Rogério César Peixoto Fragoso

Algoritmos de Seleção de Características Personalizados por Classe para Categorização de Texto

Recife
2016

Rogério César Peixoto Fragoso

Algoritmos de Seleção de Características Personalizados por Classe para Categorização de Texto

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre Profissional em 26 de agosto de 2016.

Universidade Federal de Pernambuco – UFPE

Centro de Informática

Programa de Pós-Graduação

Orientador: George Darmiton da Cunha Cavalcanti

Recife

2016

Catalogação na fonte
Bibliotecário Jefferson Luiz Alves Nazareno CRB 4-1758

F811a Fragoso, Rogério César Peixoto.
Algoritmos de seleção de características personalizados por classe para categorização de texto / Rogério César Peixoto Fragoso – 2016.
74f.: fig., tab.

Orientador: George Darmiton da Cunha Cavalcanti.
Dissertação (Mestrado profissional) – Universidade Federal de Pernambuco. CIn. Ciência da Computação, Recife, 2016.
Inclui referências e apêndices.

1. Inteligência artificial. 2. Mineração de dados. 3. Aprendizagem de máquina. I. Cavalcanti, George Darmiton da Cunha. (Orientador). II. Titulo.

006.3 CDD (22. ed.) UFPE-MEI 2017-007

Rogério César Peixoto Fragoso

Algoritmos de Seleção de Características Personalizados por Classe para Categorização de Texto

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre Profissional em 26 de agosto de 2016.

Aprovado em 26/08/2016

BANCA EXAMINADORA

Prof. Dr. Frederico Luiz Gonçalves de Freitas
Centro de Informática / UFPE

Prof. Dr. Rafael Ferreira Leite de Melo
Universidade Federal Rural de Pernambuco

Prof. Dr. George Darmiton da Cunha Cavalcanti
Centro de Informática / UFPE
(Orientador)

Resumo

A categorização de textos é uma importante ferramenta para organização e recuperação de informações em documentos digitais. Uma abordagem comum é representar cada palavra como uma característica. Entretanto, a maior parte das características em um documento textual são irrelevantes para sua categorização. Assim, a redução de dimensionalidade é um passo fundamental para melhorar o desempenho de classificação e reduzir o alto custo computacional inerente a problemas de alta dimensionalidade, como é o caso da categorização de textos. A estratégia mais utilizada para redução de dimensionalidade em categorização de textos passa por métodos de seleção de características baseados em filtragem. Métodos deste tipo exigem um esforço para configurar o tamanho do vetor final de características. Este trabalho propõe métodos de filtragem com o intuito melhorar o desempenho de classificação em comparação com os métodos atuais e de tornar possível a automatização da escolha do tamanho do vetor final de características. O primeiro método proposto, chamado *Category-dependent Maximum f Features per Document-Reduced* (cMFDR), define um limiar para cada categoria para determinar quais documentos serão considerados no processo de seleção de características. O método utiliza um parâmetro para definir quantas características são selecionadas por documento. Esta abordagem apresenta algumas vantagens, como a simplificação do processo de escolha do subconjunto mais efetivo através de uma drástica redução da quantidade de possíveis configurações. O segundo método proposto, *Automatic Feature Subsets Analyzer* (AFSA), introduz um procedimento para determinar, de maneira guiada por dados, o melhor subconjunto de características dentre um número de subconjuntos gerados. Este método utiliza o mesmo parâmetro usado por cMFDR para definir a quantidade de características no vetor final. Isto permite que a busca pelo melhor subconjunto tenha um baixo custo computacional. O desempenho dos métodos propostos foram avaliados nas bases de dados *WebKB*, *Reuters*, *20 Newsgroup* e *TDT2*, utilizando as funções de avaliação de características *Bi-Normal Separation*, *Class Discriminating Measure* e *Chi-Squared Statistics*. Os resultados dos experimentos demonstraram uma maior efetividade dos métodos propostos em relação aos métodos do estado da arte.

Palavras-chave: Seleção de características. Redução de dimensionalidade. Categorização de textos.

Abstract

Text categorization is an important technic to organize and retrieve information from digital documents. A common approach is to represent each word as a feature. However most of the features in a textual document is irrelevant to its categorization. Thus, dimensionality reduction is a fundamental step to improve classification performance and diminish the high computational cost inherent to high dimensional problems, such as text categorization. The most commonly adopted strategy for dimensionality reduction in text categorization undergoes feature selection methods based on filtering. This kind of method requires an effort to configure the size of the final feature vector. This work proposes filtering methods aiming to improve categorization performence comparing to state-of-the-art methods and to provide a possibility of automitic determination of the size of the final feature set. The first proposed method, namely *Category-dependent Maximum f Features per Document-Reduced* (cMFDR), sets a threshold for each category that determines which documents are considered in feature selection process. The method uses a parameter to arbitrate how many features are selected per document. This approach presents some advantages, such as simplifying the process of choosing the most effective subset through a strong reduction of the number of possible configurations. The second proposed method, *Automatic Feature Subsets Analyzer* (AFSA), presents a procedure to determine, in a data driven way, the most effective subset among a number of generated subsets. This method uses the same parameter used by cMFDR to define the size of the final feature vector. This fact leads to lower computational costs to find the most effective set. The performance of the proposed methods was assessed in *WebKB*, *Reuters*, *20 Newsgroup* and *TDT2* datasets, using *Bi-Normal Separation*, *Class Discriminating Measure* and *Chi-Squared Statistics* feature evaluations functions. The experimental results demonstrates that the proposed methods are more effective than state-of-art methods.

Keywords: Text categorization. Dimensionality reduction. Feature selection.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AFSA	*Automatic Features Subsets Analyzer*
ALOFT	*At Least One FeaTure*
BNS	*Bi-Normal Separation*
BoW	*Bag of Words*
CDM	*Class Discriminating Measure*
CHI	*Chi-Squared*
cMFDR	*Category-dependent Maximum f Features per Document-Reduced*
DR	Dimensionality Reduction
FEF	*Feature Evaluation Function*
FS	*Feature Selection*
IR	*Information Retrieval*
k-NN	*k-Nearest Neighbors*
MFD	*Maximum f Features per Document*
MFDR	*Maximum f Features per Document-Reduced*
ML	*Machine Learning*
SVM	*Support Vector Machine*
TC	*Text Categorization*
TCS	*Text Categorization System*
TF-IDF	*Term Frequency-Inverse Document Fequency*
VR	*Variable Ranking*

Lista de símbolos

$\mathcal{C}$	Conjunto das categorias
c_j	Categoria de índice j
$\mathcal{D}$	Base de dados com todos os documentos
$\mathcal{D}_{tr}$	Base de dados com os documentos de treinamento
$\mathcal{D}_{tt}$	Base de dados com os documentos de teste
$\mathcal{D}_{val}$	Base de dados com os documentos de validação
d_i	Documento de índice i
f	Quantidade de características selecionadas por documento
h	Índice dos termos
i	Índice de um documento
j	Índice de uma categoria
m	Quantidade de características selecionadas por um método
Q	Total de categorias
U	Total de documentos
V	Tamanho do vocabulário
w	Um termo
w_h	Um termo de índice h
$w_{h,i}$	Um termo de índice h no documento de índice i

Sumário

1 Introdução

A evolução e o barateamento das tecnologias de armazenamento e da internet têm causado um grande aumento da disponibilidade de documentos e informações em meios eletrônicos. Diante do aumento do volume de documentos de texto na forma digital, surge a necessidade de sistemas computacionais para auxiliar na organização de conteúdo e na obtenção da informação desejada de maneira rápida e eficaz. Uma solução é dada pela Categorização de texto (TC, do inglês *Text Categorization* ou *Text Classification*). TC consiste em designar documentos de texto em linguagem natural a categorias pertencentes a um conjunto predefinido de categorias (SEBASTIANI, 2002).

A abordagem mais adotada para categorização automática de texto é a utilização algoritmos de aprendizagem de máquina (ML, do inglês *Machine Learning*) (SEBASTIANI, 2002). Esta abordagem prescinde de um conjunto de instâncias previamente categorizadas, sobre o qual um algoritmo indutor realiza o aprendizado. Um instância é um exemplo ou um caso concreto de uma entidade passível de categorização. No caso da categorização de textos, uma instância é um documento textual. Cada instância é composta por características, que são atributos que descrevem as instâncias e pela informação da categoria à qual a instância pertence. As características assumem valores variados nas instâncias. O conjunto de valores que as características assumem é o que diferencia as instâncias. No caso de TC, as características podem ser palavras, sequências de palavras ou letras, frases ou elementos semânticos, dependendo da representação utilizada (este assunto é tratado em mais detalhes na Seção 2.1.1). O algoritmo indutor aprende os padrões de valores assumidos pelas características que descrevem as categorias com a finalidade de determinar a categoria de um documento novo, ou seja, sem informação sobre categoria.

1.1 Motivação

Na abordagem de aprendizagem de máquina, uma instância é representada como um vetor composto por pares de característica e valor. Para a representar documentos textuais como de vetores de características, os termos que compõem os documentos são considerados características. Uma abordagem comum para a representação de textos na forma de vetores de características é a técnica conhecida como *Bag of Words* (BoW) (FENG et al., 2015; GUYON; ELISSEEFF, 2003; JOACHIMS, 1996; SALTON; YANG, 1973). Nela, um texto é tratado como um conjunto de palavras, sem considerar gramática ou ordem de ocorrência das palavras no texto. Cada palavra do vocabulário da base de dados é considerada uma característica e é associada à frequência desta palavra no documento. Ou seja, o tamanho do vocabulário da base de dados define a dimensionalidade dos vetores. Desta forma,

em uma base de dados de tamanho médio, é comum que os vetores de características contenham dezenas de milhares de dimensões (GABRILOVICH; MARKOVITCH, 2004). Entretanto, apesar da grande quantidade de características nos documentos, a maior parte delas é irrelevante ou redundante. A alta dimensionalidade (YU; LIU, 2003) pode tornar a categorização de textos muito dispendiosa em termos de memória e tempo de execução - ou até inviável para algumas abordagens de aprendizagem de máquina. Além de causar uma maior demanda de recursos computacionais, este grande número de características pode impactar negativamente no desempenho de classificação, especialmente em bases de dados com um número pequeno de instâncias em relação ao número de características, fenômeno conhecido como "praga da dimensionalidade" (JAIN; DUIN; MAO, 2000). Como grande parte das características são irrelevantes para a categorização, estes problemas podem ser tratados através da restrição da quantidade de características do conjunto de dados. Esta abordagem é conhecida como Redução de Dimensionalidade (DR, do inglês *Dimensionality Reduction*). Por meio da simplificação do espaço de características, DR busca aumentar a eficiência e a precisão da categorização de textos.

As duas principais técnicas de DR são a extração de características e seleção de características. Este trabalho foca na estratégia de seleção de características (FS, do inglês *Feature Selection)*. Nesta abordagem, o conjunto final é formado por parte das características do conjunto original. A utilização de métodos de filtragem é a técnica de FS considerada mais adequada para problemas de TC, devido ao custo computacional ser bem mais baixo que o de outras técnicas, como métodos *wrapper* (YU; LIU, 2003). Métodos de filtragem realizam um ordenamento das características através do uso de algoritmos determinísticos e métricas estatísticas, conhecidas com funções de avaliação de características (FEF, do inglês *Feature Evaluation Function*). Após o ordenamento, uma quantidade m, estabelecida pelo usuário, de características é selecionada para a formação do novo subconjunto.

Uma dificuldade enfrentada por métodos de filtragem é a determinação do melhor valor para o parâmetro m. Uma busca exaustiva para encontrar o valor ótimo de m levaria a um aumento do esforço computacional. Então, normalmente apenas alguns valores para m são avaliados (YANG; PEDERSEN, 1997). Outra limitação desta abordagem é sua dependência em relação à FEF adotada para realizar a avaliação das características. A precisão da seleção é influenciada pela adequação do conjunto de dados ao método de avaliação das características escolhido.

1.2 Objetivos

Esta seção descreve os objetivos deste trabalho. Abaixo são apresentados o objetivo geral e os objetivos específicos.

1.2.1 Objetivos Gerais

O objetivo deste trabalho é propor métodos de seleção de características para problemas de categorização de textos que determinem de maneira guiada por dados a quantidade de características selecionadas. Ou seja, o algoritmo deve determinar o tamanho do vetor final de características de acordo com a natureza dos documentos de texto. Adicionalmente, os métodos propostos devem apresentar desempenho melhor que os métodos do estado da arte.

1.2.2 Objetivos Específicos

- Desenvolver métodos de seleção de características para categorização de textos;
- Executar experimentos para validar os métodos propostos e avaliar seus desempenhos frente aos métodos do estado da arte;
- Publicar os resultados da pesquisa.

1.3 Organização da Dissertação

Além deste capítulo, que introduziu o problema e as motivações, esta dissertação contém mais 5 capítulos. No Capítulo 2, são apresentados os conceitos fundamentais sobre categorização de texto e a arquitetura de um sistema de categorização de textos típico. O Capítulo 3 descreve o problema da seleção de características e apresenta os métodos do estado da arte. No Capítulo 4, os métodos propostos são apresentados e detalhados. O Capítulo 5 descreve as configurações dos experimentos realizados e exibe os resultados dos mesmos. No Capítulo 6 são feitas as considerações finais, indicando as contribuições do trabalho e os potenciais trabalhos futuros.

2 Categorização de textos

De maneira geral, categorização é o ato de agrupar objetos de um determinado universo, com características semelhantes, em categorias. A tarefa de categorização de textos consiste em identificar se um documento de texto em linguagem natural pertence ou não a categorias de um conjunto predefinido de categorias, com base nas características do documento. De uma maneira mais formal, sejam d_1, d_2, ..., d_U documentos de uma base de dados $\mathcal{D}$ e seja $\mathcal{C} = \{c_1, c_2, ...,c_Q\}$ um conjunto de categorias. TC é a tarefa de designar categorias do conjunto $\mathcal{C}$ aos documentos da base de dados $\mathcal{D}$.

Categorização de textos faz parte de um conjunto de tarefas de manipulação de documentos baseadas em conteúdo, conhecidas como recuperação de informação (IR, do inglês *information retrieval*). A categorização de textos é uma importante técnica utilizada para auxiliar na organização e disseminação de conteúdo e na obtenção da informação. O surgimento desta área de pesquisa remonta ao início dos anos 1960 (MARON, 1961). Inicialmente os sistemas de categorização de textos (TCS, do inglês *text categorization systems*) eram baseados em engenharia do conhecimento, ou seja, especialistas codificavam um conjunto de regras que era utilizado para categorizar os documentos textuais. A partir dos anos 1990 surgiram sistemas baseados em aprendizagem de máquina (ML, do inglês *machine learning*). Assim, nesta abordagem um classificador é construído automaticamente utilizando um algoritmo indutivo e um conjunto de dados de treinamento que contém instâncias previamente classificadas. Segundo (SEBASTIANI, 2002), o uso da aprendizagem de máquina representa uma grande redução de custos – já que não há necessidade de intervenção humana para a construção do classificador – ao passo que atinge um nível de precisão semelhante àquele obtido por especialistas.

A categorização de textos enquadra-se no paradigma de aprendizado supervisionado de máquina. Neste paradigma, o conjunto de possíveis categorias é previamente conhecido e os documentos podem ser etiquetados com uma categoria, abordagem *single-label*, ou mais de uma categoria, abordagem *multi-label*, a depender da necessidade da aplicação. O domínio da classificação pode ser binário ou multi-classe. No domínio binário, uma instância pode pertencer a uma de duas categorias possíveis, a classe c_i ou sua classe complementar, $\bar{c}_i$. Por razões óbvias, este domínio de classificação só se adéqua à abordagem *single-label*. No caso do domínio, multi-classe, existem mais de duas categorias. Ambas as abordagens *single-label* e *multi-label* podem ser utilizadas no domínio multi-classe (SEBASTIANI, 2002).

2.1 Arquitetura Geral

Um sistema de categorização de textos típico compreende as etapas de Representação de documento, Construção de classificador e Categorização. O objetivo da categorização de textos é predizer corretamente a categoria de um documento. Uma categoria pode ser identificada por um padrão de valores dos atributos descritores dos documentos pertencentes a ela. Este padrão pode ser considerado a descrição da categoria. O propósito da construção de um classificador é descobrir os padrões que identificam cada uma das classes da base de dados e ensinar ao classificador estes padrões. As atividades de construção de classificador e categorização em um sistema de categorização de textos não diferem muito de como são feitas em outros problemas de aprendizagem supervisionada de máquina. A principal distinção dos TCS em relação a outros problemas de ML se dá na etapa de representação de documentos.

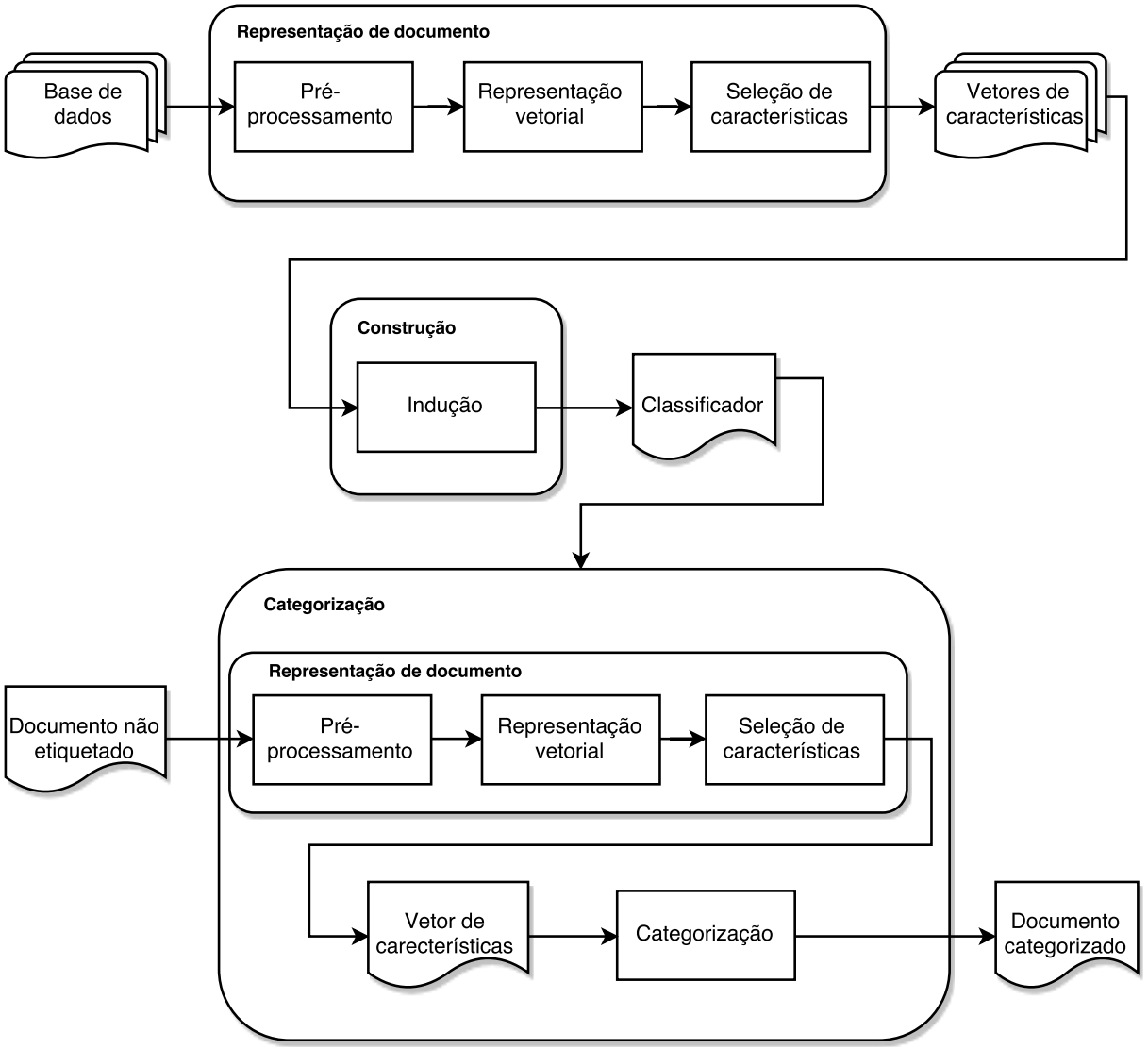

Figura 1: Arquitetura geral de um sistema de categorização de textos.

A Figura 1 apresenta a arquitetura geral de um sistema de categorização de textos. Inicialmente, o sistema recebe uma base de dados contendo documentos previamente categorizados. Estes documentos são tratados pelos procedimentos da etapa de representação de documentos com a finalidade de produzir vetores de características. Em seguida, na etapa de construção, um algoritmo de indução utiliza os vetores de características gerados na etapa de representação de documento para realizar o aprendizado e, assim, construir um classificador. Então, um documento desconhecido, ou seja, sem categorização prévia, é apresentado ao sistema na etapa de categorização. O sistema trata este documento com os procedimentos da etapa de representação de documento com a finalidade de extrair dele um vetor de características no mesmo padrão dos outros documentos da base de dados. Enfim, este vetor de características é apresentado ao classificador, que toma a decisão sobre a qual categoria o documento mais se adéqua. As atividades que compõem as etapas de um TCS são apresentadas nas seções que seguem.

2.1.1 Representação de documento

Para tornar possível a categorização de textos, é necessária a realização de alguns procedimentos anteriores à categorização propriamente dita. Os documentos textuais são recebidos pelo TCS em estado bruto, ou seja, em linguagem natural, como uma sequência de caracteres, contendo palavras, pontuações formatações, *tags* (no caso de páginas da *web*), elementos como título, assinatura, etc.

Algoritmos de aprendizagem de máquina requerem que os dados estejam em uma representação estruturada. Assim, os documentos em linguagem natural precisam passar por um tratamento que visa identificar e extrair características de modo que os documentos possam ser representados de maneira estruturada. Esta etapa envolve a quebra do texto em termos, a atribuição de pesos aos termos, a redução da quantidade de termos, entre outras ações. As atividades da etapa de representação de documento podem ser divididas em pré-processamento, representação estruturada de documentos e redução de dimensionalidade.

2.1.1.1 Pré-processamento

No pré-processamento são realizados os primeiros procedimentos nos documentos textuais, com a finalidade de remover informações irrelevantes como caracteres especiais, termos muito comuns, etc. Algumas das ações de pré-processamento, usualmente adotadas em sistemas de categorização de textos, são apresentadas a seguir.

- **Análise léxica ou tokenização:** Consiste em analisar o texto original produzindo como saída *tokens* ou símbolos léxicos. Nesta atividade, também são removidos caracteres especiais, hifens, dígitos, pontuação etc. A remoção de alguns destes itens

podem acarretar na perda de informação. Em palavras compostas, a remoção do hífen, e a consequente separação das palavras que a compõem, ocasiona uma perda do significado original. Adicionalmente, a simples separação dos termos na expressão "banco de dados" ocasionaria um perda na semântica da expressão. Uma alternativa para contornar este problema é o uso de grupos nominais, abordagem através da qual um algoritmo procura sequências de palavras semanticamente relacionadas. Estes grupos são considerados um único termo na indexação dos termos da base de dados.

