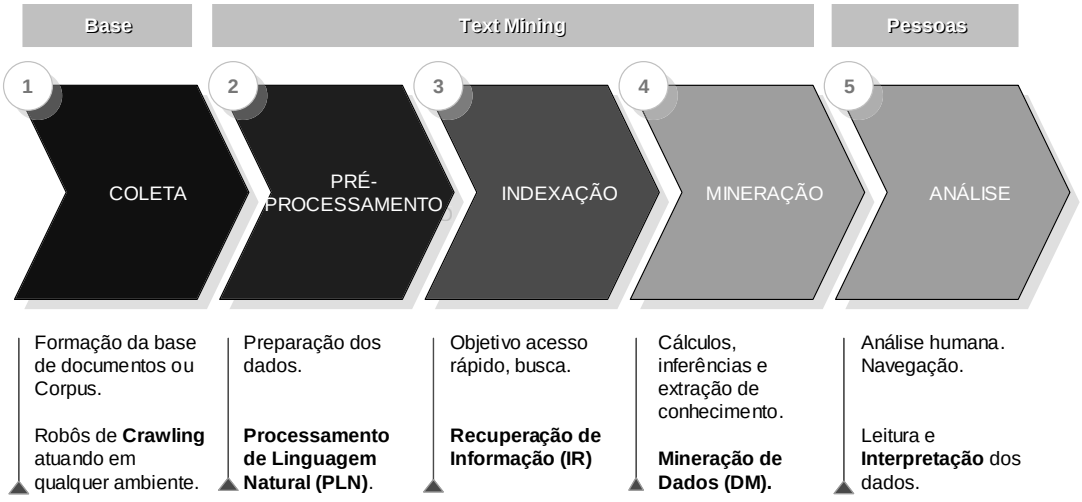

Figura 10 – Linhas cronológica das etapas de um processo de Mineração de Textos (por Aranha)

A primeira etapa a ser realizada é a de Coleta, cujo objetivo é a formação da coleção de documentos, elemento básico de qualquer processo de Mineração de Textos.

Em seguida, inicia-se a etapa de Pré-processamento. É neste momento que os documentos, obtidos na fase anterior, são submetidos a inúmeras operações capazes de obter uma forma de representá-los estruturadamente.

Após o Pré-Processamento, inicia-se a fase de Indexação. Indexação é o processo responsável pela criação de estruturas auxiliares denominadas índices e que garantem rapidez e agilidade na recuperação dos documentos e seus termos.

Uma vez indexados, documentos e termos são submetidos a algoritmos de Aprendizado de Máquina e de Estatística para que seja realizada a extração de conhecimento dos mesmos. A extração de conhecimento tem a finalidade de descobrir padrões úteis e desconhecidos presentes nos documentos.

Finalizando o processo de Mineração de Textos, há a etapa de Análise. Na etapa de Análise é realizada a avaliação e interpretação de todo o conhecimento obtido pelo processo.

3.1. Coleta de Dados

A primeira etapa de um processo de Descoberta de Conhecimento em Textos é a Coleta de Dados (SHOLOM, INDURKHYA, ZHANG, & DAMERAU, 2005) (KONCHADY, 2006) (ARANHA C. N., 2007) (FELDMAN & SANGER, 2007). Esta etapa envolve a seleção dos textos que irão compor a Coleção de Documentos, elemento básico de qualquer processo de Mineração de Textos. É interessante ressaltar que documentos devem ser relevantes ao domínio da aplicação do conhecimento a ser extraído, pois a seleção de documentos irrelevantes para fazer parte da Coleção de Documentos pode prejudicar o processo de Mineração de Textos, além de aumentar a dimensionalidade dos dados desnecessariamente.

Quanto à origem, documentos podem ser obtidos das mais diversas fontes, mas, em geral, são três os principais ambientes de localização dos mesmos: pastas de arquivos encontradas no disco rígido de usuários, tabelas de diversos bancos de dados e a *Web*.

Na *Web*, a coleta de dados pode ser realizada de forma automatizada através de *crawlers* (HEATON, 2002). Um *crawler* é um robô que visita todo e qualquer documento *Web* disponível e repassa as informações coletadas para outro componente responsável pela indexação desses documentos. Atualmente, visando paralelismo e escalabilidade, a arquitetura mais moderna de varredura na *Web*

utiliza vários desses robôs de forma distribuída trabalhando de maneira cooperativa (WEN, 2006).

(SOARES, 2008) propõe uma metodologia de coleta inteligente de dados na *Web* baseada em técnicas de Mineração Textos para a construção de um *crawler* focado. Um *crawler* focado é altamente efetivo na construção de coleções de documentos de qualidade sobre tópicos específicos e oriundos da *web*, usando modestos computadores "caseiros" (DOM, CHAKRABARTI, & BERG, 1999).

3.2. Pré-Processamento

Sistemas de Mineração de Textos não submetem aos seus algoritmos de descoberta de conhecimento coleções de textos despreparadas (GOMES, 2008). Uma vez realizada a Coleta de Dados, o próximo passo é a preparação dos textos para que os mesmos possam ser manipulados pelos algoritmos de Mineração de Textos. Esta segunda etapa denomina-se Pré-Processamento e é responsável por criar uma representação do texto mais estruturada, capaz de alimentar algoritmos de Máquinas de Aprendizado (GONÇALVES, SILVA, QUARESMA, & VIEIRA, 2006), muitos destes também utilizados em Mineração de Dados.

