4
Tarefas de Mineração de Textos

No presente capítulo são introduzidas as principais Tarefas de Mineração de Textos, as quais se prestam a cumprir determinados objetivos do processo. É necessário então que se tenha definido claramente o que se quer obter de informação a partir da massa textual preparada até o momento, para que se possa definir a tarefa e, posteriormente, o algoritmo que a implementa.

4.1.
Categorização de Textos

A Categorização de Textos é a tarefa que tem como finalidade identificar os tópicos principais em um documento e associá-los a uma ou mais categorias predefinidas [20]. Categorizar textos é similar ao processo de categorização de livros em uma biblioteca. Livros sobre o mesmo assunto devem estar próximos, registrados sob um mesmo número ou prefixo de chamada; livros que possuam assuntos similares ou próximos devem estar em prateleiras adjacentes; e livros que não possuam qualquer relacionamento entre si devem estar bem afastados, possivelmente até em andares diferentes.

O principal objetivo de se categorizar alguma coisa, sejam livros ou textos, é permitir que o acesso a determinado conteúdo seja feito de forma facilitada, sem que haja a necessidade de se despender muito esforço.

A Categorização de Textos envolve uma série de etapas intermediárias, como: a determinação dos atributos discriminantes de cada amostra, como finalidades léxicas ou freqüência de termos em um documento; a definição das categorias e suas *assinaturas*, isto é, que características presentes em cada categoria são determinantes para que seja possível indicar uma relação com as amostras; execução do processo de treinamento, aonde determinado algoritmo é *calibrado*, "aprendendo" como classificar um documento; e, finalmente, o processo de classificação em si, seguido da avaliação dos resultados. A forma mais simples de classificação é a **binária**, aonde um documento é classificado

como pertencente ou não a determinado tópico, porém algumas máquinas de aprendizado podem ser capazes de categorizar documentos entre diversas classes distintas. A Figura 10 ilustra um algoritmo de categorização que “decide” em qual das três classes predefinidas um novo documento deve ser associado.

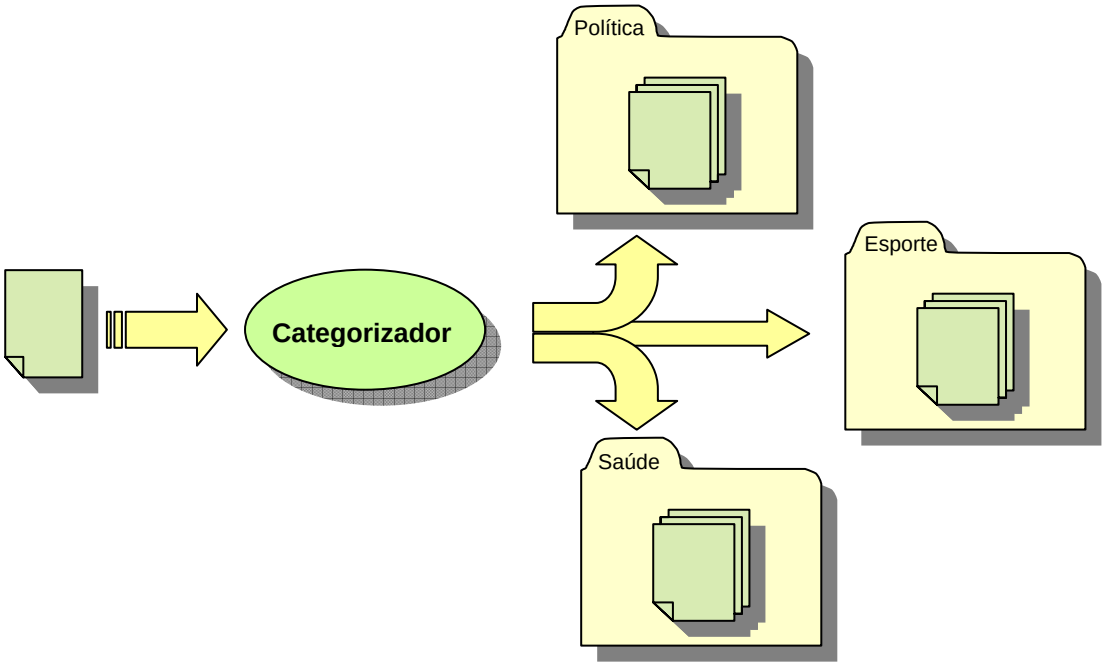

Figura 10 – Classificação ternária de documentos.

Existem alguns critérios na literatura para avaliação dos sistemas de classificação textual [22], dentre eles estão:

- Precisão, a habilidade de prever a classe dos documentos. Isto é feito ao comparar rótulos atribuídos pelo classificador com rótulos atribuídos por um ser humano.

