7
Estudos de Caso

7.1.Coleta

A coleção de documentos contém textos jornalísticos do *corpus* citado em 6.6. Os textos foram extraídos dos cadernos de esporte, imóveis, informática, política e turismo. Foram selecionados trezentos documentos de cada caderno, perfazendo um total de 1500 documentos.

7.2.
Treinamento

A metodologia de treinamento que será utilizada é a Validação Cruzada citada em 5.8.2: 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.

7.3.
Resultados

7.3.1.
Tokenização

No *framework*, o início do processo de Categorização é dado pela execução da tarefa *a* que corresponde à etapa de Tokenização (Tabela 15). Essa tarefa fará uma

transformação na apresentação dos dados: recebe os dados com o formato *A0* (conjunto de cadeia de caracteres ou documentos) e os transforma em *A1* (conjunto de *tokens*).

Tabela 15 - Planejamento de ações: Tokenização

Id	Etapa	Tarefa	Antecedente	Tipo Entrada	Tipo Saída
a	*Pré-Processamento*	*Início de etapa*	*N/D*	*N/D*	*N/D*
b	**Pré-Processamento**	**Tokenização**	**a**	***A0***	***A1***

Geralmente, o processo de Tokenização é muito rápido. Porém, a metodologia empregada nesta Tese para a realização desta tarefa é baseada em consultas aos léxicos construídos além de grande processamento estatístico (cálculo do coeficiente de correlação) para automatizar o processo de Identificação de *tokens* multivocabulares, o que torna o processo bastante oneroso.

Para efeito de comparação, a realização da tokenização utilizando apenas o passo *1* da metodologia proposta nesta Tese, que consiste na geração simples de *tokens* baseada no conjunto de *tokens* delimitadores, como espaço e fim de linha, consome em média cinco minutos. Esta é a abordagem utilizada pela maioria dos softwares de MT. A metodologia empregada neste trabalho utiliza aproximadamente oitenta minutos de processamento.

Ao final do processo, onze mil *tokens* são gerados. Pela consulta realizada pelo *framework* ao léxico, trezentas e quinze abreviações foram identificadas e substituídas pelo termo não abreviado correspondente.

Além disso, aproximadamente quatrocentos *tokens* multivocabulares foram encontrados. Como o processo de identificação desses *tokens* é realizado sem intervenção do usuário, utiliza-se limite de 40 no valor de correlação entre termos para que os mesmos sejam unificados em um único termo. Contudo, caso necessário, o usuário poderá selecionar manualmente os termos justapostos que irão ser compilados em um único *token*.

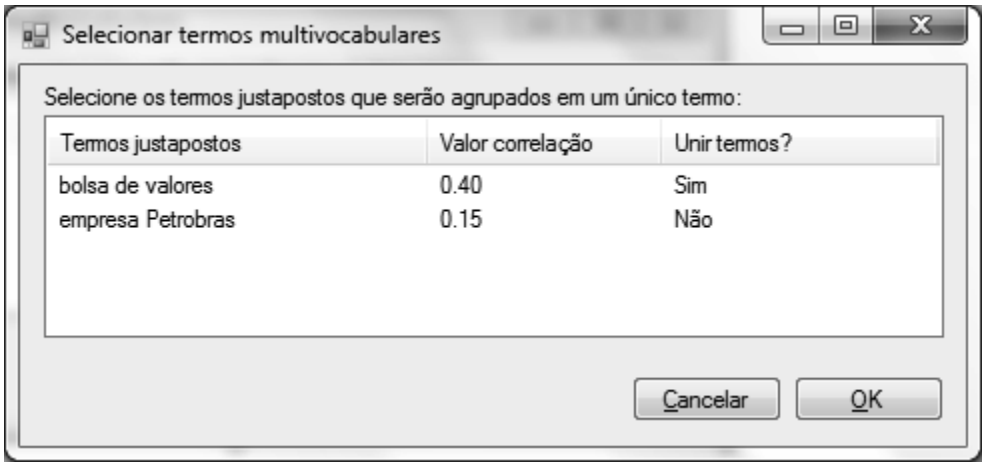


Figura 33 - Exemplo de documento do corpus CETENFolha

A etapa de identificação de números foi responsável pelo reconhecimento de 680 termos que expressam grandezas numéricas. Números e datas na forma extensa são substituídos pela sua representação numérica. A fase de identificação de símbolos de Internet encarregou-se de discriminar cerca de 535 *tokens* distintos. A maioria proveniente do caderno de informática. A etapa de identificação de palavras combinadas contribuiu de forma insignificante para o processo de tokenização.

