UNIVERSIDADE FEDERAL DO ESPÍRITO SANTO
CENTRO TECNOLÓGICO
PROGRAMA DE PÓS-GRADUAÇÃO EM INFORMÁTICA

CARIBE ZAMPIROLLI DE SOUZA

MEDIDA DE CERTEZA NA CATEGORIZAÇÃO MULTI-RÓTULO DE TEXTO E SUA UTILIZAÇÃO COMO ESTRATÉGIA DE PODA DO RANKING DE CATEGORIAS

VITÓRIA
2010

CARIBE ZAMPIROLLI DE SOUZA

MEDIDA DE CERTEZA NA CATEGORIZAÇÃO MULTI-RÓTULO DE TEXTO E SUA UTILIZAÇÃO COMO ESTRATÉGIA DE PODA DO RANKING DE CATEGORIAS

Dissertação apresentada ao Programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para obtenção do Grau de Mestre em Informática.

Orientadora: Profa. Dra. Claudine Santos Badue Gonçalves

VITÓRIA
2010

BIBLIOTECA

CARIBE ZAMPIROLLI DE SOUZA

MEDIDA DE CERTEZA NA CATEGORIZAÇÃO MULTI-RÓTULO DE TEXTO E SUA UTILIZAÇÃO COMO ESTRATÉGIA DE PODA DO RANKING DE CATEGORIAS

Dissertação submetida ao programa de Pós-Graduação em Informática do Centro Tecnológico da Universidade Federal do Espírito Santo, como requisito parcial para a obtenção do Grau de Mestre em Informática.

Aprovada em 27 de agosto de 2010.

COMISSÃO EXAMINADORA

Profa. Dra. Claudine Santos Badue Gonçalves - Orientador
Universidade Federal do Espírito Santo

Prof. Dr. Alberto Ferreira De Souza – Co-orientador
Universidade Federal do Espírito Santo

Prof. Dr. Elias de Oliveira
Universidade Federal do Espírito Santo

Prof. Dr. Wagner Meira Jr.
Universidade Federal de Minas Gerais

EPÍGRAFE

“A maior de todas as torres começa aqui no solo.” (Provérbio Chinês)

DEDICATÓRIA

Dedico este trabalho aos meus pais Carlos e Fátima que me guiaram pelos caminhos corretos e que de muitas formas me incentivaram e ajudaram para que fosse possível a concretização deste trabalho.

AGRADECIMENTOS

Agradeço primeiramente a Deus pelo sol que ilumina nossos dias ofertando-nos claridade para guiarmos nossos passos; e por todas as oportunidades de crescimento e amadurecimento.

Aos meus pais pela vida, pela segurança dos primeiros passos, pelo direcionamento na vida, pelo abraço amigo e palavras de conforto.

À minha orientadora Prof. Dr. Claudine Badue pelo suporte teórico e confiável no decorrer do projeto, paciente e dedicado fundamentais nas várias etapas de desenvolvimento deste trabalho.

À equipe do SCAE de pesquisadores pelo apoio nos estudos, em especial ao Prof. Dr. Alberto de Souza, Bruno Zanetti Melotti e Felipe Thomaz Pedroni.

Aos meus irmãos pelo carinho e pelo apoio de sempre.

À minha namorada pelo carinho, compreensão nas horas de ausência e pelas horas no celular para me tirar da frente do computador (hehehe).

Aos meus amigos Paulim e Vello pelo companheirismo e incentivo no decorrer desta caminhada.

RESUMO

Dado um documento de entrada, um sistema de categorização multi-rótulo de texto tipicamente computa graus de crença para as categorias de um conjunto pré-definido, ordena as categorias por grau de crença, e atribui ao documento as categorias com grau de crença superior a um determinado limiar de poda. Idealmente, o grau de crença deveria informar a probabilidade do documento de fato pertencer à categoria. Infelizmente, ainda não existem categorizadores que computam tais probabilidades e mapear graus de crença em probabilidades é um problema ainda pouco explorado na área de RI.

Neste trabalho, propomos um método baseado na regra de *Bayes* para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto. Propomos também uma estratégia para determinar limiares de poda baseada na medida de certeza de categorização - *bayesian cut* (BCut) - e uma variante para BCut - *position based bayesian CUT* (PBCut). Avaliamos experimentalmente o impacto dos métodos propostos no desempenho de duas técnicas de categorização multi-rótulo de texto, k-vizinhos mais próximos multi-rótulo (*ML-kNN*) e rede neural sem peso do tipo *VG-RAM* com correlação de dados (*VG-RAM WNN-COR*), no contexto da categorização de descrições de atividades econômicas de empresas brasileiras segundo a Classificação Nacional de Atividades Econômicas (CNAE). Investigamos também o impacto no desempenho de categorização multi-rótulo de texto de três métodos de poda comumente usados na literatura de RI - RCut, PCut, e SCut – e uma variante de RCut - RTCut. Além disso, propomos novas variantes para PCut e SCut – PCut* e SCut*, respectivamente – para tratar problemas existentes nestas abordagens. Nossos resultados experimentais mostram que, usando nosso método de geração de medidas de certeza de categorização, é possível prever o quão certo está o categorizador de que as categorias por ele preditas são de fato pertinentes para um dado documento. Nossos resultados mostram também que o uso de nossas estratégias de poda BCut e PBCut produz desempenho de categorização superior ao de todas as outras estratégias consideradas em termos de precisão.

ABSTRACT

A multi-label text categorization system typically computes degrees of belief when it comes to the categories of a pre-defined set, orders the categories by degree of belief, and attributes to the document categories with a higher degree of belief to determined threshold cut. It would be ideal if the degree of belief could inform the probability of the document be part of this category. Unfortunately, there isn't a categorization system that computes such probabilities and to map degrees of belief in probabilities is still a problem that isn`t well explored in IR. In this paper we propose a method based on Bayes rules to map degrees of belief in terms of multi-label text measures of categorization. There are other contributions in this work such as an strategy to determine the limits of threshold cut based on bayesian cut (BCut) and a variant for PBCut (position based bayesian CUT).

As an experience, we evaluated the impact of the proposed methods when performing the two techniques of the multi-label text categorization. The first technique is called k-nearest neighbor multi-label (ML-KNN) and the second technique is called VG-RAM weightless Neural Networks. Theses evaluations were made in the context of the categorization of economic activities description of Brazilian enterprises, according to the Economic Activities Classification in Brazil (CNAE).

In this work we also investigated the impact in the performance of multi-label text categorization of the three cut methods commonly used in the IR literature: RCut, PCut, SCut and RTCut. Moreover, we propose a new variant for the so called PCut* and a new variant for SCut*. Finally, this work shows that the cut approach proposed, BCut and PBCut, produces a categorization performance superior to the other strategies presented in the literature of IR.

SUMÁRIO

LISTA DE FIGURAS

LISTA DE TABELAS

1 INTRODUÇÃO

Com o advento da comunicação, um grande volume de informação - crescente a taxas exponenciais - trafega pela *World Wide Web* e pelas *Intranets* corporativas espalhadas em todo o mundo. Apenas a quantidade de dados textuais disponíveis na *Web* é estimada ser da ordem de bilhares de documentos de texto (http://www.worldwidewebsize.com/). Isto dispara a necessidade de ferramentas eficientes para gerenciar, recuperar, e filtrar informação dessas grandes bases de dados textuais.

Categorização de texto, a atividade de rotular textos em linguagem natural com categorias temáticas a partir de um conjunto pré-definido [Sebastiani02], é uma dessas ferramentas importantes para a gestão de dados sob a forma de texto. Contudo, a categorização manual é um processo demorado e custoso, o que limita sua aplicabilidade. Conseqüentemente, existe um grande interesse no meio acadêmico e industrial em desenvolver técnicas para categorização automática de texto [Sebastiani02].

Categorização automática de texto é ainda um problema computacionalmente muito desafiador para as comunidades de Recuperação de Informação (RI), tanto no contexto acadêmico quanto no industrial. A maioria dos trabalhos sobre categorização de texto na literatura está focada nos problemas de categorização de texto com um uni-rótulo (*single-label*), nos quais cada documento pode ter um único rótulo (ou pertencer a uma única categoria) [Sebastiani02]. Entretanto, em problemas do mundo real, a categorização multi-rótulo (*multi-label*), na qual os documentos podem receber mais de um rótulo, é freqüentemente necessária [McCallum99, Schapire00, Clare01, Elisseeff02, Comité03, Ueda03, Boutell04, Kazawa05, Zhang06, Zhang07].

Diversas técnicas têm sido propostas para atacar o problema de categorização multi-rótulo, tais como árvores de decisão (*decision trees* [Clare01, Comité03]), máquinas de vetores de suporte (*support vector machines* - SVM [Elisseeff02, Boutell04, Kazawa05]), redes neurais (*neural networks* [Romero04, Zhang06, DeSouza09a, DeSouza09b]), *k*-vizinhos mais próximos (*k-nearest neighbors* – ML-*k*NN [Zhang07]), *boosting* [Schapire99], e muitas delas especificamente para categorização multi-rótulo de texto [McCallum99, Schapire99, Ueda03, Gao04, Romero04, Zhang06, Zhang07, DeSouza09a, DeSouza09b].

Dado um documento de entrada d_j, um sistema de categorização multi-rótulo de texto tipicamente computa um valor real $f(d_j, c_i)$ para cada categoria c_i de um conjunto pré-definido.

Este valor $f(d_j, c_i)$ indica o grau de crença com que o sistema atribui a categoria c_i ao documento d_j. O sistema ordena as categorias por grau crença, formando um *ranking* de categorias para o documento de entrada. As categorias c_i posicionadas no *ranking* acima de um determinado limiar de poda τ_i são então atribuídas ao documento de entrada d_j, ou seja, c_i é predita para d_j se $f(d_j, c_i) \geq \tau_i$.

Idealmente, o grau de crença $f(d_j, c_i)$ computado por um categorizador deveria informar qual é a probabilidade do documento d_j de fato pertencer à categoria c_i. Infelizmente, ainda não existem categorizadores que computam tais probabilidades e mapear graus de crença em probabilidades é um problema ainda pouco explorado na área de RI.

Neste trabalho, propomos um método para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto. Nosso método é baseado na regra de *Bayes*, que permite alterar as probabilidades *a priori* tendo em conta novas evidências de forma a obter as probabilidades *a posteriori*. Em nosso método, as probabilidades *a priori* são estimadas empiricamente, através de experimentos de calibração, e a regra de *Bayes* é utilizada para produzir as probabilidades *a posteriori* de interesse, que denominamos "medida de certeza de categorização". Ou seja, em nossa abordagem, dado um documento de entrada d_j, um categorizador computa $f(d_j, c_i)$ para cada categoria c_i e, usando resultados experimentais prévios e a regra de *Bayes*, uma medida de certeza da categorização de d_j em c_i é estimada.

Neste trabalho, propomos também uma estratégia para determinar limiares de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto descrita acima, a qual denominamos *bayesian cut* (BCut). Na estratégia de poda BCut, um único limiar de poda, τ, para todas as categorias c_i é escolhido de modo a maximizar o desempenho de categorização, i.e., sua habilidade de atribuir todas e apenas as categorias pertinentes a um dado documento. Um único limiar de poda pode produzir bom desempenho para todas as categorias devido à nossa metodologia de mapear graus de crença em medidas de certeza de categorização – o limiar de poda, τ, está associado à probabilidade da categorização estar correta, independente da categoria c_i. Além disso, propomos uma variante para BCut que utiliza diferentes limiares de poda τ_p para diferentes posições p do *ranking*, a qual denominamos *position based bayesian cut* (PBCut). A estratégia de poda PBCut pode produzir um desempenho superior ao de BCut, porque a medida de certeza de categorização em uma dada categoria diminui à medida que a posição da categoria no *ranking* aumenta.

Avaliamos experimentalmente o impacto dos métodos propostos no desempenho de duas técnicas de categorização multi-rótulo de texto, k-vizinhos mais próximos multi-rótulo

(*multi-label k-nearest neighbors - ML-kNN*) [Zhang07] e rede neural sem peso do tipo *VG-RAM* com correlação de dados (*data correlated virtual generalizing random access memory weightless neural networks – VG-RAM WNN-COR*) [Aleksander98, Badue08, DeSouza08, DeSouza09a, DeSouza09b], no contexto da categorização de descrições de atividades econômicas de empresas brasileiras segundo a Classificação Nacional de Atividades Econômicas (CNAE) [CNAE03]. Investigamos também o impacto no desempenho de categorização multi-rótulo de texto de três métodos de poda comumente usados na literatura de RI [Yang01, Lee02, Fan07]: (i) RCut, baseada na posição das categorias no *ranking*; (ii) PCut, baseada na popularidade das categorias no conjunto de treinamento; (iii) SCut, baseada no grau de crença com que o sistema atribui as categorias aos documentos; e (iv) uma variante de RCut - RTCut [Yang01]. Ademais, propomos novas variantes para PCut e SCut – PCut* e SCut*, respectivamente – para tratar problemas existentes nestas abordagens. Em nossa análise experimental, utilizamos as métricas mais relevantes de avaliação de desempenho de categorização multi-rótulo de texto empregadas pela comunidade de RI: *exact match* [Kazawa05], *precision* [Sebastiani02, Manning08], *recall* [Sebastiani02, Manning08], e F_1 [Sebastiani02, Manning08]. Nossos resultados experimentais mostram que, usando nosso método de geração de medidas de certeza de categorização, é possível prever o quão certo está o categorizador de que as categorias por ele preditas são de fato pertinentes para um dado documento. Nossos resultados mostram também que uso o de nossas estratégias de poda BCut e PBCUT produz desempenho de categorização superior ao de todas as outras estratégias de poda consideradas em termos das métricas *precision* e *exact match*.

1.1 Motivações

Dado um documento de entrada d_j, um sistema de categorização multi-rótulo de texto tipicamente computa um valor real $f(d_j, c_i)$ para cada categoria c_i de um conjunto pré-definido. Este valor $f(d_j, c_i)$ indica o grau de crença com que o sistema atribui a categoria c_i ao documento d_j e, idealmente, deveria informar a probabilidade de d_j de fato pertencer a c_i. Infelizmente, ainda não existem categorizadores que computam tais probabilidades e mapear graus de crença em probabilidades é um problema ainda pouco explorado na área de RI. Assim, a principal motivação para o desenvolvimento deste trabalho foi desenvolver um

método para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto, tal medida facilitaria a compreensão do operador do sistema.