- Velocidade e escalabilidade do treinamento.

- Facilidade, velocidade e escalabilidade para inserção, deleção e alteração de documentos no corpus de treinamento.

- Facilidade de diagnosticar, interpretar os resultados e adicionar julgamento humano para melhorar o classificador.

Existem diversos algoritmos de categorização textual sendo desenvolvidos na literatura, divididos basicamente em duas categorias: a dos baseados em **Aprendizado de Máquina** e a dos baseados em **Recuperação de Informação**. Como exemplos da primeira estão as Árvores de Decisão, Redes Neurais Artificiais, classificadores probabilísticos (*Naive Bayes)* e as Máquinas de Vetor de Suporte (SVM). Já na segunda categoria incluem-se os classificadores lineares e classificadores de conjuntos de exemplos genéricos. O algoritmo utilizado como estudo de caso nesta dissertação é o *Naive Bayes*, detalhado na seção 4.1.3.

4.1.1. Treinamento e Teste

Existem diversas estratégias para a execução do treinamento e, posteriormente, aplicação de testes. O treinamento tem como objetivo apresentar ao classificador exemplos que o farão conhecer e aprender sobre a massa textual. A aplicação de testes possibilita a avaliação da performance, descrita na próxima seção. A seguir, a descrição das principais estratégias relatadas na literatura:

- ***Holdout***: Consiste em separar do conjunto de treinamento uma determinada porção, compondo o conjunto de teste. Usualmente, o teste utiliza 1/3 do conjunto total, mantendo o restante para treinamento. A Figura 11 ilustra o funcionamento desta estratégia que, apesar de simples e rápida de se aplicar, recebe críticas por não usar de forma otimizada o conjunto total de amostras – o classificador poderia ficar melhor construído se utilizado todo o conjunto de treinamento – e pela própria aleatoriedade dos dados, isto é, o conjunto de teste pode acabar ficando "favorecido", levando a uma falsa conclusão da real adequação do treinamento.

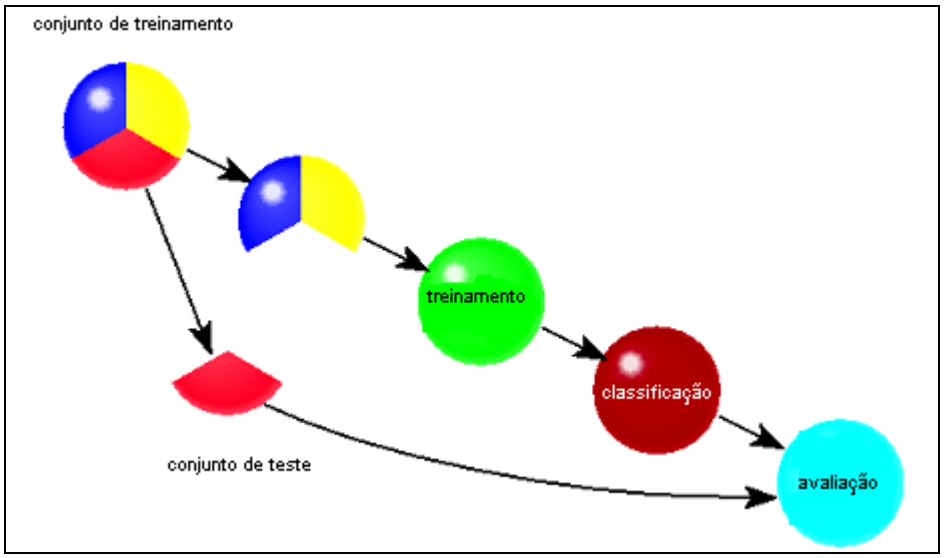

Figura 11 – Utilização da estratégia *holdout* para treinamento e validação de classificadores.

- ***K-Fold Cross Validation* (Validação Cruzada):** Validação Cruzada é a metodologia de treinamento e teste que trabalha com o conceito de *folds*, conforme introduzido por Geisser em [49]. Desta forma, o conjunto de amostras inicial é dividido em *k* subamostras. Destas *k* subamostras, uma subamostra é retida para ser utilizada na validação do modelo (conjunto de teste) e as *k*-1 subamostras compõem o conjunto de treinamento. O processo é então repetido *k* vezes, de modo que cada uma das *k* subamostras seja utilizado ao menos uma vez como teste. O resultado final é a média do desempenho do classificador nas *k* iterações. O objetivo desta estratégia é aumentar a confiabilidade da avaliação, com o ônus de se despender mais tempo que a técnica anterior. Vale ressaltar que nada impede que as duas estratégias possam ser combinadas, com a aplicação da técnica de *holdout* como mais uma forma de validar os resultados conseguidos com a Validação Cruzada, com o ônus de se despender muito mais tempo para a execução dos ciclos e de ser necessário mais dados para que os conjuntos (treinamento e teste) formados possam ter tamanho significante. A Figura 12 ilustra a utilização da Validação Cruzada com 3–*folds*.