É na etapa de Pré-processamento que é criado o modelo de representação dos documentos, ou seja, a transformação de textos em dados estruturados. Existem diversos modelos para representação estruturada de documentos textuais na literatura de RI, entretanto, o mais utilizado em Mineração de Textos é o **Modelo de Espaço Vetorial**[10] (SILVA A. A., 2007), e será visto em detalhes no item 4.3.1.2.

Uma vez criado o modelo de representação dos textos, é necessário que este seja computacionalmente tratável, e para isto são realizadas algumas operações de Análise de Dados que visam selecionar somente as características que melhor expressam o conteúdo dos textos. Este processo é ilustrado na Figura 11.

[10] Do termo inglês, *Vector Space Model*.

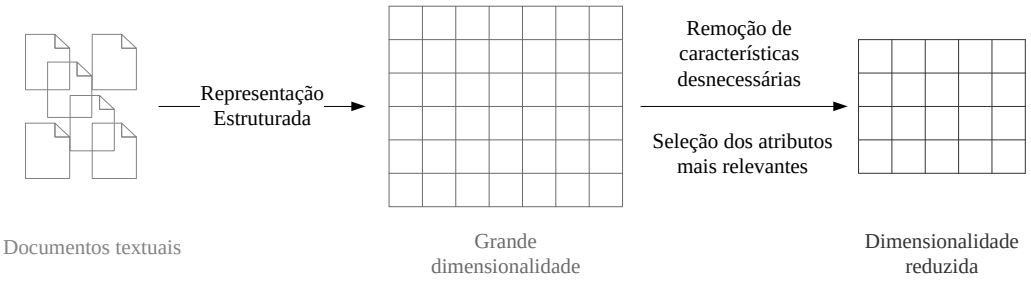

Figura 11 – Processo de representação estruturada de um texto

Pré-processar textos é, por muitas vezes, o processo mais oneroso da metodologia de Mineração de Textos, uma vez que não existe uma única técnica que possa ser aplicada para a obtenção de uma representação satisfatória em todos os domínios, sendo necessária a realização de muitos experimentos empíricos para se chegar à representação adequada (CARRILHO, 2007).

3.2.1. Tokenização

O primeiro passo de uma operação de Pré-processamento é a **tokenização**[11] ou **atomização**[12] e sua execução tem como finalidade seccionar um documento textual em unidades mínimas, mas que exprimam a mesma semântica original do texto. O termo *token* é utilizado para designar estas unidades, que geralmente correspondem a somente uma palavra do texto, porém há casos em que estas unidades textuais não podem ser consideradas palavras ou apresentam mais de uma palavra: “21/10/2007”, “PM”, “R$100,00” e “couve-flor”.

O processo de tokenização é auxiliado pelo fato das palavras serem separadas por caracteres de controle de arquivo ou de formatação, tais como espaços ou sinais de pontuação, que em alguns casos podem ser considerados *tokens* delimitadores (FELDMAN & SANGER, 2007). A criação de *tokens* de um texto baseada em seus delimitadores é uma estratégia simples e que apresenta

[11] Do inglês, *tokenization*.

[12] Alguns autores de língua portuguesa utilizam o termo atomização para fazer referência à tarefa de tokenização (FINATTO, 2005) (LINGUATECA, 2007).

bons resultados. Entretanto, a tarefa de identificação de *tokens*, que é relativamente simples para o ser humano, pode ser bastante complexa de ser executada por um computador. Este fato, segundo (CARRILHO, 2007), é atribuído ao grande número de papéis que os delimitadores podem assumir. Por exemplo, o "ponto" pode ser usado para marcar o fim de uma sentença, mas também é usado em abreviações e números ("A Av. Brasil possui 58 km de extensão.").

Em processos de Mineração de Textos que são assistidos por um dicionário de dados, este pode ser utilizado a fim de verificar as sequencias de caracteres que compõem um termo e validar sua existência, bem como corrigir possíveis erros ortográficos.

Um algoritmo muito utilizado para verificar a corretude de um termo é o de Distância de Edição (FONSECA & REIS, 2002), pois informa quantas operações (deleção, substituição ou inserção de caracteres) são necessárias para que um termo seja transformado em outro. O exemplo abaixo exibe os passos necessários para transformar o termo "casas" em "massa", definindo a distância de edição em três:

1. casas ^ masas — substituição de 'c' por 'm'
2. masas ^ mass — eliminação de 'a'
3. mass ^ massa — inserção de 'a'

A Figura 12 ilustra a metodologia proposta em (KONCHADY, 2006) para a identificação de *tokens*, que, com o uso de dicionários de dados e regras de formação de palavras, procura manter o mesmo nível semântico apresentado pelos *tokens* de um texto antes do processo de tokenização.