O sistema de categorização multi-rótulo de texto tipicamente ordena as categorias por grau crença, formando um *ranking* de categorias para o documento de entrada. As categorias c_i posicionadas no *ranking* acima de um determinado limiar de poda τ_i são então atribuídas ao documento de entrada d_j, ou seja, c_i é predita para d_j se $f(d_j, c_i) \geq \tau_i$. Estratégias para determinar limiares de poda para o *ranking* de categorias é um tópico de pesquisa pouco explorado na área de RI. Assim, outra motivação para o desenvolvimento deste trabalho foi desenvolver uma estratégia de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto.

A motivação para este trabalho surgiu durante o desenvolvimento do Sistema Computacional de Codificação Automática de Atividades Econômicas (SCAE). Tal sistema se propõe a categorizar automaticamente, segundo a CNAE, descrições, na forma de texto livre, de atividades econômicas de empresas brasileiras.

A CNAE lista todas as atividades econômicas legalmente reconhecidas no Brasil. Correntemente, a CNAE contempla 1.301 atividades econômicas, cada uma possuindo um código específico. Empresas podem ser categorizadas dentro de um ou mais códigos; ou seja, categorizar empresas segundo a CNAE é um problema de categorização multi-rótulo. Devido à grande quantidade de categorias, este é um problema complexo e incomum na literatura.

1.2 Objetivos

Ainda não existem categorizadores que computam a probabilidade de um dado documento de entrada d_j de fato pertencer a uma categoria c_i de um conjunto pré-definido. Por esta razão, o principal objetivo deste trabalho foi (i) desenvolver um método para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto. Foi também objetivo deste trabalho (ii) desenvolver uma estratégia para determinar limiares de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto, (iii) e avaliar experimentalmente o impacto dos métodos propostos no desempenho da categorização multi-rótulo de texto.

1.3 Contribuições

As principais contribuições deste trabalho foram:

- Elaboração de um método para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto;
- Elaboração de uma estratégia para determinar limiares de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto;
- Elaboração de novas variantes para estratégias comumente utilizadas na literatura para tratar problemas existentes nestas abordagens;
- Implementação dos métodos para mapeamento dos graus de crença em medidas de certeza de categorização e determinação de limiares de poda para o *ranking* de categorias baseada nas medidas de certeza de categorização;
- Avaliação experimental do impacto dos métodos propostos no desempenho da categorização multi-rótulo de texto.

1.4 Organização da Dissertação

Após esta introdução, esta dissertação está organizada da seguinte forma:

- O Capítulo 2 discute a categorização multi-rótulo de texto, apresentando (i) uma definição de categorização multi-rótulo de texto, (ii) a representação do conteúdo do documento por um vetor de pesos de termos, (iii) os métodos de categorização multi-rótulo de texto usados neste trabalho; e (iv) o problema de categorização multi-rótulo de atividades econômicas utilizado como estudo de caso;
- O Capítulo 3 apresenta os três métodos de poda de *ranking* de categorias comumente usados na literatura de RI e avaliados neste trabalho, além de suas variantes;
- O Capítulo 4 propõe um método para mapear graus de crença em medidas de certeza de categorização baseada na regra de Bayes e uma estratégia para determinar limiares de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto;

- O Capítulo 5 descreve nossa metodologia experimental, apresentando (i) as bases de dados usadas neste trabalho, (ii) o pré-processamento das bases de dados, (iii) a validação cruzada empregada na avaliação do desempenho dos categorizadores; (iv) a calibração dos categorizadores, (v) os parâmetros do método baseado na regra de *Bayes* para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto, (vi) o procedimento de validação das medidas de certeza de categorização multi-rótulo de texto, e (vii) a calibração das estratégias de poda do *ranking* de categorias;
- O Capítulo 6 descreve nossos resultados experimentais, apresentando (i) a avaliação da medida de certeza de categorização multi-rótulo de texto; e (ii) a análise do impacto das estratégias de poda no desempenho dos categorizadores multi-rótulo de texto;
- O Capítulo 7 discute este trabalho de pesquisa, apresentando trabalhos correlatos e uma análise crítica desta dissertação;
- Finalmente, o Capítulo 8 apresenta nossas conclusões e direções para trabalhos futuros.

2 CATEGORIZAÇÃO MULTI-RÓTULO DE TEXTO

Neste capítulo, apresentamos uma definição de categorização multi-rótulo de texto e descrevemos os métodos de categorização de texto *k*-vizinhos mais próximos multi-rótulo (*multi-label k-nearest neighbors* - ML-*k*NN) [Zhang07] e rede neural sem peso do tipo VG-RAM com correlação de dados (*data correlated virtual generalizing random access memory weightless neural networks* – VG-RAM WNN-COR) [Aleksander98, Badue08, DeSouza08, DeSouza09a, DeSouza09b]. Apresentamos, também, o domínio do problema de descrições de atividades econômicas de empresas brasileiras segundo a Classificação Nacional de Atividades Econômicas (CNAE) [CNAE03], e como essas descrições de atividades são representadas internamente, segundo o modelo vetorial [Salton75, Baeza99], nas técnicas de categorização multi-rótulo de texto. O conteúdo deste capítulo foi fundamentalmente extraído de [Melotti09].

2.1 Categorização Multi-Rótulo de Texto

Sejam D um domínio de documentos e $C = \{c_1, \ldots, c_{|C|}\}$ um conjunto de categorias pré-definido. Na categorização multi-rótulo de texto, os documentos de D podem ser categorizados dentro de uma ou mais categorias de C.

Seja $\Omega = \{d_1, \ldots, d_{|\Omega|}\} \subset D$ um *corpus* inicial de documentos previamente categorizados manualmente por especialistas no domínio dentro de subconjuntos de C. Em sistemas automáticos para categorização multi-rótulo, um subconjunto de Ω, denominado conjunto de treinamento (e calibração), $TV = \{d_1, \ldots, d_{|TV|}\}$, pode ser utilizado para treinar (e calibrar) categorizadores implementados segundo técnicas de aprendizado de máquina [Sebastiani02] (neste trabalho empregamos somente categorizadores automáticos baseados em técnicas de aprendizado de máquina). O conjunto de teste, $Te = \{d_{|TV|+1}, \ldots, d_{|\Omega|}\} = \Omega - TV$, por outro lado, consiste dos documentos não empregados no treinamento dos sistemas de categorização e somente submetidos a estes em fase de teste. Depois de ser treinado (e calibrado) com TV, um sistema de categorização pode ser utilizado para predizer o conjunto de categorias de cada documento em Te.

Sistemas automáticos para categorização multi-rótulo tipicamente implementam uma função $f : D \times C \rightarrow \Re$ que retorna o grau de crença para cada par $\langle d_j, c_i \rangle \in D \times C$, ou seja, um número entre 0 e 1 que, a grosso modo, representa evidência de que o documento de teste d_j deve ser categorizado dentro da categoria c_i. A função $f(.,.)$ pode ser transformada em uma função ranqueadora $r(.,.)$, tal que, se $f(d_j, c_i) > f(d_j, c_k)$, então $r(d_j, c_i) < r(d_j, c_k)$, e se $f(d_j, c_i) < f(d_j, c_k)$, então $r(d_j, c_i) > r(d_j, c_k)$. Esta forma de usar o categorizador de texto é conhecida como "categorização orientada a documento" ("*document-pivoted categorization*" [Sebastiani02]), que consiste em, dado um documento d_j, encontrar todas as categorias $c_i \in C$ pertinentes a d_j. Alternativamente, outra forma de usar o categorizador é, dada uma categoria $c_i \in C$, encontrar todos os documentos d_j associados a c_i, conhecida como "categorização orientada a categoria" ("*category-pivoted categorization*" [Sebastiani02]). Neste trabalho, avaliamos categorização orientada a documento (ver Seções 2.3.2 e 2.4.2).

Seja C_j o conjunto de categorias pertinentes (categorias especificadas pelos especialistas no domínio) ao documento de teste d_j e $\hat{C}_j$ o conjunto de categorias preditas para d_j por um categorizador automático. Um bom categorizador automático tenderá a posicionar as categorias de C_j em posições mais elevadas no *ranking* do que aquelas que não pertencem a C_j. As categorias c_i cujo grau de crença é superior ao limiar de poda τ_i são então preditas para o documento de teste d_j, isto é, $\hat{C}_j = \{c_i \mid f(d_j, c_i) \geq \tau_i\}$. Diferentes limiares τ_i são tipicamente escolhidos para diferentes categorias c_i.

2.2 Representação Vetorial de Documentos

Os documentos, em seu formato original (texto livre), usualmente não podem ser tratados diretamente por técnicas de aprendizado de máquina empregadas na construção de categorizadores automáticos de texto. Na maioria das técnicas de aprendizado de máquina, cada documento do conjunto Ω é representado por um vetor de números na representação ponto-flutuante; esta forma de representação de documentos é conhecida na literatura como representação vetorial de documentos [Salton75, Baeza99]. Cada elemento deste vetor

quantifica a freqüência com que um termo, pertencente a um vocabulário de termos conhecidos pelo categorizador, aparece em *TV* (*bag-of-words representation* – [Baeza99, Sebastiani02]). Um termo é simplesmente uma ou mais palavras cujo significado, ou semântica, é representativo para o documento [Baeza99, Sebastiani02].

Formalmente, no modelo vetorial de representação de documentos [Salton75, Baeza99], os documentos são representados por vetores no espaço $\Re^n$, onde n representa o número de termos do vocabulário de termos conhecidos pelo categorizador. Cada documento d_j do conjunto Ω é representado por um vetor de pesos $\vec{d}_j = \langle w_{1j}, w_{2j}, \ldots, w_{|T|j} \rangle$, onde T é o conjunto dos termos que ocorrem pelo menos uma vez nos documentos de *TV* e w_{kj} representa o peso do termo t_k do documento d_j; a ordem dos termos em $\vec{d}_j$ é a mesma para qualquer *j* [Sebastiani02].

A Figura 2-1 mostra um exemplo de um *corpus* formado pelo conjunto de documentos $\Omega = \{d_1, d_2, d_3\}$, representados vetorialmente por meio de vetores tridimensionais, onde cada dimensão está associada aos pesos dos termos do conjunto $T = \{t_1, t_2, t_3\}$ nos documentos. O documento d_1 é representado pelo vetor $\vec{d}_1 = \langle w_{11}, w_{21}, w_{31} \rangle$, d_2 por $\vec{d}_2 = \langle w_{12}, w_{22}, w_{32} \rangle$ e d_3 por $\vec{d}_3 = \langle w_{13}, w_{23}, w_{33} \rangle$.

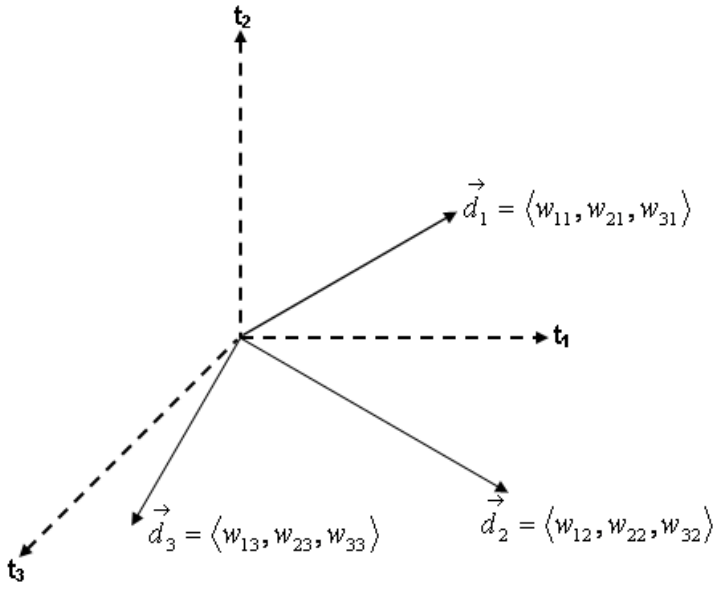


Figura 2-1 - Representação gráfica de três documentos de acordo com o modelo vetorial.

Para determinar o peso w_{kj} do termo t_k no documento d_j, diversas formulações podem ser utilizadas. Empregamos a função de ponderação conhecida como *tfidf* (*term frequency, inverse document frequency*), definida por [Sebastiani02]:

$$tfidf(t_k, d_j) = \#(t_k, d_j) * \log\left(\frac{|TV|}{\#TV(t_k)}\right) \tag{2.1}$$

onde $\#(t_k, d_j)$ representa o número de vezes que o termo t_k ocorre no documento d_j, chamada de freqüência do termo (*term frequency – tf*); $\#TV(t_k)$ denota o número de documentos do conjunto TV em que o termo t_k ocorre; e o termo $\log\left(\frac{|TV|}{\#TV(t_k)}\right)$ é chamado de freqüência inversa do documento (*inverse document frequency– idf*).

A função *tfidf* codifica a intuição de que (i) quanto mais freqüente um termo em um documento, maior é a importância semântica dele para o documento, e (ii) quanto mais freqüente um termo no conjunto de documentos TV, menor é o poder de discriminação dele. Esta formulação leva em consideração apenas a ocorrência dos termos, não considerando a ordem na qual eles aparecem nos documentos e o papel sintático que eles possuem. É importante observar que os pesos dos termos são mutuamente independentes, isto é, o peso w_{kj} calculado para o par (t_k, d_j) não diz nada a respeito do peso w_{k+1j} calculado para o par (t_{k+1}, d_j) [Baeza99].

Para que os pesos estejam no intervalo [0, 1] e para que os documentos sejam representados por vetores de mesma magnitude, os pesos calculados por *tfidf* são freqüentemente normalizados pela função de normalização de co-seno, definida por [Sebastiani02]:

$$w_{kj} = \frac{tfidf(t_k, d_j)}{\sqrt{\sum_{s=1}^{|T|} (tfidf(t_s, d_j))^2}} \tag{2.2}$$

O procedimento de transformar os textos dos documentos em uma forma que possa ser interpretada pelas técnicas de categorização de texto é chamado de indexação (*indexing*). A função de indexação *tfidf* foi escolhida para ser utilizada neste trabalho por ser a mais empregada na literatura [Sebastiani02], ou seja, as técnicas de categorização multi-rótulo

examinadas neste trabalho têm como entrada documentos (ou descrições de atividades econômicas) representados por vetores cujos pesos dos termos são calculados pela função *tfidf*. Estas técnicas são apresentadas nas seções a seguir.

2.3 Categorizador *kNN*

A técnica *k*-vizinhos mais próximos (*k-nearest neighbor* – *kNN*) é baseada em exemplos (*instance-based* [Mitchell97]), isto é, nenhum modelo é criado para extrair as características de um documento e associá-las a um conjunto de categorias na base de treinamento. Métodos baseados em exemplo são, algumas vezes, referenciados como métodos de aprendizado "preguiçosos" (*lazy learning methods*) porque eles somente processam a base de treinamento ao receber uma nova requisição de categorização para um novo documento [Mitchell97].