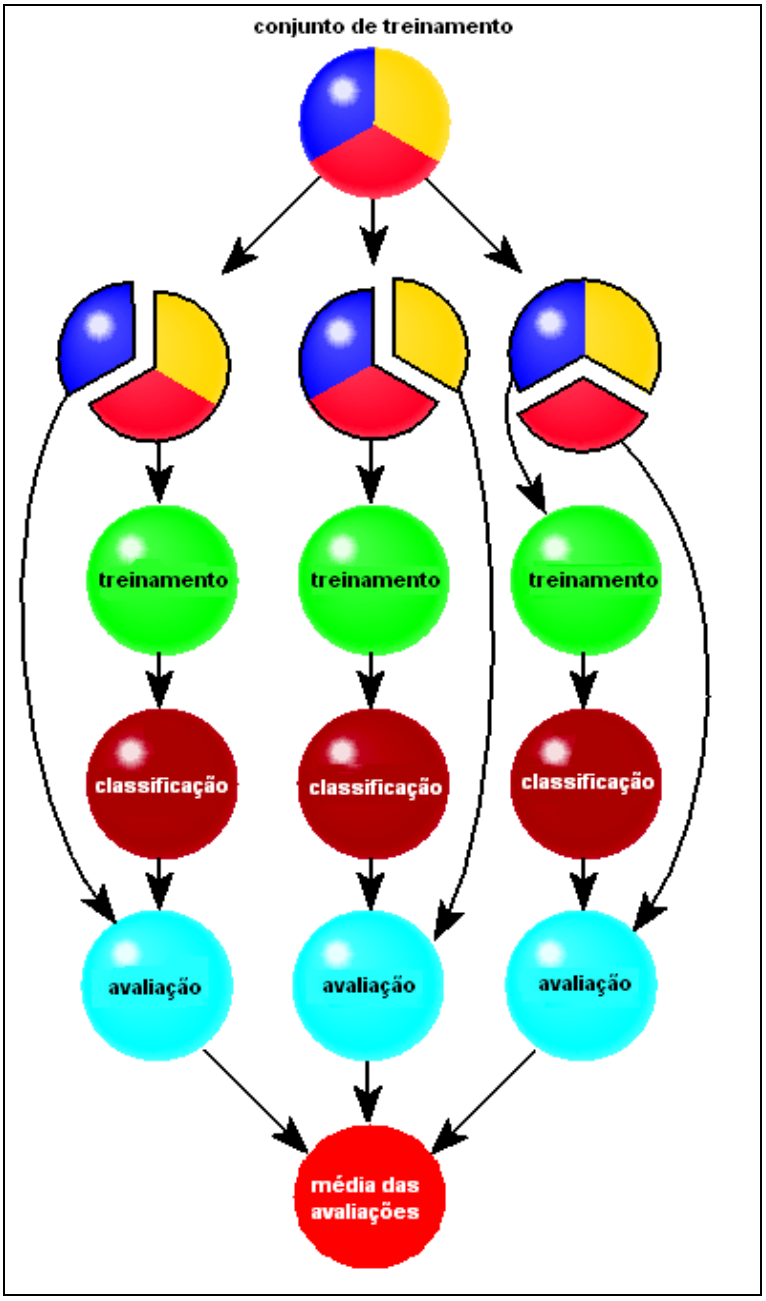

Figura 12 – Validação Cruzada com 3-*folds*.

4.1.2. Avaliação de Performance

Avaliar a performance do classificador é verificar o quão este é capaz de discriminar um novo exemplo, quando lhe é apresentado. A avaliação deve ser feita logo após o treinamento, utilizando o resultado da classificação do conjunto

de teste. Diversas são as métricas que apóiam esta etapa, provenientes principalmente da área de Recuperação de Informação, conforme listadas a seguir:

- **Precisão**: Mede a porção de exemplos de uma classe que foi corretamente classificada.

$$precisão(A) = \frac{total\ de\ exemplos\ corretamente\ classificados\ da\ classe\ A}{total\ de\ exemplos\ corretamente\ classificados}$$

- ***Recall*** (Eficiência): Proporção de amostras classificadas como sendo de uma classe em relação ao total de amostras da classe.

$$recall(A) = \frac{total\ de\ exemplos\ corretamente\ classificados\ da\ classe\ A}{total\ de\ exemplos\ da\ classe\ A}$$