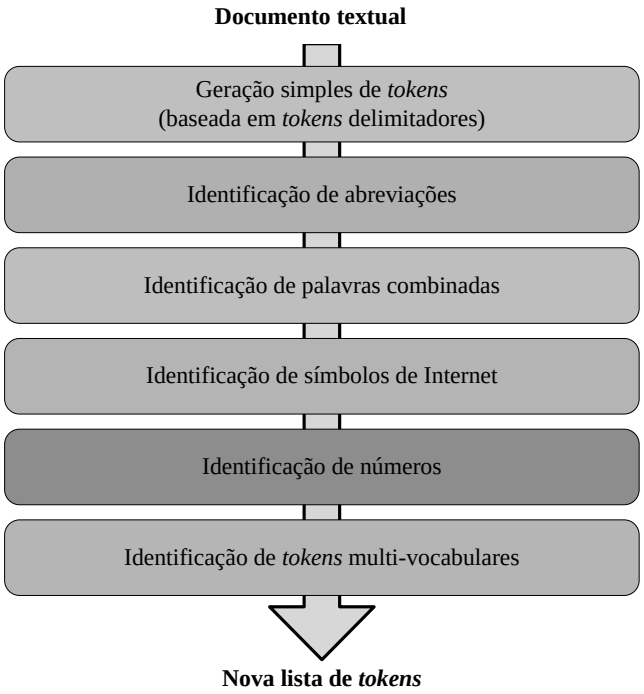

Figura 12 - Metodologia de identificação de tokens proposta por KONCHADY

3.2.2. Remoção de *stopwords*

Buscando sempre tornar possível o processamento computacional de textos, uma vez realizado o processo de tokenização, o passo seguinte é a identificação do que pode ser desconsiderado nos passos posteriores do processamento dos dados. É a tentativa de retirar tudo que não constitui conhecimento nos textos.

Em um documento, existem muitos *tokens* que possuem pouco valor semântico, sendo úteis apenas para o entendimento e compreensão geral do texto. Estes *tokens* são palavras classificadas como ***stopwords***[13]e fazem parte do que é chamado de *stoplist* de um sistema de Mineração de Textos (BASTOS, 2006).

Geralmente, fazem parte de uma *stoplist* termos como conjunções, preposições, pronomes e artigos, pois são considerados termos de menor relevância, ou seja, sua presença pouco contribuiu para a determinação do valor semântico de um documento. Uma *stoplist* bem elaborada permite a eliminação de

[13] Em "http://linguateca.di.uminho.pt/Paulo/stopwords/" há uma lista de trezentas *stopwords* da Língua Portuguesa.

muitos termos irrelevantes, tornando mais eficiente o resultado obtido pelo processo de Mineração de Textos. Normalmente, 40 a 50% do total de palavras de um texto são removidas com uma *stoplist* (SILVA A. A., 2007). A Figura 13 ilustra o exemplo de um processo de tokenização seguido por outro de remoção de *stopwords*.

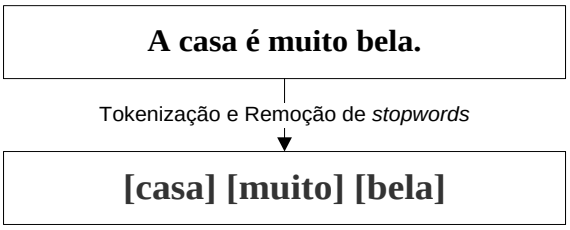

Figura 13 - Processo de tokenização seguido por remoção de stopwords

3.2.3. Processamento de Linguagem Natural

O uso de técnicas de Processamento de Linguagem Natural em Mineração de Textos tem o objetivo de identificar a real importância de cada termo em determinados contextos, possibilitando um ganho na qualidade dos resultados produzidos. O PLN é utilizado para agregar valores semânticos que poderão beneficiar o processo de descoberta de conhecimento em etapas posteriores. Embora muitas abordagens ao processo de Mineração de Textos não façam uso de PLN, a sua utilização tem incrementado os resultados obtidos e justificado o esforço computacional adicional, como em (ARANHA C. N., 2007).

3.2.3.1. Identificação de Colações

A ordem ou disposição dos vocábulos em uma sentença pode incrementar ou, até mesmo, alterar totalmente o significado de alguns termos. Palavras que expressam essa relação são conhecidas como colocações. Certas colocações têm a mesma forma que termos complexos, como por exemplo, “proteção ambiental” e “proteção do meio ambiente”. Em outras, a ordem dos elementos componentes de sua formação pode

variar, como por exemplo, “grande amigo” e “amigo grande”. Muitos são os tipos e possibilidades de uma colação e é interessante que ao criar um *token*, este seja composto por todo este conjunto de palavras que traduz uma ideia diferente.

3.2.3.2. Identificação de Classes Gramaticais

Entende-se por classe gramatical como a forma de classificação de um termo segundo seu significado e função (CEGALLA, 2005). Na língua Portuguesa há dez classes gramaticais, sendo que seis são variáveis (substantivo, artigo, adjetivo, numeral, pronome e verbo) e quatro, invariáveis (advérbio, preposição, conjunção e interjeição). A identificação de classes de palavras ou **etiquetagem de classes gramaticais**[14] presentes em uma sentença facilita o entendimento desta e muitas vezes soluciona alguns problemas simples de ambiguidade.

Cadeias de Markov Escondidas (HARPER & THEDE, 1999) e **TBL**[15] (BRILL, 1995) têm sido utilizados com sucesso em tarefas de identificação de classes gramaticais. Outro método simples para a execução desta tarefa é a simples consulta a dicionário de dados, porém, este método não é dotado de nenhuma heurística que garanta a identificação correta de uma classe gramatical quando uma mesma palavra pode assumir mais de uma classe gramatical.

3.2.3.3. Análise de Discurso

Análise de Discurso, também conhecida por Resolução de Referências (RUSSELL & NORVIG, 2004), é a interpretação de um pronome ou de um sintagma nominal que se refere a um objeto presente no texto. Em linguística, a referência a algo que já foi apresentado é chamado de anáfora. Descobrir anáforas em um texto é o principal objetivo da tarefa de Análise de Discurso e exige conhecimentos sobre o contexto e partes anteriores do texto.