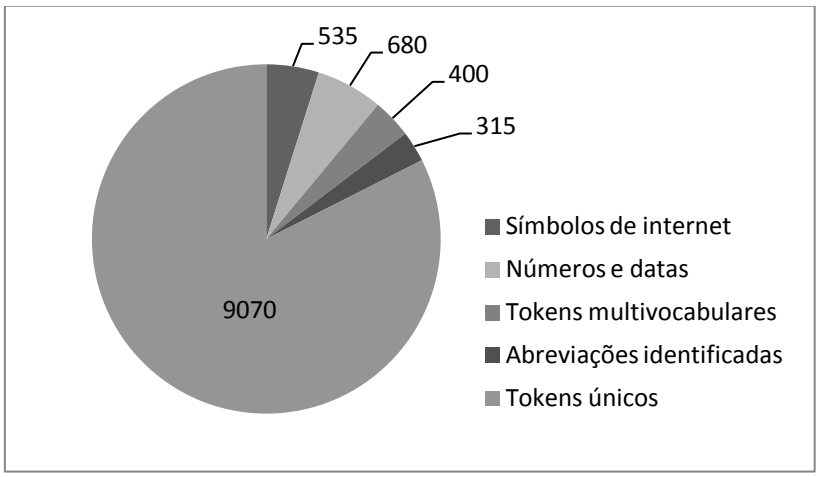


Figura 34 - Exemplo de documento do corpus CETENFolha

O final do processo de tokenização obtém-se a situação ilustrada na Figura 35: um conjunto de documentos em sua representação baseada em *tokens*, também chamada de *bag of words* ou saco de palavras.

Figura 35 - Representação de documentos na forma de *bag of words*

7.3.1. Remoção de *stopwords*

Em seguida, inicia-se o processo de remoção e análise de *stopwords.* Essa fase corresponde às tarefas *c* e *d* da tabela de planejamento de ações (Tabela 14). A tarefa *d* só é executada após a conclusão da tarefa *c* (*c* é antecedente de *d)*. Nessa fase, os documentos textuais já estão segmentados em *tokens* (tipo de saída *A1*), resultado da execução da tarefa *b* (Tokenização), portanto os dois processos, *c* e *d*, recebem como entrada o formato de dados *A1* (Tabela 16).

Tabela 16 - Planejamento de ações: Remoção de *stopwords*

Id	Etapa	Tarefa	Antecedente	Tipo Entrada	Tipo Saída
b	Pré-Processamento	Tokenização	a	*A0*	*A1*
c	**Pré-Processamento**	**Remoção de stopwords**	**b**	***A1***	***A1***
d	**Pré-Processamento**	**Remoção de stopwords (domínio)**	**c**	***A1***	***A1***

Automaticamente, são gerados três subconjuntos de *tokens* resultantes desse processo, conforme ilustrado na Figura 36. Todos são o resultado da utilização das três listas de *stopwords* fornecidas para a coleção: uma com cem termos, a segunda com duzentos e a última com trezentos termos. Cada um desses subconjuntos será mantido para que sejam avaliados pelos métodos de Categorização de Textos posteriormente.

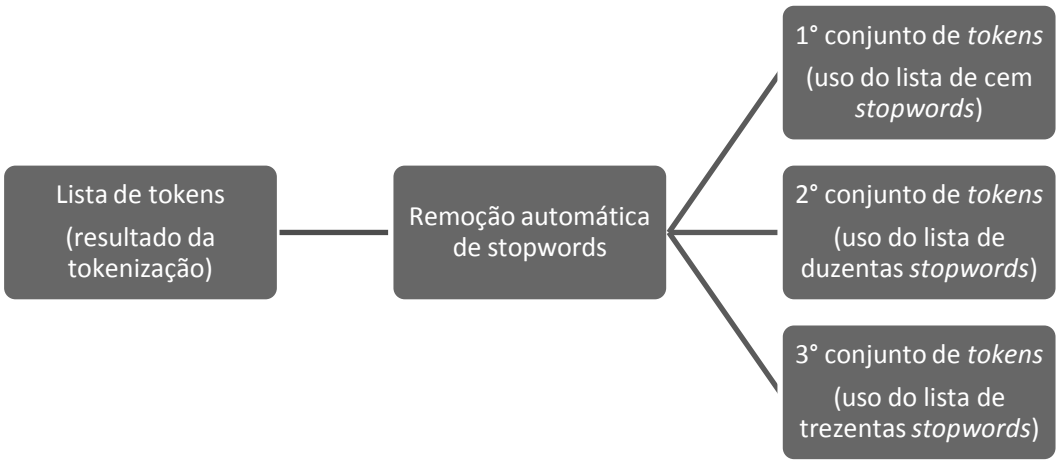


Figura 36 - Resultado do processo de remoção de *stopwords* baseado em listas

Sob os três subconjuntos de *tokens* gerados pela tarefa *c*, será aplicada a tarefa *d*, que consiste na remoção de *stopwords* relacionadas ao domínio. Esse procedimento é necessário, pois muitos termos relacionados ao domínio da aplicação são considerados irrelevantes, pois devido á alta frequência com que estão presentes nos documentos, possuem pouco caráter discriminatório.

Portanto, termos frequentes em mais de oitenta por cento dos documentos ou com frequência menor ou igual a três, devem ser eliminados, segundo (JOACHIMS, 1998).

Ao fim do término da execução das duas tarefas de remoção de *stopwords*, o número de termos foi reduzido substancialmente. O primeiro subconjunto de dados, oriundo da aplicação da lista de cem *stopwords*, possui cerca de 3600 termos. O segundo subconjunto de dados, e o terceiro conjunto, oriundo da aplicação da lista de trezentas *stopwords*, possuem cerca de 3500 termos.