- **Acurácia**: Denota a proporção total de classificações corretas.

$$acurácia = \frac{total\ de\ amostras\ classificadas\ corretamente,\ independente\ da\ classe}{total\ de\ exemplos\ do\ conjunto\ de\ teste}$$

- ***F-Measure*** (Medida F): Média harmônica entre Precisão e *Recall*. Bastante utilizada quando as predições de um classificador estão desbalanceadas, ou seja, eficaz para uma determinada classe e não para a outra. *F-Measure* também é interessante por fornecer uma medida única de comparação.

$$F - Measure = \frac{2}{(1/\ precisão) + (1/\ recall)}$$

4.1.3.
Naive Bayes

O classificador ***Naive Bayes*** é um dos mais utilizados em *Machine Learning*, apresentando excelentes resultados para a categorização de textos. O termo "*naive*" que, em português significa **ingênuo**, é atribuído à **independência condicional dos atributos**, ou seja, a informação de um evento não tem nenhuma relação com a informação de outro. É baseado no **Teorema de Bayes**, formulado no século XVIII por Thomas Bayes [50][51] e, como classificador, é considerado um dos mais eficientes em tempo de processamento e precisão quando da rotulação de novas amostras, características justificadas pela abordagem simplória com que trata as características dos exemplos. A seguir são introduzidos os fundamentos teóricos do classificador, bem como o desenvolvimento da teoria dos dois modelos existentes: o modelo **binário** e o modelo **multinomial**.

4.1.3.1.
Fundamentos Teóricos

O treinamento do classificador *Naive Bayes* envolve o cálculo de uma distribuição geradora *Pr(d|c)* para cada classe $c \in \{-1,1\}$. Na fase de classificação, simplesmente é calculada qual distribuição tem a maior probabilidade de ter gerado cada documento.

Nas máquinas *Bayesianas*, a criação de documentos é modelada como o seguinte processo:

1. Cada classe c possui uma probabilidade a priori associada $\Pr(c)$ tal que $\sum_c \Pr(c) = 1$.

2. Dado que existe uma distribuição de documentos *Pr(d|c)* associada a classe c escolhida, esta distribuição é usada para gerar o documento.

Sendo assim, a probabilidade de se gerar um documento da classe *c* é *Pr(c) Pr(d|c)*. Finalmente, dado um documento d a probabilidade a *posteriori* de que *d* foi gerado da classe *c* é:

$$\Pr(c \mid d) = \frac{\Pr(c)\Pr(d \mid c)}{\sum_{\gamma} \Pr(\gamma)\Pr(d \mid \gamma)}$$

Onde γ itera sobre ambas as classes de forma que Pr(c|d) torna-se uma medida de probabilidade apropriada.

Pr(γ|c) é estimado através da modelagem das distribuição dos termos sobre as classes gerando um conjunto de parâmetros que chamaremos de *Θ*. A estimativa de *Θ* é baseada nos termos contidos nos documentos de treinamento.

Após ter sido observado os dados de treinamento *D*, a distribuição a *posteriori* para *Θ* pode ser expressa por *Pr(Θ|D)*.

Dadas estas definições e dado um documento d, a probabilidade de d pertencer à classe c é dada por:

$$\Pr(c \mid d) = \sum_{\Theta} \Pr(c \mid d, \Theta) \Pr(\Theta \mid D)$$

$$= \sum_{\Theta} \frac{\Pr(c \mid \Theta) \Pr(d \mid c, \Theta)}{\sum_{\gamma} \Pr(\gamma, \Theta) \Pr(d \mid \gamma, \Theta)} \Pr(\Theta \mid D)$$

4.1.3.2. Modelo Binário

No modelo binário, assumimos que cada documento é representado por um vetor de atributos binários de modo que cada atributo indica a ocorrência ou não de determinado *token* no documento.

Neste modelo, $\Phi_{c,t}$ indica a probabilidade de um documento da classe *c* mencionar o termo *t* pelo menos uma vez. Assim:

$$\Pr(d \mid c) = \prod_{t \in d} \phi_{c,t} \prod_{t \in W, t \notin d} (1 - \phi_{c,t})$$

Onde W é o conjunto total de *tokens*. Para que seja evitado calcular $\prod_{t \in W, t \notin d}(1-\phi_{c,t})$ para cada documento classificado, a equação acima é reescrita:

$$\Pr(d \mid c) = \prod_{t \in d} \frac{\phi_{c,t}}{1 - \phi_{c,t}} \prod_{t \in W} (1 - \phi_{c,t})$$

É computado previamente $\prod_{t \in W} (1-\phi_{c,t})$ pata todo *c*, e somente computado primeiro produtório em tempo de classificação. Com isso, o tempo de

classificação de um termo fica linear em função do número de *tokens* presentes no documento ao invés de linear em função de | W |.