[14] Do termo inglês, *part of speech tagging.*

[15] Acrônimo do termo inglês, *Transformation Based Learner*.

Contexto é a situação histórico-social de um texto, envolvendo não somente as instituições humanas, como ainda outros textos que sejam produzidos em volta e que se relacionem. Todo contexto envolve elementos tanto da realidade do autor quanto do receptor e a análise destes elementos ajuda a determinar o seu sentido (FOUCAULT, 2002). No exemplo da Figura 14, a anáfora só pode ser bem definida quando levado em consideração o contexto.

Figura 14 - Reconhecimento de anáfora com informações do contexto

Para entender que "ele" na segunda sentença faz referência a Luís, é necessário identificar que a primeira sentença menciona duas pessoas e que Luís é quem representa o papel de cliente, logo, é provável que ele faça um pedido em vez do frentista.

3.2.3.4. Lematização

Qualquer documento textual apresenta muitas palavras flexionadas nas mais diversas formas. Na língua Portuguesa, um substantivo pode ser flexionado em gênero, número e grau, e apresentar o mesmo valor semântico. O processo de formação de palavras é, na maior parte das vezes, realizado pela derivação de radicais, resultando na criação de palavras que também exprimem o mesmo significado (CEGALLA, 2005).

Lematização ou *stemming* é o processo de reduzir ao radical original palavras derivadas ou flexionadas deste. O principal objetivo da utilização de um processo de *stemming* é reduzir a grande dimensionalidade das aplicações de Mineração de Textos, pois, com a remoção de prefixos e sufixos de palavras derivadas de um mesmo radical, e que, antes, seriam consideradas como *tokens* distintos, obtém-se um único *token* para a representação de todas elas.

Embora utilizem técnicas de linguística, o que os torna dependentes do idioma, algoritmos de *stemming* não buscam chegar às regras básicas da

linguística do idioma ao radicalizar uma palavra, mas sim, melhorar o desempenho das aplicações, o que pode resultar em tipos de erros que devem ser observados e controlados durante a execução do *stemming*:

- *Overstemming*: ocorre quando o conjunto de caracteres removidos de uma palavra não faz parte de uma derivação ou flexão desta, mas, sim, de seu radical. A ocorrência deste erro pode fazer com que se obtenha um mesmo radical para palavras distintas.
- *Understemming*: ocorre quando os caracteres resultantes do processo de *stemming* ainda fazem parte de uma derivação ou flexão da palavra original. A ocorrência deste erro pode fazer com que sejam obtidos radicais distintos para palavras de mesma origem.

No exemplo da Figura 15, temos o exemplo destes dois tipos de erros comuns no processo de *stemming*. O algoritmo de *stemming* utilizado, neste exemplo, foi o de Porter (PORTER, 1980).

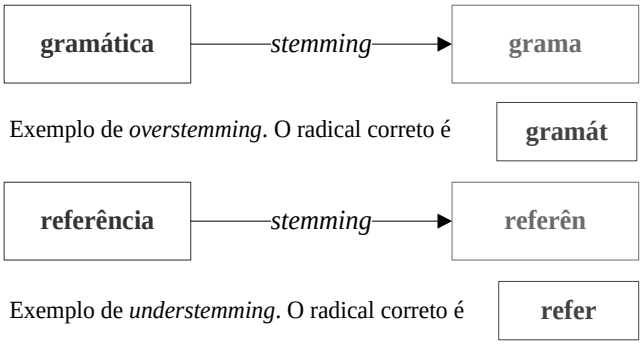

Figura 15 – Erros de um processo de stemming: overstemming e understemming

O algoritmo de *stemming* mais utilizado na Língua Portuguesa é o de Porter (PORTER, 1980). Além deste, há outros algoritmos de *stemming* disponíveis na literatura e que são apresentados a seguir. Embora a maior parte destes algoritmos tenha sido desenvolvida para o idioma inglês, é frequente encontrar adaptações de alguns deles para os mais diversos idiomas:

- Método do Stemmer S: Constitui um dos mais simples métodos de *stemming*. Apenas uns poucos finais de palavras (“ies”, “es" e “s”)

são removidos, com exceções. Embora, simples, este algoritmo é utilizado em razão do seu elevado nível de conservadorismo.

- Método de Porter: O funcionamento do algoritmo de *stemming* de Porter (PORTER, 1980) consiste na identificação e substituição das diversas inflexões e derivações de uma mesma palavra por um mesmo radical. Como, em geral, termos que derivam de um mesmo radical possuem significados semelhantes, consegue-se reunir em um único *token* a importância de todas as suas derivações, como no exemplo da Figura 16. Este algoritmo remove cerca de sessenta sufixos diferentes.

Figura 16 – Derivações de um mesmo radical identificadas pelo algoritmo de Porter

- Método de Lovins: O algoritmo de *stemming* desenvolvido por Lovins (LOVINS, 1968) é capaz de remover cerca de duzentos e cinquenta sufixos diferentes em um único passo. Sensível ao contexto, este algoritmo remove no máximo um sufixo por palavra, geralmente, o mais longo. Embora vários sufixos não sejam removidos por este método, este é o mais agressivo dos três algoritmos de *stemming* apresentados.