Pode-se concluir que apesar de utilizarem listas de *stopwords* diferentes, todos os subconjuntos de termos foram reduzidos a valores próximos. A execução da tarefa *d*, remoção de *stopwords* do domínio, foi capaz de obter representações semelhantes aos três subconjuntos de dados.

Ao final dessa tarefa, a situação dos documentos é ilustrada na Figura 37. Há, portanto, seis subconjuntos de dados disponíveis para os métodos de Categorização de Textos.

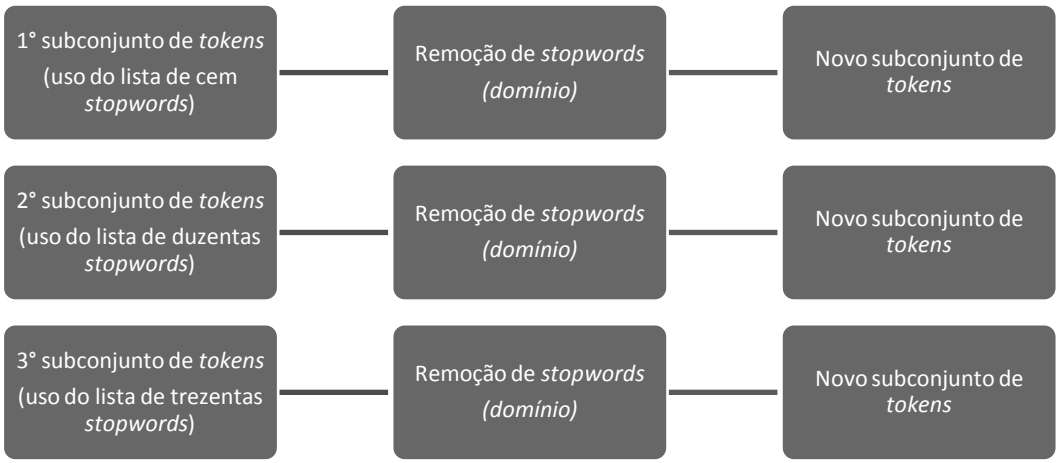


Figura 37 - Resultado do processo de remoção de *stopwords* do domínio

7.3.2. PLN - Identificação de classes gramaticais

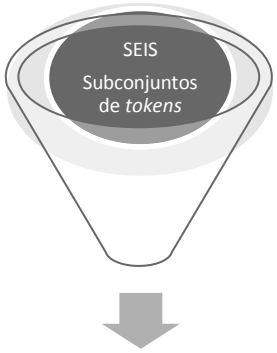
A fase de Processamento de Linguagem Natural é executada. As tarefas *b* e *c* podem ser executadas em paralelo, pois possuem as mesmas tarefas antecedentes (*c* ou *d*). Essas tarefas esperam como entrada o tipo de dados *A1* e ao final do processo irão gerar uma representação dos *tokens* que contém a identificação da classe gramatical a que cada um pertence (formato de dados *A2).* Essas tarefas serão executadas nos seis subconjuntos de dados existentes até o momento (Tabela 17).

Tabela 17 - Planejamento de ações: PLN - Identificação de classes gramaticais

Id	Etapa	Tarefa	Antecedente	Tipo Entrada	Tipo Saída
aa	*PLN*	*Início de etapa*	*N/D*	*N/D*	*N/D*
bb	**PLN**	**POS tagging (verbo/não verbo)**	**b ou c**	***A1***	***A2***
cc	**PLN**	**POS tagging (completo)**	**b ou c**	***A1***	***A2***

A tarefa *bb* distingue apenas *tokens* verbais de não verbais para que todos os verbos sejam reduzidos a sua forma canônica por meio de consulta ao léxico, ignorando os algoritmos tradicionais de *stemming.* A tarefa *cc* irá identificar as dez classes gramaticais ilustradas na Figura 29 para que as tarefas de seleção de

termos possam selecionar os termos com base nas classes gramaticais de cada um. De forma geral, o processo pode ser ilustrado como na Figura 38.



Identificação de classes gramaticais
Verbos e não verbos
Classes gramaticais básicas

Figura 38 - Identificação de classes gramaticais

7.3.3. PLN - Lematização

Após a identificação das classes gramaticais, a etapa de lematização é iniciada. Cinco tarefas, *dd*, *ee*, *ff*, *gg* e *hh*, serão executadas para cumprir esta etapa (Tabela 18). Todas as tarefas irão trabalhar com o formato de dados gerado pela etapa de Identificação de Classes Gramaticais, isto é, o formato *A2*¸ que é uma representação de termos e suas respectivas gramaticais, e irão retornar novamente um conjunto de *tokens* (formato *A1*).