4.1.3.3. Modelo Multinomial

No modelo multinomial, assume-se que cada documento é representado por um vetor de atributos inteiros caracterizando o número de vezes que cada *token* ocorre no documento.

É possível imaginar o modelo multinomial como uma ”roleta” com |W| faixas. Ao gerar um documento, o gerador primeiro escolhe um comprimento l para um documento e depois gira a roleta l vezes para definir quais *tokens* colocará no texto. Durante a fase de treinamento são calculadas as dimensões de cada faixa das roletas das classes geradoras.

Seja $\phi_{c,t}$ a probabilidade da faixa t $\in W$ da roleta ser sorteada num giro. Seja $n(d, t)$ o número de vezes que t ocorre no documento d que possui comprimento $l_d = \sum_t n(d, t)$. O comprimento do documento é uma variável aleatória denominada L e assume-se que segue uma distribuição apropriada para cada classe neste modelo:

$$\Pr(d \mid c) = \Pr(L = l_d \mid c)\Pr(l_d, c)$$

$$= \Pr(L = l_d \mid c)\binom{l_d}{\{n(d,t)\}}\prod_{t \in d}\theta_t^{n(d,t)}$$

Onde $\binom{l_d}{\{n(d,t)\}} = \frac{l_d!}{n(d,t_1)!n(d,t_2)!\ldots}$ é o coeficiente multinomial que pode ser desprezado uma vez que tem o mesmo valor para todo c. Assume-se, para este caso, uma mesma distribuição de comprimento para ambas as classes. Apesar das avaliações positivas serem em média maiores, não é necessário considerar este critério para estimar a qualidade dos modelos. Portanto, é ignorado também o termo $Pr(L = l_d|c)$ nos experimentos.

Todos os fundamentos teóricos apresentados acerca dos Modelos Binário e Multinomial, bem como a teoria que o inspira, tiveram como fonte os seguintes trabalhos: [12][20][22].

4.2. *Clusterização*

Clusterização ou agrupamento é a tarefa que determina uma organização, em grupos chamados *clusters*, para uma coleção de documentos. Para tanto, os documentos devem ser agrupados seguindo alguma métrica que mede a similaridade entre estes. Normalmente estas métricas são computadas sob números, logo é necessário que, dado um documento, este seja mapeado para a forma vetorial, conforme visto na seção 3.3.1 - Representação de Documentos.

A Tarefa de *clusterização* pode ser considerada bastante complexa, característica atribuída principalmente pela quantidade de parâmetros exigidos pelos algoritmos que a realizam e a não existência de uma resposta "correta", como na Tarefa de Classificação, justamente por não se conhecer o número de *clusters* nem suas características a priori. A Figura 13 ilustra o esquema básico de uma *clusterização* de documentos.