3.2.3.5. Reconhecimento de Entidades Nomeadas

Segundo (SUTTON & MCCALLUM, 2006)**,** o **Reconhecimento de Entidades Nomeadas**[16] é o problema de identificar e classificar nomes próprios em textos, incluindo localizações, tais como Brasil e Rio de Janeiro; pessoas, tais com Dilma e Miriam; organizações, tais como Ministério da Educação e Ministério da Cultura; tempo, tais como uma data ou um período de duração; além de outras entidades.

No exemplo da Tabela 5, a sentença contém quatro entidades diferentes: Mário como pessoa, trezentos como número (quantidade), Petrobras como organização e 2006 como tempo (data). A saída marcada com *tags* foi realizada pelo software ENAMEX que foi desenvolvido para a "Message Understanding Conference" na década de noventa.

Tabela 5 - Marcação de *tags* para Reconhecimento de Entidades Nomeadas

Entrada	Mário comprou 300 ações da Petrobras em 2006.
Saída	<ENAMEX TYPE="PERSON">Mário</ENAMEX> comprou <NUMEX TYPE="QUANTITY">300</NUMEX> ações da <ENAMEX TYPE="ORGANIZATION">Petrobras</ENAMEX> em <TIMEX TYPE="DATE">2006</TIMEX>.

O grande desafio deste problema é que muitas entidades nomeadas, mesmo em grandes conjuntos de treinamento, possuem pouca frequência; portanto, o sistema deve identificar tais entidades utilizando apenas o contexto em que esta é empregada. Outra peculiaridade é que a mesma entidade pode apresentar classificações diferentes dependendo do contexto em que se encontra.

A Tabela 6 exibe um exemplo em que, na primeira sentença a entidade Brasil representa um local, enquanto na segunda sentença, essa entidade representa uma organização (Governo Brasileiro).

[16] Do termo inglês, Named Entity Recognition (NER)

Tabela 6 - Exemplo de classificações distintas de uma mesma entidade

Sentença 1	Visitarei o *Brasil* no próximo ano.
Sentença 2	A proposta apresentada pelo *Brasil* na ONU...

Inicialmente os algoritmos de NER eram baseados em regras escritas manualmente que indicavam a existência de uma entidade em determinado contexto. Atualmente, aprendizado supervisionado tem sido a técnica predominante na tarefa de reconhecimento de entidades nomeadas (NADEAU; SEKINE, 2007).

3.2.3.6. Análise Sintática

A análise sintática tem como objetivo examinar a estrutura de um período e das orações que compõem esse período (NEVES, 2012). O fato a ser considerado é a impossibilidade, em certas sentenças, de se obter sentido sem o emprego de funções gramaticais.

Na análise sintática, cada palavra ou grupo de palavras da oração é chamado de termo da oração. Um termo é classificado de acordo com a função sintática que exerce na oração.

De acordo com a Nomenclatura Gramatical Brasileira, os termos da oração podem ser:

1. Essenciais: Também chamados de fundamentais: Sujeito e Predicado.
2. Integrantes: Completam o sentido dos verbos e dos nomes:
   a. Complemento Verbal - Objeto Direto e Objeto Indireto
   b. Complemento Nominal
   c. Agente da Passiva
3. Acessórios: Desempenham função secundária (especificam o substantivo ou expressam circunstância):
   a. Adjunto Adnominal
   b. Adjunto Adverbial

c. Aposto

No exemplo abaixo, por meio da análise sintática pode-se compreender que a oração possui um predicado nominal (o verbo estar denota estado, logo é um verbo de ligação) sobre cujo sujeito simples (a manhã) é revelada uma característica (ensolarada) por meio do predicativo do sujeito (revela uma característica sobre o mesmo), pois se tem um predicado nominal.

Sentença	A manhã está ensolarada.	
Análise	Sujeito simples	a **manhã (núcleo)**
	Predicado nominal	está ensolarada
	Predicativo do sujeito	ensolarada

De maneira geral, a análise sintática, também conhecida como *parsing* consiste da utilização de dois componentes principais. Primeiramente, uma gramática contendo os fatos sintáticos da linguagem utilizada é exigida. Esta gramática servirá como base de atuação do segundo componente do processamento sintático: o analisador. Também conhecido como *parser*, o analisador compara as formalizações descritas na gramática com a sentença de entrada. O resultado é a geração da estrutura hierárquica contendo as unidades de significado da sentença.

3.3. Indexação

Após a etapa de Pré-Processamento, independente da utilização de Processamento de Linguagem Natural, documentos textuais, antes fracamente estruturados, agora possuem representação estruturada, esta, baseada em um dos diversos modelos de representação disponíveis (ver item 4.3.1). Entretanto, para que uma simples consulta seja realizada é necessário percorrer toda a coleção de documentos, analisando documento a documento, o que demanda tempo e esforço computacional.

Indexação é fase responsável por criar estruturas de dados denominadas índices, capazes de permitir que uma consulta seja realizada sem que seja

necessário analisar toda uma base de dados (MANNING, RAGHAVAN, & SCHÜTZE, 2007). Técnicas de indexação de documentos foram bastante difundidas pela demanda e crescimento da área de Recuperação de Informação desde a década de sessenta. Contudo, muitas pessoas acreditam que esta é uma área nova. Esta ideia talvez tenha surgido com a grande popularização das máquinas de buscas que tornaram possível a pesquisa do conteúdo de páginas *web*, ou seja, documentos textuais. No entanto, segundo (BAEZA-YATES & BERTIER, 1999), há aproximadamente quatro mil anos já são praticadas técnicas de catalogação manual por índices.