Tabela 18- Planejamento de ações: PLN - Lematização

Id	Etapa	Tarefa	Antecedente	Tipo Entrada	Tipo Saída
aa	*PLN*	*Início de etapa*	*N/D*	*N/D*	*N/D*
dd	**PLN**	**Lematização verbal**	**bb**	***A2***	***A1***
ee	**PLN**	**Lematização PORTER**	**bb**	***A2***	***A1***
ff	**PLN**	**Lematização PORTER**	**bb**	***A2***	***A1***
gg	**PLN**	**Lematização LOVINS**	**bb**	***A2***	***A1***
hh	**PLN**	**Lematização RSLP**	**bb**	***A2***	***A1***

A tarefa *dd*, isto é, a Lematização verbal, será responsável somente pela lematização dos verbos e, portanto só possui como antecedente a tarefa *bb*. Desta forma, todo *token* diferente de verbo será ignorado. *Tokens* verbais serão lematizados por consulta ao léxico. Esse processo está ilustrado na Figura 39. A execução da tarefa *dd* irá gerar um novo subconjunto de *tokens* que possuem verbos reduzidos à sua forma canônica.

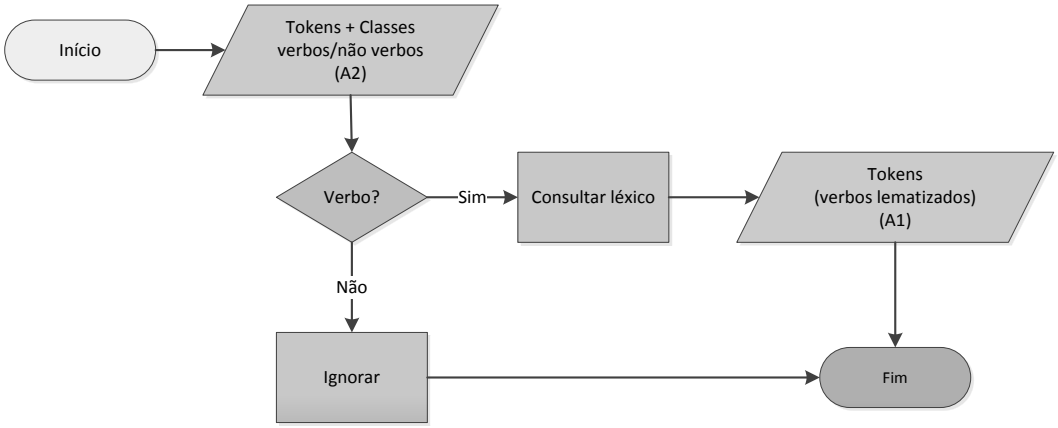


Figura 39 - Fluxograma da lematização verbal

As tarefas *ee*, *ff*. *gg* e *hh*, serão responsáveis pela lematização de todos os termos não verbais. Possuem como antecedente a tarefa *bb* que é responsável pela identificação de termos verbais e não verbais, pois termos verbais não deverão ser lematizados por esses algoritmos. Esse processo está ilustrado na Figura 40 em que o processo *Lematizar* envolve a aplicação dos quatro algoritmos de lematização implementados no *framework*. Desta forma, ao final desse processo haverá quatro novos subconjuntos de *tokens* gerados por cada um dos algoritmos.

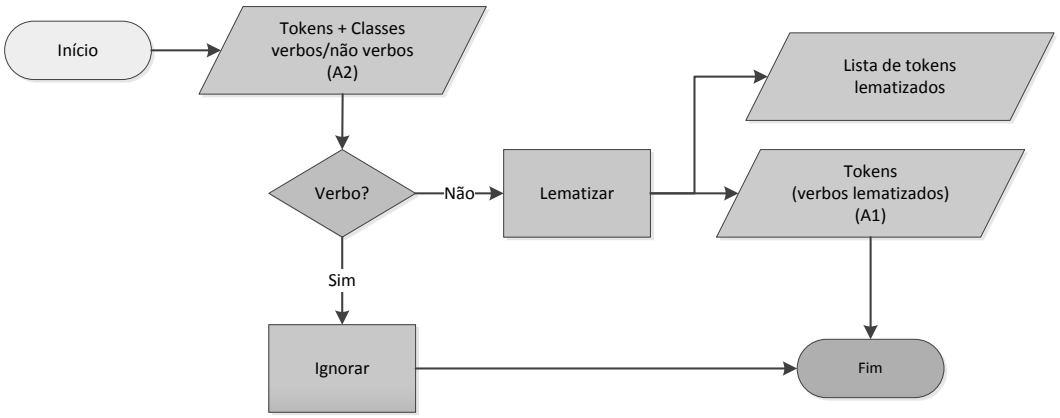


Figura 40 - Fluxograma da lematização não verbal

Além dos *tokens* substituídos pela forma canônica ou lematizados por um dos algoritmos de *stemming* também é gerada uma lista de *tokens* que foram modificados. Essa lista é necessária para que seja possível identificar todos os *tokens* que sofrerão alguma transformação. A cada um dos quatro subconjuntos de *tokens* lematizados pelos algoritmos serão incluídos os verbos substituídos, já que esses algoritmos ignoraram os verbos durante a sua execução.