- **Remoção de *stopwords*:** Palavras que estão presentes na maioria dos documentos de uma base de dados não adicionam informação para o processo de separação deles em categorias. Deste modo, artigos, preposições, conjunções e outras palavras sem relevância semântica podem ser removidas. Esta atividade exclui da base de dados os termos que constam numa lista de *stopwords* do idioma do documento em questão.

- ***Stemming*:** A atividade de *stemming* visa reduzir as palavras a seus radicais, ou seja, remover as variações de palavras como plural, prefixos, sufixos, gênero e conjugação (no caso de verbos). Por exemplo, os termos "livro", "livraria" e "'livreiro" seriam reduzidas ao radical "livr". Um efeito positivo é a redução da quantidade de termos para indexação na base de dados, visto que várias palavras podem ser reduzidas a um mesmo radical. Por outro lado, a fusão de várias palavras de mesmo radical em um único termo acarreta em perda de informação das palavras inteiras.

2.1.1.2 Representação estruturada de documentos

Após o pré-processamento, os documentos são transformados para um formato estruturado. Existem vários modelos de representação estruturada de documentos, porém, o modelo mais utilizado em problemas de TC é o modelo de espaço vetorial. Nele, cada documento é representado como um vetor em um espaço euclidiano V-dimensional, no qual V é o total de termos do vocabulário e cada dimensão equivale a um termo. Cada posição do vetor é composta por um termo e um valor associado que define a importância do termo no documento.

Assim, a atividade de representação estruturada de documento tem como objetivo transformar os termos dos documentos em dimensões dos vetores que os representam. Esta atividade consiste em extrair características dos documentos e envolve indexação e ponderação dos pesos dos termos nos documentos. A indexação pode considerar cada palavra isoladamente, usar sentenças ou ainda grupos de palavras ou caracteres (n-grams) para compor um termo do vocabulário. Algumas abordagens mais elaboradas fazem uso de técnicas de Processamento de Linguagem Natural, com objetivo de agregar importância semântica e, assim, melhorar os resultados.

Diferentes técnicas podem ser utilizadas para a ponderação da importância do

termo no documento. No modelo booleano, cada termo pode receber os pesos 1 ou 0 (*true* ou *false*), significando, respectivamente, a presença ou ausência do termo no documento. O modelo *Term Frequency* (LUHN, 1957) considera o número de ocorrências do termo no documento como sendo seu peso. Um modelo mais elaborado chamado *Term Frequency-Inverse Document Fequency* (WU et al., 2008), leva em conta duas observações para realizar a ponderação:

- Quanto maior o número de ocorrências de um termo em um documento, maior sua relevância para o tema deste documento.
- Quanto maior o número de documentos em que um termo ocorre, menor seu poder discriminatório.

Deste modo, no TF-IDF o peso de um termo em um documento é proporcional à frequência do termo no documento e inversamente proporcional ao número de documentos em que o termo ocorre. Outras abordagens levam em conta o contexto em que o termo se insere na estrutura do texto (título, negrito, posição do termo no texto etc) na ponderação dos pesos (CHAU; CHEN, 2008; COHEN; SINGER, 1999; YEPES et al., 2015). Este tratamento pode ser aplicado, principalmente, em um ambiente onde os documentos são bem estruturados, como páginas web, artigos científicos, registros de patente, etc.

Este trabalho adota apenas a representação vetorial *bag of words* (BoW) (JOACHIMS, 1996). Esta abordagem considera que a sequência das palavras no texto não é relevante. Assim, os documentos são representados como vetores de pares atributo-valor, de uma forma semelhante aos outros problemas de aprendizagem de máquina. Deste modo, cada palavra componente do vocabulário é uma característica e a frequência da palavra no documento é o valor. Em outras palavras, dada um base de dados $\mathcal{D}$ com vocabulário de tamanho V, um documento $d_i \in \mathcal{D}$ é representado por um vetor de dimensionalidade V, no qual cada dimensão é composta por uma palavra e a frequência desta palavra no documento d_i. Caso uma palavra w_h não esteja presente no documento d_i, a w_h será associado o valor 0 no vetor que representa d_i. A Figura 2 exibe um exemplo de representação vetorial de uma base de dados hipotética, formada por dois documentos textuais, usando a técnica *bag of words*.

Como o foco deste trabalho é na seleção de características, não se vê necessidade de avaliar outras formas de geração de características. Assim, decidiu-se pelo uso de *bag of words*, como forma de facilitar a comparação dos métodos aqui propostos com outros trabalhos, já que esta é a técnica mais utilizada em problemas de TC (TAŞCI; GÜNGÖR, 2013; TANG; KAY; HE, 2016).

A importância da representação estruturada de documentos em um TCS se dá na organização e estruturação dos dados dos documentos, de modo a tornar mais fácil

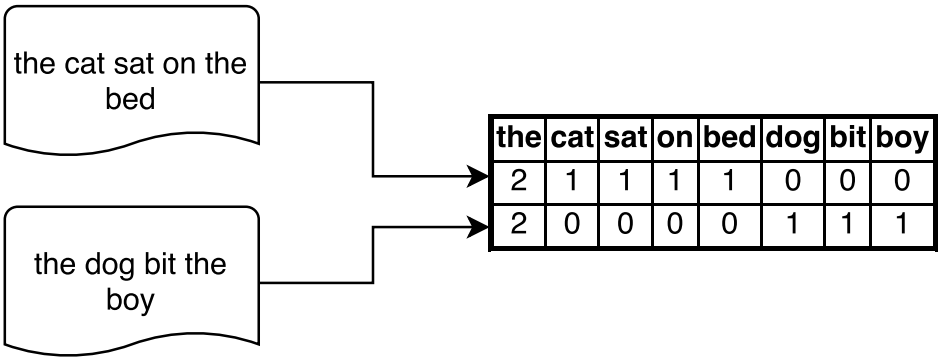

Figura 2: Exemplo de representação de documentos de texto utilizando *Bag of Words*.

o processamento por parte dos algoritmos classificadores. Uma vantagem adicional é a redução da quantidade de dados a serem processados, que contribui para um melhor desempenho em termos de consumo de processador e memória.

2.1.1.3 Redução de dimensionalidade

Na representação vetorial *bag of words*, cada palavra de uma base de dados é considerada uma característica. Deste modo, é comum que bases de dados de tamanho médio possuam dezenas de milhares de características. No entanto, a grande maioria delas é irrelevante para a categorização. Assim, a redução de dimensionalidade é considerada uma ação essencial para reduzir o esforço computacional e melhorar o desempenho de classificação.

Os métodos de Redução de Dimensionalidade são divididos, de acordo com a natureza dos termos resultantes, em métodos de extração de características e seleção de características (SEBASTIANI, 2002). A extração de características consiste em formar um novo conjunto de características, menor que o original, contendo novos termos gerados a partir da combinação ou transformação dos termos originais. Os métodos de seleção de características, por sua vez, não modificam as características originais, mas selecionam um subconjunto formado pelas características mais relevantes do conjunto original e desprezam as demais. Desta forma, seleção de características é definida como um problema de busca onde cada estado do espaço de busca especifica um subconjunto das características possíveis (PINHEIRO; CAVALCANTI; REN, 2015). As abordagens mais importantes de seleção de características, que é o foco deste trabalho, são detalhadas no Capítulo 3.

2.1.2 Construção

Para determinar automaticamente as categorias dos documentos, é necessário construir um classificador. A construção pode ser realizada através do uso de regras desenvolvidas por especialistas ou usando técnicas de aprendizagem de máquina. Nesta abordagem, os vetores produzidos na etapa de representação de documento, contendo

documentos previamente classificados e suas características, são fornecidos a um algoritmo indutor.

Seja uma base de dados $\mathcal{D}_{tr}$ e uma categoria c_j em $\mathcal{D}_{tr}$, com $j \in \{1, 2, .., Q\}$, onde Q é o total de categorias de $\mathcal{D}_{tr}$. O objetivo do indutor é utilizar os vetores de características para "aprender" o padrão de valores das características que definem a categoria c_j. Uma função hipotética $f(x)$ determina corretamente se uma instância x pertence à categoria c_j. Deste modo, o indutor procura gerar uma função $h(x)$ que seja o mais semelhante possível de $f(x)$. Este processo é realizado para todas as categorias da base de dados.

2.1.3 Categorização

Na etapa de categorização, é apresentado ao sistema um documento desconhecido, ou seja, não classificado. A finalidade é etiquetar este documento com uma ou mais categorias, dependendo do domínio de classificação adotado (*single-label* ou *multi-label*), através da utilização do classificador construído na etapa anterior (Seção 2.1.2). Para realizar esta atividade, é necessário que o documento passe pelos procedimentos da etapa de representação de documento (descritos na Seção 2.1.1) com a finalidade de se obter uma representação estruturada do documento no mesmo padrão dos demais exemplos da base de dados. Após obter o documento na estrutura padrão, o classificador anteriormente construído analisa o vetor que representa o documento e determina sua categoria com base nos valores dos seus atributos descritores. Mais detalhes sobre o processo de categorização de textos são apresentados na Seção 2.2

2.2 Algoritmos Classificadores

O propósito do uso de aprendizagem de máquina é reproduzir as capacidades de aprendizagem humana, particularmente a habilidade de reconhecer padrões complexos e tomar decisões inteligentes baseado em dados. O classificador é a parte do sistema de categorização de textos que efetivamente toma a decisão sobre a qual(is) categoria(s) determinado documento pertence.

Várias técnicas de aprendizagem de máquina podem ser utilizadas para categorização de textos. Na literatura, bons resultados em categorização de textos foram reportados com uso de k-NN (do inglês, *k-Nearest Neighbors*) (UĞUZ, 2011; YANG, 1999), SVM(do inglês, *Support Vector Machine*) (RAKOTOMAMONJY, 2003; ZHANG; YOSHIDA; TANG, 2008), *Naïve Bayes* (CHEN et al., 2009; LEWIS, 1998), redes neurais (GHIASSI; SKINNER; ZIMBRA, 2013; YU; XU; LI, 2008) etc.

Neste trabalho adotamos o classificador *Naïve Bayes*. A escolha deveu-se à combinação de simplicidade e bons resultados do classificador (RISH, 2001) e devido ao fato de que o mesmo apresenta uma relevante suscetibilidade à seleção de características (CHEN

et al., 2009; PINHEIRO et al., 2012). O classificador *Support Vector Machine* (SVM, do inglês, máquina de vetores de suporte) foi utilizado nos experimentos para realizar uma comparação de tempo de execução dos métodos propostos, já que este classificador é conhecido por ter bom desempenho em problemas de alta dimensionalidade (JOACHIMS, 1998; LEOPOLD; KINDERMANN, 2002). Os detalhes do funcionamento do classificador *Naïve Bayes* são descritos na Seção 2.2.1. A Seção 2.2.2 apresenta o SVM.

2.2.1 *Naïve Bayes*

Classificadores bayesianos tem sido amplamente utilizados para problemas de categorização de textos (CHEN et al., 2009; FENG et al., 2015; PINHEIRO; CAVALCANTI; REN, 2015; SUN et al., 2013; YANG et al., 2012). Este tipo de classificador utiliza probabilidades conjuntas de termos e categorias para a determinação da probabilidade de um documento pertencer a uma categoria.

Naïve Bayes baseia-se em dois pressupostos considerados simplórios. O classificador considera que probabilidade de um documento pertencer a uma determinada categoria depende unicamente das probabilidades dos termos em relação a esta categoria. A outra hipótese assumida é que os termos em um documento são independentes umas das outras. Apesar de partir destes pressupostos frágeis, o classificador consegue atingir resultados satisfatórios (SCHNEIDER, 2003).

Dois modelos de classificadores *Naïve Bayes* tem sido utilizados para tarefas de recuperação de informações: *Naïve Bayes Multivariate Bernoulli* e *Naïve Bayes Multinomial* (CHEN et al., 2009; MCCALLUM; NIGAM, 1998). A diferença básica entre os dois modelos se dá em relação à quantidade de informação que é capturada sobre os termos em um dado documento. *Naïve Bayes* em seu modelo Multinomial utiliza as ocorrências dos termos selecionados do vocabulário nos documentos, enquanto o modelo *Multivariate Bernoulli* utiliza os termos selecionados do vocabulário sem informações sobre o número de ocorrências dos mesmos em cada documento, contabilizando apenas ausência ou presença. Segundo Chen et al. (2009), os dois modelos apresentam desempenhos equivalentes. Entretanto, *Naïve Bayes Multinomial* exibe um desempenho superior, principalmente em bases de dados com maior dimensionalidade. Por esta razão, o classificador *Naïve Bayes Multinomial* foi escolhido para ser utilizado neste trabalho.

A base dos classificadores bayesianos á a regra de Bayes, que calcula a probabilidade $P(c_j|d_i)$ de que, dado um documento d_i, este pertença à categoria c_j. A regra de Bayes é dada pela Equação 2.1.

$$\mathrm{P(c_j|d_i)} = \frac{P(d_i|c_j)P(c_j)}{P(d_i)} \tag{2.1}$$

$P(c_j)$ é a probabilidade a priori da categoria c_j, ou seja, a probabilidade de que

um documento aleatoriamente selecionado da base de dados pertença a c_j. $P(d_i)$ é a probabilidade a priori do documento d_i, ou seja, a probabilidade de que um documento aleatoriamente selecionado da base de dados seja d_i. O algoritmo de *Naïve Bayes* computa o valor de $P(c_j|d_i)$, com $j \in \{1, 2, ..., Q\}$, onde Q é o número de categorias da base de dados. O documento é considerado pertencente à categoria que retornar o maior valor de $P(c_j|d_i)$. Como $P(d_i)$ é igual para todos as categorias, este termo pode ser removido da Equação 2.1.

$$\mathrm{P(c_j|d_i)} = P(d_i|c_j)P(c_j) \tag{2.2}$$

O cálculo da probabilidade $P(d_i|c_j)$ é realizado, na versão *Multinomial* de *Naïve Bayes* (KIBRIYA et al., 2004), através da Equação 2.3.

$$\mathrm{P(d_i|c_j)} = (\sum_{h=1}^{V} n_{ih})! \prod_{h=1}^{V} \frac{P(w_h|c_j)^{n_{ih}}}{n_{ih}!} \tag{2.3}$$

Nesta equação, n_{ih} representa o número de ocorrências do termo w_h no documento d_i, V é o tamanho do vocabulário e $P(w_h|c_j)$ é a probabilidade do termo w_h dada a categoria c_j. Esta probabilidade é estimada a partir dos documentos de treinamento como

$$\mathrm{P(w_h|c_j)} = \frac{1 + N_{jh}}{V + N_j}, \tag{2.4}$$

no qual N_{jh} é a soma das ocorrências do termo w_h em documentos pertencentes à categoria c_j, N_j é a soma das ocorrências de todos os termos em documentos da categoria c_j.

2.2.2 *Support Vector Machine*

SVM é uma abordagem de aprendizagem de máquina relativamente nova (VLADIMIR; VAPNIK, 1995) e tem sido popularmente utilizado para categorização de textos (JOACHIMS, 1998; GUYON; ELISSEEFF, 2003; ZHANG; YOSHIDA; TANG, 2008; SUN et al., 2013).

SVM, em sua forma mais básica, foi proposto para resolver problemas de classificação de domínio binário. O classificador busca definir um espaço de decisão que separe da melhor maneira possível os pontos de uma categoria positiva dos pontos de uma categoria negativa, assumindo que os exemplos de uma base de dados são representados no espaço euclidiano. A forma mais simples de SVM trata de problemas linearmente separáveis. Nestes casos, a superfície de decisão é um hiperplano. Desta forma, ao mesmo tempo que SVM tentar colocar do mesmo lado do hiperplano o maior número de exemplos da mesma categoria, ele busca maximizar a distância de cada categoria ao hiperplano, conhecida

como margem de separação. Esta margem é definida pelos pontos de cada categoria mais próximos ao hiperplano.

A superfície de decisão para um problema linearmente separável é dada por

$$\vec{d_i} \cdot \vec{x} - b = 0, \tag{2.5}$$

no qual $\vec{d_i}$ é um documento arbitrário a ser categorizado, o vetor $\vec{x}$ é o vetor normal ao hiperplano e o viés b é aprendido a partir do conjunto de treinamento. Assumindo que $\mathcal{C} =$ {-1, 1} é o conjunto de categorias (-1 para a categoria negativa e 1 para a categoria positiva), o problema de SVM é encontrar um vetor $\vec{x}$ e um valor para b que satisfaça

$$\vec{d_i} \cdot \vec{x} - b \gg 1, \tag{2.6}$$

caso $\vec{d_i}$ pertença à categoria positiva e

$$\vec{d_i} \cdot \vec{x} - b \ll -1, \tag{2.7}$$

caso $\vec{d_i}$ pertença à categoria negativa.

Este modelo de SVM pode ser estendido para lidar com problemas multi-classe e não linearmente separáveis. Para este caso, SVM introduz uma tolerância ao erro, admitindo, embora penalizando, exemplos do lado errado do hiperplano (VLADIMIR; VAPNIK, 1995).

2.2.3 Critérios de Avaliação

O objetivo da avaliação de classificadores é verificar o nível de adequação de suas decisões. *Micro-F1* e *Macro-F1* (SEBASTIANI, 2002) são medidas popularmente adotadas para avaliação de classificadores (PINHEIRO et al., 2012; TAŞCI; GÜNGÖR, 2013; WANG et al., 2014; UYSAL; GUNAL, 2012; UYSAL, 2016). Ambas as medidas são calculadas a partir da Equação 2.8:

$$\mathcal{F}1 = \frac{2\text{x}\mathcal{P}\text{x}\mathcal{R}}{\mathcal{P} + \mathcal{R}}, \tag{2.8}$$

onde $\mathcal{P}$ é uma medida chamada precisão (do inglês *precision*) e $\mathcal{R}$ é cobertura (do inglês *recall*) (CHANG; CHEN; LIAU, 2008). As fórmulas para calcular a precisão $\mathcal{P}(c_j)$ e a cobertura $\mathcal{R}(c_j)$ da categoria c_j são exibidas a seguir.

$$\mathcal{P}(\text{c}_\text{j}) = \frac{TP_j}{TP_j + FP_j} \tag{2.9}$$

$$\mathcal{R}(\mathrm{c_j}) = \frac{TP_j}{TP_j + FN_j} \tag{2.10}$$

TP_j é a quantidade de instâncias corretamente categorizadas como pertencentes à categoria c_j, FP_j é a quantidade de instâncias incorretamente categorizadas como pertencentes à categoria c_j e FN_j é a quantidade de instâncias incorretamente categorizadas como não pertencentes à categoria c_j.

Micro-F1 e *Macro-F1* se diferenciam no modo como calculam as médias de $\mathcal{P}$ e $\mathcal{R}$ para toda a base de dados. No cálculo de *Micro-F1*, cada categoria recebe um peso de acordo com seu tamanho, o que favorece o desempenho das categorias com mais documentos no resultado final. Já para *Macro-F1*, todas as categorias recebem o mesmo peso, o que favorece a performance das categorias com menos documentos. As fórmulas 2.11 à 2.14 referem-se aos cálculos da precisão e cobertura utilizadas nos cômputos de *Micro-F1* (μ) e *Macro-F1* (M). Nelas, Q representa o número de categorias da base de dados.

$$\mathcal{P}_\mu = \frac{\sum_{j=1}^{Q} TP_j}{\sum_{j=1}^{Q}(TP_j + FP_j)} \tag{2.11}$$

$$\mathcal{R}_\mu = \frac{\sum_{j=1}^{Q} TP_j}{\sum_{j=1}^{Q}(TP_j + FN_j)} \tag{2.12}$$

$$\mathcal{P}_\mathrm{M} = \frac{\sum_{j=1}^{Q} \mathcal{P}(c_j)}{Q} \tag{2.13}$$

$$\mathcal{R}_\mathrm{M} = \frac{\sum_{j=1}^{Q} \mathcal{R}(c_j)}{Q} \tag{2.14}$$

2.3 Considerações Finais

Este capítulo introduziu conceitos da atividade de categorização de textos bem como definições para TC, domínios de classificação e abordagens de etiquetagem. A arquitetura de um sistema de categorização de textos típico foi retratada e suas etapas e atividades foram explanadas. O classificador *Naïve Bayes Multinomial*, que é utilizado nos experimentos deste trabalho, foi descrito e foram apresentadas as medidas de avaliação de classificadores *Micro-F1* e *Macro-F1*. O próximo capítulo apresenta conceitos de seleção de características, bem como métodos anteriores que foram importantes para o desenvolvimento deste trabalho.

3 Seleção de Características

A categorização automática de textos a partir do uso de aprendizagem de máquina impõe uma necessidade de utilização de técnicas que permitam reduzir a quantidade de variáveis a serem analisadas durante o processo. A seleção de características (FS, do inglês *feature selection*) é amplamente adotada como abordagem de redução de dimensionalidade no processo de categorização de textos. Este capítulo destina-se a apresentar os conceitos fundamentais sobre seleção de características.

O objetivo da seleção de características é, dado um conjunto de características, selecionar um subconjunto de alto poder discriminatório de modo a aumentar o desempenho da classificação em termos de tempo de processamento e uso de memória. Desta forma, a seleção de características remove atributos irrelevantes ou redundantes para melhorar o desempenho de classificação. Além de reduzir o tamanho da base de dados, FS promove um aumento da compreensão dos resultados e provoca uma melhoria da precisão, pois evita um ajuste excessivo à amostra de dados, que prejudica a generalização do classificador - problema conhecido como *overfitting* (DIETTERICH, 1995).

A seleção de características pode ser feita através de métodos de filtragem ou métodos *wrapper*. Os métodos *wrapper* (KOHAVI; JOHN, 1997) constroem, aleatoriamente ou com base em regras, vários subconjuntos do espaço de características original e determinam o melhor subconjunto através da taxa de acerto alcançada na classificação de cada um dos subconjuntos. Esta técnica alcança maior precisão que métodos de filtragem, já que utiliza o próprio algoritmo classificador para avaliar as alternativas de subconjuntos de características, podendo assim, teoricamente, chegar ao subconjunto ótimo. Entretanto, devido ao alto número de execuções do algoritmo classificador que o processo de busca exige, seu custo computacional é alto, tornando a aplicação de métodos *wrapper* inviável para classificadores mais complexos ou problemas de altíssima dimensionalidade, como costumeiramente é TC.

Alternativamente aos métodos *wrapper*, o processo de seleção de características através de filtragem independe do uso de classificadores. Os métodos de filtragem (GUYON; ELISSEEFF, 2003) realizam um ordenamento dos termos, geralmente com o uso de funções de avaliação de características (FEF, do inglês, *Feature Evaluation Function*). A Seção 3.1 apresenta conceitos básicos sobre FEFs. Um número m de termos do conjunto original é selecionado para compor um novo subconjunto, m determinado pelo usuário. Neste passo são considerados os m termos identificados como os mais relevantes no ordenamento. A estratégia para seleção de características mais adotada, no âmbito da categorização de textos, é a filtragem de atributos. A popularidade destes métodos para TC deve-se

ao fato de que eles apresentam um melhor desempenho que os métodos *wrapper* em problemas de alta dimensionalidade, como TC (FORMAN, 2003; PINHEIRO et al., 2012; SEBASTIANI, 2002). Esta vantagem deve-se ao fato de não haver necessidade de interação, durante a construção do conjunto de características, com o algoritmo indutivo usado para classificação.