O *kNN* tradicional é utilizado na literatura em contextos uni-rótulo, mas o problema em que estamos interessados é multi-rótulo. Para empregar o categorizador *kNN* em problemas multi-rótulo, ele precisa ser alterado. A seção 2.3.1 apresenta o *kNN* uni-rótulo e a seção 2.3.2 o *k*-vizinhos mais próximos multi-rótulo (*multi-label k-nearest neighbor* – ML-kNN) proposto por Zhang e Zhou (2007).

2.3.1 Categorizador *kNN* Uni-Rótulo

O algoritmo *kNN* é muito simples: dado um documento de teste d_j, o sistema busca, empregando uma métrica de distância, os k vizinhos (documentos) mais próximos a d_j no conjunto de treinamento, *TV*, e escolhe a categoria c_i a ser atribuída a d_j dentre as categorias desses *k* vizinhos [Yang99]. Várias métricas de distância podem ser utilizadas, mas a mais freqüente é o co-seno do ângulo entre o vetor que representa d_j, $\vec{d}_j$, e cada documento d_m de *TV*, $\vec{d}_m$ [Sebastiani02]:

$$\cos(\vec{d}_j, \vec{d}_t) = \frac{\vec{d}_j \bullet \vec{d}_m}{|\vec{d}_j| \times |\vec{d}_m|} \quad (2.3)$$

O valor do co-seno do ângulo entre o vetor de d_j e o de cada vizinho d_n é usado como o grau de crença do categorizador de que a categoria de d_n deve ser atribuída a d_j. A partir deste grau de crença, duas estratégias podem ser utilizadas para predizer uma categoria a d_j: maioria dos votos (*majority votes*) e a soma dos co-senos (similaridades) (*similarity score summing*) [Baoli03, Yavuz98, Yang99]. Na primeira, a categoria mais freqüente, entre as dos k vizinhos mais próximos, é a escolhida. Na segunda, a categoria com a maior soma dos co-senos, entre as dos k vizinhos mais próximos, é a escolhida.

2.3.2 Categorizador *ML-kNN*

O *Multi-Label k-Nearest Neighbor* (*ML-k NN*) [Zhang07] é um categorizador multi-rótulo baseado no algoritmo k NN uni-rótulo. Dado um documento de teste d_j, o *ML-k NN* identifica os k documentos da base de treinamento mais similares a d_j utilizando a métrica de distância co-seno (Equação **(2.3)**). Posteriormente, o algoritmo identifica a freqüência de cada categoria nestes k documentos. Utilizando esta informação, o *ML-k NN* prediz um conjunto de categorias para d_j utilizando o *maximum a posteriori principle* (MAP) [Sparacino00].

Formalmente, dado o documento $d_m \in TV$ e o conjunto de categorias pertinentes de d_m, $C_m \subseteq C$, podemos definir: (i) o vetor de categorias de d_m, $\vec{y}_{d_m}$, de tamanho igual |*C*|, onde $y_{d_m}(\vec{c}_i)$ recebe 1 se $c_i \in C_m$ e zero caso contrário; e (ii) o conjunto dos k vizinhos mais próximos a d_m no conjunto de treinamento TV, $N(d_m)$.

Durante a fase de treinamento, baseado no conjunto de categorias associadas aos documentos d_m pertencentes a $N(d_m)$, um vetor de contagem de associação (*membership counting vector* [Zhang07]), $\vec{C}_{d_m}$, de tamanho igual |*C*|, é computado segundo a Equação **(2.4)** para cada $d_m \in TV$:

$$\vec{C}_{d_m}(c_i) = \sum_{a \in N(d_m)} \vec{y}_a(c_i) \tag{2.4}$$

O vetor $\vec{C}_{d_m}$ sumariza a vizinhança de d_m em *TV* com respeito às categorias associadas aos documentos em $N(d_m)$.

Na fase de teste, para cada documento d_j em Te, o *ML-k NN* primeiramente identifica os k vizinhos mais próximos à d_j, $N(d_j)$, no conjunto TV. Seja $H_1^{c_i}$ um evento no qual a categoria c_i está associada a d_j; $H_0^{c_i}$ um evento no qual a categoria c_i não está associada a d_j; e $E_n^{c_i}(n \in \{0,1,\ldots,k\})$ um evento no qual existem exatamente n documentos associados à categoria c_i. Baseado no vetor de contagem de associação de d_j, $\vec{C}_{d_j}$, o vetor de categorias $\vec{y}_{d_j}$ pode ser determinado pelo *MAP*, conforme Equação **(2.5)**:

$$\vec{y}_{d_j}(c_i) = \arg\max_{b \in \{0,1\}} P(H_b^{c_i} \mid E^{c_i}_{\vec{C}_{d_j}(c_i)}) \tag{2.5}$$

Pela regra de *Bayes*, a Equação **(2.5)** pode ser reescrita conforme Equação **(2.6)**:

$$\vec{y}_{d_j}(c_i) = \arg\max_{b \in \{0,1\}} \frac{P(H_b^{c_i}) P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H_b^{c_i})}{P(E^{c_i}_{\vec{C}_{d_j}(c_i)})} \tag{2.6}$$

Eliminando o denominador $P(E^{c_i}_{\vec{C}_{d_j}(c_i)})$, pois é independente de $P(H_b^{c_i})$, temos a equação final para a obtenção do vetor de categorias preditas para d_j:

$$\vec{y}_{d_j}(c_i) = \arg\max_{b \in \{0,1\}} P(H_b^{c_i}) P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H_b^{c_i}) \tag{2.7}$$

A Equação **(2.7)** mostra que, para determinar o vetor de categorias preditas $\vec{y}_{d_j}$, toda a informação sobre as probabilidades *a priori*, $P(H_b^{c_i})$, e *a posteriori*, $P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H_b^{c_i})$, são

necessárias. Na verdade, essas probabilidades podem ser estimadas a partir da freqüência das categorias no conjunto de treinamento. A Figura 2-2 mostra o pseudocódigo do *ML-k NN* [Zhang07].

$[\vec{y}_{d_t}, \vec{r}_{d_t}] = ML-KNN(TV, k, d_j, s)$

%Computa a probabilidade a priori $P(H_b^{c_i})$

(1) **para** $c_i \in C$ **faça**

(2) $P(H_1^{c_i}) = (s + \sum_{i=1}^{|TV|} \vec{y}_{x_i}(c_i)) / (sx2 + |TV|);\ \ P(H_0^{c_i}) = 1 - P(H_1^{c_i}),)$

%Computa a probabilidade a posterior $P(E^{c_i}_{\vec{C}_{d_t}(c_i)} \mid H_b^{c_i})$

(3) **Identifica** $N(x_i),\ \ i \in \{1,2,\ldots,|TV|\};$

(4) **para** $c_i \in C$ **faça**

(5) **para** $j \in \{0,1,\ldots,k\}$ **faça**

(6) $c[j] = 0;\ \ c'[j] = 0;$

(7) **para** $l \in \{0,1,2,\ldots,|TV|\}$ **faça**

(8) $\delta = \vec{C}_{x_l}(c_i) = \sum_{a \in N(x_l)} \vec{y}_a(c_i);$

(9) **se** $(\vec{y}_{x_l}(c_i) == 1)$ **então** $c[\delta] = c[\delta] + 1;$

(10) **senão** $c'[\delta] = c'[\delta] + 1;$

(11) **para** $j \in \{0,1,\ldots,k\}$ **faça**

(12) $P(E_j^{c_i} \mid H_1^{c_i}) = (s + c[j]) / (sx(k+1) + \sum_{p=0}^{k} c[p]);$

(13) $P(E_j^{c_i} \mid H_0^{c_i}) = (s + c'[j]) / (sx(k+1) + \sum_{p=0}^{k} c'[p]);$

%Computa $\vec{y}_{d_t}$ e $\vec{r}_{d_t}$

(14) **Identifica** $N(d_j);$

(15) **para** $c_i \in C$ **faça**

(16) $\vec{C}_{d_t}(c_i) = \sum_{a \in N(d_t)} \vec{y}_a(c_i);$

(17) $\vec{y}_{d_t}(c_i) = \arg\max_{b \in \{0,1\}} P(H_b^{c_i}) P(E^{c_i}_{\vec{C}_{d_t}(c_i)} \mid H_b^{c_i});$

(18) $\vec{r}_{d_t}(c_i) = P(H_1^{c_i} \mid E^{c_i}_{\vec{C}_{d_t}(c_i)}) = (P(H_1^{c_i}) P(E^{c_i}_{\vec{C}_{d_t}(c_i)} \mid H_1^{c_i})) / P(E^{c_i}_{\vec{C}_{d_t}(c_i)});$

Figura 2-2 - Pseudocódigo do algoritmo *ML-k NN*.

Os parâmetros de entrada do algoritmo são TV, k, d_j e s. O parâmetro s controla a suavização da probabilidade a priori e, neste trabalho, optamos em utilizar o valor $s = 1$ (suavização Laplaciana [Zhang07]). De acordo com a Figura 2-2, os passos (1) e (2) calculam a probabilidade *a priori*, $P(H_b^{c_i})$. Os passos de (3) a (13) estimam a probabilidade *a posteriori*, $P(E^{c_i}_{\vec{C}_{d_j}(c_i)} \mid H_b^{c_i})$, onde $c[j]$ contabiliza o número de documentos entre os k documentos similares no conjunto de treinamento que possuem a categoria c_i. Correspondentemente, $c'[j]$ contabiliza o número de documentos entre os k documentos similares no conjunto de treinamento que não possuem a categoria c_i. Finalmente, os passos

(14) a (18) são a predição do algoritmo, isto é, a atribuição de um grau de crença para cada categoria $c_i \subseteq C$ referente ao documento de teste d_j, $f(.,c_i)$.

2.4 Categorizador VG-RAM WNN

Uma rede neural artificial é um modelo de computação inspirado na forma como a estrutura paralela e densamente conectada do cérebro dos mamíferos processa as informações. Mais formalmente, as redes neurais artificiais são sistemas paralelos distribuídos compostos por unidades de processamento simples, chamados de nós, que calculam determinadas funções matemáticas (normalmente não-lineares) [Haykin99]. Essas unidades são dispostas em uma ou mais camadas e interligadas por um número de conexões, chamadas de sinapses.

Essencialmente, um neurônio artificial é composto por um conjunto de sinapses, um somador e uma função de transferência (ou função de ativação) [Haykin99]. Conforme a Figura 2-3, cada sinapse do neurônio k está associada aos pesos $\{w_{k1}, w_{k2}, \ldots w_{km}\}$. Especificamente, ao ser apresentada uma informação ao neurônio, $\{x_1, x_2, \ldots x_m\}$, cada elemento da informação é multiplicado pelo peso w_{kj} da sinapse, e o resultado de cada entrada é somado, ou seja, é realizada uma soma ponderada da informação de entrada pelo Somador. O resultado da soma passa por uma Função de ativação, $\varphi(.)$, que computa a saída y_k do neurônio em função da saída do Somador.

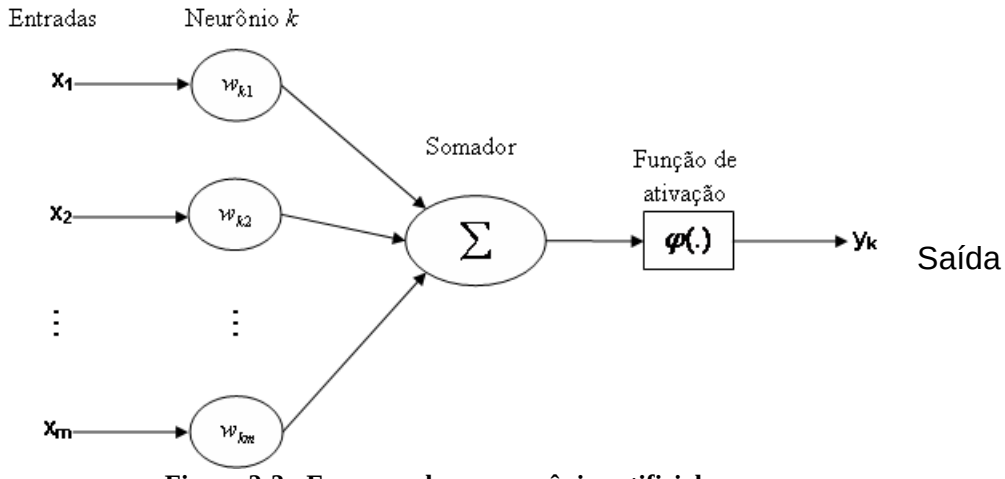


Figura 2-3 - Esquema de um neurônio artificial.

Redes neurais sem peso (*weightless neural networks* - WNN), também conhecidas como redes neurais baseadas em *Random Access Memories* (RAM), não armazenam conhecimento em suas conexões, mas em memórias do tipo RAM dentro dos nodos da rede, ou neurônios. Estes neurônios operam com valores de entrada binários e usam RAM como tabelas-verdade: as sinapses de cada neurônio coletam um vetor de bits da entrada da rede, que é usado como o endereço da RAM, e o valor armazenado neste endereço é a saída do neurônio. O treinamento pode ser feito em um único passo e consiste basicamente em armazenar a saída desejada no endereço associado com o vetor de entrada do neurônio.

Apesar da sua notável simplicidade, as WNN são muito efetivas como ferramentas de reconhecimento de padrões, oferecendo treinamento e teste rápidos, e fácil implementação. No entanto, se a entrada da rede for muito grande, o tamanho da memória dos neurônios da WNN torna-se proibitivo, dado que tem de ser igual a 2^n, onde n é o tamanho da entrada.

As redes *Virtual Generalizing RAM* (*VG-RAM*) são redes neurais baseadas em RAM que somente requerem capacidade de memória para armazenar os dados relacionados ao conjunto de treinamento. Os neurônios *VG-RAM* armazenam os pares entrada-saída observados durante o treinamento, em vez de apenas a saída. Na fase de teste, as memórias dos neurônios *VG-RAM* são pesquisadas mediante a comparação entre a entrada apresentada à rede e todas as entradas nos pares entrada-saída aprendidos. A saída de cada neurônio *VG-RAM* é determinada pela saída do par cuja entrada é a mais próxima da entrada apresentada – a função de distância adotada pelos neurônios *VG-RAM* é a distância de *Hamming*, isto é, o número de bits diferentes entre dois vetores de bits de tamanho igual. Se existir mais do que um par na mesma distância mínima da entrada apresentada, a saída do neurônio é escolhida aleatoriamente entre esses pares.