Ao fim de todo o processo de lematização, há, portanto, cinco novas opções de subconjuntos (Lematização verbal, Porter, Stemmer S, Lovins e RSLP) para cada um dos seis subconjuntos iniciais.

7.3.4. Thesaurus

A tarefa *ii* compreende a execução desse processo em que termos com o mesmo valor semântico são identificados e substituídos por um termo preferencial (Tabela 19). O léxico construído durante esta Tese e a base de Thesaurus Eletrônico para o Português do Brasil são utilizados.

Tabela 19- Planejamento de ações: PLN - Thesaurus

Id	Etapa	Tarefa	Antecedente	Tipo Entrada	Tipo Saída
Aa	*PLN*	*Início de etapa*	*N/D*	*N/D*	*N/D*
ii	**PLN**	**Thesaurus**	**dd ou ee ou ff ou gg ou hh**	***A1***	***A1***

Os dicionários utilizados, além dos termos na forma original, contém uma representação dos mesmos lematizados segundo cada um dos métodos utilizados na etapa anterior para que seja possível encontrá-los. Essa etapa foi responsável, em média, pela substituição de vinte por cento dos *tokens*, conforme ilustrado na Figura 41.

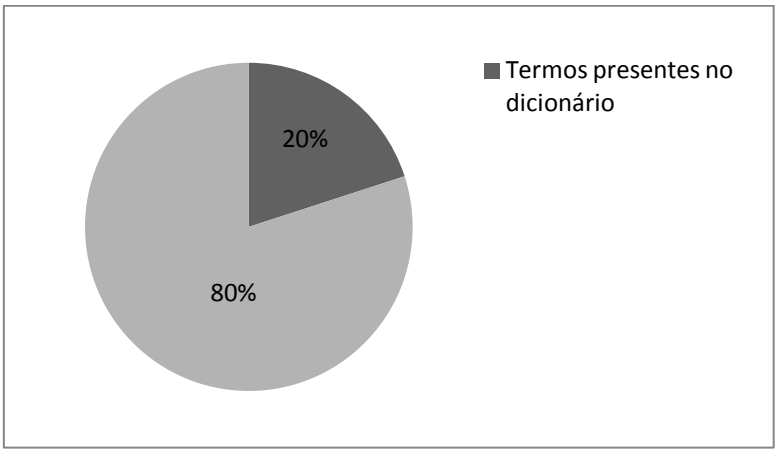


Figura 41 - Substituição de termos por consulta ao Thesaurus

Ao fim de todo o processo de Thesaurus, há, portanto, duas novas opções de subconjuntos (com ou sem uso de Thesaurus) para cada um dos trinta subconjuntos do início do processo.

7.3.5. Seleção de características

Em seguida, inicia-se o processo de seleção ou redução de características. Essa fase corresponde às tarefas *jj*, *kk*, *ll* e *mm* da tabela de planejamento de ações (). Essas tarefas visam diminuir a alta dimensionalidade dos modelos de representação de documentos.

A tarefa *mm*, seleção de características POS (*Part of Speech*), irá fazer a seleção de características em que a informação linguística define a importância dos termos. Utiliza a abordagem de seleção de características baseada em padrões morfossintáticos, ou seja, utiliza as classes gramaticais dos termos para fazer a seleção do que será considerado na representação reduzida dos documentos.

Portanto, esta tarefa é obrigatoriamente executada após a conclusão da tarefa *cc* que é a identificação de classes gramaticais. Logo, a tarefa *mm* possui como entrada, o modelo de dados *A2* que é o modelo que fornece os *tokens* associados às suas respectivas classes gramaticais. Além disso, na elaboração do planejamento de tarefas do *framework*, incluiu-se também que esta tarefa aguarde a execução de pelo menos uma das tarefas de lematização (tarefas de *dd* ou *ee* ou *ff* ou *gg* ou *hh*), conforme representado na Tabela 20.

As tarefas restantes (*kk, ll* e *mm)* irão fazer a extração de características baseadas em critérios estatísticos (item 4.3.1.3), a saber: TF-IDF, Ganho de Informação e Escore de Relevância. Podem ser iniciadas tão logo a tokenização seja concluída, porém no planejamento de ações do *framework*, optou-se por permitir a execução dessas tarefas somente após a execução dos métodos de lematização. (tarefas de *dd* ou *ee* ou *ff* ou *gg* ou *hh).*

Tabela 20- Planejamento de ações: Seleção de características

Id	Etapa	Tarefa	Antecedente	Tipo Entrada	Tipo Saída
aa	*PLN*	*Início de etapa*	*N/D*	*N/D*	*N/D*
jj	**PLN**	**Seleção de características: TF/IDF**	**dd ou ee ou ff ou gg ou hh**	***A1***	***A1***
kk	**PLN**	**Seleção de características: Ganho de Informação**	**dd ou ee ou ff ou gg ou hh**	***A1***	***A1***
ll	**PLN**	**Seleção de características: Escore de relevância**	**dd ou ee ou ff ou gg ou hh**	***A1***	***A1***
mm	**PLN**	**Seleção de características: POS**	**cc e (dd ou ee ou ff ou gg ou hh)**	***A2***	***A1***

A tarefa de seleção de características POS utilizará a combinação de classes gramaticais, como em (CAMARGO, 2007), para escolher os termos. Serão formados sete subconjuntos constituídos de: substantivo, substantivo + adjetivo, substantivo + nome próprio, substantivo + verbo, substantivo + verbo + adjetivo, substantivo + nome próprio + adjetivo, nome próprio + adjetivo e verbo.