Métodos de filtragem e métodos *wrapper* podem ser usados conjuntamente. Pode-se realizar uma filtragem inicial das características e posteriormente se valer de métodos *wrapper* para – em um espaço de características já reduzido – conseguir um melhor desempenho de classificação (DAS, 2001).

A Seção 3.1 introduz conceitos de FEFs, a Seção 3.2 apresenta os trabalhos anteriores importantes para o desenvolvimento dos métodos propostos neste trabalho e as conclusões do capítulo são exibidas na seção 3.3.

3.1 Funções de Avaliação de Características

Funções de avaliação de características (FEF) são métricas estatísticas usadas para determinar a capacidade discriminatória de uma característica em relação a um conjunto de dados. O cálculo normalmente leva em conta as probabilidades de um termo w com relação aos documentos da categoria c_j. Forman (FORMAN, 2003) realizou um estudo extensivo sobre FEFs costumeiramente utilizadas em categorização de textos. Adicionalmente, outras métricas foram propostas (CHEN et al., 2009; SHANG et al., 2007; UYSAL; GUNAL, 2012; YANG et al., 2012).

As FEFs adotadas nos experimentos deste trabalho, bem como suas fórmulas, são detalhadas na seção 5.1.2.

3.2 Métodos de Seleção de Características

Métodos clássicos de filtragem geralmente utilizam uma estratégia de ordenamento de características. Nesta abordagem, cada característica tem sua importância avaliada através de métricas estatísticas (funções de avaliação de características). São selecionadas aquelas características com as melhores avaliações, ou seja, com maiores valor FEF. O algoritmo clássico utilizado para selecionar as características mais determinantes, de acordo com o ordenamento, é conhecido como *Variable Ranking* (VR, em português, ordenamento de variáveis) (GUYON; ELISSEEFF, 2003).Este método surgiu com Lewis e Ringuette (LEWIS; RINGUETTE, 1994), que utilizaram a FEF *Information Gain* como métrica para ranquear as características. Posteriormente outras métricas, como *Mutual Information* (GUYON; ELISSEEFF, 2003; YANG; PEDERSEN, 1997), *Chi-Square* (DEBOLE; SEBASTIANI, 2003; YANG; PEDERSEN, 1997), *Bi-Normal Separation* (FORMAN, 2003), *Improved*

Gini Index (SHANG et al., 2007), *Class Discriminating Measure* (CHEN et al., 2009), foram usadas em conjunto com VR. Sua simplicidade, escalabilidade e os bons resultados apresentados ajudaram a difundir a utilização de VR. Entretanto uma importante limitação deste método é a dificuldade de determinar o melhor valor para o parâmetro m. Para cada valor possível de m existe um subconjunto de características que deve ser avaliado, através do uso de classificadores, para que se encontre o melhor conjunto existente. Este fato faria com que o método de filtragem perdesse sua vantagem sobre métodos *wrapper*, que é o melhor desempenho em termos de tempo e uso computacional. Para contornar este problema normalmente testa-se somente alguns valores para m e escolhe-se o melhor dos subconjuntos testados. Outra dificuldade desta abordagem é em relação à representatividade dos documentos no conjunto final de características. Caso o valor de m seja muito baixo, alguns documentos podem não ter nenhuma característica selecionada (nenhum dos m termos mais importantes fazem parte do documento), impossibilitando a correta categorização do mesmo.

Estudos tem sido realizados e métodos de seleção de características tem sido propostos com a finalidade de tratar questões como o desbalanceamento de bases de dados (LU; LIU; HE, 2015; OGURA; AMANO; KONDO, 2011; ZHENG; WU; SRIHARI, 2004), o uso combinado de métodos de seleção de características (GUNAL, 2012; UĞUZ, 2011; UYSAL, 2016), grupos de características discriminatórios (SUN et al., 2013) e a determinação, de maneira automática, do valor ideal do parâmetro m (PINHEIRO et al., 2012; PINHEIRO; CAVALCANTI; REN, 2015).

Em Pinheiro et al. (2012) o método, *At Least One FeaTure* (ALOFT, em português, ao menos uma característica), é apresentado. ALOFT representa cada documento com no mínimo uma característica do conjunto de dados original. Em estudo mais recente (PINHEIRO; CAVALCANTI; REN, 2015), um método de seleção menos agressiva (MFD, *Maximum f Features per Document*, em português, máximo de f características por documento) foi proposto com a finalidade de melhorar o desempenho classificatório em relação a ALOFT. Cada documento pode ser representado por mais de uma característica. Ainda no mesmo estudo, os autores apresentaram outro método (MFDR, *Maximum f Features per Document-Reduced*, máximo de f características por documento - reduzido) onde apenas os documentos que contêm características de alto valor FEF são considerados para a seleção de características. Estes três métodos são descritos em detalhes nas seções 3.2.1 a 3.2.3.

3.2.1 *At Least One FeaTure*

O método de seleção de características *At Least One FeaTure* (ALOFT) (PINHEIRO et al., 2012) se propõe a sanar dois problemas do método clássico de filtragem, *Variable Ranking*: a determinação automática do tamanho do subconjunto final de características e

garantir que todos os documentos da base de dados sejam representados no subconjunto final.

O método calcula o valor discriminatório de cada característica, de acordo com alguma FEF. Em seguida, ALOFT realiza o ordenamento global das características baseando-se nos valores FEF das características. Após esta etapa, o método percorre cada documento da base de dados. Para cada documento, o algoritmo verifica no ranking qual é a característica valorada (com frequência maior que zero no documento) melhor ranqueada. Esta característica é, então, armazenada no vetor FS. Caso esta característica já esteja presente em FS, o algoritmo a ignora e segue para o próximo documento. Desta forma, ALOFT assegura que todos os documentos serão representados por ao menos uma característica no subconjunto final de características. O número de termos selecionados por ALOFT é no máximo igual ao número de documentos da base de dados, já que o método seleciona apenas um termo por documento. Entretanto, como uma mesma característica pode ser a melhor ranqueada em mais de um documento, geralmente o tamanho do subconjunto final de características é bem menor que o número de documentos da base de dados. A Figura 3 exibe o fluxograma do ALOFT.

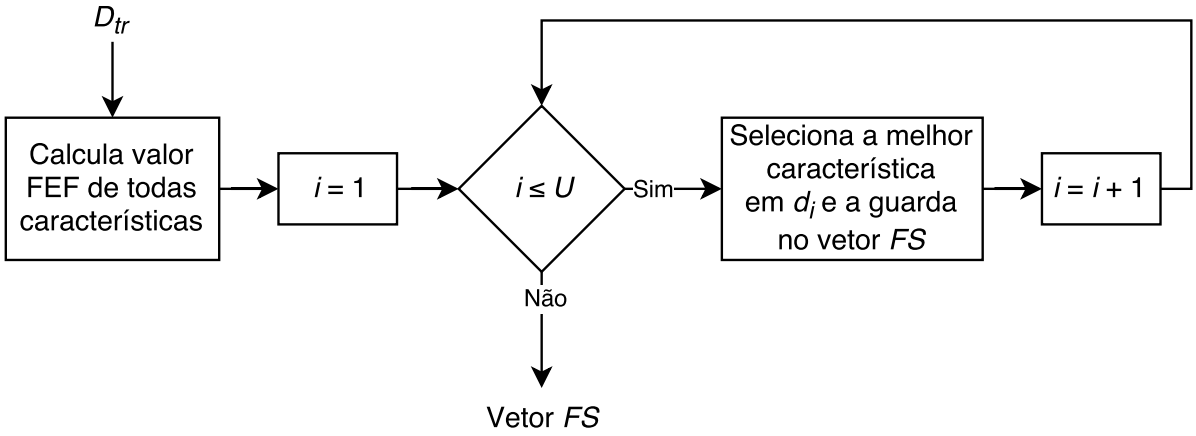

Figura 3: Fluxograma do método ALOFT. $\mathcal{D}_{tr}$ é o conjunto de documentos de treinamento, U é o número de documentos em $\mathcal{D}_{tr}$ e FS é o vetor que armazena as características selecionadas.

Uma importante vantagem do ALOFT é a determinação automática do tamanho do conjunto final de características, visto que nenhuma parametrização é necessária. Os resultados obtidos com ALOFT são comparáveis com métodos clássicos de filtragem (PINHEIRO et al., 2012). Entretanto, a limitação de selecionar apenas uma característica por documento, por vezes, dificulta a separação das categorias.

3.2.2 *Maximum f Features per Document*

As limitações de desempenho do método ALOFT, devido à sua restrição de selecionar apenas uma característica por documento, motivaram a proposição do método

Maximum f Features per Document, MFD (PINHEIRO; CAVALCANTI; REN, 2015). Este método utiliza o parâmetro f para determinar quantas características devem ser selecionadas por documento. Assim como ALOFT, o MFD inicialmente realiza um ordenamento global das características de acordo com alguma FEF. O algoritmo então percorre os documentos selecionando as f características valoradas com maiores valores FEF de cada documento para compor o vetor FS, que representa o subconjunto final de características. As características selecionadas que já estejam presentes no vetor FS (que foram selecionadas previamente em outros documentos) são ignoradas pelo algoritmo e as outras características selecionadas são incluídas normalmente no vetor FS. O fluxograma do MFD é apresentado na Figura 4.

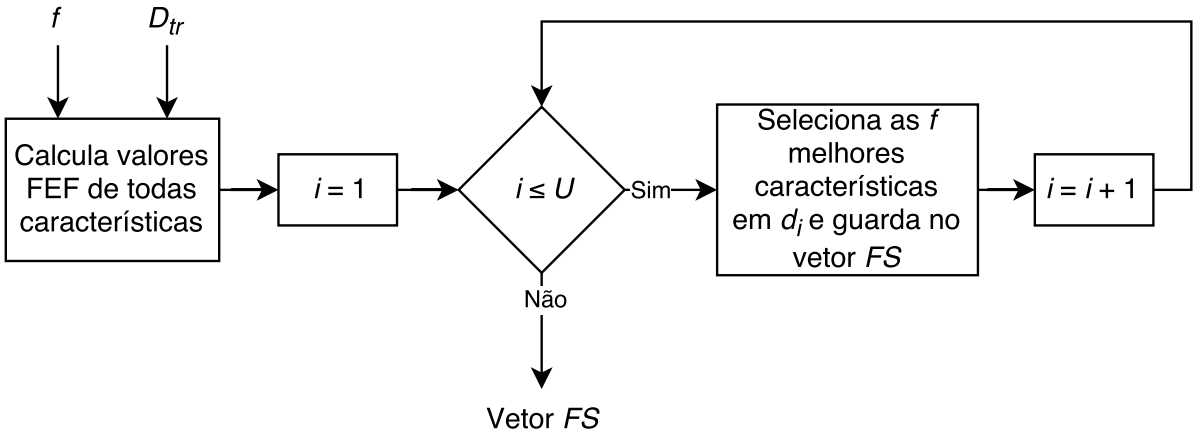

Figura 4: Fluxograma do método *Maximum f Features per Document*. $\mathcal{D}_{tr}$ é o conjunto de documentos de treinamento, f é o número de características a serem selecionadas por documento, U é o número de documentos em $\mathcal{D}_{tr}$ e FS é o vetor que armazena as características selecionadas.

MFD flexibiliza a quantidade de características a serem selecionadas por documento em relação a ALOFT. Este fato garante melhores resultados para o MFD (PINHEIRO; CAVALCANTI; REN, 2015). Porém, esta vantagem é conseguida com a inclusão do parâmetro f, que determina a quantidade de características selecionadas por documento. Entretanto, a configuração deste parâmetro é bem menos custosa que a busca do melhor valor para o parâmetro m, dos métodos clássicos de filtragem de características. Os melhores resultados de MFD são alcançados com valores para f entre 1 e 10, que selecionam entre 11 e 5938 características, dependendo da base de dados e FEF utilizado (PINHEIRO; CAVALCANTI; REN, 2015). Logo, para os métodos clássicos seria necessário avaliar valores para o parâmetro m neste intervalo.

3.2.3 *Maximum f Features per Document - Reduced*

Um aspecto negativo do MFD é o tamanho dos melhores subconjuntos, que são consideravelmente maiores que aqueles gerados com ALOFT. Para tratar esta desvantagem,

o método *Maximum f Features per Document - Reduced* (MFDR) impõe um limiar para determinar se um documento será considerado no processo de seleção de características ou se ele será descartado. A hipótese é que é possível reduzir o tamanho do subconjunto final de características removendo alguns documentos da seleção de características, seguindo algum indicativo de que eles apresentam pouca contribuição para separação das categorias. Neste caso, o indicativo é a relevância do documento DR, valor que é calculado como o somatório dos valores FEF das características presentes no documento, conforme a equação que segue:

$$DR_i = \sum_{h=1}^{V} \Big(\mathrm{S_h} \times \mathrm{valued}(w_h, d_i) \Big), \tag{3.1}$$

no qual d_i é um documento, V é o tamanho do vocabulário, S é um vetor que armazena o valor FEF do termo w_h na sua $h$ésima posição, $w_{h,i}$ é o $h$ésimo termo do $i$ésimo documento. A função $valued(\cdot)$ retorna 1 se um termo específico é valorado, ou seja, está presente em um determinado documento, caso contrário, retorna 0. Após isto, a média dos DRs de todos os documentos é calculada. Esta média define o limiar T. Este cálculo é realizado como

$$T = \frac{\sum_{i=1}^{U} \Big(\mathrm{DR}_i \Big)}{U}, \tag{3.2}$$

no qual U é o total de documentos em $\mathcal{D}_{tr}$. Somente documentos com relevância $DR > T$ são considerados na seleção de características. Documentos com relevância $DR \leqslant T$ são desprezados no processo de seleção de características.

A Figura 5 apresenta o fluxograma do método MFDR. Seu processo de seleção de características inicia-se com o cálculo do valor FEF de cada característica e o ordenamento das mesmas. Em seguida é realizado o cálculo das relevâncias dos documentos (DR) e do limiar T. Posteriormente à definição de T, o algoritmo percorre cada documento d_i da base de dados verificando se sua relevância DR_i é maior que o limiar T. Caso seja, MFDR seleciona as f características de maior valor FEF e as adiciona ao vetor FS. Caso contrário, o documento é ignorado.

A restrição da quantidade de documentos considerados na seleção de características acarreta em uma redução do tamanho do subconjunto final de características. Esta diminuição, entretanto, não afeta o desempenho classificatório. MFDR apresenta performance semelhante ao MFD (PINHEIRO; CAVALCANTI; REN, 2015). Porém, percebe-se que, com uso do MFDR, o desempenho de algumas categorias fica bem abaixo de outras. Isto deve-se ao fato de que o método utiliza um único limiar para toda a base de dados. Assim, se uma categoria é composta por documentos que contém termos de baixo valor FEF

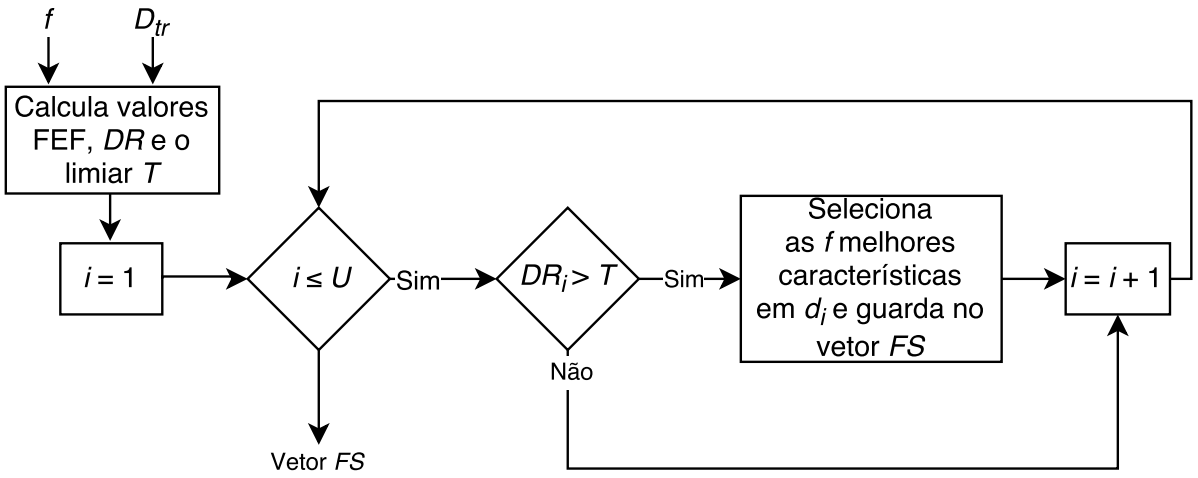

Figura 5: Fluxograma do método *Maximum f Features per Document-Reduced*. $\mathcal{D}_{tr}$ é o conjunto de documentos de treinamento, f é o número de características a serem selecionadas por documento, U é o número de documentos em $\mathcal{D}_{tr}$, DR é a relevância do documento e FS é o vetor que armazena as características selecionadas.

possuem, as relevâncias DR dos seus documentos dificilmente superarão o limiar e, deste modo, o desempenho classificatório com documentos da categoria pode ser prejudicado.

3.3 Considerações Finais

Neste capítulo foram apresentados os conceitos básicos sobre seleção de características, métodos *wrapper* e *filter*, suas vantagens, desvantagens e aplicações. Também foram expostos os métodos tradicionais de seleção de características e os métodos que inspiraram o desenvolvimento dos métodos propostos neste trabalho: *At Least One FeaTure*, *Maximum f Features per Document* e *Maximum f Features per Document-Reduced*. A Tabela 1 apresenta os pontos fortes e os pontos fracos dos métodos apresentados nesta seção.

Tabela 1: Comparativo entre os métodos de seleção de características VR, ALOFT, MFD e MFDR

Método	Pontos fortes	Pontos fracos
VR	Simplicidade Bons resultados	Pode produzir vetores vazios Dificuldade de definir tamanho do vetor final
ALOFT	Tamanho do vetor final definido guidao por dados Todos os documentos são representados no vetor final com pelo menos uma característica Desempenho equivalente ou superior a VR	Número de características selecionadas pode ser pequeno, dependendo da FEF utilizada
MFD	Todos os documentos são representados no vetor final com pelo menos f características Desempenho equivalente ou superior a ALOFT Parâmetro f de fácil configuração	Melhores resultados são alcançados com conjunto grande de características Alguns documentos pioram a qualidade do subconjunto selecionado, para valores altos de f
MFDR	Seleciona menos características que MFD Desempenho equivalente ou superior a MFD	O cálculo de DR pode beneficiar documentos com muitas características O limiar global pode causar subrepresentação de categorias na seleção

4 Métodos Propostos

Este capítulo apresenta os métodos de seleção de características propostos neste trabalho. Os métodos apresentados aqui foram desenvolvidos com o intuito de melhorar o desempenho e a eficiência da classificação. O primeiro método proposto restringe quais documentos de cada categoria são considerados na seleção de características através da definição de limiares baseados nas relevâncias dos documentos. O segundo método proposto tem por objetivo automatizar a definição do número ideal de características selecionadas por documento, dado pelo parâmetro f nos métodos MFD e MFDR. O método *Category-dependent Maximum f Features per Document - Reduced* é apresentado na Seção 4.1 e o método *Automatic Features Subsets Analyzer* é descrito na Seção 4.2. A Seção 4.3 trás as considerações finais deste capítulo.

4.1 *Category-dependent Maximum f Features per Document - Reduced*

O método proposto, Category-dependent Maximum f Features per Document (cMFDR) (FRAGOSO; PINHEIRO; CAVALCANTI, 2016a), é um método de seleção de características que visa aprimorar o desempenho alcançado pelo método MFDR (Seção 3.2.3). MFDR seleciona f características por cada documento que satisfaça uma condição baseada na sua relevância DR. Apenas documentos com DR maior que a média dos DR de todos os documentos da base de dados são considerados pelo algoritmo de seleção de características.

Porém, esta abordagem empregada por MFDR apresenta alguns problemas. Caso uma categoria possua muitos termos com baixo valor FEF em comparação a outras, as relevâncias DR dos documentos desta categoria dificilmente irão superar a média geral dos DR da base de dados. Assim, a categoria pode ser sub-representada no processo de seleção de características. Em outras palavras, um documento considerado irrelevante (DR menor que a média de todos DRs) pode ser importante para determinada categoria. Adicionalmente, o cálculo da relevância DR no MFDR consiste na soma dos valores FEF dos termos presentes no documento. Assim, documentos com um grande número de termos podem ser privilegiados, mesmo sem possuir termos de relevantes para a categorização. Documentos com uma grande quantidade de termos com baixos valores FEF podem superar o limiar, sendo assim considerados relevantes, enquanto documentos com uma pequena quantidade de termos com alto valor FEF podem ser descartados. Por exemplo, um documento d_1 que possui 10 termos, todos com valor FEF igual a 1, tem $DR = 10$.

Outro documento d_2 que possui apenas um termo, com valor FEF igual a 9, tem $DR = 9$. Assim, d_1 é considerado mais relevante que d_2, mesmo possuindo apenas termos de baixo valor FEF. Caso a média dos DRs de todos os documentos seja um valor entre 9 e 10, o documento d_1 é considerado relevante enquanto d_2 é descartado. Deste modo, a importante informação discriminatória contida no único termo do documento d_2 é perdida.

Portanto, o objetivo do cMFDR é melhorar o desempenho de classificação através da resolução destes problemas: definição do limiar e cálculo de DR. O método cMFDR define limiares diferentes para cada categoria, baseado nas relevâncias DR. O limiar CT para uma dada categoria é calculado como a média das relevâncias dos documentos desta categoria. Apenas documentos com DR maior que CT da categoria à qual pertencem são designados para participar da seleção de características. Em relação ao cálculo da relevância DR, cMFDR busca evitar a influência do tamanho do documento. Para garantir uma melhor comparação entre as relevâncias dos documentos, o valor de DR é dado pela média dos valores FEF dos termos presentes no documento.

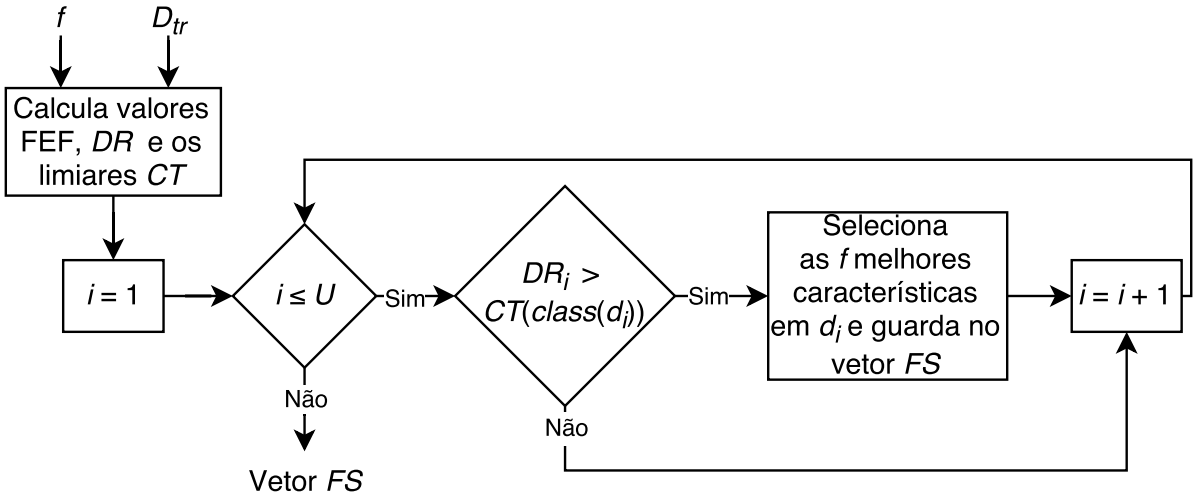

Figura 6: Fluxograma do método *Category-dependent Maximum f Features per Document-Reduced*. $\mathcal{D}_{tr}$ é o conjunto de documentos de treinamento, f é o número de características a serem selecionadas por documento, D é o número de documentos em $\mathcal{D}_{tr}$, DR é a relevância do documento, CT é o limiar da categoria e FS é o vetor que armazena as características selecionadas.