Semelhantes ao sumário de um livro que é uma lista detalhada, com a indicação de localização no texto, dos principais tópicos abordados no interior deste, índices são utilizados para otimizar a velocidade e o desempenho da busca por um documento relevante em relação a um termo buscado. O custo pelo ganho de tempo durante a recuperação de informação é o espaço de armazenamento computacional adicional necessário para armazenar o índice.

É importante ressaltar que a etapa de Indexação é diretamente influenciada pela etapa de Pré-Processamento, pois, todo o conteúdo que será indexado, ou não, foi determinado por esta etapa. Desta forma, quando a etapa de Pré-Processamento faz uso de PLN e, com isso, fornece características linguísticas do texto processado, a etapa de Indexação utiliza estes dados ricos em semântica na construção do índice. Além disso, a etapa de Indexação também pode fazer uso de PLN, tornando possíveis duas abordagens distintas ao processo de criação de índices: Indexação Textual ou Indexação Temática.

3.3.1. Indexação Textual

O processo de Indexação Textual é realizado pela indexação dos termos presentes em um documento. É um procedimento automático e não utiliza informações externas, como por exemplo, um dicionário de palavras. Dependendo do algoritmo utilizado na construção do índice, é possível a realização de consultas com a utilização de operadores de proximidades e operadores booleanos.

Atualmente, a técnica mais utilizada para a indexação textual é a de **índices invertidos**[17] (KONCHADY, 2006) (FELDMAN & SANGER, 2007). Em sua apresentação básica, um índice invertido é uma estrutura de dados composta de uma lista ordenada, geralmente denominada vocábulo ou vocabulário, que armazena todas as palavras distintas encontradas nos textos e os documentos em que elas ocorrem. Informações adicionais como a frequência e posição de ocorrência da palavra no texto também podem ser armazenadas. Em (FONSECA & FIDALGO, 2002), há o exemplo de um índice invertido construído com auxílio de técnicas de Processamento de Linguagem Natural, reproduzido na Figura 17. Nota-se que, neste exemplo, todas as palavras foram mantidas em sua forma singular e em caixa baixa. Além disso, foram removidas do processo de indexação todas as palavras com pouco poder discriminatório (*stopwords*).

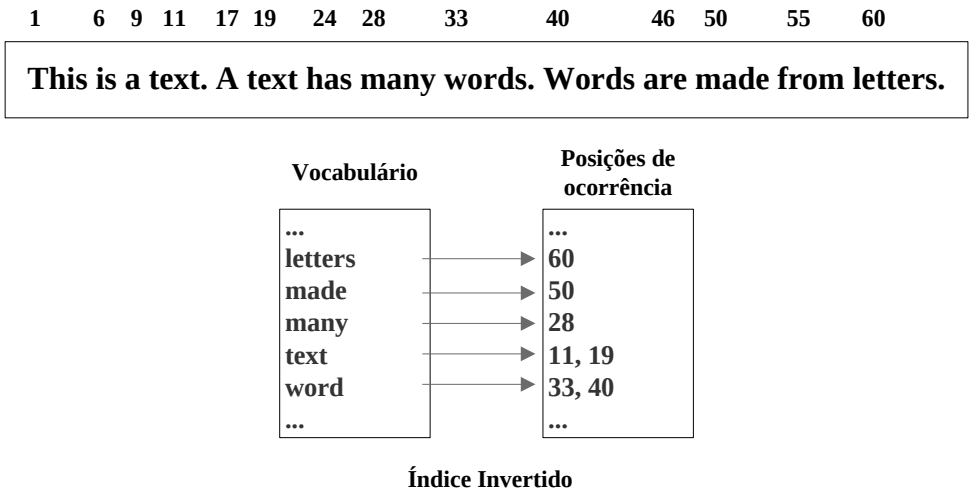

Figura 17 - Representação de um índice invertido

3.3.2. Indexação Temática

O procedimento de Indexação Temática é caracterizado pela constante consulta a um dicionário de termos. Este dicionário é conhecido pelo nome de *thesaurus* e sua função é simples: após receber uma palavra encontrada no texto, realiza uma consulta em sua base de dados e indica ao indexador o termo correto a

[17] Recebem essa denominação por inverter a hierarquia da informação. No lugar de uma lista de documentos contendo termos, tem-se uma lista de termos referenciando documentos.

ser utilizado na indexação da palavra recebida. Isto é possível devido a sua estrutura, ilustrada na Figura 18, que mapeia em único termo, este denominado termo preferido, todo um conjunto de termos sinônimos, estes denominados de termos não preferenciais. Como exemplo, podemos citar as palavras “carro”, “automóvel”, “veículo”, que poderiam ser associadas a uma única palavra que é “carro”.

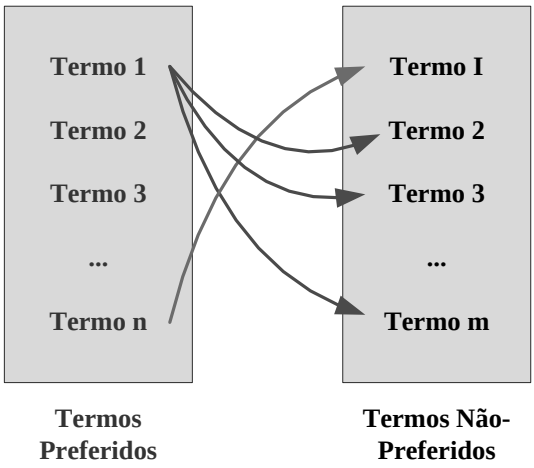

Figura 18 - Estrutura básica de um Dicionário Thesaurus

A utilização desta técnica, além de tornar o índice mais compacto, permite a localização de documentos grafados de forma diferente, mas que apresentam mesmo valor semântico. Porém, a maior dificuldade para a utilização deste mecanismo reside na criação do próprio dicionário.