Os resultados estatísticos obtidos pelas tarefas *kk*, *ll* e *mm* serão utilizados em ordem decrescente para determinar os termos que irão fazer parte dos modelos reduzidos de representação dos documentos.

Assim que concluído todo o processo de cálculo das métricas, é necessário definir a quantidade de termos que será utilizada na representação reduzida dos documentos para cada uma das quatro tarefas de seleção de características. Há dois critérios para isso: seleção global e seleção local. Esse processo está ilustrado na Figura 42.

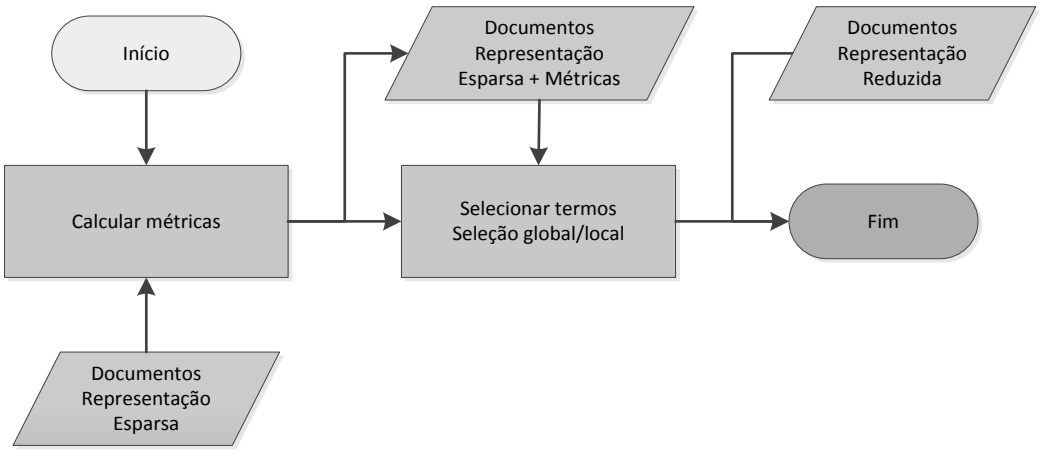


Figura 42 - Seleção de características

A seleção global compreende a escolha dos termos mais representativos de toda a coleção segundo as métricas calculadas nos processos de seleção de características: TF/IDF, Ganho de Informação e Escore de Relevância. Os experimentos utilizam os 30, 60, 90, 120 e 150 termos mais relevantes, conforme sugerido em (SEBASTIANI, 2002) e (JOACHIMS, 1998).

A seleção local compreende a escolha dos termos mais representativos de cada categoria segundo as mesmas métricas calculadas no processo de seleção de características: TF/IDF, Ganho de Informação e Escore de Relevância. Foram selecionados os 6, 12, 18, 24 e 30 termos mais relevantes de cada categoria; como há cinco categorias, serão construídas ao final da seleção dos termos mais relevantes da cada categoria representações de documentos com os seguintes números de termos: 30, 60, 90, 120 e 150.

Ao fim de todo o processo de seleção de características, há, portanto, para cada um dos três métodos baseados em estatísticas para seleção de características (seleção por TF/IDF, Ganho de Informação, Escore de Relevância) dez .novos subconjuntos de dados: cinco conjuntos formados por seleção global e cinco formados por seleção local. O método baseado em classes gramaticais para

seleção de características possui também dez novos subconjuntos de dados, formados por seleção global e seleção local, para cada uma das sete combinações de classes gramaticais.

7.3.6. Mineração

Compreende a execução das tarefas *bbb, cccc e dddd*, ou seja, a aplicação dos algoritmos classificadores *k*-NN, SVM e Bayesiano (Tabela 21) . Os conjuntos de dados que serão utilizados para treinar os classificadores foram definidos na etapa anterior.

Tabela 21- Planejamento de ações: Mineração

Id	Etapa	Tarefa	Antecedente	Tipo Entrada	Tipo Saída
aaaa	*Mineração*	*Início de etapa*	*N/D*	*N/D*	*N/D*
bbbb	**Mineração**	**KNN**	**jj ou kk ou ll ou mm**	***A3***	***A4***
cccc	**Mineração**	**SVM**	**jj ou kk ou ll ou mm**	***A3***	***A4***
dddd	**Mineração**	**Classificador Bayesiano**	**jj ou kk ou ll ou mm**	***A3***	***A4***
zzzz	*Mineração*	*Fim de etapa*	*N/D*	*N/D*	*N/D*

É nesta etapa também que os algoritmos serão ajustados para obter o melhor desempenho em cada uma das representações obtidas ao término da etapa de PLN. A Tabela 22 exibe a modelagem das características de execução do algoritmo SVM e a Tabela 23 exibe a modelagem referente ao algoritmo *k*-NN.