A Figura 6 exibe o fluxograma do método cMFDR. O método requer duas entradas: uma base de dados de treinamento $\mathcal{D}_{tr}$ e o número f de características que devem ser selecionadas por documento. O valor de f deve ser um inteiro maior que zero. A base de dados $\mathcal{D}_{tr}$ contém $d \in \mathbb{N}^V$ documentos, onde V é o tamanho do vocabulário. O algoritmo calcula o valor FEF para cada termo w_h contido na base $\mathcal{D}_{tr}$. Os valores são armazenados no vetor S. Em seguida, o vetor DR, que armazena a importância de cada documento, é

calculado como segue

$$\mathrm{DR}_i = \frac{\sum_{h=1}^{V}\Big(\mathrm{S_h} \times \mathrm{valued}(w_h, d_i)\Big)}{\sum_{h=1}^{V} \mathrm{valued}(w_h, d_i)}, \tag{4.1}$$

no qual d_i é um documento, V é o tamanho do vocabulário, S é um vetor que armazena o valor FEF do termo w_h na sua $h$ésima posição, $w_{h,i}$ é o $h$ésimo termo do $i$ésimo documento. A função $valued(\cdot)$ retorna 1 se um termo específico é valorado, ou seja, está presente em um determinado documento, caso contrário, retorna 0. O próximo passo é calcular o vetor CT, que armazena o valor do limiar de cada categoria. O cálculo de CT é dado por

$$\mathrm{CT}(c_j) = \frac{\sum_{i=1}^{U}\Big(\mathrm{DR}_i \times \mathrm{belongs}(d_i, c_j)\Big)}{\sum_{i=1}^{U} \mathrm{belongs}(d_i, c_j)}, \tag{4.2}$$

no qual c_j é uma categoria, U é o número de documentos em $\mathcal{D}_{tr}$, d_i é o $i$ésimo documento. A função $belongs(\cdot)$ retorna 1 se o documento d_i pertence à categoria c_j, caso contrário, retorna 0. O subconjunto de características é computado na próxima etapa. Um documento é ignorado se sua relevância DR for menor que o valor de CT da categoria à qual pertence. Para cada documento, as f características com maior valor FEF são avaliadas, em ordem descendente. Caso a característica não esteja presente no vetor FS, o algoritmo a adiciona neste vetor. Após analisar as f características, o algoritmo segue para o próximo documento. Ao final desta etapa, o vetor FS, que representa o subconjunto final de características, conterá as m características selecionadas.

4.1.1 Exemplo

Nesta seção, é apresentado um exemplo prático do funcionamento do método cMFDR, bem como é demonstrada a vantagem da adoção de um limiar por categoria, em contraste com o limiar global de seu predecessor MFDR. Uma base de dados de treinamento é apresentada na Tabela 2. Ela contém 13 linhas, cada uma representando um documento. Cada documento possui 9 características booleanas (1 significa presença do termo no documento e 0 significa ausência). O vetor S, exibido na última linha, contém o valor FEF de cada característica. As relevâncias DR, calculada pelo método MFDR e DR', calculada por cMFDR, são exibidas na penúltima e na última colunas, respectivamente.

O primeiro passo, para ambos os métodos, é computar os valores FEF das características e armazená-los no vetor S. Neste exemplo, adotamos o cálculo do valor FEF S_h de uma característica w_h como

$$S_h = 2 \times sum(w_h) + |diff(w_h)|, \tag{4.3}$$

Tabela 2: Base de dados de treinamento

$\mathcal{D}_{tr}$	w_1	w_2	w_3	w_4	w_5	w_6	w_7	w_8	w_9	C	MFDR DR	cMFDR DR'
d_1	0	0	0	0	0	0	0	1	0	A	16	16
d_2	1	0	0	1	1	0	0	1	1	A	71	14,2
d_3	1	0	0	1	0	0	0	1	1	A	61	15,3
d_4	0	0	0	0	0	1	0	1	0	A	33	16,5
d_5	0	1	0	0	0	1	1	1	0	A	46	11,5
d_6	0	0	0	1	0	0	0	0	0	B	23	23
d_7	0	0	0	1	0	1	0	0	1	B	51	17
d_8	1	0	1	0	1	0	0	1	0	B	43	10,8
d_9	0	0	0	1	0	1	1	0	0	B	50	16,7
d_{10}	0	0	0	1	0	1	1	0	1	B	61	15,3
d_{11}	1	0	1	1	1	1	1	0	0	B	77	12,8
d_{12}	0	0	0	1	1	0	0	0	1	B	44	14,7
d_{13}	1	0	0	1	0	1	0	0	0	B	51	17
S	11	3	6	23	10	17	10	16	11	-	-	-

onde $sum(w_h)$ é o número de documentos em que w_h está presente, independentemente de categoria, e $diff(w_h)$ é a diferença entre o número de documentos da categoria A em que w_h está presente e o número de documentos da categoria B em que w_h está presente. O próximo passo é calcular a relevância DR. Enquanto cMFDR realiza este cálculo usando a Equação 4.1, MFDR utiliza a soma dos valores FEF das características presentes no documento, como mencionado na Seção 3.2.3. Em seguida, um limiar define se um documento participará da seleção de características ou não. MFDR calcula um limiar global dado pela média das relevâncias DR de todos os documentos da base de treinamento. Neste exemplo, o valor deste limiar é $48,2$. Já cMFDR, calcula um limiar CT para cada categoria, considerando os valores de DR dos documentos pertencentes à categoria em questão (Equação 4.2). A ordem dos documentos na base de treinamento não afeta este cálculo. Para este exemplo, os valores dos limiares são $CT(A) \approx 14.7$ para a categoria A e $CT(B) \approx 15.9$ para a categoria B.

O passo final é a seleção das características. MFDR seleciona as f características de valor FEF mais alto naqueles documentos que possuem DR maior que o limiar global. cMFDR considera documentos que tem DR maior que o limiar da categoria à qual pertencem e seleciona as f características de maior valor FEF em cada documento. Neste exemplo, MFDR considera documentos com DR maior que $48,2$: d_2, d_3, d_7, d_9, d_{10}, d_{11} and d_{13}. Neste exemplo, estamos considerando $f = 1$. Assim, para d_2, MFDR seleciona a característica que possui maior valor no vetor S, que é w_4, e atualiza o vetor FS para $\{4\}$. Para d_3, w_4 é selecionada novamente e, como ela já está presente no vetor FS, nenhuma atualização é realizada. O mesmo acontece com d_7, d_9, d_{10}, d_{11} e d_{13}. Logo, o vetor FS usando MFDR é igual a $\{4\}$.

Já cMFDR considera os documentos da categoria A com DR maior que $CT(A) = 14,7$ (d_1, d_3 e d_4) e documentos da categoria B com DR maior que $CT(B) = 15,9$ (d_6, d_7, d_9 e d_{13}). Para d_1, w_8 é selecionada e o vetor FS é atualizado para $\{8\}$. Para d_3, cMFDR seleciona w_4 e atualiza o vetor FS para $\{4, 8\}$. Para d_4, w_6 é selecionada e o vetor FS é atualizado para $\{4, 6, 8\}$. cMFDR seleciona w_4 for d_6, d_7, d_9 e d_{13}. Como esta característica já está presente em FS, não é feita nenhuma atualização. Em seguida, uma nova base de dados é criada baseada nos índices das características selecionadas. A Tabela 3 exibe os resultados de ambos os métodos.

Tabela 3: Características selecionadas por MFDR e cMFDR.

	MFDR	cMFDR			
$\mathcal{D}'_{tr}$	w_4	w_4	w_6	w_8	C
d_1	0	0	0	1	A
d_2	1	1	0	1	A
d_3	1	1	0	1	A
d_4	0	0	1	1	A
d_5	0	0	1	1	A
d_6	1	1	0	0	B
d_7	1	1	1	0	B
d_8	0	0	0	1	B
d_9	1	1	1	0	B
d_{10}	1	1	1	0	B
d_{11}	1	1	1	0	B
d_{12}	1	1	0	0	B
d_{13}	1	1	1	0	B

Este exemplo prático mostrou que o limiar global utilizado por MFDR despreza documentos que pertencem a categorias que possuem características com baixo valor FEF. Então, o número de características selecionadas não foi suficiente para discriminar as duas categorias. Além disso, no MFDR, um documento que contém um grande número de características pode ser considerado relevante mesmo que todas as características apresentem um baixo valor FEF, por que o método calcula a relevância DR como a soma dos valores FEF das características presentes no documento. Este problema é ainda maior em categorias que possuem documentos pequenos, porque grande parte deste documentos não irão contribuir para a seleção de características. Por outro lado, cMFDR previne estes problemas calculando DR como a média dos valores FEF das características presentes no documento e computando limiares diferentes para cada categoria.

4.2 *Automatic Features Subsets Analyzer*

Os métodos MFD, MFDR e cMFDR facilitam a configuração em relação ao tamanho do vetor final de características através da utilização do parâmetro f, que representa o

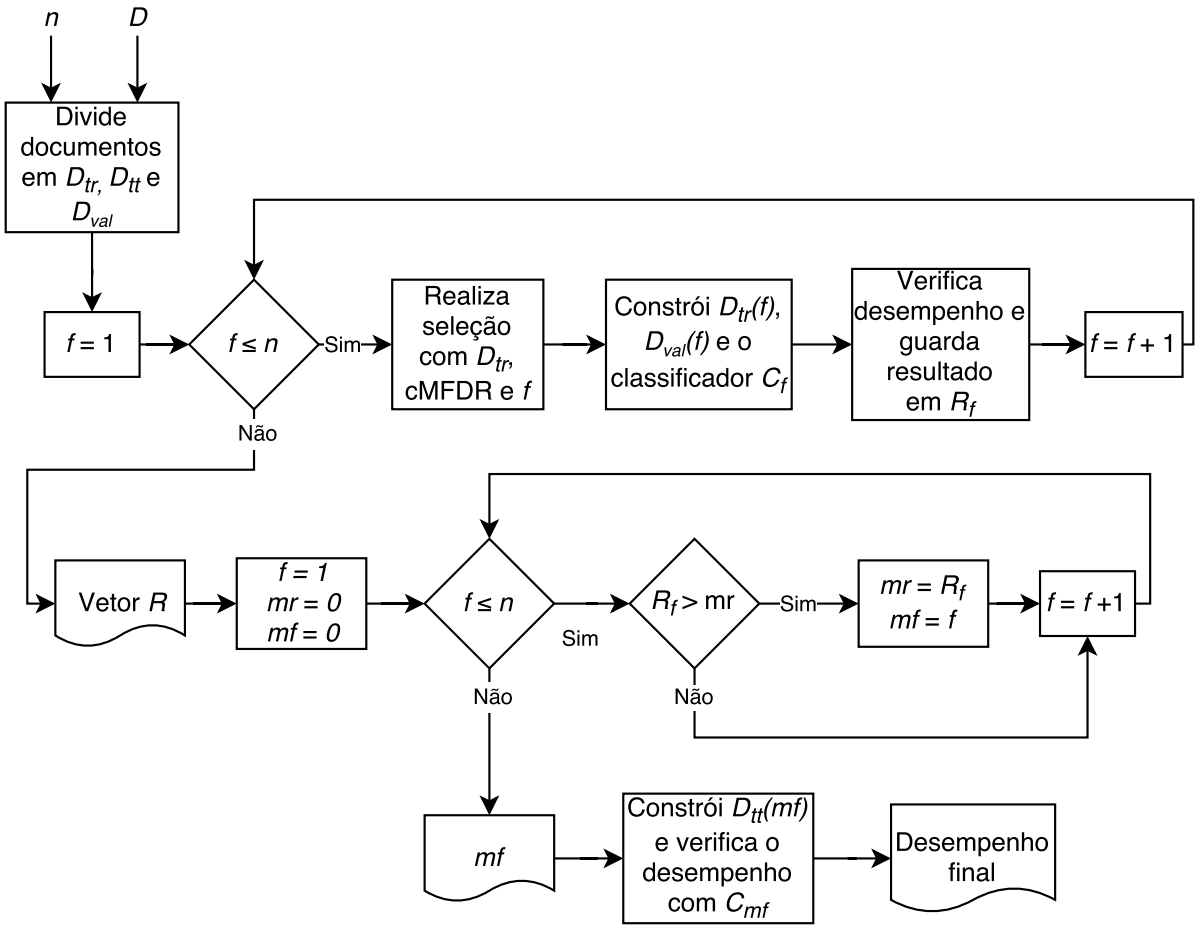

Figura 7: Fluxograma do método *Automatic Features Subsets Analyzer*. $\mathcal{D}$ é a base de dados, n é o número de subconjuntos a serem gerados.

número de características selecionadas por documento. Porém, estes métodos ainda exigem um esforço para determinar o valor do parâmetro f que produz o subconjunto com melhor desempenho de classificação. A configuração deste valor pode ter um custo impeditivo. Em um ambiente de produção, com novos documentos sendo apresentados, é necessário reanalisar a base de dados, de tempos em tempos, para incluir estes novos documentos na seleção de características. Sempre que esta análise for realizada, o usuário precisa verificar manualmente os subconjuntos gerados por estes métodos para determinar qual apresenta melhor desempenho classificatório. Deste modo, vários valores do parâmetro f precisam ser verificados, o que pode ser um processo demorado. Neste sentido, o método *Automatic Features Subsets Analyzer* (AFSA) (FRAGOSO; PINHEIRO; CAVALCANTI, 2016b) é proposto com o objetivo de automatizar a configuração deste parâmetro, determinando, de maneira guiada por dados, qual o subconjunto mais efetivo, ou seja, qual o melhor valor do parâmetro f. AFSA adota como estratégia o uso de uma de base de validação. O algoritmo gera um número de subconjuntos, usando o processo de FS de cMFDR, realiza uma pré avaliação do desempenho de cada subconjunto usando um conjunto de validação e, por fim, escolhe o valor de f que gera o subconjunto mais efetivo. Então, este valor

de f é passado para cMFDR que constrói o conjunto final de características usando os dados de treinamento. O desempenho do conjunto final é verificado usando um conjunto de testes. A Figura 7 apresenta o funcionamento do método AFSA.

O método requer um parâmetro n, que determina o número de subconjuntos a serem gerados e pré avaliados. A base de dados $\mathcal{D}$ é dividida em conjunto de treinamento $\mathcal{D}_{tr}$, conjunto de validação $\mathcal{D}_{val}$ e conjunto de teste $\mathcal{D}_{tt}$. Em seguida, os dados de treinamento são passados ao algoritmo do cMFDR, que realiza a seleção de características (conforme descrito na Seção 4.1) e gera um subconjunto de treinamento $\mathcal{D}_{tr}(f)$ e outro de validação $\mathcal{D}_{val}(f)$ para cada valor fornecido para o parâmetro f, de 1 até n. Após isto, o algoritmo constrói n classificadores $\mathcal{C}_f$, usando $\mathcal{D}_{tr}(f)$, um para cada valor de f. Na sequência, cada conjunto é pré-avaliado utilizando $\mathcal{C}_f$ e seu respectivo conjunto de validação $\mathcal{D}_{val}(f)$ e o resultado é armazenado na posição f do vetor R. No próximo passo, cada valor do vetor R é analisado em busca do subconjunto com melhor desempenho de classificação. O valor de f que produziu o melhor desempenho, ou seja, A posição do vetor R que guarda o melhor desempenho, é armazenado em mf. O algoritmo então gera o conjunto de testes $\mathcal{D}_{tt}(mf)$ e o utiliza, junto com o classificador $\mathcal{C}_{mf}$ para realizar a avaliação da performance final.

4.3 Considerações Finais

Este capítulo apresentou os métodos propostos neste trabalho: *Category-dependent Maximum f Features per Document - Reduced* e *Automatic Features Subsets Analyzer*. O primeiro trata cada categoria de maneira diferenciada para determinar quais documentos participam da selção de características. O segundo automatiza o processo de escolha do melhor valor para o parâmetro f, que discrimina quantas características devem ser selecionadas por documento. O próximo capítulo irá exibir os experimentos realizados com os métodos propostos.

5 Experimentos

Neste capítulo são descritos os experimentos realizados para validar os métodos propostos. A Seção 5.1 apresenta as configurações dos experimentos e a Seção 5.2 exibe os resultados dos métodos propostos. Na Seção 5.3, são descritas as conclusões do capítulo.

5.1 Configurações dos Experimentos

Esta seção descreve as configurações para execução dos experimentos. O algoritmo *Naïve Bayes Multinomial*, adotado nos experimentos, trabalha com vetores de características contendo a frequência dos termos nos documentos, conforme a técnica *Term Frequency*, apresentada na Seção 2.1.1.2. Por esta razão, esta foi a abordagem de classificador bayesiano empregada nos experimentos. Adicionalmente, o classificador SVM foi utilizado como base de comparação para o tempo de execução dos métodos propostos. A escolha do SVM para este comparativo deveu-se à sua reconhecida habilidade de tratar de problemas de alta dimensionalidade (JOACHIMS, 1998; LEOPOLD; KINDERMANN, 2002). Assim, nos experimentos, SVM é executado sem uso de nenhum tipo de redução de dimensionalidade.

A validação cruzada estratificada (do inglês, *stratified cross-validation*) foi utilizada como método para estimativa de acurácia. Esta técnica é adotada para avaliar a capacidade de generalização de um modelo, a partir de um conjunto de dados. Neste trabalho utilizou-se a variação *10 fold stratified cross-validation*, na qual a base de dados $\mathcal{D}$ é particionada em 10 subconjuntos (*folds*), de tamanhos semelhantes, mantendo a proporção de documentos por categorias equivalente à proporção encontrada no conjunto original. Então, são construídos 10 classificadores, cada um utilizando uma parcela dos *folds* para treinamento e outra parcela para realizar o teste do mesmo, de modo a gerar diferentes combinações dos *folds*. A acurácia final é dada pela média das acurácias obtidas em cada uma das 10 execuções (KOHAVI et al., 1995). Nos experimentos realizados com o método cMFDR, nove partições foram utilizadas para treinamento e uma partição foi utilizada para teste. Os experimentos executados com AFSA utilizaram oito partições para treinamento, uma para validação e uma para teste.

O teste estatístico chamado *t-test* (PINHEIRO et al., 2012) foi utilizado para verificar a eficácia dos métodos propostos. Este teste é aplicado na média e no desvio padrão dos valores de *Micro-F1* e *Macro-F1* obtidos por cada método em experimento utilizando a validação cruzada. Nos experimentos deste trabalho, a seguinte convenção foi adotada:

- *P-value* menor ou igual a 0,01 indica uma forte evidência de que um método possui

maior eficácia que outro;

- *P-value* maior que 0,01 e menor que 0,05 indica uma evidência fraca de que um método é mais eficaz que o outro;
- *P-value* maior que 0,05 indica que não existe evidência de diferença de eficácia entre os métodos.

Para os experimentos, foi empregada uma máquina dedicada exclusivamente para este fim com processador Intel Core i5 de 2.7 Giga Hertz, memória RAM de 8 Gigabytes e sistema operacional OSX El Capitan. As seções que seguem detalham as bases de dados utilizadas e funções de avaliação de características (FEFs) utilizadas nos experimentos.

5.1.1 Bases de Dados

Bases de dados são conjuntos de documentos previamente categorizados. Estes conjuntos são essenciais para o desenvolvimento e avaliação de sistemas de categorização de textos. Os experimentos foram realizados utilizando quatro bases de dados amplamente mencionadas na literatura de TC (FORMAN, 2003; FENG et al., 2015; PINHEIRO; CAVALCANTI; REN, 2015; TANG; KAY; HE, 2016; UĞUZ, 2011; YANG et al., 2012): 20 Newsgroup e subconjuntos das bases WebKB, Reuters-21578 e TDT2. Estas bases de dados apresentam diferentes tamanhos, em termos de documentos e vocabulário, diferentes proporções de documentos por categoria e conteúdos diversos. A seguir, são apresentados detalhes de cada uma das bases utilizadas. A Tabela 4 sintetiza as informações sobre as bases de dados.

- WebKB
  O *WebKB corpus* contém por documentos da *web* de quatro faculdades norte americanas obtidos em 1997. O conjunto original é composto por 7 categorias e 8.282 documentos. Entretanto, neste trabalho foi utilizado um subconjunto contendo apenas 4 categorias (*course*, *faculty*, *project*, *student*) e 4.199 documentos e um vocabulário de 7.770 termos. Esta configuração ignora categorias com poucos documentos e é utilizada em vários trabalhos (MCCALLUM; NIGAM, 1998; YANG et al., 2011). A distribuição de documentos pelas categorias da base de dados é bem heterogênea, com a maior categoria representando aproximadamente 39% do tamanho total da base enquanto a menor, possui 12% dos documentos. A base de dados é disponibilizada pré-processada[1]. Os documentos da base passaram por pré-processamento com remoção de pontuação e números, formatação de todas as letras em caixa baixa, a remoção termos com duas ou menos letras, remoção de *stopwords* (SALTON; MC-

[1] Disponível em http://ana.cachopo.org/datasets-for-single-label-text-categorization.

GILL, 1971) e *stemming*, utilizando o algoritmo *Porter Stemmer* (PORTER, 1980). Os títulos dos documentos foram simplesmente adicionados no início do mesmo.

- *20 Newsgroup*
  20 Newsgroup[1] é uma base de dados formada por documentos extraídos de grupos de discussão da Usenet. Algumas categorias tratam de assuntos altamente correlacionados com o assunto de outras, como é o caso de *comp.sys.ibm.pc.hardware* e *comp.sys.mac.hardware*. Também existem categorias que tratam de assuntos completamente distintos, como *misc.forsale* e *soc.religion.christian*. O balanceamento dos documentos nas 20 categorias da base é praticamente equânime. Os documentos desta base de dados passaram pelo mesmo pré-processamento realizado com *WebKB*.

- Reuters 10
  Esta base de dados é um subconjunto da coleção *Reuters-21578*[2], que é uma das bases mais utilizadas em trabalhos sobre TC (DEBOLE; SEBASTIANI, 2005). A base é composta por documentos coletados do *Reuters newswire* de 1987 e apresenta 135 categorias. Entretanto, neste trabalho foi adotado um subconjunto composto pelas 10 maiores categorias da base. O subconjunto *Reuters 10* contém 9.980 documentos e seu vocabulário abarca por 10.987 termos. Uma abordagem semelhante também foi utilizada em trabalhos de TC (CHANG; CHEN; LIAU, 2008; CHEN et al., 2009; YANG et al., 2011). A distribuição dos documentos é bastante desbalanceada, apresentando categorias representando desde 2,3% até 39% do tamanho total da base. Nesta base foram aplicados os seguintes procedimentos de pré-processamento: remoção termos com duas ou menos letras, remoção de *stopwords* (utilizando a lista de palavras listadas no Apêndice A) e *stemming*, com o algoritmo *Iterated Lovins Stemmer* (LOVINS, 1968).

