6
Sistema Proposto

"Quem nunca errou nunca experimentou nada novo."

Albert Einstein

O presente projeto visa a ser um estudo de caso para validação e comprovação da utilidade do uso de técnicas de inteligência computacional e aprendizado de máquina no tratamento de ambigüidades de sentido de palavras e dessa forma se constitui em uma ferramenta enquadrada na área de mineração de textos.

6.1
Desambiguação baseada em agrupamento de documentos

Uma abordagem para o tratamento de ambigüidade lexical, a descoberta de palavras com a mesma grafia que contém diferentes significados, pode ser o emprego de técnicas de agrupamento. Supondo-se que termos ambíguos com significados distintos tendam a pertencer a documentos de diferentes assuntos pode-se detectar a existência de tais ambigüidades com base na distribuição do termo ambíguo em questão por diferentes grupos ou classes de assuntos. Tais grupos podem ser encontrados com base na similaridade gerada pelos termos mais comumente encontrados em determinadas classes e menos aparentes em outras. Intuitivamente, essa abordagem pode ser enxergada levando-se em consideração que cada significado, de uma mesma palavra, ambígua, costuma estar mais fortemente associado a um pequeno grupo de outras palavras. Um exemplo disso é dado pela palavra lula, que pode se referir ao presidente do Brasil, e nesse caso está mais associada a termos como política, presidência, Brasil, etc; e também pode se referir ao molusco marinho, e se associar a termos como mar, pesca, dentre outros.

Maior precisão durante a recuperação de informações pode ser obtida com base na abordagem mencionada. Para isso, durante a fase de indexação, os termos devem ser indexados levando consigo informações sobre a quais grupos

pertencem e, dessa forma, sempre que uma consulta for submetida, uma avaliação prévia sobre possíveis ambigüidades deve ser realizada. Essa avaliação é obtida pela análise dos grupos associados aos termos buscados, e, caso sejam constatadas diferenças substanciais, o termo deve ser considerado ambíguo e opções de desambiguação devem ser retornadas ao usuário, constituindo um sistema semi-automático, já que não existe outra forma de "adivinhar" o que o usuário deseja realmente, caso ele não seja mais específico. Assim, após o refinamento da consulta, baseado na detecção automática de ambigüidades, o resultado deve se tornar mais preciso.

A desambiguação de termos durante a recuperação de informação é amplamente utilizada em sistemas de lojas virtuais, por exemplo. Porém as informações, em tais sistemas, costumam estar completamente estruturadas, armazenadas em bancos de dados e ainda separadas por categorias bem definidas, pré-classificadas manualmente e utilizadas para o tratamento de ambigüidades. A diferença na abordagem proposta é a utilização de documentos em formato livre de qualquer estrutura ou semi-estruturados e de agrupamento automatizado pela utilização de alguma técnica inteligência computacional ou de aprendizado de máquina. A visão geral do sistema é apresentada na figura 19.

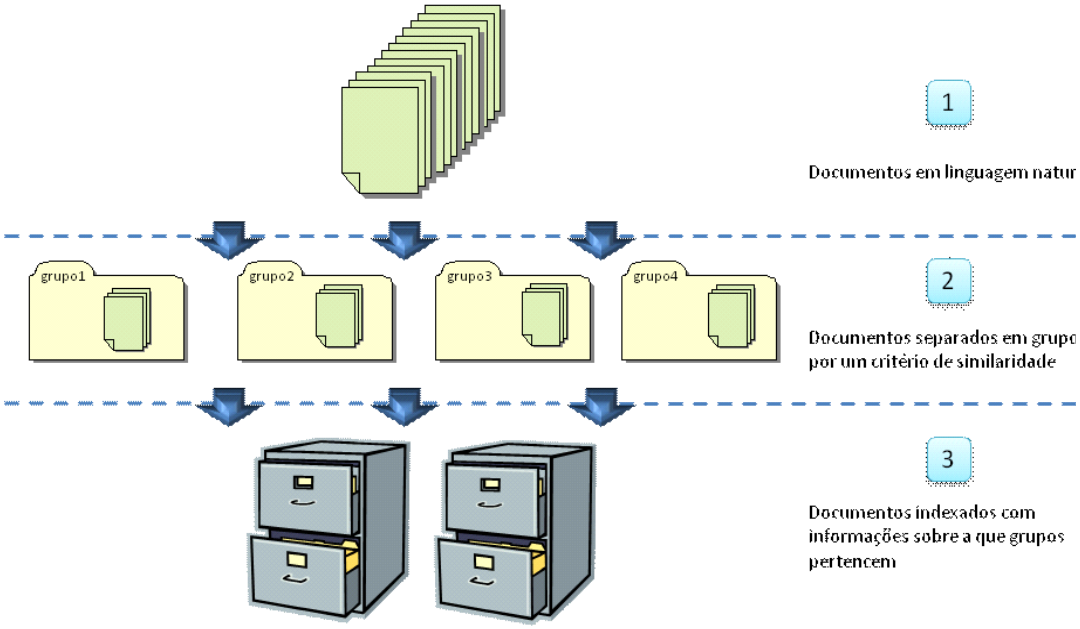

Figura 19 – Indexação de documentos com informações de agrupamento.