Tabela 22 - Configurações de execução do algoritmo SVM

SVM	
Parâmetros	Tipo
Parâmetro C	Numérico
Função	Função

Valor mínimo	Valor máximo	Valor inicial	Incremento	Decremento
0,1	100	1	x 10	/ 10

Função	Definição	Ordem execução
1. Linear	...	1
2. Polinomial	...	2
3. RBF	...	3

Tabela 23 - Configurações de execução do algoritmo *k*-NN

KNN	
Parâmetros	Tipo
Parâmetro k	Numérico
Distância	Função

Valor mínimo	Valor máximo	Valor inicial	Incremento	Decremento
1	15	5	+2	-2

Função	Definição	Ordem execução
1. Euclidiana	...	1
2. Cosseno	...	2
3. Jaccard	...	3
4. Manhattan	...	4

Há disponível para treinamento dos classificadores seis mil subconjuntos de dados, conforme ilustrado na Figura 43. Encontrar o melhor subconjunto para um determinado classificador é desejável, pois além da melhoria da eficácia do classificador, maior facilidade de compreensão e visualização das características representativas dos documentos é obtida.

Figura 43 - Subconjuntos disponíveis

Escolher a melhor representação de características pode ser visualizado como um problema de busca. A Figura 44 mostra um diagrama de estados com quatro características (CHAGAS, 2009). Cada estado determina um subconjunto de características escolhido em um determinado instante. O círculo branco indica a ausência de uma determinada característica, enquanto que o círculo preto indica a presença. No contexto do *framework*, cada coluna representa uma etapa do processo de MT em que se faz uso, ou não, das técnicas disponíveis para a etapa.

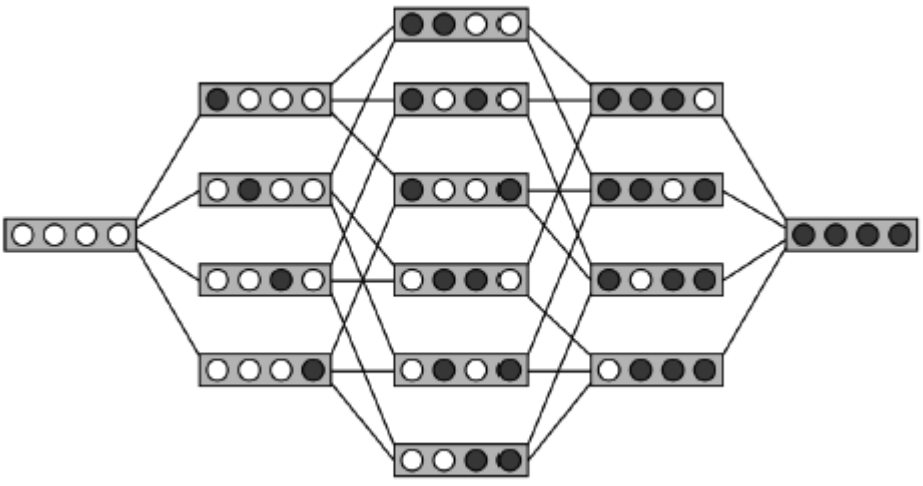


Figura 44 - Diagrama de estados

A busca completa ou exaustiva possibilita avaliar subconjuntos ótimos. Para isso, todos os subconjuntos de características possíveis são avaliados. Entretanto, em muitos casos, o espaço de busca é grande demais para ser explorado exaustivamente tornando esse algoritmo computacionalmente intratável (essa solução é NP-Completa). Nessas situações, algoritmos subótimos devem ser utilizados para encontrar a melhor solução possível.

A base de conhecimentos sobre Categorização de Textos construída para o *framework* dispõe de vinte e três caminhos do espaço de busca (soluções) que podem ser considerados subótimos. Ao iniciar uma Categorização de Textos, esses caminhos são avaliados. Baseado nos melhores resultados obtidos por essas soluções, a cada processo de treinamento de um novo classificador, características são acrescentadas ou substituídas até que um determinado critério de parada seja atingido. Um critério de parada pode ser, por exemplo, um valor *k* que determine quantas novas soluções alternativas serão avaliadas.

7.3.6.1. Resultados

Abaixo são apresentados os melhores resultados obtidos pelo *framework* no processo de Categorização Automática de Textos, isto é, a atribuição de cada notícia a um dos cinco cadernos do corpus. A Tabela 24 exibe os valores obtidos. Foram realizadas cinquenta e três tentativas (critério de parada *k = 30*) de encontrar a melhor representação dos documentos no espaço de busca. A melhor alternativa foi encontrada após a realização de trinta e cinco experimentos.