- *TDT2 top 30*
  Esta base é um subconjunto da base original, *TDT2*, que é formada por dados coletados de 6 agências de notícias, no primeiro semestre de 1998. Do conjunto original, que contém 11.201 documentos divididos em 96 categorias, foram removidos os documentos etiquetados com mais de uma categoria e as 66 categorias com menor número de documentos. Ao fim, *TDT2 top 30*[3] é composto por 9.394 documentos espalhados por 30 categorias. Outros trabalhos (CAI et al., 2008; SAHA; SINDHWANI, 2010; PINHEIRO; CAVALCANTI; REN, 2015) utilizam esta configuração. O vocabulário é composto por 36.093 termos. Esta base de dados é bastante desbalanceada, apresentando categorias que variam de 0,5% a 19,6% do tamanho total da base. A base foi pré-processada utilizando os mesmo procedimentos adotados com a base *Reuters 10*, listados anteriormente.

[2] Disponível em http://disi.unitn.it/moschitti/corpora.htm.
[3] Disponível em http://www.cad.zju.edu.cn/home/dengcai/Data/TextData.html.

Tabela 4: Descrição das bases de dados.

Base	Nº de categorias	Nº de documentos	Nº de termos
WebKB	4	4.199	7.770
Reuters 10	10	9.980	10.987
20 Newsgroup	20	18.821	70.216
TDT2	30	9.394	36.093

5.1.2 Funções de Avaliação de Características

Funções de avaliação de características (FEF) são funções que baseiam-se em medidas estatísticas para determinar a importância de uma característica em relação a uma base de dados. O cálculo leva em conta as probabilidades de um termo w em relação aos documentos da categoria c_j com a finalidade de calcular o poder discriminatório deste termo w. Três FEFs que apresentam bons resultados em problemas multi-classe (PINHEIRO et al., 2012) foram escolhidas para a realização dos experimentos deste trabalho. Uma breve explanação sobre elas bem como suas fórmulas são apresentadas a seguir. A seguinte nomenclatura é adotada nas fórmulas:

- $P(w|c_j)$ é a probabilidade de que um documento da categoria c_j contenha o termo w;
- $P(\bar{w}|c_j)$ é a probabilidade de que um documento da categoria c_j não contenha o termo w;
- $P(w|\bar{c_j})$ é a probabilidade de que um documento que não pertença à categoria c_j contenha o termo w;
- $P(\bar{w}|\bar{c_j})$ é a probabilidade de que um documento que não pertença à categoria c_j não contenha o termo w;
- $P(w)$ é a probabilidade de que um documento qualquer contenha o termo w;
- $P(c_j)$ é a probabilidade de que um documento qualquer pertença à categoria c_j.

Bi-Normal Separation (BNS) é uma métrica proposta por (FORMAN, 2003), que utiliza a função de probabilidade acumulativa inversa de uma distribuição normal padrão (F^{-1}) para calcular a distância entre dois limiares. Sua fórmula é apresentada na Equação 5.1.

$$\mathrm{BNS}(w) = \sum_{j=1}^{Q} \left| F^{-1}(P(w|c_j)) - F^{-1}(P(w|\bar{c_j})) \right| \tag{5.1}$$

Class Discriminating Measure (CDM) é uma FEF originada de uma simplificação da FEF *Multi-class Odss Ratio*. Esta, por sua vez, é uma variação de *Odds Ratio* para

trabalhar em problemas multi-classe (CHEN et al., 2009). A Equação 5.2 descreve a fórmula de CDM.

$$\mathrm{CDM}(w) = \sum_{j=1}^{Q} \left| \log \frac{P(w|c_j)}{P(w|\bar{c_j})} \right| \tag{5.2}$$

Chi-Squared ou X^2 *Statistic* mede o grau de dependência de um termo w em relação à classe c_j. O valor zero significa que w e c_j são totalmente independentes (YANG; PEDERSEN, 1997). A fórmula de CHI é demonstrada na Equação 5.3.

$$\mathrm{CHI}(w) = \sum_{j=1}^{Q} \frac{\left[P(w|c_j)P(\bar{w}\bar{c_j}) - P(w|\bar{c_j})P(\bar{w}|\bar{c_j})\right]^2}{P(w)P(\bar{w})P(c_j)P(\bar{c_j})} \tag{5.3}$$

5.2 Resultados dos experimentos

Esta seção apresenta os resultados dos experimentos realizados com os métodos propostos neste trabalho. As seções 5.2.1 e 5.2.2 expõe análises de desempenho dos métodos cMFDR e AFSA, respectivamente, bem como comparações do desempenho dos métodos com métodos anteriores. A Seção 5.2.3 exibe uma análise do tempo de execução dos métodos.

5.2.1 Resultados obtidos com cMFDR

Neste trabalho, método cMFDR teve seu desempenho avaliado em comparação com o método MFDR devido a este ter apresentado resultados superiores aos métodos do estado da arte (PINHEIRO; CAVALCANTI; REN, 2015).

Tanto cMFDR, como seu predecessor MFDR, requerem um inteiro positivo para o parâmetro f, que determina quantas características devem ser selecionadas por documento. Em Pinheiro, Cavalcanti e Ren (2015), foi demonstrado que os melhores resultados de MFDR são alcançados com f assumindo valores entre 1 e 10. Para cMFDR, foram realizados experimentos com objetivo de determinar os melhores valores para o parâmetro f, onde valores de 1 a 20 foram analisados. Verificou-se que, para valores maiores que 10, não se percebe melhoria na performance. Algumas poucas combinações de base de dados e FEF apresentaram um desempenho levemente superior com $f > 10$. Entretanto, nestes casos, o tamanho do vetor final de características (m) aumenta significativamente. Assim, apenas valores de f entre 1 e 10 foram utilizados nos experimentos, para ambos os métodos.

A Figura 8 exibe a quantidade de características selecionadas, ou seja, o tamanho do vetor de características, para cada valor empregado para o parâmetro f, usando os métodos MFDR e cMFDR combinados com as quatro bases de dados e três FEFs. De uma maneira geral, a FEF que resultou nos maiores vetores de características selecionadas,

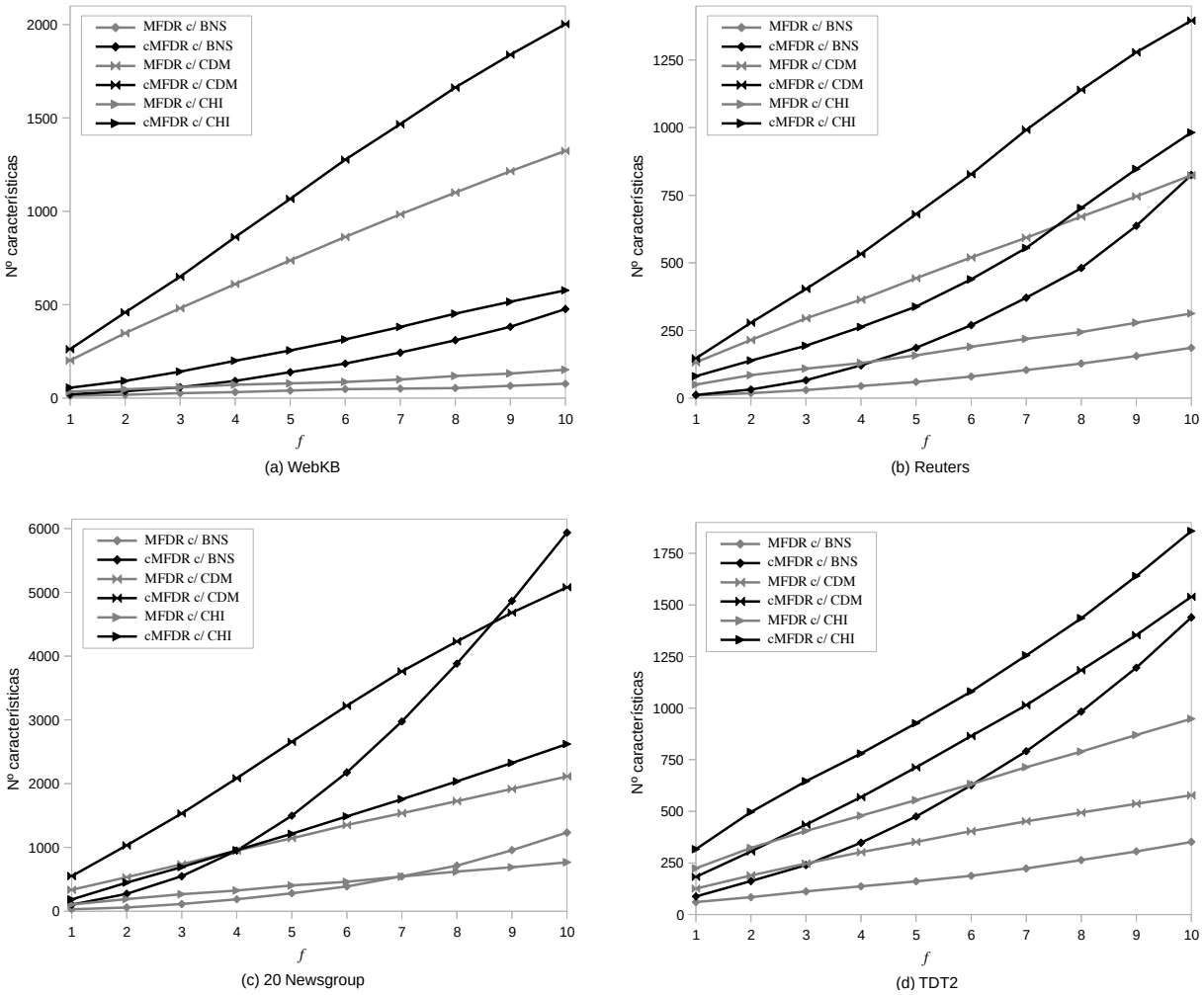

Figura 8: Número de características selecionadas para cada base de dados, MFDR e cMFDR usando as três FEFs com f variando de 1 a 10.

independentemente de base de dados ou método de seleção utilizado, foi CDM. A única exceção foi a base TDT2, na qual CHI resultou em mais características selecionadas. Independentemente de FEF ou base de dados, cMFDR seleciona mais características que MFDR, principalmente para valores maiores de f.

As Figuras 9, 10, 11, e 12 ilustram o desempenho, em termos de *Micro-F1* e *Macro-F1*, de MFDR e cMFDR para as bases *WebKB*, *Reuters*, *20 Newsgroup* e *TDT2*, respectivamente. Para todos os valores de f avaliados na base de dados *WebKB* (Figura 9), cMFDR apresentou um desempenho superior ao MFDR, utilizando BNS e CHI, tanto para *Micro-F1* quanto para *Macro-F1*. Usando a FEF CDM, cMFDR apresentou resultados superiores de *Micro-F1* para valores de f entre 1 e 6. Com $f > 6$, o desempenho dos métodos foi semelhante. Em termos de *Macro-F1*, cMFDR foi só não foi superior ao MFDR para $f = 8$ e $f = 9$.

O desempenho dos métodos em termos de *Mircro-F1 Reuters* (Figura 10) apresentou comportamento semelhante ao desempenho na base de dados *WebKB*. Usando CDM, os

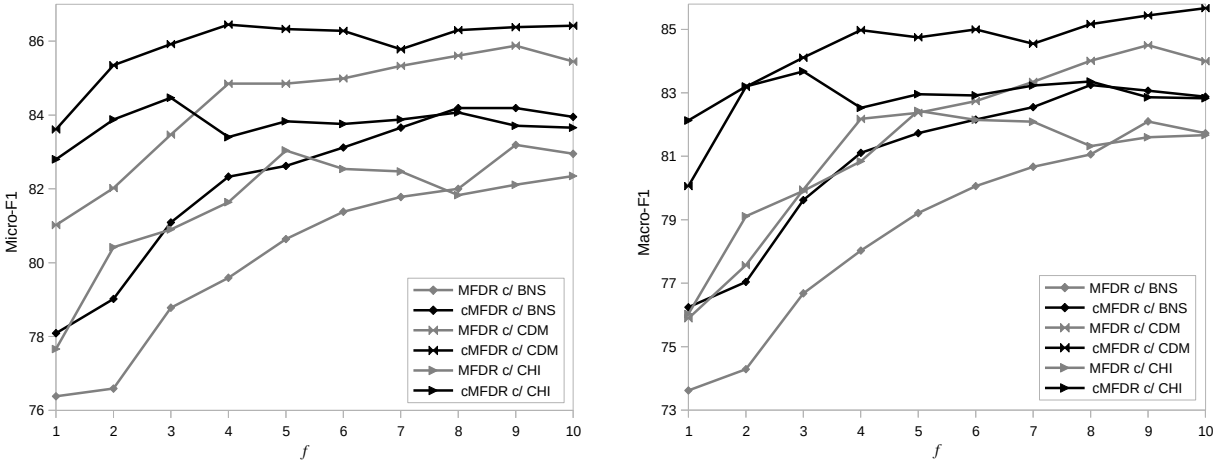

Figura 9: Resultados de MFDR e cMFDR para a base *WebKB* usando cada FEF.

métodos apresentaram performances equivalentes para $f > 7$. Para valores menores de f ou usando as outras FEFs, cMFDR teve desempenho superior ao MFDR. Com respeito a *Macro-F1*, os métodos atingem seus melhores resultados com f em torno de 3 e com $f \geqslant 6$ há uma deterioração do desempenho, independente de FEF.

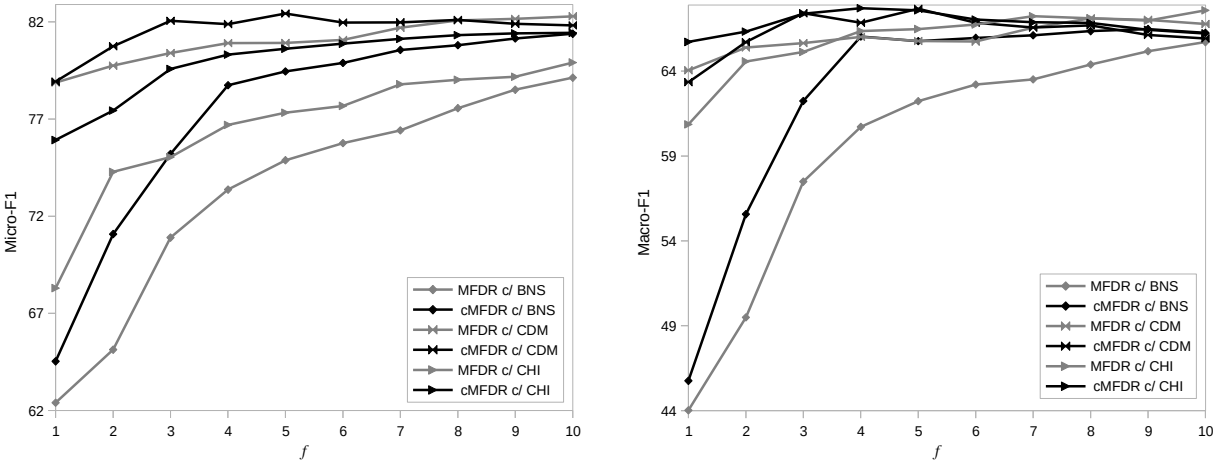

Figura 10: Resultados de MFDR e cMFDR para a base *Reuters* usando cada FEF.

O método cMFDR apresentou melhores resultados que MFDR na base *20 Newsgroup* (Figura 11), tanto em termos de *Micro-F1* quanto de *Macro-F1*. A superioridade foi observada comparando-se os resultados utilizando as três FEFs e todos os valores de f. A FEF que produziu os melhores resultados foi CDM.

Os resultados dos experimentos com a base *TDT2* (Figura 12) mostram uma superioridade do método cMFDR em realação ao MFDR, usando as três FEFs. Os valores de *Micro-F1* e *Macro-F1* estabilizam, usando cMFDR, com $f \geq 2$, enquanto com MFDR apresenta um crescimento mais lento do desempenho. Para *Micro-F1*, BNS e CDM apresentam bons resultados com $f \geq 6$ e CHI estabiliza apenas com $f = 9$. Os resultados de *Macro-F1* de cMFDR são superiores aos resultados de MFDR para as três FEFs e

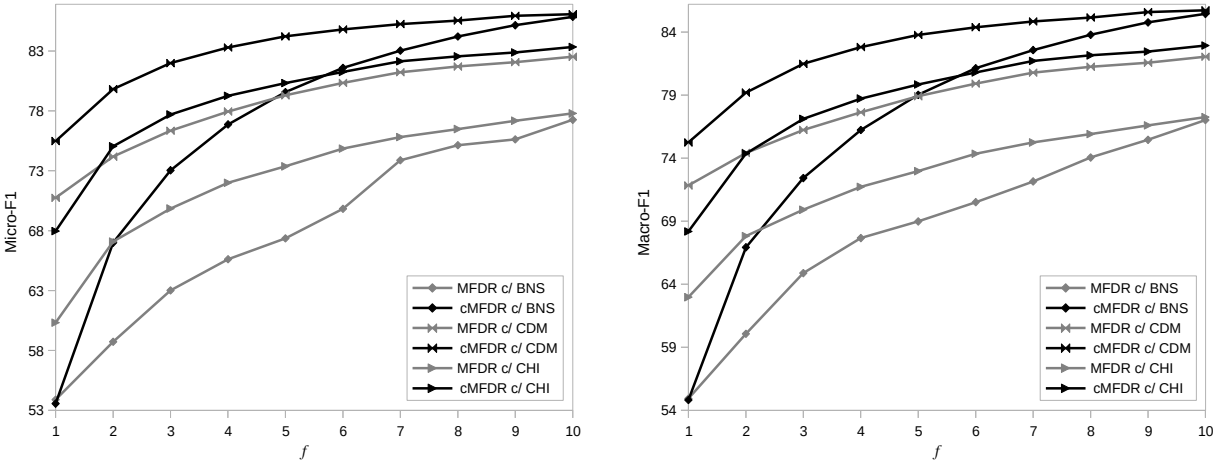

Figura 11: Resultados de MFDR e cMFDR para a base *20 Newsgroup* usando cada FEF.

qualquer valor de f, de modo mais perceptível com valores baixos de f.

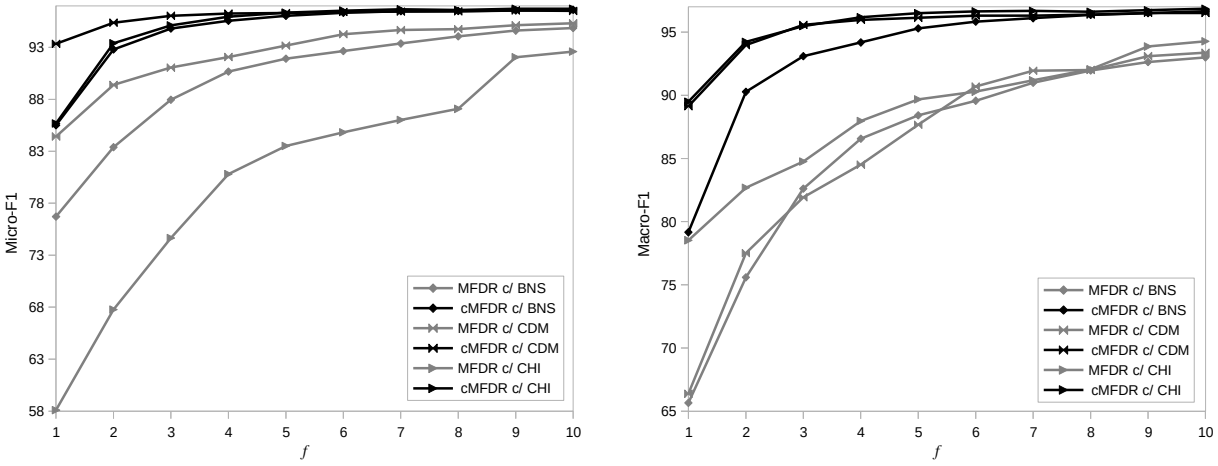

Figura 12: Resultados de MFDR e cMFDR para a base *TDT2* usando cada FEF.

As Tabelas 5 e 6 apresentam os melhores resultados para *Micro-F1* e *Macro-F1*, respectivamente, de ambos os métodos, para cada combinação de base de dados e FEF. De uma maneira geral, os melhores resultados de cMFDR são alcançados com valores mais altos de m (tamanho do vetor de características), em comparação com MFDR. Porém, em vários casos, confrontando o melhor resultado de MFDR com resultados de cMFDR alcançados valores mais baixos de f (que geram vetores com m também mais baixos) ainda se percebe a vantagem de cMFDR. Por exemplo, para a base *TDT2* e a FEF CHI, ambos os métodos atingem seus melhores resultados de *Micro-F1* com $f = 10$. Com esta configuração MFDR produz um vetor de características de tamanho $m = 948$ enquanto cMFDR gera um vetor de características de tamanho $m = 1.858$. Entretanto, com $f = 2$, gerando um vetor de características de tamanho $m = 497$, cMFDR apresenta desempenho superior à melhor configuração de MFDR nesta base. O mesmo ocorre com outras combinações de base de dados e FEF. Em uma análise mais pormenorizada dos

resultados (Apêndice B), percebe-se que em mais da metade das possíveis combinações de base de dados e FEF, cMFDR precisou de menos características que MFDR para superar o desempenho deste. Então, cMFDR não necessita um maior número de características para obter bons resultados, mas com mais características ele atinge um desempenho ainda melhor que MFDR.

Tabela 5: Melhores resultados para *Micro-F1* de cMFDR e MFDR, com o valor de f que gerou o resultado, o tamanho m do vetor final de características.

Base de dados	FEF	MFDR			cMFDR		
		f	m	Micro-F1	f	m	Micro-F1
WebKB	BNS	9	68	83,19 (2,20)	9	382	84,19 (0,78)
	CDM	9	1215	85,88 (1,66)	4	862	86,45 (2,05)
	CHI	5	79	83,04 (1,66)	3	142	84,47 (1,68)
Reuters	BNS	10	186	79,13 (1,46)	10	825	81,39 (1,03)
	CDM	10	824	82,29 (0,77)	5	680	82,43 (0,72)
	CHI	10	313	79,91 (0,99)	10	982	81,44 (1,15)
20 Newsgroup	BNS	10	1190	77,26 (1,14)	10	5937	85,85 (0,63)
	CDM	10	1964	82,52 (0,54)	10	5081	86,07 (0,45)
	CHI	10	758	77,79 (0,94)	10	2621	83,34 (0,61)
TDT2	BNS	10	352	94,86 (0,59)	10	1440	96,61 (0,44)
	CDM	10	578	95,32 (0,74)	9	1354	96,55 (0,40)
	CHI	10	948	92,60 (1,67)	9	1640	96,72 (0,36)

Tabela 6: Melhores resultados para *Macro-F1* de cMFDR e MFDR, com o valor de f que gerou o resultado, o tamanho m do vetor final de características.