2.4.1 VG-RAM WNN

A Tabela 2-1 ilustra a tabela-verdade de um neurônio *VG-RAM* com três sinapses (X_1, X_2 e X_3). Esta tabela-verdade contém três pares entrada-saída que foram armazenados durante a fase de treinamento ($par\ \#1$, $par\ \#2$ e $par\ \#3$). Durante a fase de teste, quando um vetor de entrada é apresentado à rede, o algoritmo de teste *VG-RAM* calcula a distância entre este vetor de entrada e cada entrada dos pares entrada-saída armazenados na tabela-verdade. No exemplo da Tabela 2-1, a distância de *Hamming* entre o vetor de entrada (*input*)

e o *par* #1 é dois, porque ambos os bits X_2 e X_3 não são semelhantes aos bits X_2 e X_3 do vetor de entrada. A distância do *par* #2 é um, porque X_1 é o único bit diferente. A distância do *par* #3 é três. Portanto, para este vetor de entrada, o algoritmo avalia a saída do neurônio, Y, como "*categoria* 2", pois é o valor de saída armazenado no *par* #2.

Tabela 2-1 - Exemplo de tabela-verdade de um neurônio da *VG-RAM WNN* [SCAE08].

Tabela-verdade	X_1	X_2	X_3	Y
par #1	1	1	0	*categoria* 1
par #2	0	0	1	*categoria* 2
par #3	0	1	0	*categoria* 3
	↑	↑	↑	↓
vetor de entrada	1	0	1	*categoria* 2

Para categorizar documentos de texto usando uma *VG-RAM WNN*, um documento é representado por um vetor multidimensional $V = \{v_1, v_2, \ldots, v_{|V|}\}$, onde cada elemento v_i corresponde a um peso associado a um termo específico do vocabulário de interesse. Uma *VG-RAM WNN* de uma única camada (Figura 2-4) é utilizada, de forma que as sinapses $X = \{x_1, x_2, \ldots, x_{|X|}\}$ de seus neurônios são conectadas aleatoriamente à entrada da rede $N = \{n_1, n_2, \ldots, n_{|N|}\}$, que tem o mesmo tamanho de um vetor que representa um documento, isto é, $|N| = |V|$. Note que $|X| < |V|$ (nossos experimentos demonstraram que $|X| < |V|$ provê melhor desempenho). Cada sinapse x_i de um neurônio forma uma célula *Minchinton* com a próxima x_{i+1} ($x_{|X|}$ forma uma célula *Minchinton* com x_1) [Mitchell98]. O tipo de célula *Minchinton* usada retorna 1 se a sinapse x_i da célula é conectada a um elemento de entrada n_j cujo valor é maior do que aquele do elemento n_k ao qual a sinapse x_{i+1} é conectada (isto é, $n_j > n_k$); caso contrário, ela retorna zero.

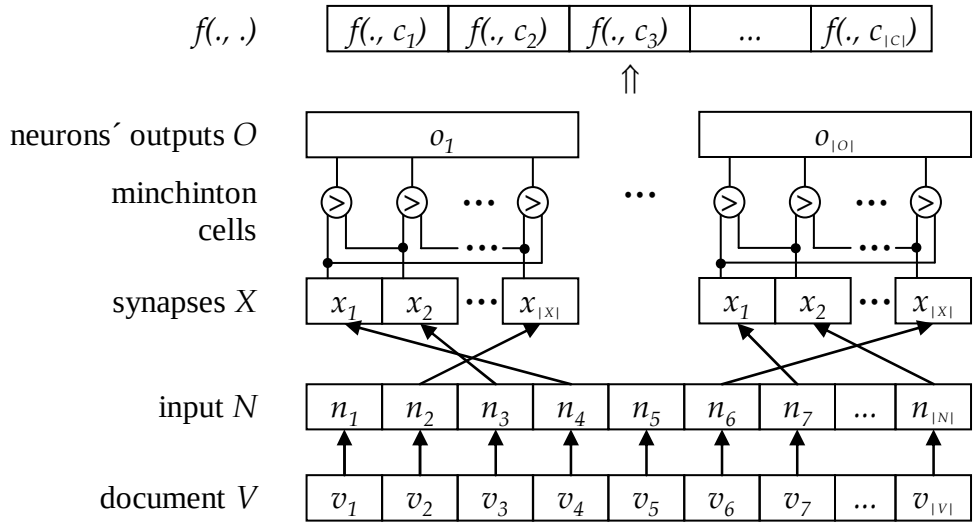


Figura 2-4 – Arquitetura para categorização de texto da *VG-RAM WNN* [SCAE08].

Durante a fase de treinamento, para cada documento no conjunto de treinamento, o vetor correspondente V é conectado à entrada N da *VG-RAM WNN* e as saídas $O = \{o_1, o_2, \ldots, o_{|O|}\}$ dos neurônios a uma das categorias do documento. Todos os neurônios da *VG-RAM WNN* são então treinados para retornar como saída esta categoria com este vetor de entrada. O treinamento para este vetor de entrada é repetido para cada categoria associada ao documento correspondente. Durante a fase de teste, para cada documento de teste, a entrada é conectada ao vetor correspondente e o número de neurônios retornado para cada categoria é contabilizado. A saída da rede é computada dividindo-se a contagem de cada categoria pelo número de neurônios da rede.

A saída da rede é reorganizada como um vetor cujo tamanho é igual ao número de categorias existentes. O valor de cada elemento deste vetor varia entre 0 e 1 e representa a porcentagem de neurônios que exibiram a categoria correspondente como saída (a soma dos valores de todos os elementos deste vetor é sempre 1). Desta forma, a saída da rede reorganizada deste modo implementa a função $f(.,.)$, que apresenta valores no domínio dos números reais e que mapeia a múltipla pertinência de um documento frente a um dado conjunto de categorias existentes. Finalmente, um valor limiar τ_i para cada categoria c_i pode ser usado com a função $f(.,.)$, a fim de definir o conjunto de categorias a serem atribuídas a um documento de teste d_j: se $f(d_j, c_i) \geq \tau_i$, então c_i é atribuída a d_j.

2.4.2 VG-RAM WNN-COR

Enquanto numa *VG-RAM WNN* cada neurônio é treinado para retornar como saída uma única categoria para cada vetor de entrada, numa *VG-RAM WNN* com Correlação de Dados (*VG-RAM WNN-COR* [DeSouza08, DeSouza09b]) cada neurônio pode ser treinado para retornar como saída um conjunto de categorias para cada vetor de entrada. A Tabela 2-2 ilustra a tabela-verdade de uma *VG-RAM WNN-COR* com três sinapses X_1, X_2 e X_3 e três pares entrada-saída armazenados durante a fase de treinamento (par #1, par #2 e par #3). Semelhante à *VG-RAM WNN*, quando um vetor de entrada é apresentado à rede na fase de teste, o algoritmo de teste da *VG-RAM WNN-COR* computa a distância entre este vetor de entrada e cada entrada dos pares entrada-saída na tabela-verdade. No exemplo da Tabela 2-2, a distância de *Hamming* entre o vetor de entrada (*input*) e os pares #1, #2, e #3 é dois, um e três, respectivamente. Como o par #2 da tabela-verdade é o mais próximo da entrada da rede, a saída do neurônio da *VG-RAM WNN-COR* é dada pelas categorias 1 e 3, isto é, o valor de Y representa ambas as categorias, 1 e 3.

Tabela 2-2 - Exemplo de tabela-verdade de uma rede neural VG-RAM WNN-COR [SCAE08].

Tabela-verdade	X_1	X_2	X_3	Y
par #1	1	1	0	$categoria$ 1
par #2	0	0	1	$categoria$ 1; $categoria$ 3
par #3	0	1	0	$categoria$ 1; $categoria$ 2; $categoria$ 3
	↑	↑	↑	↓
vetor de entrada	1	0	1	$categoria$ 1; $categoria$ 3

Para categorizar documentos de texto usando uma *VG-RAM WNN-COR*, a mesma configuração da *VG-RAM WNN*, ilustrada na Tabela 2-2, é usada. Na fase de treinamento, para cada documento no conjunto de treinamento, o vetor correspondente V é conectado à entrada da *VG-RAM WNN-COR*, N, e as saídas dos seus neurônios, O, ao conjunto de categorias atribuído ao documento. Cada neurônio da *VG-RAM WNN-COR* é treinado para retornar como saída este conjunto com este vetor de entrada. Durante a fase de teste, para cada documento de teste, o vetor correspondente V é conectado à entrada da rede, N. A função $f(.,.)$ é computada ao dividir o número de votos para cada categoria pelo número total de categorias retornadas pela rede. O número de votos para cada categoria é obtido ao contar suas ocorrências em todos os conjuntos retornados pelos neurônios da rede.

A implementação de rede neural *VG-RAM WNN-COR* explora a correlação das categorias associadas a cada documento para melhorar o desempenho da categorização multi-rótulo de texto. Resultados experimentais apresentados em [DeSouza08, DeSouza09b] mostram que a implementação *VG-RAM WNN-COR* tem um desempenho global superior ao da *VG-RAM WNN* em termos das métricas mais relevantes de avaliação de desempenho de categorização multi-rótulo de texto empregadas pela comunidade de RI. Por esta razão, neste trabalho, utilizamos a implementação *VG-RAM WNN-COR* [DeSouza08, DeSouza09b] para avaliar a estratégia de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto.

2.5 Aplicação de Categorização Multi-Rótulo de Texto

Devido ao aumento da disponibilidade do número de documentos de texto no formato digital, e pela conseqüente necessidade de organizá-los, a categorização de texto tornou-se uma das técnicas chave para manipular e organizar dados no formato texto [Sebastiani02]. Hoje em dia, a categorização de texto pode ser aplicada em diversos problemas, tais como: organização de documentos, filtragem de texto, geração automatizada de metadados, desambiguação do sentido da palavra, categorização de páginas Web baseados em um catálogo hierárquico, entre outras. No entanto, existem muitas outras importantes aplicações às quais pouca atenção tem sido dada. Um exemplo é a categorização de atividades econômicas baseada na descrição das atividades econômicas realizadas por uma empresa [Badue08, Ciarelli08, Ciarelli09, DeSouza07, DeSouza08, DeSouza09a, DeSouza09b, Oliveira08a, Oliveira08b]. Neste trabalho, verificamos o impacto da nossa estratégia de poda para o *ranking* de categorias, baseada na medida de certeza de categorização multi-rótulo de texto, utilizando bases de texto contendo descrições de atividades econômicas de empresas brasileiras.

2.5.1 Categorização de Atividades Econômicas

A categorização de companhias de acordo com as respectivas atividades exercidas é uma etapa importante do processo de obtenção de informação para a realização de análises estatísticas das atividades econômicas de uma cidade ou país. Com as companhias

categorizadas, é possível realizar uma análise estruturada de cada setor da economia, auxiliando empresas e governos em suas decisões.

Para facilitar e melhorar a qualidade de categorização das empresas de acordo com as atividades econômicas, o governo brasileiro está criando uma biblioteca digital centralizada com as declarações de propósitos de todas as empresas no país. Esta biblioteca vai ajudar as três esferas de governo – federal, os 27 Estados, e os mais de 5.000 municípios brasileiros – na tarefa de categorizar as empresas de acordo com a lei Brasileira vigente.

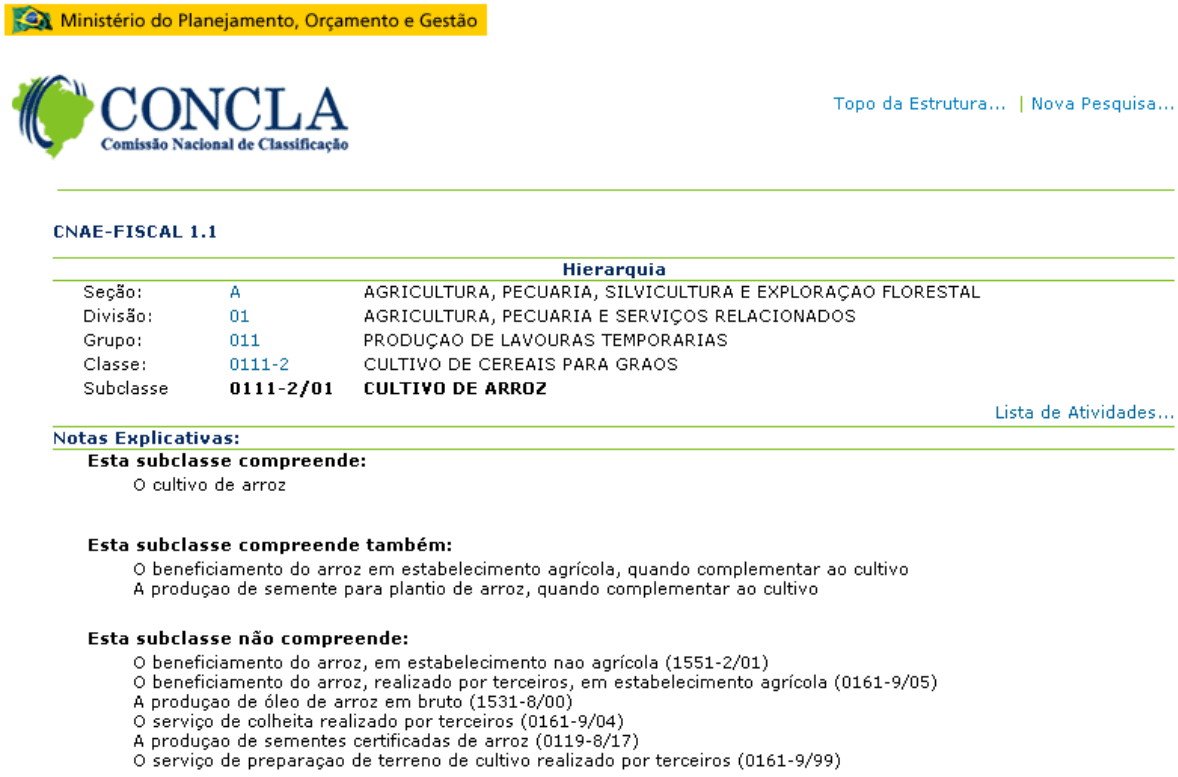
A categorização oficial das atividades econômicas adotada pelos órgãos da administração federal é baseada na Classificação Nacional de Atividades Econômicas (CNAE). A CNAE foi desenvolvida tendo como referência a *International Standard Industrial Classification of All Economic Activities - ISIC*, 3ª revisão, das Nações Unidas. A *ISIC* é uma padronização internacional definida pelas Nações Unidas para a disseminação das estatísticas econômicas no mundo. A partir da elaboração da CNAE foi derivada outra classificação, a CNAE-FISCAL, ou CNAE-Subclasses [CNAE03], que é um detalhamento das Classes da CNAE para uso nos cadastros da administração pública, em especial da administração tributária, nas três esferas do governo. A Tabela 2-3 apresenta sumariamente a CNAE-Subclasses Versão 1.1.