Tabela 24 - Configuração do melhor resultado obtido

Termos	Métrica de seleção	Número de termos	% Erro
Substantivo + Nome próprio + Adjetivo	Ganho de informação	150	5,20

O resultado obtido utiliza a combinação de termos que teve o melhor resultado em (CAMARGO, 2007) e a métrica de seleção que conseguiu o

desempenho mais alto em (CHAGAS, 2009). A solução de melhor desempenho é ilustrada na Figura 45. Houve utilização de todas as técnicas de tratamento linguístico fornecidas pelo *framework*.

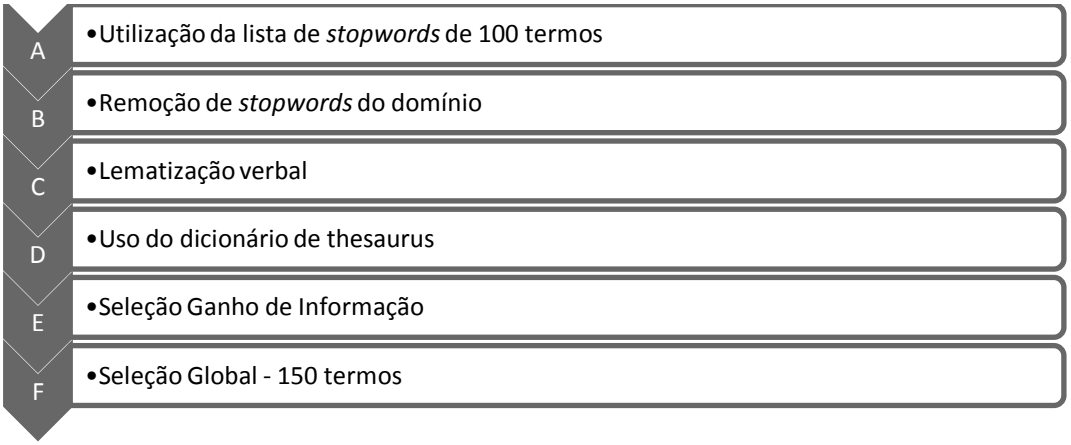


Figura 45 - Solução de melhor desempenho

Das cinco categorias, três foram categorizadas por SVM e as duas restantes pelo *k*-NN. A Tabela 25 exibe a relação "classificador x categoria".

Tabela 25 - Relação categoria x classificador

Categoria	Classificador
Esportes	SVM
Imóveis	SVM
Informática	KNN
Política	KNN
Turismo	SVM

Para as duas categorias que foram categorizadas pelo *k*-NN, obteve-se o valor de *k = 15*. Esse era o valor máximo definido para *k*; talvez, seja necessário ampliar a faixa de valores de *k*.

Mudando apenas o técnica de lematização verbal, verifica-se que a lematização verbal obteve o melhor resultado seguida pela lematização RSLP, conforme ilustrado na Figura 46.

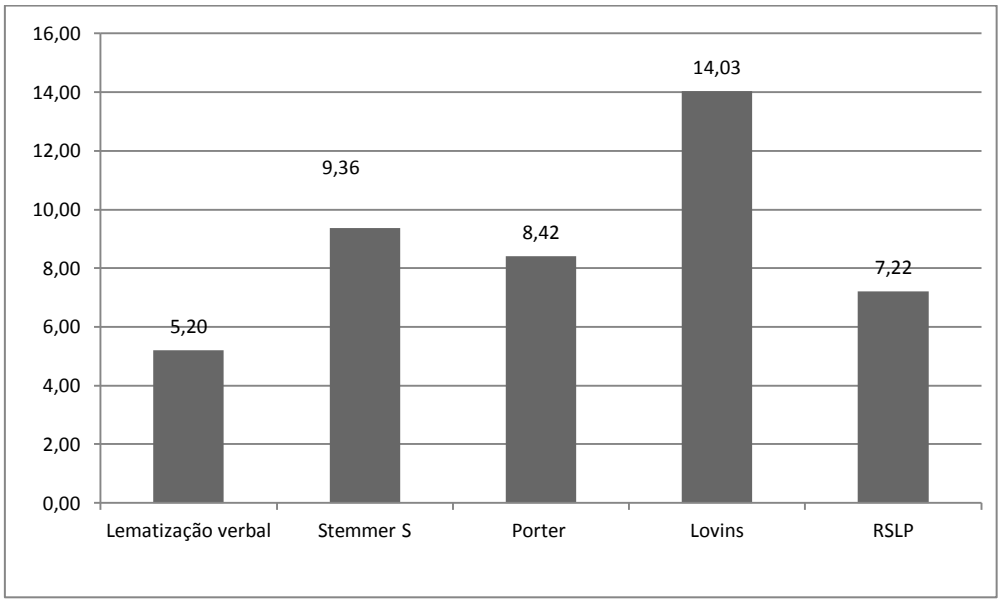


Figura 46 - Desempenho lematização

A utilização da lista de *stopwords* de 100, 200 ou 300 termos não exerceu grande influência nos resultados obtidos quando é acompanhada da remoção de *stopwords* do domínio. A consulta ao dicionário de Thesaurus incrementou o resultado de todos os experimentos quando passou a ser utilizado.

O resultado atingido (5,20%) supera o obtido em (CAMARGO, 2007): 7,49%.