Base de dados	FEF	MFDR			cMFDR		
		f	m	Macro-F1	f	m	Macro-F1
WebKB	BNS	9	68	82,10 (2,05)	9	382	83,07 (0,99)
	CDM	9	1215	84,50 (2,08)	4	862	84,98 (2,14)
	CHI	5	79	82,43 (1,77)	3	142	83,68 (1,88)
Reuters	BNS	10	186	65,71 (2,60)	10	825	66,26 (1,48)
	CDM	8	671	67,11 (1,60)	5	680	67,68 (1,08)
	CHI	10	313	67,58 (1,56)	10	982	66,22 (2,16)
20 Newsgroup	BNS	10	1190	77,02 (1,26)	10	5937	96,65 (0,35)
	CDM	10	1964	82,05 (0,60)	10	5081	96,51 (0,24)
	CHI	10	758	77,26 (0,95)	10	2621	96,85 (0,47)
TDT2	BNS	10	352	93,00 (1,05)	10	1440	85,44 (0,64)
	CDM	10	578	93,37 (1,04)	9	1354	85,58 (0,53)
	CHI	10	948	94,27 (1,18)	9	1640	82,45 (0,74)

Para realizar um comparativo da performance de cMFDR frente ao MFDR, foi utilizado o teste estatístico *t-test*. A análise dos resultados do teste revela que o desempenho de cMFDR é superior ou semelhante a MFDR em 100% dos casos, tanto para *Micro-F1* quanto *Macro-F1*. *WebKB* foi a base de dados na qual cMFDR apresentou desempenho mais fraco em termos de *Micro-F1*, enquanto *Reuters* foi a base em que cMFDR teve mais

Tabela 7: Resultados do *t-test* comparando o desempenho de cMFDR e MFDR. Convenção adotada: "$\gg$"e "$\ll$"significam uma forte evidência de que um método apresenta maior ou menor valor de efetividade que outro, respectivamente; ">"e "<"significam uma fraca evidência de que um método apresenta maior ou menor valor de efetividade que outro, respectivamente; "$\sim$"significa que a diferença entre os métodos não é relevante.

Base de dados		Micro-F1			Macro-F1		
		BNS	CDM	CHI	BNS	CDM	CHI
WebKB	$f = 1$	$\sim$	$>$	$\gg$	$>$	$\gg$	$\gg$
	$f = 2$	$>$	$\gg$	$\gg$	$>$	$\gg$	$\gg$
	$f = 3$	$>$	$\gg$	$\gg$	$>$	$\gg$	$\gg$
	$f = 4$	$>$	$\sim$	$>$	$>$	$\gg$	$>$
	$f = 5$	$\sim$	$\sim$	$\sim$	$>$	$\gg$	$\sim$
	$f = 6$	$\sim$	$\sim$	$\sim$	$>$	$>$	$\sim$
	$f = 7$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$	$\sim$
	$f = 8$	$>$	$\sim$	$\gg$	$>$	$\sim$	$>$
	$f = 9$	$\sim$	$\sim$	$>$	$\sim$	$\sim$	$\sim$
	$f = 10$	$\sim$	$\sim$	$>$	$\sim$	$>$	$\sim$
Reuters	$f = 1$	$\gg$	$\sim$	$\gg$	$\sim$	$\sim$	$\gg$
	$f = 2$	$\gg$	$>$	$\gg$	$\gg$	$\sim$	$>$
	$f = 3$	$\gg$	$\gg$	$\gg$	$\gg$	$>$	$>$
	$f = 4$	$\gg$	$>$	$\gg$	$\gg$	$\sim$	$>$
	$f = 5$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\sim$
	$f = 6$	$\gg$	$>$	$\gg$	$\gg$	$\sim$	$\sim$
	$f = 7$	$\gg$	$\sim$	$\gg$	$\gg$	$\sim$	$\sim$
	$f = 8$	$\gg$	$\sim$	$\gg$	$\sim$	$\sim$	$\sim$
	$f = 9$	$\gg$	$\sim$	$\gg$	$\sim$	$\sim$	$\sim$
	$f = 10$	$\gg$	$\sim$	$\gg$	$\sim$	$\sim$	$\sim$
20 Newsgroup	$f = 1$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 2$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 3$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 4$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 5$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 6$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 7$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 8$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 9$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 10$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
TDT2	$f = 1$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 2$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 3$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 4$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 5$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 6$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 7$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 8$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 9$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$
	$f = 10$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$	$\gg$

dificuldades para superar MFDR em termos de *Macro-F1*. Na base *WebKB*, cMFDR obteve desempenho de *Micro-F1* semelhante a MFDR em 16 casos e superior em nos outros 14 casos. Já a observação dos resultados desta base de dados em termos de *Macro-F1* reportou desempenhos semelhantes dos métodos em 11 casos e melhor desempenho de cMFDR em 19 casos. Na base *Reuters*, para *Micro-F1*, cMFDR foi superior a MFDR em 25 casos e os métodos obtiveram desempenhos equivalentes em 5 casos. Para *Macro-F1*, cMFDR apresentou resultados melhores em 12 casos e em 18 casos não foi detectada diferença significativa entre os resultados dos métodos. A comparação dos desempenhos dos métodos nas bases de dados *20 Newsgroup* e *TDT2* torna a superioridade de cMFDR fica mais evidente. Nestas duas bases, o teste estatístico indica um melhor desempenho de cMFDR em 100% dos casos, tanto para *Micro-F1* quanto para *Macro-F1*. Uma sumarização dos dados aponta que, para *Micro-F1*, cMFDR apresentou desempenho superior a MFDR em 82,5% dos casos e semelhante a MFDR em 17,5% dos casos. Para *Macro-F1*, cMFDR obteve desempenho superior em 75,8% dos casos e em 24,2% dos casos os métodos apresentaram desempenho sem diferenças significativas. A Tabela 7 apresenta todos os resultados do comparativo de desempenho entre cMFDR e MFDR, realizado através do *t-test*.

5.2.2 Resultados obtidos com AFSA

O método AFSA requer um inteiro maior que zero para o parâmetro n. Este parâmetro define quantos subconjuntos serão gerados e pré-avaliados. Cada subconjunto é gerado através da utilização do algoritmo do cMFDR, passando valores de 1 a té n para o parâmetro f de cMFDR. Portanto, nos experimentos adotou-se $n = 10$, pelos mesmo motivos da escolha de valores para o parâmetro f para cMFDR, expostos anteriormente na Seção 5.2.1.

O desempenho do método foi comparado com o desempenho de cMFDR, devido ao fato de que este apresenta resultados superiores aos demais métodos. Como AFSA retorna apenas um subconjunto de características (o mais efetivo dos n gerados), o desempenho de classificação deste subconjunto é comparado com o desempenho do subconjunto mais efetivo de cMFDR. Esta comparação é exibida nas Tabelas 8 e 9, para *Micro-F1* e *Macro-F1*, respectivamente.

Desta comparação, pode-se perceber que a quantidade de características selecionadas por AFSA é menor que aquela selecionada pelo melhor subconjunto de cMFDR na maioria dos casos. Nas tabelas ainda pode-se ver que, a performance dos métodos é semelhante, tanto em termos de *Micro-F1* quanto de *Macro-F1*. cMFDR, entretanto, apresenta um desempenho levemente superior na maioria dos casos. Uma exceção é com *Reuters* e CHI, quando AFSA obteve desempenho de *Macro-F1* superior a cMFDR.

As Figuras 13, 14, 15 e 16 ilustram o desempenho do método AFSA, em termos *Micro-F1* e *Macro-F1*, para as quatro bases de dados. Como AFSA retorna apenas um

Tabela 8: Resultados de *Micro-F1* de AFSA e melhores resultados de cMFDR, com o número de características selecionadas (m) e parâmetro f.

Dataset	FEF	cMFDR			AFSA		
		f	m	Micro-F1	f	m	Micro-F1
WebKB	BNS	9	382 (9.69)	84.19 (0.78)	9	366 (13.92)	84.04 (1.01)
	CDM	4	862 (11.86)	86.45 (2.05)	5	991 (25.01)	86.07 (1.47)
	CHI	3	142 (3.33)	84.47 (1.68)	3	137 (5.58)	84.35 (1.84)
Reuters	BNS	10	825 (36.82)	81.39 (1.03)	10	781 (25.15)	81.05 (0.95)
	CDM	5	680 (9.72)	82.43 (0.72)	6	775 (9.97)	82.09 (0.71)
	CHI	10	982 (13.06)	81.44 (1.15)	8	683 (9.03)	81.23 (0.98)
20 Newsgroup	BNS	10	5937 (70.39)	85.85 (0.63)	10	5544 (93.81)	85.42 (0.67)
	CDM	10	5081 (21.86)	86.07 (0.45)	10	4707 (29.56)	85.44 (0.62)
	CHI	10	2621 (19.04)	83.34 (0.61)	10	2521 (25.42)	83.09 (0.62)
TDT2	BNS	10	1440 (19.69)	96.61 (0.44)	9	1089 (86.12)	96.38 (0.58)
	CDM	9	1354 (10.12)	96.55 (0.40)	10	1456 (10.77)	96.39 (0.38)
	CHI	9	1640 (15.26)	96.72 (0.36)	10	1805 (28.56)	96.68 (0.42)

Tabela 9: Resultados de *Micro-F1* de AFSA e melhores resultados de cMFDR, com o número de características selecionadas (m) e parâmetro f.

Dataset	FEF	cMFDR			AFSA		
		f	m	Macro-F1	f	m	Macro-F1
WebKB	BNS	8	310 (8.11)	83.25 (1.68)	9	366 (13.92)	83.00 (1.45)
	CDM	10	2003 (13.27)	85.67 (1.02)	5	991 (25.01)	84.58 (1.83)
	CHI	3	142 (3.33)	83.68 (1.88)	3	137 (5.58)	83.44 (1.91)
Reuters	BNS	9	637 (35.35)	66.26 (1.41)	10	781 (25.15)	65.60 (1.36)
	CDM	5	680 (9.72)	67.68 (1.08)	6	775 (9.97)	66.84 (1.77)
	CHI	4	263 (4.03)	67.71 (1.58)	8	683 (9.03)	66.34 (2.03)
20 Newsgroup	BNS	10	5937 (70.39)	85.44 (0.64)	10	5544 (93.81)	85.03 (0.67)
	CDM	10	5081 (21.86)	85.72 (0.51)	10	4707 (29.56)	85.07 (0.71)
	CHI	10	2621 (19.04)	82.93 (0.66)	10	2521 (25.42)	82.67 (0.70)
TDT2	BNS	10	1440 (19.69)	96.65 (0.35)	9	1089 (86.12)	96.23 (0.30)
	CDM	9	1354 (10.12)	96.52 (0.41)	10	1456 (10.77)	96.28 (0.38)
	CHI	10	1558 (19.49)	96.85 (0.47)	10	1805 (28.56)	96.65 (0.49)

subconjunto de características, o desempenho da classificação deste é representado nos gráficos por uma linha horizontal pontilhada. O desempenho do cMFDR é descrito por uma linha contínua.

Com a base *WebKB* (Figura 13), o método AFSA apresentou resultado médio superior a cMFDR em 21 dos 30 casos de *Micro-F1*. A FEF em que AFSA apresentou maior vantagem foi CHI, obtendo desempenho superior a cMFDR em 9 dos 10 casos. Com relação a *Macro-F1*, a vantagem de AFSA se deu também em 21 dos 30 casos, com a mesma distribuição entre as FEFs.

Para a base de dados *Reuters* (Figura 14), AFSA apresenta superioridade em 23 das 30 configurações para *Micro-F1*. Usando as FEFs BNS e CDM, o método AFSA obteve desempenho superior a cMFDR em 8 dos 10 casos, para cada FEF. Usando a FEF CHI, AFSA foi superior em 7 dos 10 casos. AFSA obteve desempenho superior a cMFDR em 12

dos 30 casos de *Macro-F1*.

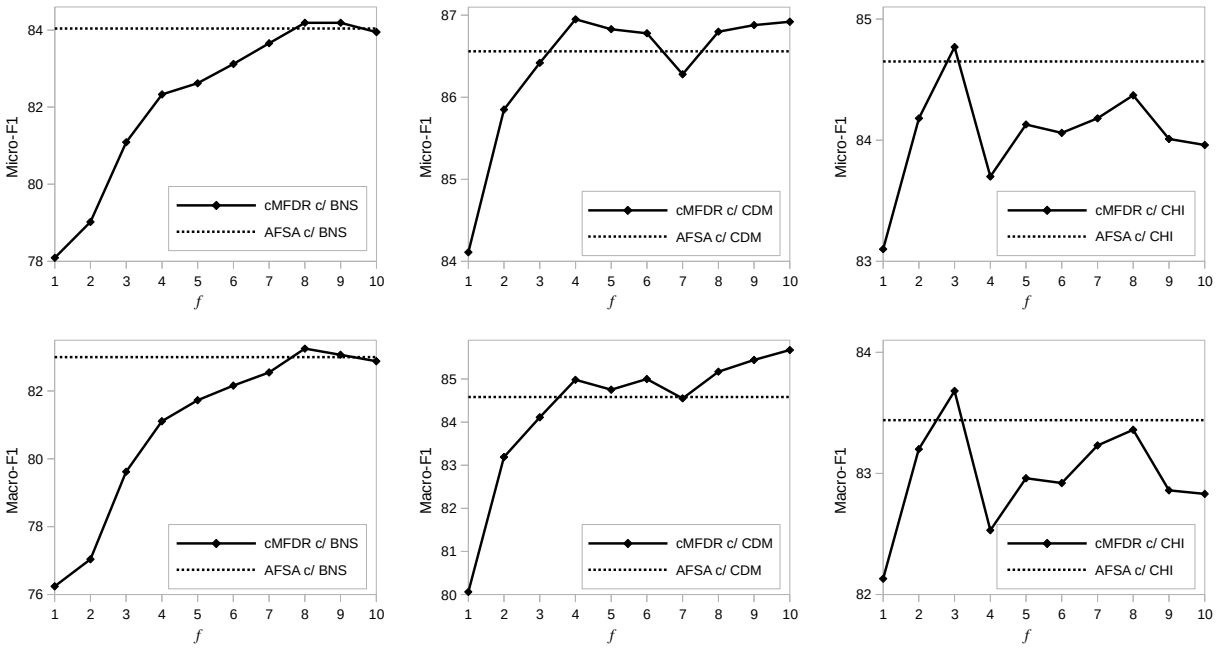

Figura 13: Resultados para a base de dados *WebKB* em termos de *Micro-F1* e *Macro-F1* para AFSA, com $n = 10$, e cMFDR, com f variando de 1 a 10.

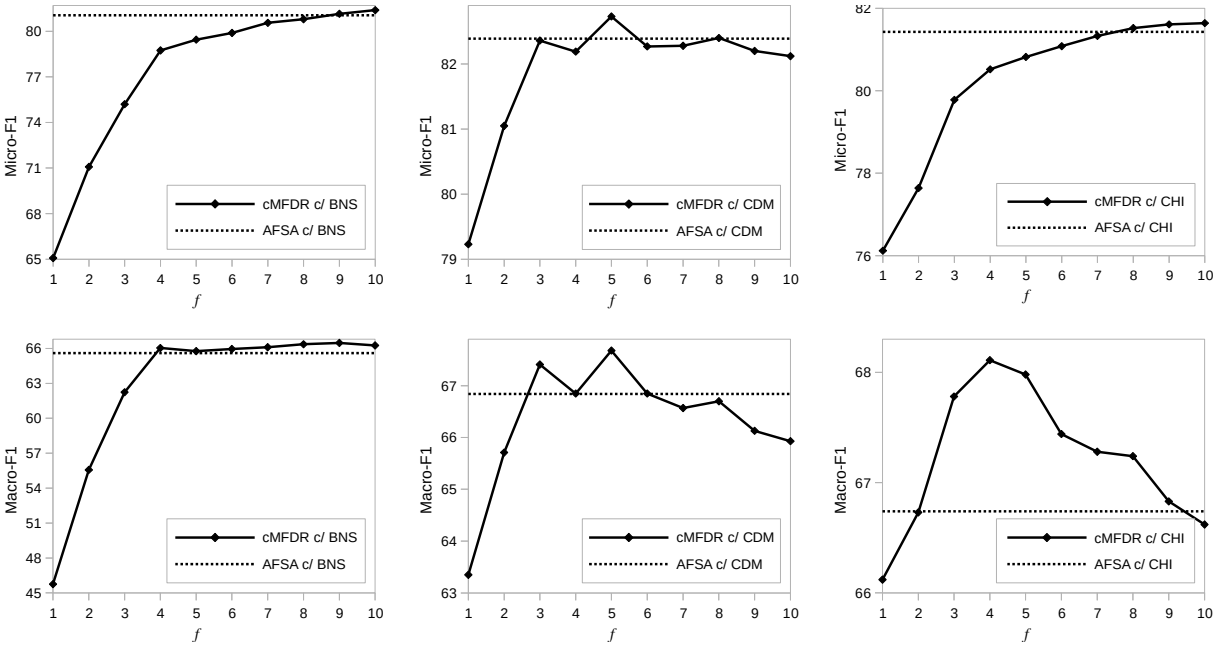

Figura 14: Resultados para a base de dados *Reuters* em termos de *Micro-F1* e *Macro-F1* para AFSA, com $n = 10$, e cMFDR, com f variando de 1 a 10.

O gráfico da Figura 15 apresenta os resultados para *20 Newsgroup*. Para esta base, AFSA em 83% dos casos, tanto de *Micro-F1* como de *Macro-F1*. Para ambas as medidas, AFSA foi superior em 9 dos 10 casos de BNS e CHI e 7 dos 10 casos de CDM.

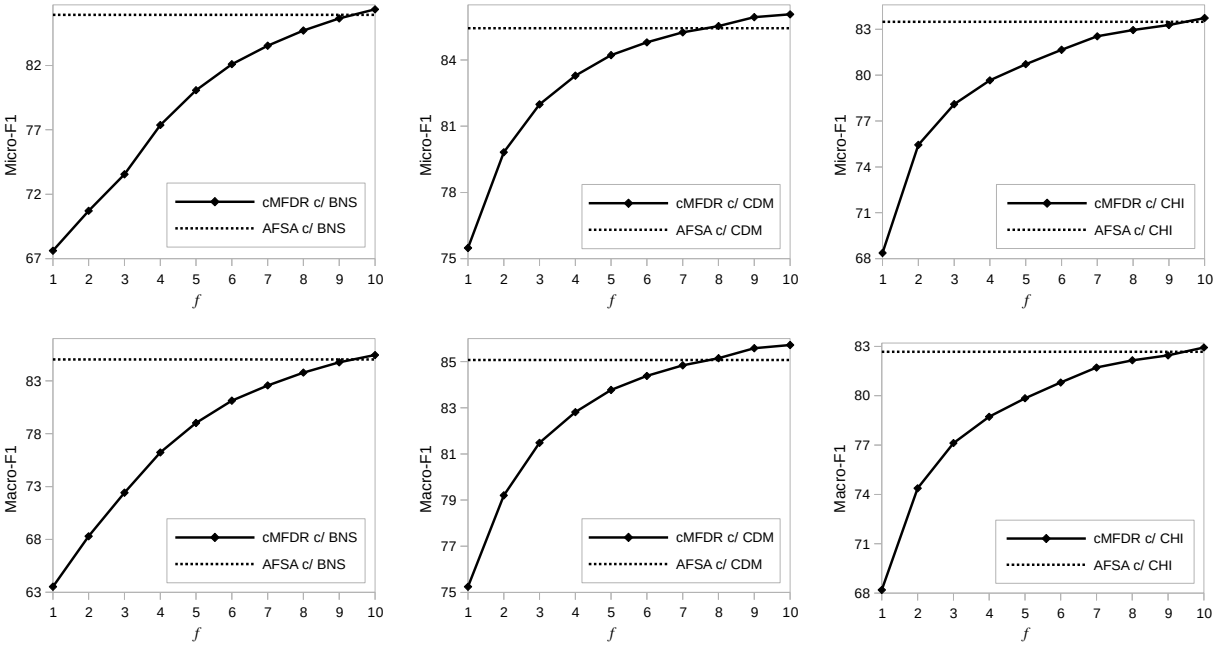

Figura 15: Resultados para a base de dados *20 Newsgroup* em termos de *Micro-F1* e *Macro-F1* para AFSA, com $n = 10$, e cMFDR, com f variando de 1 a 10.

Para *TDT2* (Figura 16), AFSA apresentou desempenho superior a cMFDR em 19 e 18 casos, dos 30 casos de *Micro-F1* e *Macro-F1*, respectivamente. Para *Micro-F1*, CHI foi a FEF com a qual AFSA apresentou maior vantagem, obtendo 8 vezes desempenho superior contra 2 vezes de cMFDR. Com respeito a *Macro-F1*, a FEF com a qual AFSA conseguiu melhores resultados em relação a cMFDR foi BNS.

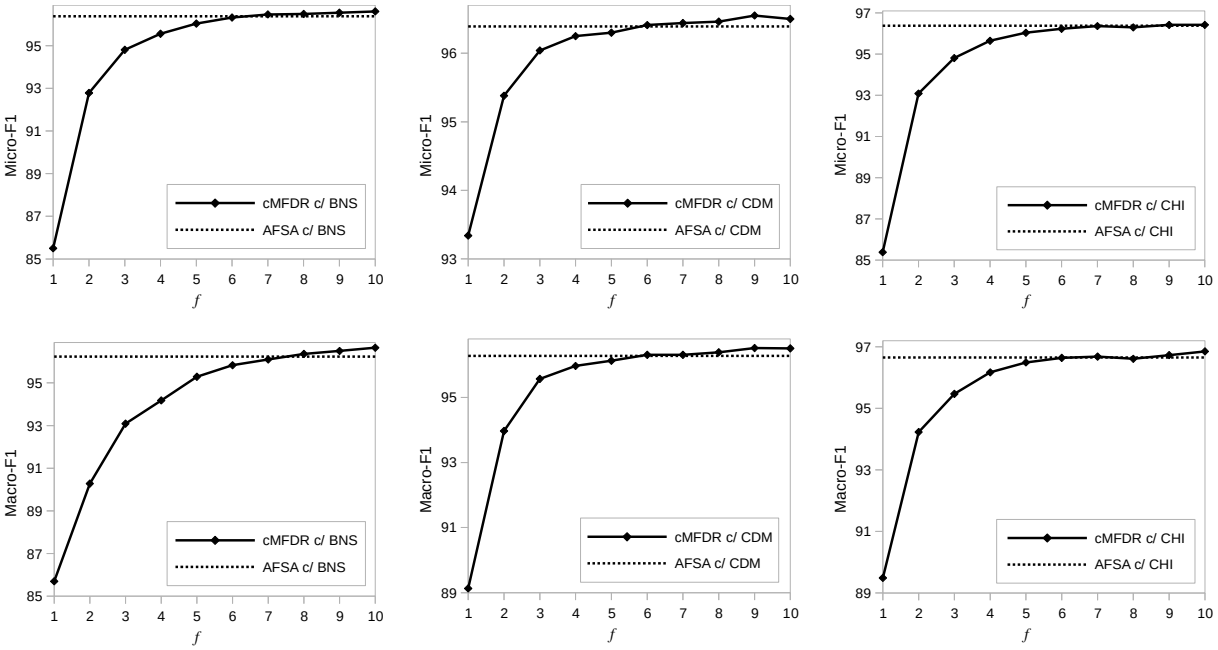

Figura 16: Resultados para a base de dados *TDT2* em termos de *Micro-F1* e *Macro-F1* para AFSA, com $n = 10$, e cMFDR, com f variando de 1 a 10.