A transição do ambiente 1 para o ambiente 2 necessita de alguns analisadores específicos para cada formato de documentos. São ferramentas de *parsing* capazes de *tokenizar* (veja seção 2.2.1 - Tokenização) e extrair os termos a serem utilizados pela ferramenta de agrupamento para análise de similaridade entre documentos (veja seção 4.3 – Medidas de Similaridade). Informações sobre os grupos, obtidas no ambiente 2 são passadas para a ferramenta de indexação (veja seção 2.3 - Indexação) no ambiente 3 de maneira que seja possível localizar os termos sabendo os respectivos grupos aos quais estão contidos. Dessa maneira, é possível aumentar a precisão, com base na desambiguação de termos, que, tendo significados diferentes, tendam estar em grupos distintos. Durante a recuperação da informação o comportamento do sistema pode ser visto pela figura 20.

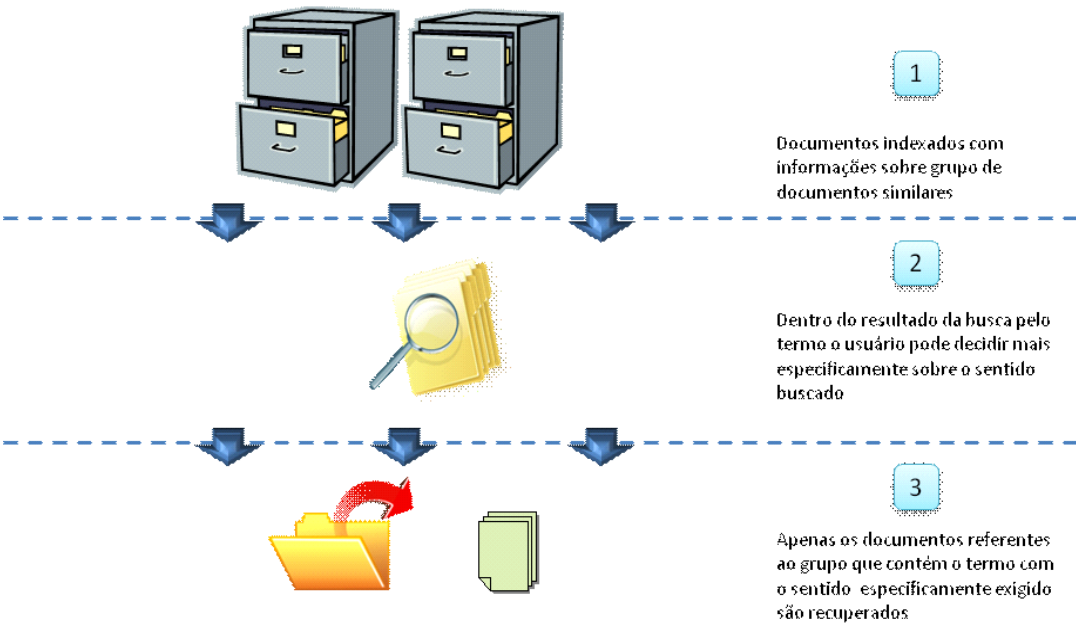

Figura 20 – Recuperação de informações desambiguadas por informações de agrupamento.

A transição do ambiente 1 para o ambiente 2 é dada por um processo comum de recuperação de informações, porém com base em uma análise dos grupos recuperados são submetidas ao usuário opções de escolha. Essa opção pode ser dada apresentando-se os termos mais comuns dentro de cada grupo para que assim o usuário possa escolher o sentido associado ao termo buscado. Dessa forma, o sistema semi-automatizado é capaz de melhorar a precisão conseguindo desambiguar o termo buscado através da separação entre grupos.

Podem-se agregar ao sistema outras ferramentas com o intuito de melhorar ainda mais o processo como um todo. Por exemplo, um módulo capaz de fazer extração de entidades (veja seção 1.3.5 – Extração de Informações) entre o ambiente 1 e o ambiente 2 na fase de indexação pode ser de grande valia, e nesse caso a busca poderia ser mais precisa ainda, sendo efetuada em segmentos dentro dos documentos. A tarefa de extração de entidade fica bastante facilitada com a utilização de textos etiquetados ou ainda de dicionário de termos. Outro proveito de uma ferramenta de extração de entidades é a indexação em banco de dados, que torna a busca mais rápida e fácil. Determinando-se a entidade de um termo o armazenamento em um banco de dados já preparado para tal se torna uma tarefa simples.