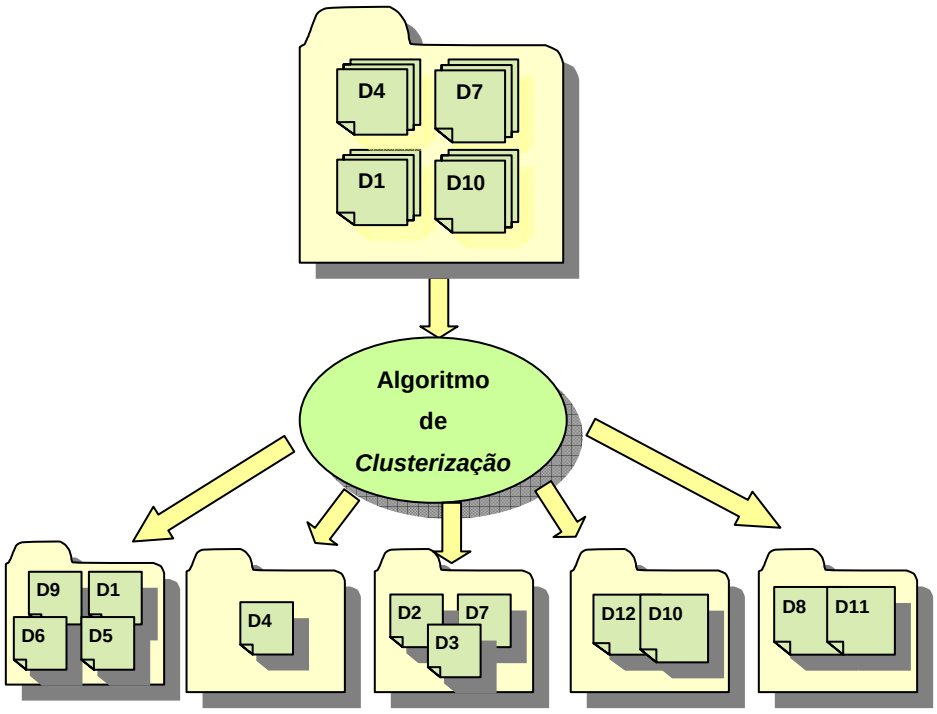

Figura 13 – Esquema básico da Tarefa de *Clusterização.*

Os métodos de *clusterização* também podem ser classificados quanto ao tipo de estrutura gerada, sendo chamados de **hierárquicos** ou **não-hierárquicos**. Os **não-hierárquicos** são os mais simples, aonde é feita uma divisão de N

documentos em M grupos sem sobreposição. Cada documento é associado ao grupo que melhor o representa, isto é, aquele grupo o qual contém documentos similares ao mesmo. Apesar de ser considerado simples, normalmente é requerido algum tipo de informação *a priori* sobre a coleção, como o número de grupos a serem formados e parâmetros "ideais" para a execução do algoritmo.

Cada item pertence ao grupo que melhor represente suas características. Esses métodos são heurísticos por natureza, pois certas decisões precisam ser tomadas *a priori* como número de grupos e critério de agrupamento. São mais utilizados quando os recursos computacionais são limitados.

Já os **hierárquicos** são mais complexos, produzindo um conjunto de dados aninhados, onde pares de itens são sucessivamente ligados até que todos os itens do conjunto estejam conectados. Os métodos hierárquicos podem ser **aglomerativos**, onde aos N documentos são aplicados N-1 junções de pares de documentos ou grupos anteriormente não agrupados, ou **divisivos**, começando com todos os objetos em um mesmo grupo e N-1 divisões progressivas em grupos menores. A representação mais natural para agrupamentos hierárquicos corresponde a uma árvore conhecida como **dendograma**, que mostra como os documentos são agrupados. A Figura 14 a seguir apresenta um dendograma para agrupamento de seis documentos [52].

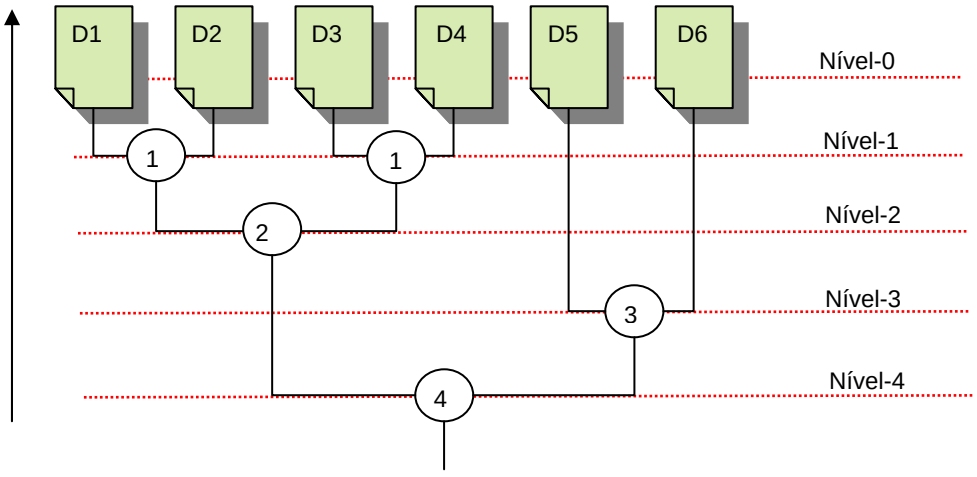

Figura 14 – Agrupamento de não-hierárquico aglomerativo de documentos.

O nível-0 mostra os grupos compostos por cada documento separado, conhecidos como *singleton*. Já no nível-1 se formaram dois grupos, ambos com um par de documentos cujos graus de similaridade entre eles mais se destacava,

um agrupando os documentos D1 e D2, e outro com os documentos D3 e D4. Procede-se dessa forma até que todos os documentos pertençam a um mesmo grupo (método aglomerativo). É possível medir a similaridade dos agrupamentos em cada nível, que vai caindo na medida em que grupos com mais documentos vão sendo formados. Pode-se dizer que no nível-0 a escala de similaridade seja máxima, ou 100 se estivermos tratando em termos percentuais, pois todo documento é 100% similar a si mesmo [52].