Tabela 2-3 – Apresentação sumária da Tabela CNAE-Subclasses, Versão 1.1.

Seções	Divisões	Grupos	Classes	Subclasses	Denominação
A	2	7	25	91	Agricultura, pecuária, silvicultura e exploração florestal
B	1	1	2	11	Pesca
C	4	7	14	42	Indústrias extrativas
D	23	104	286	395	Indústrias de transformação
E	2	4	7	8	Produção e distribuição de eletricidade, gás e água
F	1	6	16	43	Construção
G	3	19	72	223	Comércio; Reparação de veículos automotores, objetos pessoais e domésticos
H	1	2	7	16	Alojamento e alimentação
I	5	14	29	76	Transporte, armazenagem e comunicações
J	3	11	27	65	Intermediação financeira, seguros, previdência complementar e serviços relacionados
K	5	24	38	80	Atividades imobiliárias, aluguéis e serviços prestados às empresas
L	1	3	10	10	Administração pública, defesa e seguridade social
M	1	4	10	17	Educação
N	1	3	9	35	Saúde e serviços sociais
O	4	11	26	69	Outros serviços coletivos, sociais e pessoais
P	1	1	1	1	Serviços domésticos
Q	1	1	1	1	Organismos internacionais e outras instituições extraterritoriais
Total	**59**	**222**	**580**	**1.183**	

A CNAE-Subclasses é uma tabela hierárquica de descrição de atividades econômicas com os respectivos códigos associados. Conforme a Tabela 2-3 mostra, a CNAE-Subclasses 1.1 está organizada hierarquicamente em 5 níveis: Seção, Divisão, Grupo, Classe e Subclasse, contendo 17 Seções, 59 Divisões, 222 Grupos, 580 Classes e 1.183 Subclasses. O campo Denominação representa a descrição textual do código de Seção. Cada código nos níveis Divisão, Grupo, Classe e Subclasse também está associado a uma denominação [CNAE03].

Os códigos da CNAE-Subclasses são constituídos por 7 dígitos, sendo os 5 primeiros dígitos referentes ao nível de Classe e os dois últimos referentes ao detalhamento de cada Classe CNAE. Por exemplo, a Figura 2-5 apresenta o nível de Subclasse da Seção A para o código 0111-2/01 com a denominação "CULTIVO DE ARROZ". Como podemos perceber, a Classe é identificada pelo código 0111-2 e pela denominação "CULTIVO DE CEREAIS PARA GRAOS".


Ministério do Planejamento, Orçamento e Gestão

CONCLA
Comissão Nacional de Classificação

Topo da Estrutura... | Nova Pesquisa...

CNAE-FISCAL 1.1

Hierarquia		
Seção:	A	AGRICULTURA, PECUARIA, SILVICULTURA E EXPLORAÇAO FLORESTAL
Divisão:	01	AGRICULTURA, PECUARIA E SERVIÇOS RELACIONADOS
Grupo:	011	PRODUÇAO DE LAVOURAS TEMPORARIAS
Classe:	0111-2	CULTIVO DE CEREAIS PARA GRAOS
Subclasse	**0111-2/01**	**CULTIVO DE ARROZ**

Lista de Atividades...

Notas Explicativas:

Esta subclasse compreende:
O cultivo de arroz

Esta subclasse compreende também:
O beneficiamento do arroz em estabelecimento agrícola, quando complementar ao cultivo
A produçao de semente para plantio de arroz, quando complementar ao cultivo

Esta subclasse não compreende:
O beneficiamento do arroz, em estabelecimento nao agrícola (1551-2/01)
O beneficiamento do arroz, realizado por terceiros, em estabelecimento agrícola (0161-9/05)
A produçao de óleo de arroz em bruto (1531-8/00)
O serviço de colheita realizado por terceiros (0161-9/04)
A produçao de sementes certificadas de arroz (0119-8/17)
O serviço de preparaçao de terreno de cultivo realizado por terceiros (0161-9/99)

Figura 2-5 – Um exemplo da tabela CNAE para o nível de Subclasse.

Os códigos Subclasse também carregam a identificação dos níveis de Divisão e Grupo. Por exemplo, para o código 0111-2/01 (Figura 2-5), os dois primeiros dígitos, 01, identificam o nível de Divisão, com a denominação "AGRICULTURA, PECUARIA E

SERVIÇOS RELACIONADOS", e os três primeiros, 011, o de Grupo, com a denominação "PRODUÇAO DE LAVOURAS TEMPORARIAS".

Além da denominação do código de um determinado nível, existem notas explicativas para agregar mais informação àquele nível. No caso do nível de Subclasse, as notas explicativas mostram o que a Subclasse compreende ("Esta subclasse compreende:"), o que ela compreende também ("Esta subclasse compreende também:"), e o que ela não compreende ("Esta subclasse não compreende:").

Atualmente, em muitos órgãos usuários, a determinação de quais códigos devem ser atribuídos a cada empresa - a codificação em CNAE-Subclasses - é feita manualmente por codificadores humanos treinados para tal e apoiados por ferramentas computacionais de busca em versões eletrônicas da tabela CNAE-Subclasses. O codificador (ou categorizador) humano treinado deve associar/combinar a descrição da atividade da empresa com a informação na tabela CNAE-Subclasses e com seu conhecimento, fruto de seus vários anos de educação e experiência profissional, para atribuir códigos CNAE-Subclasse.

Conforme as características apresentadas anteriormente, o problema de categorização de atividades econômicas consiste em dada uma descrição textual do propósito de uma empresa, categorizá-la em um ou mais dos 1.183 possíveis códigos (ou categorias) CNAE-Subclasse. O grande número de possíveis categorias torna este problema complexo quando comparado com outros apresentados na literatura [Sebastiani02]. Neste contexto, é de grande interesse uma ferramenta para produzir uma medida de certeza para as categorias preditas para um dado documento de entrada. Dessa forma, o sistema de categorização pode ser acionado por um operador humano, no caso de ser obtida uma medida de certeza abaixo de um determinado limiar pré-definido. Por essa razão, escolhemos este problema de categorização para este trabalho.

3 ESTRATÉGIAS DE PODA DE *RANKING* DE CATEGORIAS

Neste capítulo, apresentamos três métodos de poda de *ranking* de categorias comumente usados na literatura de RI [Yang01, Lee02, Fan07]: (i) RCut, baseada na posição das categorias no *ranking*; (ii) PCut, baseada na popularidade das categorias no conjunto de treinamento; (iii) SCut, baseada no grau de crença com que o sistema atribui as categorias aos documentos; e (iv) uma variante de RCut - RTCut [Yang01]. Além disso, propomos novas variantes para PCut e SCut – PCut* e SCut*, respectivamente – para tratar problemas existentes nestas abordagens.

3.1 Estratégia RCut

A estratégia de poda RCut [Yang01], baseada na posição das categorias no *ranking* (*ranking based*), para cada documento de teste d_j, ordena as categorias por grau de crença e atribui as t categorias a partir do topo do *ranking* para d_j. O valor do parâmetro t (um número inteiro entre 1 e $|C|$) pode ser especificado pelo usuário ou automaticamente ajustado para otimizar o desempenho "global" do categorizador em um conjunto de calibração (o valor de t é automaticamente ajustado ao variá-lo até que o desempenho global do categorizador seja otimizado para o conjunto de validação). RCut com t = 1 é comumente usada pela comunidade de aprendizado de máquina em problemas uni-rótulo, nos quais um documento tem uma única categoria [Joachims98].

A Tabela 3-1 apresenta um exemplo de poda de *ranking* de categorias utilizando a estratégia RCut. Na Tabela 3-1, a coluna "Posição" indica a posição das categorias no *ranking* retornado pelo categorizador *VG-RAM WNN-COR* para um dado documento de teste d_i; a coluna "Categorias preditas a d_i" mostra os identificadores das categorias no *ranking* retornado pelo categorizador para d_i; a coluna "Graus de crença para d_i" mostra os graus de crença com que o categorizador atribuiu as categorias a d_i; e a coluna "Categorias pertinentes a d_i" mostra a lista de categorias de fato pertinentes a d_i. As colunas 5, 6, e 7 são análogas às colunas 2, 3, e 4, porém, estão associadas ao documento de teste d_j. Finalmente, a coluna

"Parâmetro *t*" mostra o valor do parâmetro *t* otimizado para um conjunto de validação. No exemplo da Tabela 3-1, as *t=2* categorias a partir do topo do *ranking* são atribuídas aos documentos de teste.

Tabela 3-1 – Exemplo de poda de *ranking* de categorias utilizando RCut

Posição	Categorias preditas para *di*	Grau de crença *di*	Categorias pertinentes *di*	Categorias preditas para *dj*	Grau de crença *dj*	Categorias pertinentes *dj*	Parâmetro *t*
1	C9	0,109859	C3	C58	0,333576	C1	
2	C8	0,100939	C7	C89	0,307021	C58	
3	C7	0,089202	C9	C59	0,302292		
4	C14	0,069484	C54	C78	0,003638		
5	C10	0,053991		C90	0,003638		
6	C54	0,045541		C20	0,003274		$t = 2$
7	C11	0,039906		C76	0,002183		
8	C67	0,039906		C56	0,001819		
9	C13	0,025822		C63	0,001819		
10	C12	0,022066		C18	0,001455		

Como pode ser observado na Tabela 3-1, a estratégia RCut não é adequada para problemas nos quais os diferentes documentos têm diferentes números de categorias pertinentes, uma vez que RCut atribui o mesmo número de categorias a todos os documentos [Yang01].

3.2 Estratégia RTCut

RCut impõe um compromisso duro entre as métricas de avaliação de desempenho revocação (*recall*) e precisão (*precision*), ao retornar ou um grande número de categorias ou pequeno número de categorias. Para atenuar esse compromisso, Yang (2001) propôs a estratégia de poda RTCut, que atribui graus de crença sintéticos às categorias computados a partir de suas posições no *ranking* e de seus graus de crença para um dado documento. Estes graus de crença sintéticos são computados por:

$$s(d_j, c_i) = r(d_j, c_i) + \frac{f(d_j, c_i)}{\max_{c' \in C} \{f(c' | d_j)\} + 1} \qquad (3\text{-}1)$$

onde d_j é o documento de teste, $r(d_j, c_i)$ é a posição da categoria c_i no *ranking* retornado para d_j, $f(d_j, c_i)$ é o grau de crença original de que a categoria c_i deve ser atribuída a d_j, e $\max_{c' \in C}\{f(c' \mid d_j)\}$ retorna o grau de crença de valor máximo atribuído a uma determinada categoria $c' \in C$ no *ranking* retornado para d_j. Para cada documento de teste, a estratégia RTCut ordena as categorias pelo grau de crença sintético e atribui ao documento as categorias com grau de crença sintético superior a um determinado limiar τ. O valor do parâmetro τ pode ser especificado pelo usuário ou automaticamente ajustado para otimizar o desempenho "global" do categorizador em um conjunto de calibração.

A Tabela 3-2 apresenta um exemplo de poda de *ranking* de categorias utilizando a estratégia RTCut. Na Tabela 3-2, a coluna "Posição" indica a posição das categorias no *ranking* retornado pelo categorizador *VG-RAM WNN-COR* para um dado documento de teste d_i; a coluna "Categorias preditas a d_i" mostra os identificadores das categorias no *ranking* retornado pelo categorizador para d_i; a coluna "Graus de crença originais" mostra os graus de crença com que o categorizador atribuiu as categorias a d_i; a coluna "Categorias preditas a d_i por RTCut" mostra os identificadores das categorias no *ranking* retornado pela estratégia RTCut; a coluna "Graus de crença sintéticos" mostra os graus de crença sintéticos atribuídos às categorias pela estratégia RTCut; a coluna "Categorias pertinentes a d_i" mostra a lista de categorias de fato pertinentes a d_i; e, finalmente, a coluna "Parâmetro τ" mostra o valor do parâmetro τ otimizado para um conjunto de validação. No exemplo da Tabela 3-2, as categorias com grau de crença sintético superior ao limiar $\tau = 2{,}4$ são atribuídas aos documentos de teste.

Tabela 3-2 – Exemplo de poda de *ranking* de categorias utilizando a estratégia RTCut.

	***Ranking* Original**		***Ranking* Sintético**			
Posição	**Categorias preditas para *di***	**Grau de crença *di***	**Categorias preditas para *di***	**Grau de crença *di***	**Categorias pertinentes**	**Parâmetro τ**
1	C9	0,109859	C9	1,09898465	C3	
2	C8	0,100939	C8	2,09094759	C7	
3	C7	0,089202	C7	3,08037237	C9	
4	C14	0,069484	C14	4,06260615	C54	
5	C10	0,053991	C10	5,04864672		
6	C54	0,045541	C54	6,04103314		τ= 2,4
7	C11	0,039906	C11	7,03595592		
8	C67	0,039906	C67	8,03595592		
9	C13	0,025822	C13	9,02326602		
10	C12	0,022066	C12	10,0198818		

RTCut preserva a ordem das categorias no *ranking*, mas permite a distinção entre categorias com a mesma posição nos *rankings* atribuídos a diferentes documentos. Ao podar o *ranking* no grau de crença sintético, ao invés da posição da categoria no *ranking*, compromissos entre revocação e precisão de granularidade mais fina podem ser alcançados [Yang01].

3.3 Estratégia PCut

A estratégia de poda PCut [Yang01], baseada na popularidade das categorias no conjunto de treinamento (*proportion-based*), (i) recebe como entrada as categorizações do conjunto de teste que consiste, para cada $\langle d_j, c_i \rangle \in Te \times C$, dos graus de crença $f(d_j, c_i)$ de que o documento de teste d_j deve ser categorizado dentro da categoria c_i. Dada uma categoria c_i, PCut (ii) ordena os documentos de teste por grau de crença $f(., c_i)$ (gerando um *ranking* de documentos para c_i) e (iii) categoriza os $k_i = p(c_i) * x * |C|$ documentos do topo do *ranking* dentro de c_i, onde $p(c_i)$ é a probabilidade *a priori* (estimada utilizando um conjunto de treinamento) de um documento arbitrário pertencer a c_i e x é um parâmetro cujo valor (um número real) pode ser especificado pelo usuário ou automaticamente ajustado para otimizar o desempenho "global" do categorizador em um conjunto de calibração. PCut foi usada em várias avaliações publicadas de categorizadores probabilísticos, tais como *Naive Bayes*,

decision tree (DTree), kNN, e métodos de regressão *linear least squares fit* (LLSF) [Lewis92, Lewis94, Yang99].