O teste estatístico *t-test* foi aplicado para realizar um comparativo da performance de AFSA em relação a cMFDR. Os resultados apresentados na Tabela 10, demonstram que AFSA apesentou valor de efetividade superior ou semelhante a cMFDR em 100% dos casos. Na grande maioria dos casos, não foi observada diferença significativa entre os métodos. Para *20 Newsgroup*, usando CDM, o teste apresentou evidência de superioridade do desempenho de AFSA, tanto para *Micro-F1* como *Macro-F1*.

Tabela 10: Resultados do *t-test* comparando o desempenho de AFSA e cMFDR. Convenção adotada: "$\gg$"e "$\ll$"significam uma forte evidência de que um método apresenta maior ou menor valor de efetividade que outro, respectivamente; "$>$"e "$<$"significam uma fraca evidência de que um método apresenta maior ou menor valor de efetividade que outro, respectivamente; "$\sim$"significa que a diferença entre os métodos não é relevante.

Dataset	Measure	FEF		
		BNS	CDM	CHI
WebKB	Micro-F1	$\sim$	$\sim$	$\sim$
	Macro-F1	$\sim$	$\sim$	$\sim$
Reuters	Micro-F1	$\sim$	$\sim$	$\sim$
	Macro-F1	$\sim$	$\sim$	$\sim$
20 Newsgroup	Micro-F1	$\sim$	$>$	$\sim$
	Macro-F1	$\sim$	$>$	$\sim$
TDT2	Micro-F1	$\sim$	$\sim$	$\sim$
	Macro-F1	$\sim$	$\sim$	$\sim$

5.2.3 Análise de tempo de execução

Esta seção descreve o comportamento dos métodos propostos quanto ao tempo de execução. O tempo de execução do processo de seleção de características do método cMFDR é comparado com MFDR. Esta comparação é exibida na Tabela 11. O método AFSA não foi incluído no comparativo pelo fato de que ele utiliza o algoritmo do cMFDR para realizar a seleção e, deste modo, os tempos da seleção de características dos métodos são equivalentes. Pela forma como foram implementados os métodos, os tempos de execução de MFDR e cMFDR independem do valor do parâmetro f. Assim, os tempos apresentados na Tabela 11 representam o tempo de execução com $f = 10$.

O método cMFDR apresenta uma maior complexidade no cálculo dos seus limiares locais em relação ao cálculo do limiar global de MFDR. Apesar deste fato, o comparativo do tempo de execução dos métodos mostra desempenhos semelhantes na maioria dos casos. O desempenho na base de dados *Reuters* usando a FEF BNS é uma exceção. Com esta configuração, MFDR obtém um desempenho 64% mais rápido que cMFDR. Outra configuração que entregou um resultado com relativa vantagem a MFDR, 8% mais rápido que cMFDR, foi a utilização da FEF BNS na base *20 Newsgroup*. Em ambos os casos, uma diferença substancial da quantidade de características selecionadas por cada

Tabela 11: Comparativo do tempo de execução da seleção de características do método cMFDR com MFDR. O tempo é dado em milissegundos

Base	FEF	MFDR	cMFDR
WebKB	BNS	108	104
	CDM	342	318
	CHI	186	174
Reuters	BNS	166	261
	CDM	1016	979
	CHI	684	739
20 Newsgroup	BNS	502	540
	CDM	2041	2066
	CHI	1454	1548
TDT2	BNS	438	438
	CDM	1627	1579
	CHI	922	914

método justifica a diferença no tempo de execução. Para ambos os métodos, a FEF CDM apresentou os maiores tempos de execução, enquanto BNS foi a FEF com a qual ambos os métodos obtiveram os melhores desempenhos.

AFSA determina automaticamente o melhor valor para o parâmetro f de cMFDR, através da realização de pré-avaliações dos subconjuntos criados. Ou seja, AFSA utiliza um algoritmo classificador para determinar com qual valor de f cMFDR entrega o subconjunto com maior efetividade de classificação. Diante deste fato, é necessária a avaliação do tempo que AFSA leva para realizar todo este processo: seleção de características, treinamento e validação dos subconjuntos gerados por cMFDR e teste do desempenho final de classificação para o melhor valor de f. Desta forma, seria injusto realizar um comparativo do tempo de classificação de AFSA com cMFDR ou MFDR, pois estes dois métodos necessitam da configuração manual do parâmetro f. Assim, o tempo que AFSA leva para selecionar as características e realizar a classificação é comparado com o tempo de classificação da bases de dados, sem nenhuma redução de dimensionalidade, o com o algoritmo classificador *Support Vector Machine* (SVM). A Tabela 12 apresenta os resultados da comparação do tempo de execução de AFSA e SVM.

Com o método AFSA, CDM foi a FEF que apresentou maior tempo de execução em todas as bases de dados, com exceção de *TDT2*, na qual a FEF CHI teve um desempenho pior. Desta comparação, percebemos que é vantajoso, em termos de tempo de execução, realizar todo o processo de AFSA (seleção de características, escolha do melhor subconjunto gerado e classificação deste subconjunto) que simplesmente realizar a classificação usando SVM nas bases de dados sem nenhum tipo de redução de dimensionalidade. A execução do método AFSA é substancialmente mais rápida que a classificação com SVM em todos os cenários.

Tabela 12: Comparativo do tempo de execução do método AFSA (seleção de características e classificação) e tempo de classificação com SVM

	AFSA		SVM
Base de dados	FEF	ms	ms
WebKB	BNS	714	
	CDM	2285	4139
	CHI	1069	
Reuters	BNS	963	
	CDM	2055	5086
	CHI	1353	
20 Newsgroup	BNS	12721	
	CDM	17491	28063
	CHI	9221	
TDT2	BNS	7233	
	CDM	9988	21939
	CHI	11597	

5.3 Considerações Finais

Neste capítulo foram apresentados os resultados dos experimentos realizados para avaliação dos métodos propostos bem como comparativos para demonstrar a efetividade dos mesmos. Os resultados experimentos demonstraram que o método cMFDR apresenta uma superioridade de desempenho de classificação em relação a MFDR e registra tempo de execução equivalente a seu predecessor. O método AFSA atinge desempenho similar a cMFDR. AFSA utiliza menos dados de treinamento, pois reserva um *fold* para validação. Por este fato, o tempo de classificação de um subconjunto com AFSA é menor que com cMFDR. Além disso, o processo automático de escolha do melhor subconjunto torna AFSA mais eficiente em ambientes dinâmicos, com mudanças de documentos e categorias. Nestes casos, o processo de seleção de características precisa ser realizado com frequência para que o subconjunto de características reflita a atual realidade dos documentos da base de dados. O desempenho de AFSA também supera o desempenho de classificação de SVM, utilizando todas as características. O tempo de execução de AFSA também é melhor que o de SVM. A tabela 13 apresenta uma sumarização dos resultados da classificação das quatro bases de dados com os métodos utilizados nos experimentos: SVM (utilizando todas as características), MFDR, cMFDR e AFSA.

Tabela 13: Comparativo de desempenho entre AFSA, cMFDR, MFDR e SVM.

Base de dados	FEF	SVM		MFDR			cMFDR			AFSA		
		m	Micro-F1	f	m	Micro-F1	f	m	Micro-F1	f	m	Micro-F1
WebKB	BNS			9	68	83,19 (2,20)	9	382	84,19 (0,78)	9	366	84.04 (1.01)
	CDM			9	1215	85,88 (1,66)	4	862	86,45 (2,05)	5	991	86.07 (1.47)
	CHI			5	79	83,04 (1,66)	3	142	84,47 (1,68)	3	137	84.35 (1.84)
	-	7770	86,16 (1,02)									
Reuters	BNS			10	186	79,13 (1,46)	10	825	81,39 (1,03)	10	781	81.05 (0.95)
	CDM			10	824	82,29 (0,77)	5	680	82,43 (0,72)	6	775	82.09 (0.71)
	CHI			10	313	79,91 (0,99)	10	982	81,44 (1,15)	8	683	81.23 (0.98)
	-	10987	79,36 (0,78)									
20 Newsgroup	BNS			10	1190	77,26 (1,14)	10	5937	85,85 (0,63)	10	5544	85.42 (0.67)
	CDM			10	1964	82,52 (0,54)	10	5081	86,07 (0,45)	10	4707	85.44 (0.62)
	CHI			10	758	77,79 (0,94)	10	2621	83,34 (0,61)	10	2521	83.09 (0.62)
	-	70216	83,81 (1,73)									
TDT2	BNS			10	352	94,86 (0,59)	10	1440	96,61 (0,44)	9	1089	96.38 (0.58)
	CDM			10	578	95,32 (0,74)	9	1354	96,55 (0,40)	10	1456	96.39 (0.38)
	CHI			10	948	92,60 (1,67)	9	1640	96,72 (0,36)	10	1805	96.68 (0.42)
	-	36093	94,54 (0,74)									

6 Considerações Finais

Neste trabalho foram apresentados dois métodos de seleção de características para categorização e textos: cMFDR (*Category-dependent Maximum f Features per Document - Reduced* e AFSA (*Automatic Feature Subsets Analyzer*). O método cMFDR define um limiar para cada categoria com a finalidade de determinar quais documentos serão considerados na seleção de características. Este limiar é baseado na relevância DR dos documentos. O cálculo de DR foi melhorado em relação ao cálculo realizado no predecessor de cMFDR. Os resultados dos experimentos demonstram que cMFDR apresenta desempenho similar ou superior a MFDR em 100% dos casos, tanto para *Micro-F1* quanto para *Macro-F1*, sendo superior em 82,5% dos casos de *Micro-F1* e 75,8% dos casos de *Macro-F1*. Nos experimentos, verificou-se também que o tempo de execução do cMFDR é semelhante ao tempo de execução do MFDR.

O segundo método proposto, AFSA, apresenta um mecanismo para determinar automaticamente o melhor valor para o parâmetro f de cMFDR, que define quantas características devem ser selecionadas por documento. AFSA, realiza uma pré-verificação do desempenho classificatório de n subconjuntos, através de dados de validação, para determinar automaticamente qual subconjunto traz a melhor performance. Os experimentos demonstram uma que AFSA apresenta eficácia semelhante ou superior em relação ao cMFDR em 100% dos casos, sendo similar em 91,7% dos casos e superior em 8,3% dos casos. O tempo de execução do método (incluindo todo o processo de seleção de características, treinamento e validação dos subconjuntos gerados e teste do desempenho final de classificação) foi comparado com o tempo de execução usando o Classificador SVM, sem nenhum procedimento de redução de dimensionalidade. Os experimentos demonstraram que é mais rápido executar o processo do método AFSA que realizar a classificação com SVM.

6.1 Contribuições

Dentre as contribuições do presente trabalho, pode-se destacar os dois métodos propostos, cMFDR e AFSA, que: a) melhoram o desempenho de classificação dos métodos atuais; b) proveem uma solução para determinar automaticamente o tamanho do vetor final de características. Outra importante contribuição foi a publicação de dois artigos (FRAGOSO; PINHEIRO; CAVALCANTI, 2016a) e (FRAGOSO; PINHEIRO; CAVALCANTI, 2016b), cada um apresentando um dos métodos propostos neste trabalho.

6.2 Trabalhos Futuros

Para trabalhos futuros, sugere-se:

- Validar os métodos propostos utilizando uma maior diversidade de classificadores, bases de dados e FEFs;
- Estudar alternativas para computar os limiares usando outros cálculos, como métricas distintas para cada categoria;
- Avaliar outras formas de determinar automaticamente o número de características selecionadas por documento, sem necessidade de gerar e avaliar vários subconjuntos;
- Averiguar formas de determinar um valor dinâmico do parâmetro f, para cada documento.

Referências

CAI, D. et al. Modeling hidden topics on document manifold. In: ACM. *Proceedings of the 17th ACM conference on Information and knowledge management*. [S.l.], 2008. p. 911–920. Citado na página 45.

CHANG, Y.; CHEN, S.; LIAU, C. Multilabel text categorization based on a new linear classifier learning method and a category-sensitive refinement method. *Expert Systems with Applications*, Elsevier, v. 34, n. 3, p. 1948–1953, 2008. ISSN 0957-4174. Citado 2 vezes nas páginas 26 e 45.

CHAU, M.; CHEN, H. A machine learning approach to web page filtering using content and structure analysis. *Decision Support Systems*, Elsevier, v. 44, n. 2, p. 482–494, 2008. Citado na página 21.

CHEN, J. et al. Feature selection for text classification with Naive Bayes. *Expert Systems with Applications*, Elsevier, v. 36, n. 3, p. 5432–5435, 2009. Citado 6 vezes nas páginas 23, 24, 29, 30, 45 e 47.

COHEN, W. W.; SINGER, Y. Context-sensitive learning methods for text categorization. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 17, n. 2, p. 141–173, 1999. Citado na página 21.

DAS, S. Filters, wrappers and a boosting-based hybrid for feature selection. In: CITESEER. *ICML*. [S.l.], 2001. v. 1, p. 74–81. Citado na página 29.

DEBOLE, F.; SEBASTIANI, F. Supervised term weighting for automated text categorization. In: ACM. *Proceedings of the 2003 ACM Symposium on Applied Computing*. [S.l.], 2003. p. 784–788. Citado na página 29.

DEBOLE, F.; SEBASTIANI, F. An analysis of the relative hardness of reuters-21578 subsets. *Journal of the American Society for Information Science and technology*, Wiley Online Library, v. 56, n. 6, p. 584–596, 2005. Citado na página 45.

DIETTERICH, T. Overfitting and undercomputing in machine learning. *ACM computing surveys (CSUR)*, ACM, v. 27, n. 3, p. 326–327, 1995. Citado na página 28.

FENG, G. et al. Feature subset selection using naive bayes for text classification. *Pattern Recognition Letters*, Elsevier, v. 65, p. 109–115, 2015. Citado 3 vezes nas páginas 14, 24 e 44.

FORMAN, G. An extensive empirical study of feature selection metrics for text classification. *The Journal of Machine Learning Research*, JMLR. org, v. 3, p. 1289–1305, 2003. Citado 3 vezes nas páginas 29, 44 e 46.

FRAGOSO, R.; PINHEIRO, R.; CAVALCANTI, G. Class-dependent feature selection algorithm for text categorization. In: *International Joint Conference on Neural Networks*. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 36 e 61.

FRAGOSO, R.; PINHEIRO, R.; CAVALCANTI, G. A method for automatic determination of the feature vector size for text categorization. In: *2016 Brazilian Conference on Intelligent Systems*. [S.l.: s.n.], 2016. Citado 2 vezes nas páginas 41 e 61.

GABRILOVICH, E.; MARKOVITCH, S. Text categorization with many redundant features: using aggressive feature selection to make svms competitive with c4. 5. In: ACM. *Proceedings of the twenty-first international conference on Machine learning*. [S.l.], 2004. p. 41. Citado na página 15.

GHIASSI, M.; SKINNER, J.; ZIMBRA, D. Twitter brand sentiment analysis: A hybrid system using n-gram analysis and dynamic artificial neural network. *Expert Systems with applications*, Elsevier, v. 40, n. 16, p. 6266–6282, 2013. Citado na página 23.

GUNAL, S. Hybrid feature selection for text classification. *Turkish Journal of Electrical Engineering and Computer Sciences*, v. 20, n. 2, p. 1296–1311, 2012. Citado na página 30.

GUYON, I.; ELISSEEFF, A. An introduction to variable and feature selection. *The Journal of Machine Learning Research*, JMLR. org, v. 3, p. 1157–1182, 2003. ISSN 1532-4435. Citado 4 vezes nas páginas 14, 25, 28 e 29.

JAIN, A. K.; DUIN, R. P. W.; MAO, J. Statistical pattern recognition: A review. *IEEE Transactions on pattern analysis and machine intelligence*, IEEE, v. 22, n. 1, p. 4–37, 2000. Citado na página 15.

JOACHIMS, T. *A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization*. [S.l.], 1996. Citado 2 vezes nas páginas 14 e 21.

JOACHIMS, T. Text categorization with support vector machines: Learning with many relevant features. In: SPRINGER. *European conference on machine learning*. [S.l.], 1998. p. 137–142. Citado 3 vezes nas páginas 24, 25 e 43.

KIBRIYA, A. M. et al. Multinomial naive bayes for text categorization revisited. In: SPRINGER. *Australasian Joint Conference on Artificial Intelligence*. [S.l.], 2004. p. 488–499. Citado na página 25.

KOHAVI, R.; JOHN, G. Wrappers for feature subset selection. *Artificial Intelligence*, Elsevier, v. 97, n. 1-2, p. 273–324, 1997. Citado na página 28.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: *Ijcai*. [S.l.: s.n.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 43.

LEOPOLD, E.; KINDERMANN, J. Text categorization with support vector machines. how to represent texts in input space? *Machine Learning*, Springer, v. 46, n. 1-3, p. 423–444, 2002. Citado 2 vezes nas páginas 24 e 43.

LEWIS, D. Naive (Bayes) at forty: the independence assumption in information retrieval. *European Conference on Machine Learning*, Springer, p. 4–15, 1998. Citado na página 23.

LEWIS, D.; RINGUETTE, M. A comparison of two learning algorithms for text categorization. In: *3rd annual symposium on document analysis and information retrieval*. [S.l.: s.n.], 1994. v. 33, p. 81–93. Citado na página 29.

LOVINS, J. B. *Development of a stemming algorithm*. [S.l.]: MIT Information Processing Group, Electronic Systems Laboratory Cambridge, 1968. Citado na página 45.

LU, Y.; LIU, W.; HE, X. A text feature selection method based on category-distribution divergence. *Artificial Intelligence Research*, v. 4, n. 2, p. 143, 2015. Citado na página 30.

LUHN, H. P. A statistical approach to mechanized encoding and searching of literary information. *IBM Journal of research and development*, IBM, v. 1, n. 4, p. 309–317, 1957. Citado na página 21.

MARON, M. E. Automatic indexing: an experimental inquiry. *Journal of the ACM*, ACM, v. 8, n. 3, p. 404–417, 1961. Citado na página 17.

MCCALLUM, A.; NIGAM, K. A comparison of event models for naive bayes text classification. In: *Workshop on Learning for Text Categorization*. [S.l.: s.n.], 1998. p. 41–48. Citado 2 vezes nas páginas 24 e 44.

OGURA, H.; AMANO, H.; KONDO, M. Comparison of metrics for feature selection in imbalanced text classification. *Expert Systems with Applications*, v. 38, n. 5, p. 4978–4989, 2011. Citado na página 30.

PINHEIRO, R. et al. A global-ranking local feature selection method for text categorization. *Expert Systems with Applications*, v. 39, n. 17, p. 12851–12857, 2012. Citado 7 vezes nas páginas 24, 26, 29, 30, 31, 43 e 46.

PINHEIRO, R.; CAVALCANTI, G.; REN, T. Data-driven global-ranking local feature selection methods for text categorization. *Expert Systems with Applications*, v. 42, n. 4, p. 1941–1949, 2015. Citado 8 vezes nas páginas 22, 24, 30, 32, 33, 44, 45 e 47.

PORTER, M. F. An algorithm for suffix stripping. *Program*, MCB UP Ltd, v. 14, n. 3, p. 130–137, 1980. Citado na página 45.

RAKOTOMAMONJY, A. Variable selection using svm-based criteria. *Journal of machine learning research*, v. 3, n. Mar, p. 1357–1370, 2003. Citado na página 23.

RISH, I. An empirical study of the naive bayes classifier. In: IBM NEW YORK. *IJCAI 2001 workshop on empirical methods in artificial intelligence*. [S.l.], 2001. v. 3, n. 22, p. 41–46. Citado na página 23.

SAHA, A.; SINDHWANI, V. Dynamic nmfs with temporal regularization for online analysis of streaming text. In: *Proceedings of NIPS Workshop on Machine Learning for Social Computing, pp. 1C8*. [S.l.: s.n.], 2010. Citado na página 45.

SALTON, G.; MCGILL, M. *The SMART retrieval system—experiments in automatic document retrieval*. [S.l.]: Prentice Hall Inc., Englewood Cliffs, NJ, 1971. Citado na página 45.

SALTON, G.; YANG, C.-S. On the specification of term values in automatic indexing. *Journal of documentation*, MCB UP Ltd, v. 29, n. 4, p. 351–372, 1973. Citado na página 14.

SCHNEIDER, K.-M. A comparison of event models for naive bayes anti-spam e-mail filtering. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *10th Conference on European Chapter of the Association for Computational Linguistics-Volume 1*. [S.l.], 2003. p. 307–314. Citado na página 24.

SEBASTIANI, F. Machine learning in automated text categorization. *ACM Computing Surveys*, ACM, v. 34, n. 1, p. 1–47, 2002. ISSN 0360-0300. Citado 5 vezes nas páginas 14, 17, 22, 26 e 29.

SHANG, W. et al. A novel feature selection algorithm for text categorization. *Expert Systems with Applications*, Elsevier, v. 33, n. 1, p. 1–5, 2007. ISSN 0957-4174. Citado 2 vezes nas páginas 29 e 30.

SUN, X. et al. Feature selection using dynamic weights for classification. *Knowledge-Based Systems*, v. 37, p. 541–549, 2013. Citado 3 vezes nas páginas 24, 25 e 30.

TANG, B.; KAY, S.; HE, H. Toward optimal feature selection in naive bayes for text categorization. *IEEE Transactions on Knowledge and Data Engineering*, IEEE, 2016. Citado 2 vezes nas páginas 21 e 44.

TAŞCI, Ş.; GÜNGÖR, T. Comparison of text feature selection policies and using an adaptive framework. *Expert Systems with Applications*, Elsevier, v. 40, n. 12, p. 4871–4886, 2013. Citado 2 vezes nas páginas 21 e 26.

UĞUZ, H. A two-stage feature selection method for text categorization by using information gain, principal component analysis and genetic algorithm. *Knowledge-Based Systems*, v. 24, n. 7, p. 1024–1032, 2011. Citado 3 vezes nas páginas 23, 30 e 44.

UYSAL, A. K. An improved global feature selection scheme for text classification. *Expert Systems with Applications*, v. 43, p. 82–89, 2016. Citado 2 vezes nas páginas 26 e 30.

UYSAL, A. K.; GUNAL, S. A novel probabilistic feature selection method for text classification. *Knowledge-Based Systems*, Elsevier, v. 36, p. 226–235, 2012. Citado 2 vezes nas páginas 26 e 29.

VLADIMIR, V. N.; VAPNIK, V. *The nature of statistical learning theory*. [S.l.]: Springer Heidelberg, 1995. Citado 2 vezes nas páginas 25 e 26.

WANG, D. et al. T-test feature selection approach based on term frequency for text categorization. *Pattern Recognition Letters*, Elsevier, v. 45, p. 1–10, 2014. Citado na página 26.

WU, H. C. et al. Interpreting tf-idf term weights as making relevance decisions. *ACM Transactions on Information Systems (TOIS)*, ACM, v. 26, n. 3, p. 13, 2008. Citado na página 21.