A *clusterização* também pode ser utilizada em MT para reduzir o número de *tokens* em uma coleção de documentos, caracterizando o que é chamado de *clustering* de palavras [53]. Desta forma, o que se pretende agrupar não são mais os documentos, e sim as palavras que os constituem, possibilitando a representação de documentos por agrupamentos, e não mais por unidades individuais.

Finalmente, cabe aqui citar que existem diversos algoritmos de sucesso para executar a tarefa de agrupamento, como o ***k-means*** [54] e sua variação nebulosa, o ***fuzzy C-means*** [55].

4.3. Sumarização

Sumarização de documentos é tarefa que trata da redução da massa textual, a fim de se obter ganhos significativos em desempenho quando da busca por informação útil. Também conhecida por **criação automática de resumos**, esta tarefa impõe como desafio a necessidade de se eliminar dados, tanto quanto possível, entretanto, mantendo os significados-chave do texto [20].

A Sumarização de documentos pode ser classificada quanto à natureza do processo de criação, conforme descrito abaixo:

- **Sumarização por Abstração:** É a criação automática de resumos realizada de forma similar àquela feita pelo homem. Dado um texto, o resumo é criado a partir do entendimento do leitor, possivelmente com a inclusão de nova informação, ou seja, novas palavras, sentenças e estilos. Devido à complexidade, a sumarização por abstração foi preterida em favor de outros métodos.

- **Sumarização por Extração:** Esta técnica concentra-se na criação de resumos através da seleção de sentenças e parágrafos principais e importantes, copiados inteiramente do texto original. Baseia-se na medida de importância das palavras de um texto, através da identificação por alguma medida. Em média, apenas 20% de um texto são aproveitados para a criação do resumo com esta técnica. Para que o resumo possa ter o efeito necessário, outras heurísticas podem ser adicionadas, como a identificação da sentença dentro de um documento ou parágrafo, reconhecimento de palavras conclusivas (“portanto”, “definitivamente”, “resumindo”) ou até mesmo de construções conclusivas (“Minha dissertação de mestrado é sobre”).

Outra classificação para os resumos diz respeito à finalidade e ao uso dos resumos criados. Os resumos podem ser, basicamente, de dois tipos:

- **Resumos Indicativos:** Caracterizam-se pela formação do resumo com o mínimo de informação necessária para que, ao lê-lo, o leitor decida por também ler ou não o texto original por completo. Exemplos são manchetes de jornal, resenhas de livros e sinopses de filmes.

- **Resumos Informativos:** Em contrapartida, neste tipo de resumo o leitor não tem a intenção de ler o documento original completo, limitando-se a obter toda informação necessária a partir do próprio resumo.

A tarefa de Sumarização, normalmente, se utiliza de outras duas tarefas para cumprir seu objetivo: classificação e *clusterização*. O problema de decidir se determinada sentença ou parágrafo será incluído no resumo pode ser mapeado para um problema de classificação de sentenças, com o treinamento de um algoritmo como as Redes Neurais ou Classificadores *Bayesianos*. Outra abordagem é a identificação de grupos de sentenças e parágrafos, realizada em conjunto a *clusterização*.

Finalmente, cabe salientar que em um sistema de Mineração de Textos, a criação de resumos de textos é sempre desejada, de forma a ajustar o tamanho da massa textual trabalhada. Isto acelera várias etapas descritas no capítulo 3, como a **Indexação**.

4.4. Extração de Informação

A tarefa de Extração de Informação (EI), também conhecida como Extração de Características, concentra-se na obtenção automática de dados estruturados a partir de dados não-estruturados. O objetivo final é obter o preenchimento de tabelas, também chamadas de *templates*, conforme o exemplo ilustrado na Figura 15.