3.4. Mineração

É na etapa de Mineração que ocorre a busca efetiva por conhecimentos novos e úteis a partir dos dados. Compreende a aplicação de algoritmos de Aprendizado de Máquina sobre os dados de forma a abstrair o conhecimento implícito presente nestes.

A escolha do algoritmo a ser utilizado está relacionada com o objetivo da tarefa de Mineração de Textos. Este objetivo, definido no início do processo, irá determinar quais as opções possíveis de Aprendizado de Máquina que se aplicam ao problema. Além disso, outros detalhes devem ser considerados, como por

exemplo, a necessidade ou não de que o conhecimento aprendido seja facilmente interpretável, o que pode descartar da lista de opções possíveis algoritmos de Aprendizado de Máquina do tipo "caixa preta", como Redes Neurais, pois a compreensão da rede neural resultante de um processo de aprendizado não é uma tarefa trivial e requer esforço adicional para a extração das regras aprendidas por esta técnica, como em (SETIONO & LEOW, 1998).

Outro fator restritivo é a necessidade de urgência do processo. Alternativas que, embora possam apresentar excelentes resultados, muitas vezes precisam ser desconsideradas em razão do elevado tempo de processamento computacional necessário para o treinamento destas.

Em (GOLDSCHMIDT & PASSOS, 2005) é ressaltado que a dificuldade de escolha de um algoritmo de aprendizado apropriado é intensificada na medida em que surjam novos algoritmos com o mesmo propósito, aumentando a diversidade de alternativas, mas, que, geralmente, a escolha dos algoritmos se restringe às opções conhecidas pelo analista do processo, que muitas vezes, deixa de considerar muitas alternativas promissoras.

3.5. Análise

A etapa de Análise, algumas vezes chamada de Pós-Processamento, abrange o tratamento do conhecimento obtido na etapa de Mineração, através da análise, visualização e interpretação deste. Tal tratamento tem como objetivo viabilizar a avaliação da utilidade do conhecimento descoberto (ZHU & DAVIDSON, 2007).

Para analisar o resultado de um processo de Mineração de Textos são utilizadas métricas de avaliação de desempenho. O objetivo de uma métrica de desempenho é graduar a execução de uma tarefa. As principais métricas de avaliação utilizadas em Mineração de Textos foram adotadas da área de Recuperação de Informação e são baseadas na noção de relevância. Um documento é considerado relevante quando possui importância para o tópico considerado. Precisão, Abrangência e Média-F são as métricas de desempenho mais utilizadas e serão abordadas nos itens "3.5.1", "3.5.2" e "3.5.3", respectivamente.

Porém, de acordo com o objetivo de cada processo de Descoberta de Conhecimento em Textos, métricas de avaliação de desempenho diferentes das citadas acima devem ser utilizadas. Por exemplo, uma tarefa de Sumarização não será bem avaliada por medidas como Abrangência, Precisão ou Medida-F.

Muitas vezes, de forma a facilitar análise do conhecimento obtido, podem ser utilizados métodos de transformação de dados que consistem, basicamente, na conversão de uma forma de visualização para outra (KANTARDZIC, 2002). Da mesma forma que as medidas de desempenho, diferentes estratégias de visualização podem ser empregadas, cada qual mais adequada ao objetivo do processo de Mineração de Textos. Por exemplo, para melhor visualizar as regras obtidas por um processo de Mineração de Textos é comum a conversão de árvores de decisão em regras ou vice-versa. No exemplo abaixo, as regras na Tabela 7 são visualizadas sob a forma de árvore de decisão na Figura 19.

Tabela 7 – Visualização das regras para concessão de empréstimos em uma tabela

Montante	Salário	Possui Conta?	Empréstimo
médio	baixo	indiferente	**não**
médio	alto	indiferente	**sim**
alto	indiferente	sim	**sim**
alto	indiferente	não	**não**

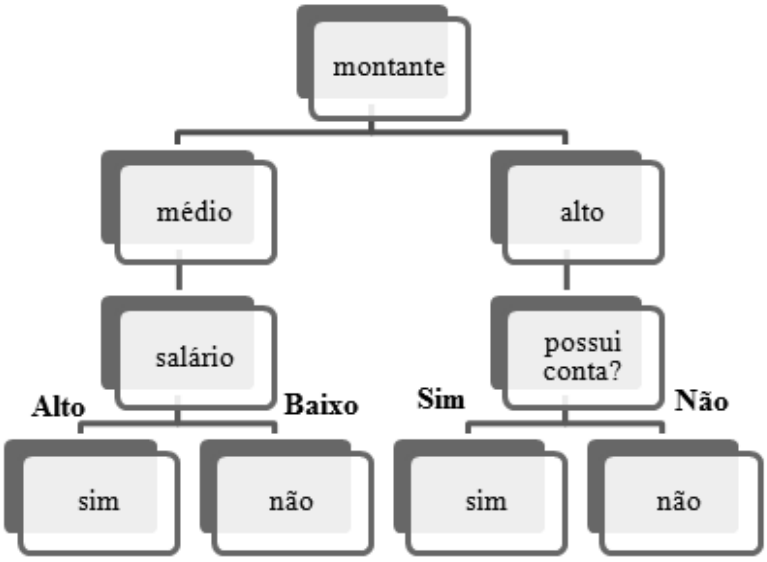

Figura 19- Visualização das regras para concessão de empréstimo em uma árvore de decisão

Usualmente, as técnicas de visualização de dados facilitam a compreensão do conhecimento obtido. Além de árvores de decisão e regras, outros recursos podem ser utilizados: gráficos bi ou tridimensionais, planilhas, tabelas e cubos de dados.