6.2
Processamento de textos

O processamento de grandes massas de texto é dificultoso e por isso buscou-se inicialmente uma forma de se avaliar a capacidade potencial do sistema. A máquina utilizada possui os seguinte itens relevantes de configuração:

1. AMD Athlon 64 X2 Dual Core Processor 4400+ 2,31 GHz;
2. 2 Gb de memória RAM.

Para essa avaliação o *corpus* utilizado foi obtido a partir do CetenFolha 1.0 (Corpus de Extractos de Textos Electrônicos NILC/Folha de S. Paulo), que é um *corpus* de cerca de 24 milhões de palavras em português brasileiro, criado pelo projeto Processamento Computacional do Português (projeto que deu origem à Linguateca) com base nos textos do jornal Folha de S. Paulo que fazem parte do corpus NILC/São Carlos, compilado pelo Núcleo Interinstitucional de Lingüística Computacional (NILC).

Um *parser* específico para lidar com os rótulos do CetenFolha foi construído de maneira a separar todos os artigos segundo seu identificador e grupo, uma vez que o sistema SID foi construído para lidar com um diretório contendo documentos sendo representados por um único arquivo. Os documentos dentro do arquivo CetenFolha são discriminados através de um identificador único e também são etiquetados segundo o caderno no qual a folha o publicou. Para

efeito de armazenar esses cadernos para utilizá-los como grupos uma estrutura tipo tabela *hash* foi construída para atribuir valores a cada artigo segundo o grupo ao qual ele pertencesse conforme a tabela 1. Esses grupos dados pela Folha foram inicialmente utilizados como os grupos naturais, ou seja, aqueles não formados por algum método de agrupamento. A classificação do texto, inspirada nas classificações do CETEM Público tem como valores possíveis: pol (política brasileira e internacional), des (desporto), eco (economia), clt (cultura), opi (opinião), agr (agricultura), vei (veículos), com (informática) e nd (não determinado). Alguns artigos pertencem a mais de uma categoria (marcados, por exemplo, como clt-soc). A estrutura construída no sistema usou potências de 2 para a codificação dos grupos pensando numa forma simples e rápida de armazenar, de maneira única, os múltiplos grupos que a Folha fornecia. Dessa forma artigos classificados em mais de uma categoria puderam ser facilmente identificados através do seu código, pois para cada valor existe uma e apenas uma combinação possível, baseada na soma dos códigos de cada categoria individualmente. Como exemplo, teríamos para o caso de um documento classificado nas categorias economia e opinião (eco + opi) o código numérico 80 (16 + 64), que garante unicidade da combinação.

Tabela 1 – Codificação das classes naturais do corpus Cetenfolha

Cód. string (rótulo no CetenFolha)	Cód. Numérico para cada categoria
Agr	**1**
CLT	**2**
Com	**4**
Dês	**8**
Eco	**16**
Nd	**32**
Opi	**64**
Pol	**128**
Soc	**256**
Vei	**512**

O sistema é constituído de estruturas de dados cuidadosamente montadas para ampliar a capacidade de processamento de textos e possibilitar a montagem de um modelo vetorial para um grande *corpus*.

6.2.1
Indexação do Corpus

O *corpus* é indexado com o uso de uma tabela *hash*, que pode ser ampliada com a aquisição de novos documentos. Como a desambiguação é calcada em grupos de documentos similares essa estrutura que indexa os documentos também armazena os resultados de agrupamentos previamente realizados e assim é capaz de responder a consultas para recuperação de informação com base nos grupos pré-estabelecidos. Essa facilidade, de armazenamento da identificação dos grupos, pode ser utilizada durante a desambiguação, partindo-se da premissa de que determinado termo ambíguo costuma estar associado a assuntos distintos dependendo do sentido empregado. A classificação pré-estabelecida, quando inexistente, é substituída pela aplicação de alguma técnica de agrupamento capaz de identifica homogenia dentro dos grupos e heterogenia entre os mesmos.