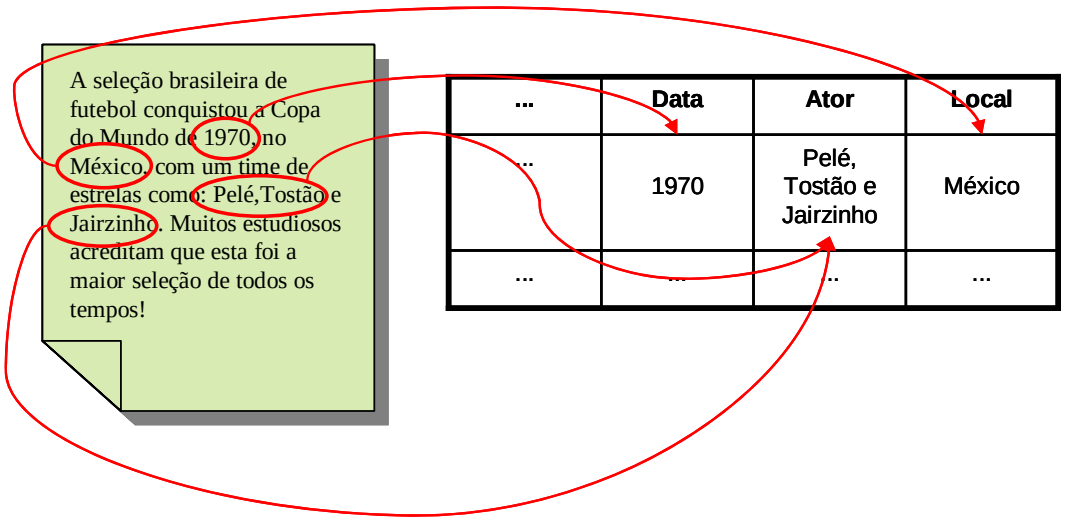

Figura 15 – Extração de Características de um documento.

Na área de EI, os textos podem ser classificados como **estruturados**, **semi-estruturados** e **não-estruturados** (ou livres). Um texto estruturado segue um formato rígido (e.g., páginas HTML geradas a partir de bancos de dados), o que possibilita que a informação seja extraída usando regras baseadas em delimitadores e/ou na ocorrência de termos. Os textos livres contêm, basicamente, sentenças em alguma língua natural, o que inviabiliza a extração com base apenas em formatação. Textos semi-estruturados, por sua vez, apresentam algum grau de estruturação (e.g., referências bibliográficas), juntamente com irregularidades,

como campos ausentes ou com valor nulo, variações na ordem dos dados, e ausências de delimitadores entre as informações a serem extraídas.

O preenchimento destes *templates* permite, dentre outros, a utilização de algoritmos clássicos de Data Mining, obtendo toda a vantagem de se trabalhar no mundo da informação estruturada. Existem diversas técnicas para a construção de sistemas de EI.

A primeira é a utilização de **Processamento de Linguagem Natural** – **PLN** - como forma de "recortar" a informação realmente necessária. Nesse campo, é comum a utilização de **autômatos finitos** [56][57][58] e **expressões regulares** [60]. Em um sistema de Mineração de Textos, técnicas de PLN são também utilizadas, entretanto, em conjunto com outras, conforme visto no capítulo 3, principalmente na etapa de Pré-processamento.

A segunda abordagem é a utilização da tarefa de Classificação para decidir a que classe determinado *token* está associado. Por exemplo, um classificador pode ser treinado para classificar *tokens* em três classes distintas: "número telefônico", "nome de empresa", "endereço" e "outros". Estes classificadores utilizam características do contexto, isto é, verificam o número de aparições no texto, presença de palavras específicas, presença de letras capitalizadas, formatação, etc.

Finalmente, a terceira abordagem baseia-se na utilização de autômatos finitos probabilísticos, utilizando os **Modelos de Markov Escondidos**. Nessa modelagem, um estado oculto é criado para cada campo de saída, e os símbolos emitidos pelos estados ocultos são definidos como os *tokens* do documento. Dada uma seqüência de *tokens*, o HMM determina os estados ocultos associados a cada um desses símbolos, ou seja, que campo de saída cada *token* deverá preencher. O HMM tem a vantagem de realizar uma classificação ótima para a seqüência completa de entrada. Por outro lado, ele não é capaz de fazer uso de múltiplas características dos *tokens* (por exemplo, formatação, tamanho e posição), como ocorre com outros classificadores [61].

4.5.
Sistemas de Busca de Informação

Embora existam autores que citem que um sistema de MT não pode ser considerado apenas o mesmo que um sistema de Recuperação de Informação, considerando a busca por informação útil e de qualidade não uma tarefa objetivo, e sim, uma etapa intermediária conforme a metodologia estudada no presente trabalho, existem outros que preferem considerar a busca por informação como sendo sim uma tarefa de MT, aonde o objetivo é encontrar a informação que se deseja da melhor forma possível, conforme mencionado em [20].

O fato é que a indexação de documentos pode sim ficar mais "inteligente" com a associação de diversas técnicas apresentadas neste trabalho, transformando sistemas simples de busca em sistemas robustos, que não apenas respondam a consultas formuladas pelo usuário, mas atuem de forma **corretiva**, **sugestiva** e **qualitativa**, introduzindo novo conhecimento ao que foi buscado pelo usuário. Um bom exemplo disto é o recurso "você quis dizer" da máquina de buscas do **Google,** que tenta verificar erros de digitação nos parâmetros de entrada em uma consulta formulada pelo usuário. Desta forma, caso um usuário procurasse por documentos que contivessem a palavra digitada "Venezue**j**a" - com um erro ortográfico - a resposta deveria ser "Você quis dizer *Venezuela?"*.