3.5.1. Precisão

Para um dado conjunto de itens recuperados, precisão é definida como a proporção entre o número de itens relevantes recuperados e o número total de itens recuperados (Equação 2).

$$\textbf{Precisão} = \frac{\mathbf{N_{recuperados} \cap N_{relevantes}}}{\mathbf{N_{recuperados}}}$$

Equação 2 - Fórmula da métrica de desempenho “Precisão”

3.5.2. Abrangência

Para um dado conjunto de itens recuperados, a abrangência é definida como a proporção entre o número de itens relevantes recuperados e o número total de itens relevantes no sistema em questão (Equação 3).

$$\textbf{Abrangência} = \frac{\mathbf{N_{recuperados} \cap N_{relevantes}}}{\mathbf{N_{relevantes}}}$$

Equação 3 - Fórmula da métrica de desempenho “Abrangência”

3.5.3.
Medida-F

A Medida-F, Média Harmônica ou *F-Mean* é a combinação de Abrangência e Precisão em uma única métrica (Equação 4) Esta função assume valores no intervalo [0, 1]. Quando o valor retornado é zero, não há documentos relevantes no conjunto de dados medido. Quanto mais próximo de um o resultado da fórmula, maior relevância possui o conjunto de dados testado.

Algumas vezes, a Medida-F é definida à priori; nesses casos, busca-se encontrar a relação ideal entre Abrangência e Precisão para que o resultado da métrica seja obtido. Também pode ser utilizada com pequenas variações nos termos visando atribuir pesos diferentes para Abrangência e Precisão.

$$\mathbf{Medida - F} = \frac{\mathbf{2}}{\frac{\mathbf{1}}{\mathbf{Abrangência}} + \frac{\mathbf{1}}{\mathbf{Precisão}}}$$

Equação 4 - Fórmula da métrica de desempenho "Medida-F"

3.5.4.
Precisão x Abrangência

Em resumo, Precisão é a porcentagem dos itens recuperados que são relevantes. Abrangência é a porcentagem dos itens relevantes que foi recuperada. Por exemplo, uma consulta com valor de Precisão igual a 0.70 significa que 70 por cento dos itens recuperados são relevantes, ao passo que uma consulta com valor de Abrangência igual a 0.70 possui apenas 70 por cento dos documentos que são ou poderiam ser relevantes.

Abrangência e Precisão são frequentemente objetivos contraditórios (BAEZA-YATES & BERTIER, 1999), pois, na medida em que se deseja obter mais itens relevantes (aumentando o nível de abrangência), mais itens irrelevantes também são recuperados (diminuindo o nível de precisão) (SILVA F. R., 2007). Estudos empíricos sobre o desempenho mostram uma tendência de declínio da precisão na medida em que a abrangência aumenta. Em (BUCKLAND & GEY, 1944) é comprovado que obter altos índices de Precisão ou Abrangência é

possível a qualquer sistema, porém não simultaneamente. Na Figura 20 é ilustrado o gráfico de equilíbrio entre Abrangência e Precisão. A linha contínua representa o mapeamento real da relação Abrangência-Precisão em uma coleção de documentos e a linha tracejada representa a relação ideal entre estas duas métricas.

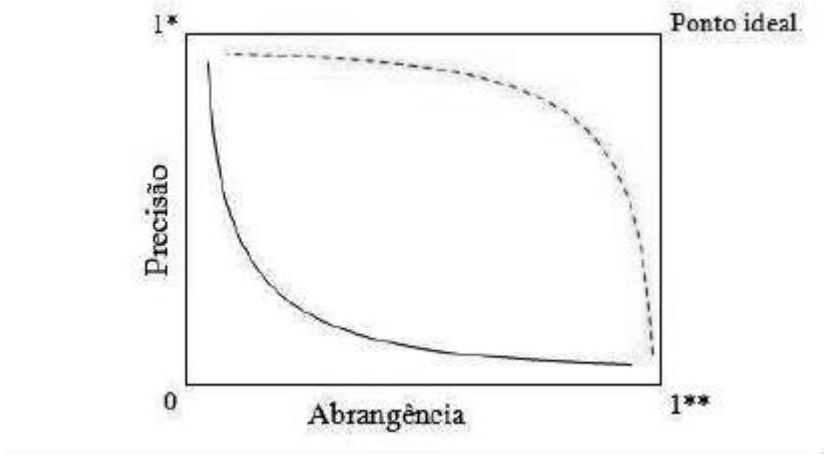

Figura 20 – Gráfico de compensação entre precisão e abrangência

Com base nesta relação, de acordo com (SILVA F. R., 2007), pode-se determinar que um Sistema A é melhor do que o Sistema B, segundo as métricas de Precisão e Abrangência destes sistemas, se, em todos os pontos de Abrangência, o valor da precisão do Sistema A for maior do que do Sistema B. Caso isso não aconteça, as médias dos valores de precisão para valores de abrangência selecionados são calculadas e comparadas.