A Tabela 3-3 apresenta um exemplo de poda de *ranking* de categorias utilizando a estratégia PCut. Na Tabela 3-3, a coluna "Documentos preditos para c_1" mostra os documentos de teste d_j categorizados dentro da categoria c_1 e seus respectivos graus de crença $f(d_j, c_1)$; a coluna "Limiar k_1" mostra o valor do limiar de poda k_1 (para a categoria c_1) otimizado para um conjunto de validação; a coluna "Documentos preditos para c_2" mostra os documentos de teste d_j categorizados dentro da categoria c_2 e seus respectivos graus de crença $f(d_j, c_2)$; a coluna "Limiar k_2" mostra o valor do limiar de poda k_2 (para a categoria c_2) otimizado para um conjunto de validação; e assim por diante. No exemplo da Tabela 3-3, os k_1 = 5 documentos do topo do *ranking* da categoria c_1 são categorizados dentro de c_1, os k_2 = 7 documentos do topo do *ranking* de c_2 são categorizados dentro de c_2, e assim por diante.

Tabela 3-3 – Exemplo de poda de *ranking* de categorias utilizando a estratégia PCut

Documentos preditos *c1*		**Limiar *k1***	**Documentos preditos *c2***		**Limiar *k2***	**...**	**Documentos preditos *cm***		**Limiar *km***
d3	0,22089		d9	0,29516			d2	0,29516	
d2	0,09801		d6	0,18523			d7	0,13241	
d4	0,04759		d3	0,15627			d4	0,13180	
d7	0,02379		d10	0,14825			d1	0,10809	
d9	0,01927		d8	0,13180			d8	0,029912	
d1	0,016781	*k1* = 5	d4	0,08187	*k2* = 7		d10	0,023793	*km* = 3
d5	0,014276		d2	0,05495			d9	0,022434	
d10	0,012237		d7	0,024176			d5	0,018611	
d8	0,011557		d5	0,019231			d6	0,014316	
d6	0,008792		d10	0,018681			d3	0,012169	

A estratégia de poda PCut é a única que utiliza a distribuição de categorias observada no conjunto de treinamento para ganhar um controle global das atribuições das categorias aos documentos no conjunto de teste [Yang01]. Isto dá a PCut poder adicional, mas sacrifica a habilidade de categorização *online*, porque os graus de crença dos documentos de teste devem ser acumulados antes de PCut ser aplicada. Diferente do PCut, as categorizações realizadas por RCut para cada documento de teste são independentes entre si; isto torna RCut mais adequado para categorização *online*. Além disso, PCut assume que a distribuição das

categorias através dos documentos permanece constante – esta é uma boa suposição apenas para certos domínios de documentos.

3.4 Estratégia SCut

A estratégia de poda SCut [Yang01], baseada no grau de crença com que o sistema atribui as categorias aos documentos (*score-based*), ajusta automaticamente um limiar de poda τ_i para cada categoria c_i, de forma a otimizar o desempenho do categorizador para c_i em um conjunto de calibração. Diferente de RCut e PCut, nas quais um único parâmetro (*t* ou *x*) é usado para otimizar o desempenho "global" do categorizador na média, SCut otimiza o desempenho "local" do categorizador para categorias individuais sem garantir um ótimo global. SCut foi usada em avaliações de muitos categorizadores, incluindo Ripper, *first order inductive learner* (FOIL), Winnow, *exponentiated gradient* (EG), kNN, *LLSF* e Rocchio [Cohen96, Lewis96, Yang99, Yang01].

Os limiares de poda em SCut são ajustados para otimizar o desempenho do sistema para o conjunto de calibração, enquanto os limiares de poda em PCut dependem apenas das probabilidades das categorias no conjunto de treinamento [Yang01]. Por isso, SCut é mais suscentível a *overfit*, fenômeno pelo qual um categorizador é ajustado também às características contingentes do conjunto treinamento ao invés de apenas as características constitutivas das categorias [Sebastiani02].

A otimização por categoria em SCut torna essa estratégia particularmente efetiva quando o desempenho do sistema em categorias raras é a função alvo a ser otimizada. RCut e PCut, por outro lado, com apenas um único parâmetro (*t* ou *x*) a ser ajustado, serão em geral menos efetivas para otimizar o desempenho do sistema em categorias raras.

Diferente de PCut (e similar a RCut), as categorizações realizadas por SCut para cada documento de teste são independentes entre si: uma vez que os limiares por categoria são otimizados (*offline*) para o conjunto de validação, as categorizações para cada documento de teste são independentes entre si. Isto torna SCut e Rcut mais adequados para categorização *online* do que PCut.

3.5 Novas Variantes para as Estratégias PCut e SCut

Nesta seção, propomos modificações para as estratégias de poda PCut e SCut - PCut* e SCut*, respectivamente - para resolver dois problemas que não foram ainda tratados pela literatura: (i) a inadequabilidade de PCut para sistemas de categorização de texto *online*; (ii) a observação, durante o processo de calibração dos limiares de poda de SCut, de intervalos de valores de limiares que produzem desempenho de categorização constante para algumas categorias.

3.5.1 Estratégia PCut*

Como visto na Seção 3.3, a estratégia PCut não é adequada para sistemas de categorização de texto *online*. Entretanto, de acordo com resultados experimentais publicados por Yang (2001), PCut lida melhor com categorias raras do que RCut e SCut e exibe um compromisso mais suave entre revocação e precisão. Dadas estas vantagens, neste trabalho, propomos uma variante da estratégia PCut para sistemas de categorização de texto *online*, a qual denominamos PCut*.

A estratégia de poda PCut* ajusta automaticamente um limiar de poda τ_i para cada categoria c_i, de forma a otimizar o desempenho do categorizador para c_i em um conjunto de calibração. Para isso, PCut* (i) recebe como entrada as categorizações de um conjunto de calibração, *Va*, que consiste, para cada par $\langle d_j, c_i \rangle \in Va \times C$, dos graus de crença $f(d_j, c_i)$ de que o documento de calibração d_j deve ser categorizado dentro da categoria c_i e, dada uma categoria c_i, (ii) ordena os documentos por grau de crença $f(., c_i)$ (gerando um *ranking* de documentos para c_i), e (iii) categoriza os $k_j = p(c_i) * x * |C|$ documentos do topo do *ranking* dentro de c_i, onde $p(c_i)$ é a probabilidade *a priori* de um documento de treinamento pertencer a c_i e x é um parâmetro cujo valor (um número inteiro) é automaticamente ajustado para otimizar o desempenho "global" do categorizador de um subconjunto do conjunto de calibração. O limiar de poda para a categoria c_i, τ_i, recebe o valor do grau de crença $f(d_j, c_i)$ de que o documento de calibração d_j na posição k_j do *ranking* de documentos de c_i deve ser categorizado dentro de c_i. Quando um documento de teste d_j for submetido ao

sistema de categorização, as categorias c_i cujo grau de crença é superior ao limiar de poda τ_i são então preditas para d_j.

A Tabela 3-4 apresenta um exemplo de poda de *ranking* de categorias utilizando a estratégia PCut*. Na Tabela 3-4, a coluna "Documentos preditos para c_1" mostra os documentos de teste d_j categorizados dentro da categoria c_1 e seus respectivos graus de crença $f(d_j, c_1)$; a coluna "Limiar τ_1" mostra o valor do limiar de poda τ_1 (para a categoria c_1) otimizado para um conjunto de calibração; a coluna "Documentos preditos para c_2" mostra os documentos de teste d_j categorizados dentro da categoria c_2 e seus respectivos graus de crença $f(d_j, c_2)$; a coluna "Limiar τ_2" mostra o valor do limiar de poda τ_2 (para a categoria c_2) otimizado para um conjunto de calibração; e assim por diante. No exemplo da Tabela 3-4, os documentos do *ranking* da categoria c_1 com grau de crença superior ao limiar $\tau_1 = 0{,}08$ são categorizados dentro de c_1, os documentos do *ranking* de c_2 com grau de crença superior ao limiar $\tau_2 = 0{,}13$ são categorizados dentro de c_2, e assim por diante.

Tabela 3-4 – Exemplo de poda de *ranking* de categorias utilizando a estratégia PCut*.

Documentos preditos para *c1*		Limiar *τ1*	Documentos preditos para *c2*		Limiar *τ1*	...	Documentos preditos para *cm*		Limiar *τm*
d3	0,220889		d9	0,29516			d2	0,29516	
d2	0,098011		d6	0,185229			d7	0,132414	
d4	0,047585		d3	0,156271			d4	0,131802	
d7	0,023793		d10	0,148252			d1	0,10809	
d9	0,019267		d8	0,131802			d8	0,029912	
d1	0,016781	*τ1* = 0,08	d4	0,081868	*τ2* = 0,13		d10	0,023793	*τm* = τ
d5	0,014276		d2	0,054945			d9	0,022434	
d10	0,012237		d7	0,024176			d5	0,018611	
d8	0,011557		d5	0,019231			d6	0,014316	
d6	0,008792		d10	0,018681			d3	0,012169	

3.5.2 Estratégia SCut*

A estratégia de poda SCut ajusta automaticamente um limiar de poda τ_i para cada categoria c_i, de forma a otimizar o desempenho local do categorizador para c_i em um conjunto de calibração. O valor de τ_i é automaticamente ajustado ao variá-lo até que o desempenho do categorizador para a categoria c_i seja otimizado para o conjunto de validação.

Em nossos experimentos de calibração, observamos que, para algumas categorias c_i, o desempenho do categorizador para c_i permaneceu constante em um dado intervalo de valores (números reais) crescentes de τ_i. Neste caso, testamos duas implementações para a estratégia SCut. Na primeira, SCut tradicional, o limiar de poda para a categoria c_i recebe o valor mínimo do intervalo de valores crescentes de τ_i, para o qual o desempenho do categorizador para c_i permaneceu constante. Na segunda, o limiar de poda para a categoria c_i recebe o valor mínimo daquele intervalo. Dessa forma, os valores dos limiares calibrados por SCut* são maiores do que aqueles calibrados por SCut. Consequentemente, SCut* é mais efetiva do que SCut para otimizar o desempenho do sistema em termos de precisão e menos efetiva em termos de revocação.

4 MEDIDA DE CERTEZA DE CATEGORIZAÇÃO

Neste capítulo, propomos um método para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto e duas estratégias para determinar limiares de poda para o *ranking* de categorias baseada na medida de certeza de categorização.

4.1 Uso da Regra de Bayes para o Cálculo da Medida de Certeza de Categorização

Seja F o conjunto dos intervalos não sobrepostos e adjacentes dos valores retornados pela função $f(d_j, c_i)$, que retorna o grau de crença do categorizador de que o documento de teste d_j deve ser categorizado dentro da categoria c_i (Seção 2.1). A probabilidade de um categorizador automático predizer corretamente uma categoria c_i para o documento de teste d_j, considerando que ele predisse c_i com grau de crença $f(d_j, c_i)$ dentro de um intervalo $y \in F$ e posicionou c_i na posição $k = r(d_j, c_i)$ do *ranking* de categorias, pode ser enunciada como $p(x|y,k)$, onde:

- a variável aleatória x pode assumir dois valores: 1, se o categorizador predisse uma categoria c_i pertinente a d_j; ou 0, se o categorizador predisse uma categoria c_i não pertinente a d_j;
- a variável aleatória y pode assumir valores específicos $y \in F$, que representam os intervalos não sobrepostos e adjacentes dos valores retornados pela função f;
- a variável aleatória k pode assumir $|C|$ valores $\{1,2,..,|C|\}$, que representam a posição de c_i no *ranking* de categorias.

A probabilidade de interesse, $p(x \mid y,k)$, é a probabilidade do categorizador predizer corretamente qualquer categoria c_i, para qualquer documento de teste d_j, dentre os casos em que o categorizador predisse c_i com grau de crença dentro de um intervalo y e posicionou c_i na posição k do *ranking*. Em outras palavras, $p(x \mid y,k)$ é o percentual dos documentos de teste categorizados corretamente dentre todos os documentos que foram categorizados com grau de

crença dentro de um intervalo *y*, ao considerar as categorias posicionadas na posição *k* do *ranking*.

Desconsiderando por enquanto a variável *k*, então temos uma quantidade de interesse desconhecida *x* (predição correta ou incorreta). A informação de que dispomos sobre *x* é observada probabilisticamente através de *p(x)*. Entretanto, podemos agregar mais informação a *p(x)* observando uma quantidade aleatória *F* (conjunto dos intervalos não sobrepostos e adjacentes dos valores retornados pela função $f(d_j, c_i)$) relacionada com *x*. A distribuição amostral *p(y| x)* define esta relação (probabilidade de um intervalo *y* acontecer dado que a predição esta correta para qualquer categoria c_i). A idéia de que após observar *F = y* a quantidade de informação sobre *x* aumenta é intuitiva e o teorema de Bayes é a regra de atualização utilizada para quantificar este aumento de informação.

$$p(x \mid y) = \frac{p(y \mid x)\, p(x)}{p(y)} \tag{4-1}$$

Para um valor fixo de *y*, a função *p(y|x)* fornece a verossimilhança de cada um dos possíveis valores de *x*, enquanto *p(x)* é chamada distribuição a priori de *x*. Estas duas fontes de informação, priori e verossimilhança, são combinadas levando à distribuição a posteriori de *x*, *p(x|y)*.

Pela Regra de *Bayes*, a probabilidade de interesse ($p(x \mid y,k)$) levando em consideração a variável *k* é dada por:

$$p(x \mid y,k) = \frac{p(y \mid x,k)\, p(x \mid k)}{p(y \mid k)} \tag{4-2}$$

Podemos utilizar a regra de *Bayes* para computar $p(x \mid y,k)$. Através de experimentos de validação, podemos obter amostras experimentais que permitem calcular os valores aproximados dos termos da regra de *Bayes* dos quais $p(x \mid y,k)$depende, i.e.:

- $p(y \mid x,k)$ é a probabilidade do categorizador predizer qualquer categoria c_i, para qualquer documento de teste d_j, com grau de crença dentro de um intervalo *y*, dentre os casos em que o categorizador predisse c_i corretamente e posicionou c_i na posição *k* do *ranking*. Em outras palavras, $p(y \mid x,k)$ é o percentual dos documentos de teste categorizados com grau de crença dentro de um intervalo *y* dentre todos os documentos que foram categorizados corretamente, ao considerar as categorias posicionadas na posição *k* do *ranking*;

- $p(x \mid k)$ é a probabilidade do categorizador predizer corretamente qualquer categoria para qualquer documento de teste, dado que as categorias foram posicionadas na posição *k* do *ranking*;
- $p(y \mid k)$ é a probabilidade do categorizador predizer qualquer categoria com grau de crença dentro de um intervalo *y*, dado que as categorias foram posicionadas na posição *k* do *ranking*.