YANG, J. et al. A new feature selection algorithm based on binomial hypothesis testing for spam filtering. *Knowledge-Based Systems*, Elsevier, v. 24, n. 6, p. 904–914, 2011. Citado 2 vezes nas páginas 44 e 45.

YANG, J. et al. A new feature selection based on comprehensive measurement both in inter-category and intra-category for text categorization. *Information Processing and Management: an International Journal*, Pergamon Press, Inc., v. 48, n. 4, p. 741–754, 2012. Citado 3 vezes nas páginas 24, 29 e 44.

YANG, Y. An evaluation of statistical approaches to text categorization. *Information retrieval*, Springer, v. 1, n. 1-2, p. 69–90, 1999. Citado na página 23.

YANG, Y.; PEDERSEN, J. A comparative study on feature selection in text categorization. In: *Proceedings of the International Conference on Machine Learning*. [S.l.: s.n.], 1997. p. 412–420. Citado 3 vezes nas páginas 15, 29 e 47.

YEPES, A. J. J. et al. Feature engineering for medline citation categorization with mesh. *BMC bioinformatics*, BioMed Central, v. 16, n. 1, p. 1, 2015. Citado na página 21.

YU, B.; XU, Z.-b.; LI, C.-h. Latent semantic analysis for text categorization using neural network. *Knowledge-Based Systems*, Elsevier, v. 21, n. 8, p. 900–904, 2008. Citado na página 23.

YU, L.; LIU, H. Feature selection for high-dimensional data: a fast correlation-based filter solution. In: *Proceedings of the International Conference on Machine Leaning*. [S.l.: s.n.], 2003. v. 20, n. 2, p. 856–863. Citado na página 15.

ZHANG, W.; YOSHIDA, T.; TANG, X. Text classification based on multi-word with support vector machine. *Knowledge-Based Systems*, Elsevier, v. 21, n. 8, p. 879–886, 2008. Citado 2 vezes nas páginas 23 e 25.

ZHENG, Z.; WU, X.; SRIHARI, R. Feature selection for text categorization on imbalanced data. *ACM Sigkdd Explorations Newsletter*, v. 6, n. 1, p. 80–89, 2004. Citado na página 30.

Apêndices

APÊNDICE A – Lista de *stopwords*

a, about, above, across, after, afterwards, again, against, all, almost, alone, along, already, also, although, always, am, among, amongst, amoungst, amount, an, and, another, any, anyhow, anyone, anything, anyway, anywhere, are, around, as, at, back, be, became, because, become, becomes, becoming, been, before, beforehand, behind, being, below, beside, besides, between, beyond, bill, both, bottom, but, by, call, can, cannot, cant, co, computer, con, could, couldnt, cry, de, describe, detail, do, done, down, due, during, each, eg, eight, either, eleven, else, elsewhere, empty, enough, etc, even, ever, every, everyone, everything, everywhere, except, few, fifteen, fify, fill, find, fire, first, five, for, former, formerly, forty, found, four, from, front, full, further, get, give, go, had, has, hasnt, have, he, hence, her, here, hereafter, hereby, herein, hereupon, hers, herse', him, himse', his, how, however, hundred, i, ie, if, in, inc, indeed, interest, into, is, it, its, itse', keep, last, latter, latterly, least, less, ltd, made, many, may, me, meanwhile, might, mill, mine, more, moreover, most, mostly, move, much, must, my, myse', name, namely, neither, never, nevertheless, next, nine, no, nobody, none, noone, nor, not, nothing, now, nowhere, of, off, often, on, once, one, only, onto, or, other, others, otherwise, our, ours, ourselves, out, over, own, part, per, perhaps, please, put, rather, re, same, see, seem, seemed, seeming, seems, serious, several, she, should, show, side, since, sincere, six, sixty, so, some, somehow, someone, something, sometime, sometimes, somewhere, still, such, system, take, ten, than, that, the, their, them, themselves, then, thence, there, thereafter, thereby, therefore, therein, thereupon, these, they, thick, thin, third, this, those, though, three, through, throughout, thru, thus, to, together, too, top, toward, towards, twelve, twenty, two, un, under, until, up, upon, us, very, via, was, we, well, were, what, whatever, when, whence, whenever, where, whereafter, whereas, whereby, wherein, whereupon, wherever, whether, which, while, whither, who, whoever, whole, whom, whose, why, will, with, within, without, would, yet, you, your, yours, yourself, yourselves.

APÊNDICE B – Resultados de cMFDR

Tabela 14: Comparação entre os resultados de cMFDR e MFDR para *WebKB*. Os melhores resultados para cada combinação de base de dados, FEF e métodos são exibidos em negrito e os desvios padrão, entre parênteses.

FEF	f	MFDR			cMFDR		
		m	Micro-F1	Macro-F1	m	Micro-F1	Macro-F1
BNS	1	13 (1,05)	76,38 (1,63)	73,62 (1,92)	20 (0.67)	78.09 (2.92)	76.24 (2.99)
	2	19 (0,63)	76,59 (2,24)	74,29 (2,67)	37 (1.10)	79.02 (1.86)	77.04 (2.26)
	3	28 (1,31)	78,78 (2,17)	76,68 (2,45)	58 (1.77)	81.09 (1.98)	79.62 (1.98)
	4	34 (0,99)	79,59 (2,50)	78,03 (2,90)	92 (1.23)	82.33 (1.77)	81.11 (1.86)
	5	41 (1,03)	80,64 (2,45)	79,21 (2,74)	139 (4.44)	82.62 (1.81)	81.73 (1.82)
	6	48 (2,22)	81,38 (1,90)	80,06 (1,99)	185 (6.93)	83.12 (1.84)	82.16 (1.76)
	7	53 (1,76)	81,78 (2,33)	80,67 (2,47)	244 (7.38)	83.66 (1.60)	82.55 (1.72)
	8	58 (1,89)	82,00 (1,75)	81,06 (1,68)	310 (8.11)	84.19 (1.44)	**83.25 (1.68)**
	9	68 (3,15)	**83,19 (2,20)**	**82,10 (2,05)**	382 (9.69)	**84.19 (0.78)**	83.07 (0.99)
	10	77 (5,36)	82,95 (2,08)	81,73 (2,09)	477 (6.03)	83.95 (0.75)	82.88 (0.92)
CDM	1	202 (5,60)	81,02 (2,25)	75,89 (3,54)	262 (5.50)	83.61 (2.06)	80.06 (2.30)
	2	348 (13,1)	82,02 (1,86)	77,57 (2,63)	459 (14.85)	85.35 (1.82)	83.19 (2.35)
	3	481 (14,7)	83,47 (1,76)	79,95 (2,05)	650 (12.84)	85.92 (1.96)	84.11 (2.24)
	4	611 (14,8)	84,85 (1,72)	82,18 (1,76)	862 (11.86)	**86.45 (2.05)**	84.98 (2.14)
	5	737 (19,4)	84,85 (1,65)	82,37 (1,80)	1067 (17.54)	86.33 (1.70)	84.75 (1.69)
	6	863 (17,9)	84,99 (1,81)	82,74 (1,87)	1277 (18.34)	86.28 (1.33)	85.00 (1.57)
	7	984 (11,3)	85,33 (1,69)	83,35 (1,80)	1467 (14.32)	85.78 (1.43)	84.55 (1.83)
	8	1101 (10,0)	85,61 (1,76)	84,01 (1,94)	1663 (17.57)	86.30 (1.34)	85.17 (1.60)
	9	1215 (15,4)	**85,88 (1,66)**	**84,50 (2,08)**	1839 (17.42)	86.38 (1.00)	85.44 (1.14)
	10	1324 (14,8)	85,45 (1,38)	84,00 (1,79)	2003 (13.27)	86.42 (1.10)	**85.67 (1.02)**
CHI	1	35 (1,55)	77,66 (2,72)	76,03 (2,57)	56 (1.33)	82.80 (1.72)	82.13 (2.06)
	2	47 (1,49)	80,42 (2,19)	79,11 (2,29)	92 (2.62)	83.88 (1.27)	83.20 (1.40)
	3	59 (1,42)	80,90 (2,11)	79,91 (2,27)	142 (3.33)	**84.47 (1.68)**	**83.68 (1.88)**
	4	71 (1,89)	81,64 (1,97)	80,84 (2,18)	200 (4.52)	83.40 (1.31)	82.53 (1.15)
	5	79 (1,40)	**83,04 (1,66)**	**82,43 (1,77)**	256 (5.52)	83.83 (1.39)	82.96 (1.37)
	6	88 (2,13)	82,54 (1,51)	82,15 (1,64)	314 (8.12)	83.76 (1.19)	82.92 (0.97)
	7	100 (2,62)	82,47 (1,09)	82,09 (1,29)	381 (10.22)	83.88 (1.84)	83.23 (1.88)
	8	115 (2,83)	81,83 (1,06)	81,32 (1,25)	452 (6.22)	84.07 (1.68)	83.36 (1.64)
	9	132 (4,19)	82,11 (1,21)	81,60 (1,17)	516 (8.73)	83.71 (1.36)	82.86 (1.29)
	10	153 (4,48)	82,35 (1,30)	81,67 (1,22)	577 (11.32)	83.66 (1.15)	82.83 (1.26)

Tabela 15: Comparação dos resultados de cMFDR frente ao MFDR para *Reuters*. Os melhores resultados para cada combinação de base de dados, FEF e métodos são exibidos em negrito e os desvios padrão, entre parênteses.

FEF	f	MFDR			cMFDR		
		m	Micro-F1	Macro-F1	m	Micro-F1	Macro-F1
BNS	1	10 (0.92)	62.40 (1.38)	44.01 (2.31)	12 (0.82)	64.53 (1.37)	45.75 (2.18)
	2	18 (2.31)	65.13 (2.01)	49.49 (2.78)	32 (1.93)	71.08 (1.52)	55.57 (2.17)
	3	30 (5.47)	70.90 (1.33)	57.49 (1.34)	66 (2.74)	75.20 (0.94)	62.24 (1.19)
	4	45 (3.09)	73.37 (1.17)	60.72 (2.07)	121 (5.70)	78.74 (1.29)	66.04 (1.77)
	5	60 (9.39)	74.88 (1.32)	62.23 (2.03)	186 (6.78)	79.45 (1.02)	65.77 (1.94)
	6	80 (7.28)	75.76 (1.05)	63.21 (2.01)	270 (14.34)	79.89 (0.82)	65.96 (1.68)
	7	104 (5.64)	76.42 (1.39)	63.51 (2.32)	371 (18.28)	80.56 (0.96)	66.11 (1.55)
	8	128 (6.22)	77.56 (1.44)	64.39 (2.51)	481 (25.38)	80.80 (0.95)	66.36 (1.37)
	9	156 (4.16)	78.51 (1.34)	65.17 (2.25)	637 (35.35)	81.15 (0.97)	**66.48 (1.65)**
	10	186 (6.98)	**79.13 (1.46)**	**65.71 (2.60)**	825 (36.82)	**81.39 (1.03)**	66.26 (1.41)
CDM	1	132 (2.45)	78.88 (0.82)	64.05 (1.24)	147 (3.01)	78.93 (1.02)	63.35 (1.71)
	2	215 (3.74)	79.75 (0.59)	65.39 (1.41)	279 (3.50)	80.75 (0.97)	65.71 (1.71)
	3	295 (4.83)	80.40 (1.01)	65.65 (1.80)	404 (5.23)	82.06 (0.91)	67.41 (1.50)
	4	364 (6.70)	80.91 (0.85)	66.02 (1.74)	533 (8.60)	81.89 (0.81)	66.85 (1.31)
	5	443 (5.40)	80.92 (0.71)	65.78 (1.68)	680 (9.72)	**82.43 (0.72)**	**67.68 (1.08)**
	6	520 (6.17)	81.07 (0.62)	65.74 (1.45)	828 (6.83)	81.97 (0.76)	66.85 (1.27)
	7	593 (4.14)	81.70 (0.76)	66.56 (1.65)	992 (11.18)	81.98 (0.57)	66.57 (1.14)
	8	671 (6.83)	82.08 (0.86)	**67.11 (1.60)**	1140 (8.10)	82.10 (0.72)	66.70 (1.55)
	9	746 (5.36)	82.16 (0.83)	67.02 (1.38)	1279 (11.21)	81.90 (0.76)	66.13 (1.59)
	10	824 (7.33)	**82.29 (0.77)**	66.78 (1.42)	1396 (10.16)	81.82 (0.61)	65.93 (1.27)
CHI	1	50 (1.64)	68.30 (1.61)	60.86 (2.75)	81 (2.08)	75.92 (0.91)	65.72 (1.61)
	2	85 (0.82)	74.28 (1.47)	64.57 (1.52)	139 (2.45)	77.44 (1.21)	66.33 (1.57)
	3	109 (1.10)	75.04 (0.93)	65.13 (1.08)	194 (5.07)	79.58 (1.01)	67.38 (2.00)
	4	130 (0.94)	76.70 (1.13)	66.36 (1.09)	263 (4.03)	80.32 (0.79)	**67.71 (1.58)**
	5	158 (2.27)	77.33 (0.86)	66.48 (1.18)	338 (7.10)	80.62 (1.04)	67.58 (1.88)
	6	190 (3.68)	77.68 (1.26)	66.75 (1.87)	439 (4.78)	80.88 (1.03)	67.04 (1.57)
	7	219 (5.19)	78.79 (0.97)	67.24 (1.45)	555 (7.41)	81.13 (0.88)	66.88 (1.50)
	8	244 (5.56)	79.02 (1.15)	67.11 (1.43)	703 (7.51)	81.32 (0.88)	66.84 (1.85)
	9	279 (5.26)	79.18 (1.11)	66.98 (1.53)	847 (10.56)	81.41 (1.18)	66.43 (2.29)
	10	313 (4.11)	**79.91 (0.99)**	**67.58 (1.56)**	982 (13.06)	**81.44 (1.15)**	66.22 (2.16)

Tabela 16: Comparação entre os resultados de cMFDR e MFDR para *20 Newsgroup*. Os melhores resultados para cada combinação de base de dados, FEF e métodos são exibidos em negrito e os desvios padrão, entre parênteses.

FEF	f	MFDR			cMFDR		
		m	Micro-F1	Macro-F1	m	Micro-F1	Macro-F1
BNS	1	30 (0.87)	53.89 (1.40)	54.93 (1.85)	102 (2.69)	53.56 (1.66)	54.82 (1.55)
	2	59 (2.22)	58.73 (1.27)	60.07 (1.38)	274 (7.94)	67.00 (1.71)	66.93 (1.78)
	3	112 (4.24)	63.01 (1.30)	64.88 (1.34)	549 (13.41)	73.04 (1.43)	72.42 (1.45)
	4	181 (6.32)	65.62 (1.36)	67.66 (1.26)	951 (16.89)	76.87 (1.13)	76.23 (1.24)
	5	274 (7.97)	67.36 (0.97)	68.98 (0.97)	1498 (26.40)	79.57 (0.98)	79.02 (1.09)
	6	383 (13.03)	69.83 (1.44)	70.51 (1.36)	2177 (34.04)	81.59 (0.91)	81.13 (0.99)
	7	528 (16.08)	73.88 (1.44)	72.15 (1.42)	2976 (42.81)	83.03 (0.59)	82.57 (0.65)
	8	699 (23.99)	75.13 (1.44)	74.05 (1.43)	3882 (34.98)	84.21 (0.54)	83.78 (0.64)
	9	921 (22.80)	75.62 (1.21)	75.45 (1.30)	4866 (46.37)	85.15 (0.67)	84.76 (0.67)
	10	1190 (25.76)	**77.26 (1.14)**	**77.02 (1.26)**	5937 (70.39)	**85.85 (0.63)**	**85.44 (0.64)**
CDM	1	320 (6.27)	70.74 (0.93)	71.83 (1.02)	552 (5.33)	75.48 (0.84)	75.24 (0.83)
	2	496 (6.83)	74.18 (0.72)	74.41 (0.78)	1033 (10.72)	79.82 (0.79)	79.20 (0.86)
	3	686 (6.73)	76.35 (0.84)	76.24 (0.89)	1534 (11.78)	81.99 (0.65)	81.48 (0.65)
	4	885 (9.55)	77.95 (0.93)	77.64 (0.96)	2086 (13.88)	83.29 (0.75)	82.81 (0.76)
	5	1062 (10.02)	79.30 (0.91)	78.91 (0.94)	2658 (14.38)	84.22 (0.72)	83.77 (0.73)
	6	1252 (9.60)	80.34 (0.66)	79.91 (0.72)	3223 (18.91)	84.80 (0.50)	84.38 (0.52)
	7	1426 (10.79)	81.22 (0.57)	80.78 (0.60)	3762 (21.95)	85.25 (0.55)	84.84 (0.60)
	8	1604 (9.30)	81.71 (0.78)	81.25 (0.82)	4232 (11.69)	85.54 (0.60)	85.15 (0.64)
	9	1785 (8.44)	82.07 (0.70)	81.57 (0.72)	4682 (12.63)	85.94 (0.48)	85.58 (0.53)
	10	1964 (10.71)	**82.52 (0.54)**	**82.05 (0.60)**	5081 (21.86)	**86.07 (0.45)**	**85.72 (0.51)**
CHI	1	104 (1.49)	60.32 (0.77)	62.97 (0.82)	184 (2.02)	67.97 (0.76)	68.19 (0.71)
	2	186 (3.40)	67.09 (1.09)	67.82 (1.15)	450 (4.78)	75.04 (0.71)	74.37 (0.83)
	3	256 (4.31)	69.85 (0.97)	69.91 (1.01)	693 (6.49)	77.70 (0.85)	77.12 (0.84)
	4	324 (6.62)	72.00 (0.90)	71.72 (1.00)	957 (7.82)	79.26 (0.82)	78.72 (0.88)
	5	394 (4.44)	73.37 (0.51)	72.97 (0.59)	1214 (8.09)	80.32 (0.74)	79.84 (0.83)
	6	460 (6.68)	74.85 (0.71)	74.35 (0.79)	1487 (12.27)	81.26 (0.78)	80.80 (0.82)
	7	535 (7.05)	75.81 (0.78)	75.24 (0.83)	1757 (16.08)	82.14 (0.61)	81.71 (0.56)
	8	611 (5.45)	76.49 (0.88)	75.91 (0.93)	2036 (17.26)	82.55 (0.61)	82.15 (0.63)
	9	681 (5.21)	77.18 (0.97)	76.60 (0.97)	2326 (15.13)	82.88 (0.67)	82.45 (0.74)
	10	758 (6.75)	**77.79 (0.94)**	**77.26 (0.95)**	2621 (19.04)	**83.34 (0.61)**	**82.93 (0.66)**

Tabela 17: Comparação entre os resultados de cMFDR e MFDR para *TDT2*. Os melhores resultados para cada combinação de base de dados, FEF e métodos são exibidos em negrito e os desvios padrão, entre parênteses.

FEF	f	MFDR			cMFDR		
		m	Micro-F1	Macro-F1	m	Micro-F1	Macro-F1
BNS	1	61 (1.87)	76.72 (2.70)	65.66 (3.66)	88 (3.71)	85.50 (1.57)	79.16 (2.21)
	2	84 (3.06)	83.39 (1.16)	75.59 (2.39)	162 (5.55)	92.79 (0.77)	90.28 (1.44)
	3	112 (3.46)	87.95 (1.13)	82.61 (2.20)	240 (3.94)	94.81 (0.49)	93.09 (0.95)
	4	137 (3.01)	90.67 (0.71)	86.56 (1.31)	348 (8.38)	95.56 (0.54)	94.18 (1.13)
	5	161 (4.88)	91.91 (1.03)	88.41 (1.93)	475 (8.63)	96.04 (0.41)	95.29 (0.74)
	6	188 (5.61)	92.64 (0.66)	89.57 (1.04)	627 (7.64)	96.32 (0.59)	95.83 (0.79)
	7	223 (7.53)	93.37 (0.81)	91.00 (1.41)	791 (12.68)	96.47 (0.61)	96.10 (0.65)
	8	264 (8.39)	94.06 (0.84)	91.96 (1.33)	983 (13.76)	96.49 (0.56)	96.36 (0.55)
	9	306 (6.40)	94.61 (0.67)	92.63 (1.22)	1196 (15.87)	96.55 (0.54)	96.50 (0.25)
	10	352 (11.25)	**94.86 (0.59)**	**93.00 (1.05)**	1440 (19.69)	**96.61 (0.44)**	**96.65 (0.35)**
CDM	1	126 (1.15)	84.42 (1.21)	66.38 (2.75)	183 (3.65)	93.34 (1.01)	89.13 (1.85)
	2	190 (2.67)	89.39 (0.81)	77.51 (1.68)	306 (5.73)	95.38 (0.57)	93.97 (1.19)
	3	247 (3.15)	91.05 (0.70)	81.93 (1.53)	436 (5.73)	96.04 (0.60)	95.57 (0.62)
	4	302 (3.48)	92.07 (0.81)	84.51 (2.07)	569 (8.31)	96.25 (0.47)	95.97 (0.51)
	5	352 (4.04)	93.17 (0.85)	87.67 (2.00)	713 (7.96)	96.30 (0.68)	96.13 (0.60)
	6	404 (6.47)	94.26 (0.79)	90.70 (1.15)	865 (7.60)	96.41 (0.63)	96.31 (0.38)
	7	452 (4.37)	94.66 (0.67)	91.95 (1.05)	1015 (10.19)	96.44 (0.46)	96.31 (0.30)
	8	494 (6.34)	94.76 (0.66)	92.01 (1.09)	1184 (11.17)	96.46 (0.45)	96.39 (0.38)
	9	537 (4.47)	95.13 (0.73)	93.10 (1.07)	1354 (10.12)	**96.55 (0.40)**	**96.52 (0.41)**
	10	578 (5.44)	**95.32 (0.74)**	**93.37 (1.04)**	1540 (8.83)	96.50 (0.42)	96.51 (0.24)
CHI	1	224 (4.14)	58.09 (1.38)	78.53 (1.14)	317 (5.14)	85.68 (2.74)	89.49 (1.10)
	2	323 (4.20)	67.77 (2.44)	82.69 (1.24)	497 (3.61)	93.39 (0.77)	94.23 (0.67)
	3	405 (5.12)	74.65 (1.11)	84.76 (0.87)	646 (6.73)	95.11 (0.66)	95.47 (0.71)
	4	479 (4.55)	80.80 (1.60)	87.97 (1.00)	780 (9.20)	95.95 (0.45)	96.17 (0.71)
	5	554 (4.37)	83.51 (1.49)	89.68 (0.90)	927 (9.10)	96.34 (0.43)	96.49 (0.67)
	6	632 (5.45)	84.82 (1.05)	90.29 (0.74)	1081 (9.78)	96.53 (0.46)	96.64 (0.54)
	7	714 (5.17)	86.02 (1.21)	91.18 (0.87)	1256 (8.56)	96.66 (0.44)	96.68 (0.56)
	8	789 (6.93)	87.08 (1.58)	92.06 (1.12)	1436 (11.93)	96.60 (0.43)	96.61 (0.56)
	9	870 (8.36)	92.05 (1.74)	93.86 (1.14)	1640 (15.26)	96.72 (0.36)	96.73 (0.49)
	10	948 (6.98)	**92.60 (1.67)**	**94.27 (1.18)**	1858 (19.49)	**96.72 (0.34)**	**96.85 (0.47)**