6.2.2
Indexação do Léxico

Pensou-se na seguinte abordagem para construção do léxico: um dicionário, isto é, uma tabela *hash*, contendo todos os termos que aparecem em todos os documentos do *corpus*, com cada linha desse dicionário indexando uma estrutura capaz de ser encadeada e armazenar um valor para identificar o documento (*D*) no qual o termo aparece (figura 21) e outro para indicar o número de vezes que o referido termo (*T*) aparece no respectivo documento. Essa estrutura (índice invertido) pode ser estendida para trabalhar com indexação posicional, ou seja, identificando cada posição que o termo aparece dentro do documento, facilitando assim a recuperação de conjunto de termos associados. Abstraindo para o modelo matricial seria como se as linhas fossem os termos e as colunas os documentos.

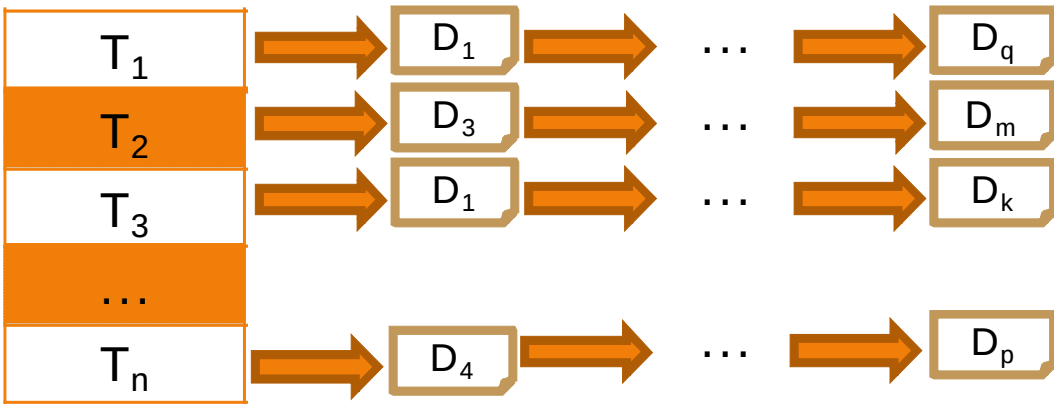

Figura 21 – Índice invertido, onde termos são índices para documentos.

6.2.3
Montagem do modelo vetorial

Considere uma matriz, baseada na representação comum de duas dimensões com muitos valores iguais a zero, muito comum em problemas de grafos como ordenação topológica, algoritmos de menor caminho e etc. Na representação vetorial de documentos de um *corpus* a esparsividade da matriz também se torna característica constante. Neste caso, uma grande quantidade de memória é desperdiçada para armazenar explicitamente esses valores zerados. Além disso, quando operações são executadas sobre matrizes assim representadas, como adição e, principalmente, multiplicação, existe um grande desperdício de tempo, já que muitas operações utilizam operandos com valor zero, o que pode ser considerado desnecessário. Em uma representação esparsa pode-se armazenar os elementos diferentes de zero e implicitamente assumir que os elementos não armazenados têm valor zero. Então em vez da utilização comum de duas dimensões para representação de uma matriz esparsa é interessante o uso de uma estrutura de dados capaz de reduzir o desperdício de espaço utilizado para armazenar os dados e facilitar as operações de modo a acelerar o tempo de execução delas.

Um modelo enxuto de armazenamento dos valores diferentes de zero foi construído, utilizando apenas as referências de posicionamento, dentro da matriz bidimensional, para cada um dos referidos valores, e dessa forma o modelo de

espaço vetorial (Salton, 1983) no qual os documentos são representados como vetores puderam ser carregados em memória com uma capacidade estendida.

O sistema foi capaz de montar o Modelo Vetorial com 4671 documentos e 29934 dimensões, testado pelos documentos gerados pelo *parser* especificamente criado para o *corpus* CetenFolha 1.0, completamente carregado em memória volátil sem estratégia de uso de *swap* entre memória RAM e disco. Optou-se por fornecer como saída o modelo vetorial no formato de matriz bidimensional esparsa como arquivo texto com o intuito de se utilizar ferramentas adicionais (MATLAB, SPSS, dentre outras) para de uso mais diversificado de técnicas de agrupamento. Neste momento cabe dizer que o MATLAB rodando na mesma máquina não foi capaz de carregar a matriz de saída do sistema por estouro de memória. Já o SPSS conseguiu ler a matriz e apresentar os dados, porém não foi capaz de rodar nenhuma das ferramentas de análise solicitadas (agrupamento hierárquico, k-meas, análise de fatores) também devido a estouro de memória. O próprio Windows teve dificuldades para indexação e apresentação dos textos por meio de sua interface gráfica. Demorava muitos minutos para apresentar listagem dos textos dentro do diretório e por isso a maioria das operações sobre os arquivos foi realizada utilizando-se o sistema *prompt* através de antigos comandos de DOS. Por esses motivos optou-se por um estudo de caso prático sobre um universo reduzido e dentro de um ambiente controlado.