As probabilidades $p(x \mid y,k)$ podem ser usadas para mapear graus de crença $f(d_j, c_i)$ em medidas de certeza de categorização da seguinte forma. Se o categorizador predisse a categoria c_i para o documento d_j com grau de crença $f(d_j, c_i)$ dentro de um intervalo *y*, e posicionou a categoria c_i na posição $r(d_j, c_i)$ do *ranking*, então a medida de certeza para essa predição pode ser expressa por $p(x|y,k)$, onde $y \subset f(d_j, c_i)$ e $k = r(d_j, c_i)$.

4.2 Uso da Medida de Certeza na Poda do *Ranking* de Categorias

Nesta seção, propomos uma estratégia para determinar limiares de poda para o *ranking* de categorias baseada na medida de certeza de categorização, a qual denominamos *bayesian cut* (BCut). Propomos também uma variante para BCut que usa diferentes limiares de poda para diferentes posições do *ranking*, a qual denominamos *position based bayesian cut* (PBCut).

4.2.1 Estratégia BCut

A estratégia de poda *bayes cut* (BCut), baseada na medida de certeza de categorização descrita na Seção 4.1, para cada documento de teste d_j, (i) ordena as categorias por grau de crença $f(d_j, c_i)$; (ii) mapeia o grau de crença $f(d_j, c_i)$ das *p* categorias do topo do *ranking* (as categorias no *ranking* em posições inferiores a *p* são desconsideradas) em uma medida de certeza de categorização $p(x \mid y,k)$ (Seção 4.1); e atribui a d_j as categorias c_i com medida de certeza $p(x \mid y,k)$ superior a um determinado limiar τ. BCut é parametrizada por *p* e τ. O parâmetro *p* denota o número de posições do *ranking* de categorias consideradas para obter amostras experimentais que permitem calcular os valores aproximados dos termos da regra de *Bayes*, $p(y \mid x,k)$, $p(x \mid k)$ e $p(y \mid k)$ (Seção 4.1). Os termos da regra de *Bayes*

diminuem (tendendo a zero) com o aumento do número de posições do *ranking* de categorias consideradas. O valor do parâmetro τ (um número real entre 0 e 1) é automaticamente ajustado para otimizar o desempenho "global" do categorizador em um conjunto de calibração, considerando apenas as p categorias do topo do *ranking* retornado para os documentos.

A Tabela 4-1 apresenta um exemplo de poda de *ranking* de categorias utilizando a estratégia BCut. Na Tabela 4-1, a coluna "Categorias preditas a d_i" mostra os identificadores das categorias no *ranking* retornado pelo categorizador para d_i; a coluna "Graus de crença" mostra os graus de crença com que o categorizador atribuiu as categorias a d_i; a coluna "Categorias preditas a d_i por BCut" mostra os identificadores das categorias no *ranking* retornado pela estratégia BCut; a coluna "Medidas de certeza" mostra as medidas de certeza de categorização atribuídos às categorias pela estratégia BCut; e, finalmente, a coluna "Parâmetro τ" mostra o valor do parâmetro τ otimizado para um conjunto de validação. No exemplo da Tabela 4-1, as categorias com medidas de certeza de categorização superior ao limiar $\tau = 0{,}59$ são atribuídas aos documentos de teste.

Tabela 4-1 – Exemplo de poda de *ranking* de categorias utilizando a estratégia BCut.

***Ranking* Original**		***Ranking* Probabilístico**		
Categorias preditas a *di*	**Graus de Crença *di***	**Categorias preditas a *dj***	***p(x|y)***	**Parâmetro τ**
d58	0,333576	d58	0,970967	
d89	0,307021	d89	0,712908	
d59	0,302292	d59	0,467745	
d78	0,003638	d78	0,000000	
d90	0,003638	d90	0,003205	$\tau = 0{,}59$
d20	0,003274	d20	0,000000	
d76	0,002183	d76	0,000000	
d56	0,001819	d56	0,000000	
d63	0,001819	d63	0,000000	

A estratégia BCut é mais efetiva, do que todas as outras estratégias de poda consideradas, para otimizar o desempenho do sistema em termos de precisão. A razão é que BCut poda o *ranking* de categorias na medida de certeza de categorização, ou seja, na probabilidade da categorização estar correta.

4.2.2 Estratégia PBCut

A estratégia de poda PBCut, uma variante de BCut, aplica diferentes limiares de poda τ_p para diferentes posições p do *ranking*, ou seja, para cada documento de teste d_j, PBCut atribui a d_j a categoria c_i na posição p do *ranking* somente se sua medida de certeza $p(x \mid y,k)$ for superior a um determinado limiar τ_p. PBCut é parametrizada por p (similar ao parâmetro p de BCut descrito na Seção 4.2.1) e τ_p. O valor do parâmetro τ_p (um número real entre 0 e 1) é automaticamente ajustado para otimizar o desempenho "global" do categorizador em um conjunto de calibração, considerando apenas as p categorias do topo do *ranking* como resposta para os documentos. A calibração dos diferentes limiares τ_p é processada através de p iterações. Na primeira iteração, o valor de τ_1 é ajustado ao variá-lo até que o desempenho global do categorizador seja otimizado para o conjunto de validação, considerando apenas a primeira categoria do topo do *ranking* como resposta para os documentos. Na segunda iteração, o valor de τ_1 é fixado no valor ótimo encontrado através da primeira iteração e o valor de τ_2 é então ajustado ao variá-lo até que o desempenho do categorizador seja otimizado, considerando apenas as 2 primeiras categorias do topo do *ranking*, e assim por diante.

A Tabela 4-2 apresenta um exemplo de poda de *ranking* de categorias utilizando a estratégia PBCut. Na Tabela 4-2, a coluna "Categorias preditas a d_i" mostra os identificadores das categorias no *ranking* retornado pelo categorizador para d_i; a coluna "Graus de crença" mostra os graus de crença com que o categorizador atribuiu as categorias a d_i; a coluna "Categorias preditas a d_i por PBCut" mostra os identificadores das categorias no *ranking* retornado pela estratégia PBCut; a coluna "Medidas de certeza" mostra as medidas de certeza de categorização atribuídos às categorias pela estratégia PBCut; e, finalmente, a coluna "Parâmetros τ_p" mostra os valores dos parâmetros τ_p para cada posição p do *ranking* otimizados para um conjunto de validação. No exemplo da Tabela 4-2, são atribuídas ao documento de teste as categorias na posição $p = 1$ do *ranking* com medidas de certeza de categorização superior ao limiar $\tau_1 = 0{,}79$, $p = 2$ com medidas de certeza superior a $\tau_2 = 0{,}58$, e $p = 3$ com medidas de certeza superior a $\tau_3 = 0{,}44$.

Tabela 4-2 – Exemplo de poda de *ranking* de categorias utilizando a estratégia BCut*.

***Ranking* Original**		***Ranking* Probabilístico**		
Categorias preditas a *di*	**Graus de Crença *di***	**Categorias preditas a *dj***	***p(x|y)***	**Parâmetro *τi***
d58	0,333576	d58	0,970967	*τ1=0,79*
d89	0,307021	d89	0,712908	*τ2=0,58*
d59	0,302292	d59	0,467745	*τ3=0,44*
d78	0,003638	d78	0,000000	*τ4=-1,0*
d90	0,003638	d90	0,003205	*τ5=-1,0*
d20	0,003274	d20	0,000000	*τ6=-1,0*
d76	0,002183	d76	0,000000	*τ7=-1,0*
d56	0,001819	d56	0,000000	*τ8=-1,0*
d63	0,001819	d63	0,000000	*τ9=-1,0*

A medida de certeza de categorização em uma categoria diminui à medida que a posição da categoria no *ranking* aumenta. Por esta razão, ao escolher diferentes limiares de poda para diferentes posições do *ranking*, a estratégia de poda PBCut pode produzir um desempenho superior ao de BCut em termos de precisão.

5 METODOLOGIA EXPERIMENTAL

Neste capítulo, descrevemos nossa metodologia experimental. Apresentamos as bases de dados empregadas em nossa avaliação experimental, compostas por descrições textuais de atividades econômicas de empresas brasileiras, e a correção ortográfica e a indexação dessas bases de dados. Apresentamos também a abordagem de validação cruzada empregada na avaliação do desempenho dos categorizadores e o ajuste (calibração) dos parâmetros dos categorizadores. Além disso, apresentamos o cálculo dos parâmetros do método baseado na regra de *Bayes* para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto, o procedimento empregado para validar as medidas de certeza de categorização, e, finalmente, o ajuste (calibração) dos parâmetros das estratégias de poda do *ranking* de categorias. O conteúdo das seções 5.1 a 5.5 foi fundamentalmente extraído de [Melotti09].

5.1 Bases de Dados

O conjunto de dados empregado em nossa avaliação experimental é composto de descrições textuais de atividades econômicas de empresas brasileiras. Todas essas descrições foram manualmente categorizadas em uma ou mais atividades econômicas por funcionários públicos Brasileiros treinados nesta tarefa. A lei brasileira determina que todas as empresas devem apresentar uma descrição textual das suas atividades econômicas para órgãos do governo para que elas sejam categorizadas de acordo com a tabela oficial de atividades econômicas, Tabela CNAE-Subclasse [CNAE03]. Chamamos de documento a descrição textual das atividades econômicas de uma empresa categorizadas em uma ou mais categorias da tabela CNAE-Sublcasses.

Neste trabalho, contamos com descrições de atividades econômicas de empresas das cidades de Vitória – Espírito Santo e Belo Horizonte – Minas Gerais. A base de dados de Vitória, chamada de VIX, possui 3.281 documentos referentes a empresas da localidade categorizados em 764 diferentes categorias CNAE-Subclasse. O número médio de categorias por documento é 4,3 (desvio padrão de 5,6).

A Figura 5-1 apresenta o histograma do número de documentos com um determinado número de categorias. No gráfico da Figura 5-1, o eixo horizontal representa o Número de categorias por documento e o eixo vertical o Número de documentos. De 1 a 35 categorias por documento, as barras do gráfico indicam exatamente o número de documentos com o respectivo número de categorias. De 36 categorias por documento em diante, só aparecem no eixo horizontal do gráfico os números de categorias por documento para os quais há documentos na base VIX.

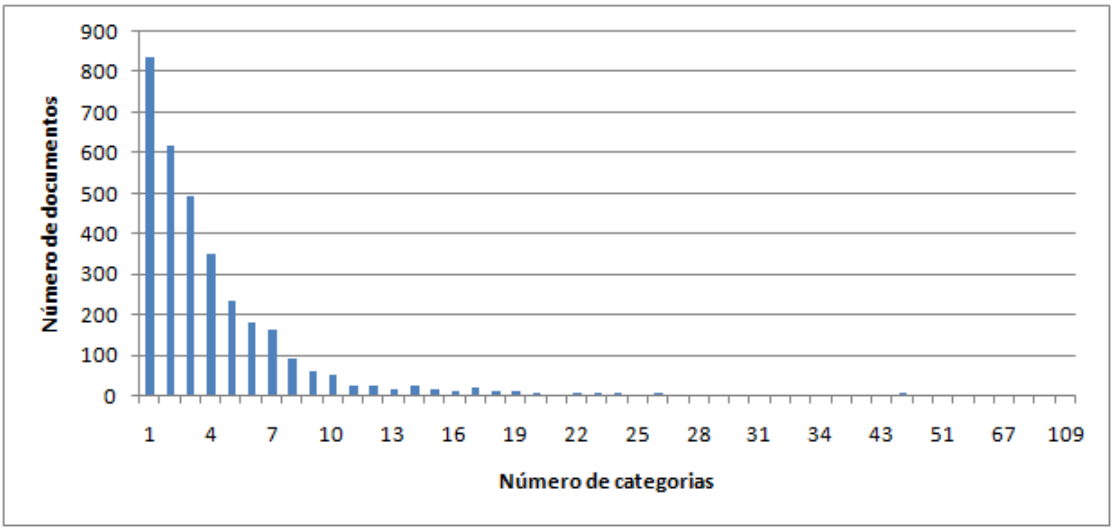


Figura 5-1 – Distribuição do número de categorias por documento na base de dados VIX.

O número de categorias por documento varia de 1 a 109, sendo que mais de 800 documentos possuem apenas uma categoria e apenas um documento possui 109 categorias. Como a Figura 5-1 mostra, a maior parte dos documentos da base VIX possui de 1 a 7 categorias por documento (87,53%).

A base de dados de Belo Horizonte, chamada BH, possui 88.000 documentos categorizados em 1.002 diferentes categorias CNAE-Subclasse. O número médio categorias por documento é 2,0 (desvio padrão de 1,7). A Figura 5-2 apresenta o histograma da base BH.

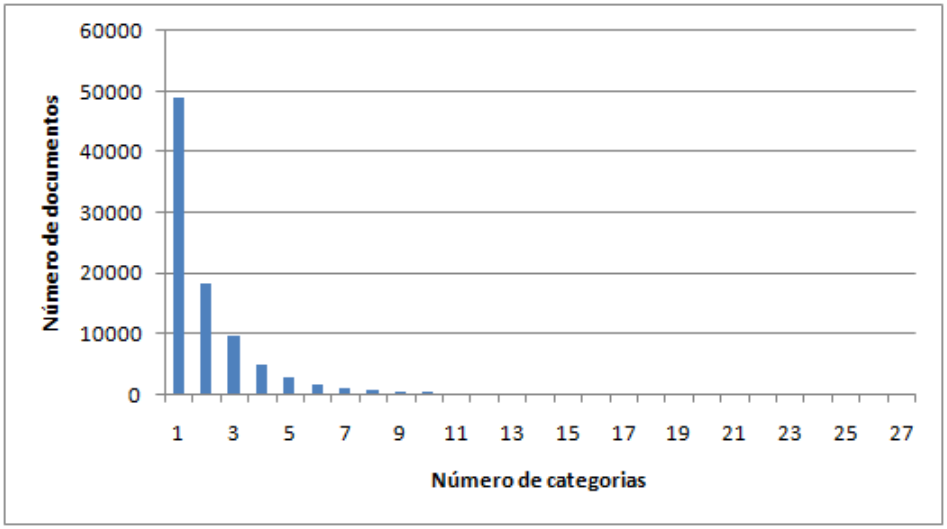


Figura 5-2 – Distribuição do número de categorias por documento na base de dados BH.

Na base BH, o número de categorias por documento varia entre 1 e 27, sendo que quase 50000 documentos possuem apenas uma categoria e apenas um documento possui 27 categorias. Como a Figura 5-2 mostra, a maior parte dos documentos da base BH possui entre 1 e 3 categorias.

A partir das bases VIX e BH, geramos duas bases de dados que utilizamos para treinar, validar, testar e avaliar o impacto dos tipos de *ranking* nos categorizadores. A primeira base gerada, chamada de EX100 (EXatamente 100), possui exatamente 100 exemplares de documentos de cada categoria. Ela é composta de 6.911 documentos selecionados aleatoriamente da união de VIX e BH; 105 categorias diferentes ocorrem na base EX100, isto é, existem exatamente 100 documentos na base categorizados dentro de cada uma destas 105 categorias. O número médio de categorias por documento é 1,52 (desvio padrão de 0,79).

As características da EX100 permitem avaliar o impacto dos *rankings* Ordinal Aleatório, Denso, Padrão e Modificado no desempenho dos categorizadores nos casos onde as categorias estão aproximadamente uniformemente distribuídas na base de treinamento. A Figura 5-3 apresenta o histograma da base EX100. Conforme a figura mostra, o número de categorias por documento varia de 1 a 6, sendo que mais de 4.000 documentos possuem apenas uma categoria e 9 documentos possuem 6 categorias. A maior parte dos documentos desta base possui entre 1 e 2 categorias (89,22%).

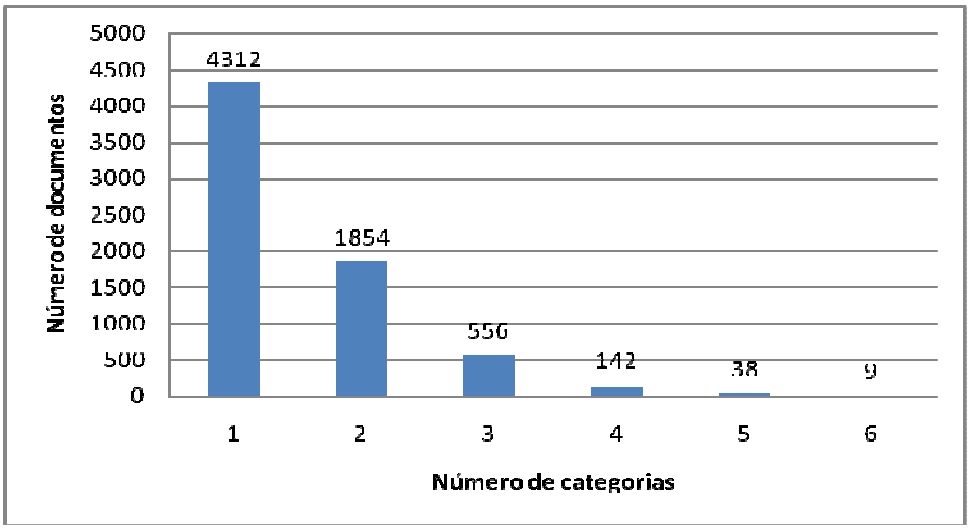


Figura 5-3 – Distribuição do número de categorias por documento na base de dados EX100.

Na segunda base gerada, chamada de AT100 (ATé 100), cada categoria ocorre em até 100 diferentes documentos, isto é, existem entre 1 e 100 exemplares de documentos de cada categoria. Ela é composta de 10.495 documentos selecionados aleatoriamente da união de VIX e BH; 692 categorias diferentes ocorrem na base AT100. O número médio de categorias por documentos é 1,49 (desvio padrão de 0,86). As características de AT100 permitem avaliar o impacto de cada tipo de *ranking* no desempenho dos categorizadores nos casos onde existem categorias raras.

A Figura 5-4 apresenta o histograma da base AT100. Conforme a figura mostra, o número de categorias por documento varia de 1 a 12, sendo que mais de 7.000 documentos possuem apenas uma categoria e um documento possui 12 categorias. A maior parte dos documentos desta base possui entre 1 e 2 categorias.

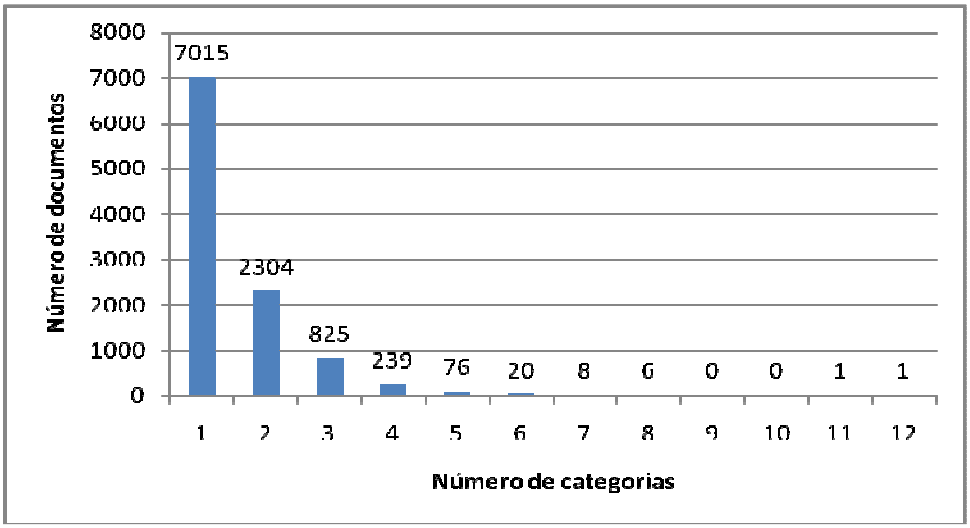


Figura 5-4 – Distribuição do número de categorias por documento na base de dados AT100.

Além das bases EX100 e AT100, utilizamos a própria tabela CNAE-Subclasse, chamada de CNAE, para treinar os categorizadores. A tabela CNAE-Subclasse possui 1183 Subclasses. Cada uma destas Subclasses possui um pequeno texto com sua denominação (ver Seção 2.5.1, pág. 37). Este texto foi utilizado, juntamente com o código CNAE correspondente, como documento de treinamento. Foram utilizados apenas os documentos cujas categorias ocorrem nas bases EX100 ou AT100. Então, temos duas bases CNAE: uma para a base EX100, chamada CNAE_EX100, com 105 documentos (códigos CNAE-Subclasse), e outra para a AT100, chamada CNAE_AT100, com 692 documentos. Estas bases foram usadas porque, no caso de problema de categorização em CNAE, esta informação estará sempre disponível e verificamos que utilizá-la melhora o desempenho dos categorizadores.

5.2 Correção Ortográfica Automática

Antes da geração das bases (EX100 e AT100) para os experimentos de 10-*fold cross-validation*, realizamos o procedimento de correção ortográfica automática das bases VIX, BH e CNAE. Foi adotada a correção automática ao invés da manual em função do grande número de documentos existentes nas bases.

A correção ortográfica está relacionada a dois principais problemas: a detecção de erro, que é o processo de encontrar uma palavra errada; e a correção de erro, que é o processo de sugerir palavras corretas para substituir uma palavra errada encontrada [Martins04]. Atualmente, existem corretores ortográficos para diversos idiomas. Dentre os existentes para o Português escolhemos o *GNU Aspell* [Aspell08] por ter código aberto e, assim, permitir a customização necessária para seu uso no SCAE [SCAE08].

A ferramenta *Aspell* faz uso de um dicionário para propor uma lista de palavras corretas para uma palavra errada. Basicamente, a ferramenta calcula a distância entre a palavra errada e cada uma das palavras existentes no dicionário, sendo que a de menor distância é colocada no topo da lista de sugestões, ou seja, a topo da lista é a considerada correta. O valor da distância é considerado pelo *Aspell* como uma pontuação (*score*).

Em testes preliminares de correção ortográfica automática, percebemos que em muitas situações a palavra correta estava na lista de sugestões do *Aspell*, mas não se encontrava no topo. Visando melhorar o desempenho, utilizamos uma lista auxiliar de palavras com as

respectivas freqüências [Crowell03]. Esta lista foi gerada a partir das palavras existentes nos documentos da base VIX corrigida manualmente.

O novo *score*, que chamamos de *rank*, é calculado a partir do *score* atribuído pelo *Aspell* e a freqüência da palavra (FP) existente na lista auxiliar, conforme Equação (5.1). Então, para que o *Aspell* retorne uma palavra correta dada uma errada, o mesmo escolhe a de menor *rank*.

$$rank = \frac{score}{1 + \ln(FP)} \tag{5.1}$$

Mais detalhes sobre o corretor ortográfico automático empregado em [SCAE08].

5.3 Indexação das Bases de Dados

O procedimento de indexação é realizado após o pré-processamento das bases corrigidas, que envolve [Sebastiani02]: *Análise léxica*; *Remoção de stopwords* (artigos, preposições, etc.); e *Redução de dimensionalidade*. A Figura 5-5 apresenta graficamente o fluxograma do pré-processamento, que está também definido/implementado na ferramenta SCAE.

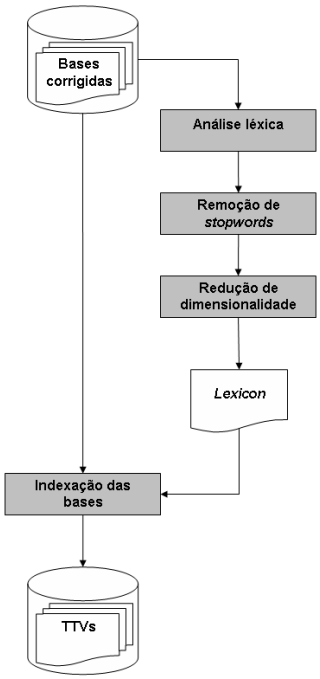


Figura 5-5 – Fluxograma do pré-processamento realizado nas Bases corrigidas anterior à indexação.

Na *Análise léxica*, os textos dos documentos são convertidos em um conjunto de palavras, que são candidatas a serem adotadas como termos dos documentos. Para isso, as palavras do texto dos documentos são separadas pelos caracteres de espaço e pontuação, ou seja, esses caracteres são delimitadores das palavras dos documentos. Por exemplo, considere o texto "*Cultivo de arroz,banana em 1995.*". O resultado da análise léxica são as palavras "*cultivo*", "*de*", "*arroz*", "*banana*" e "*em*". Note que os caracteres de dígitos são removidos e palavras maiúsculas são convertidas em minúsculas.

Stopwords são palavras que não possuem informação relevante para a discriminação dos documentos de interesse [Baeza99]. Possíveis classes gramaticais de palavras candidatas a *stopwords* são: artigo, conjunção, contração, interjeição, preposição e pronome. A *Remoção de stopwords* tem como objetivo remover palavras que não contribuem para a categorização dos documentos. Com isso, o número de palavras a serem consideradas é reduzido. Em nossos experimentos, removemos apenas preposição do conjunto *TV* (Seção 2.1, pág. 24). Escolhemos remover apenas preposições porque, em testes preliminares, foi a opção em que os categorizadores apresentaram os melhores desempenhos de categorização.

Após a *Análise léxica* e a *Remoção de stopwords*, aplicamos o pré-processamento *Redução de dimensionalidade* (*dimensionality reduction – DR*) com o objetivo de reduzir a dimensionalidade (o número de termos) do espaço vetorial de representação dos documentos. Para isso, usamos a técnica conhecida como lematização (*lemmatization*) [Manning08], em

que as palavras dos documentos são transformadas na sua forma canônica, ou lema, isto é, o singular de um substantivo ou o infinitivo de um verbo [Antiqueira05, Cherman07]. Para implementar a lematização, utilizamos o dicionário do SCAE, que possui a forma canônica de mais de 1.200.00 de palavras do Português [SCAE08].

As palavras canônicas do conjunto TV que sobrevivem à Análise léxica, Remoção de *stopwords*, e Redução de dimensionalidade são denominadas termos. Chamamos o conjunto de termos presentes em TV, ou seja, o conjunto de palavras de interesse, de *Lexicon*. Com o *Lexicon*, transformamos (ou seja, indexamos) cada documento d_j de nossas bases em sua forma vetorial, $\vec{d_j} = \langle w_{1j}, w_{2j}, \ldots, w_{|T|j} \rangle$, conforme discutido na Seção 2.2. Chamamos de Train and Test Vector (TTV) um documento na forma vetorial.

5.4 Validação Cruzada

Em problemas do mundo real, o conjunto de dados disponível para avaliar o desempenho das técnicas de categorização é limitado. Mas, para obtermos uma estimativa confiável do desempenho dos categorizadores desejamos treiná-los e testá-los com tantos documentos quanto possível. Existem muitas técnicas para tratar desse problema, mas a mais empregada na literatura, e que utilizamos neste trabalho, é a técnica n*-fold cross-validation* [Picard84].

Em n*-fold cross-validation*, o conjunto de dados é dividido em n partições mutuamente exclusivas de tamanhos aproximadamente iguais chamadas de *folds*. $n-1$ *folds* são usados para treinar, e o *fold* remanescente é usado para testar os categorizadores. Esse processo é repetido n vezes, cada vez considerando um *fold* diferente para teste. O desempenho reportado do categorizador multi-rótulo de texto segundo as métricas de avaliação de desempenho é a média dos valores das métricas obtidos em cada um dos n *folds*.

A repetição do processo de treinamento e teste permite atenuar a influência de uma amostra de treinamento e teste não representativa, tornando assim a avaliação de desempenho menos tendenciosa e mais confiável. Em experimentos da literatura, o n escolhido é freqüentemente igual a 10, pois testes extensivos sobre numerosas bases, com diferentes técnicas de categorização, têm mostrado que 10 é um número apropriado de *folds* para se obter uma estimativa confiável de desempenho [Witten05, pág. 150].

Em nossos experimentos, os 6.911 documentos da base de dados EX100 foram divididos em 10 *folds*, sendo 9 de 691 documentos e um de 692, e os 10.495 documentos da AT100 foram divididos também em 10 *folds*, sendo 9 de 1049 documentos e um de 1054. Nos experimentos com a base EX100, os categorizadores empregados foram treinados com 9 *folds* e com todos os documentos da CNAE_EX100, e testados com o décimo *fold*; enquanto que, nos experimentos com a base AT100, os categorizadores empregados foram treinados com 9 *folds* e com todos os documentos da CNAE_AT100, e testados com o décimo *fold*.

O tamanho médio do *Lexicon* para os experimentos com CNAE_EX100 e EX100 é 3609,8 termos (desvio padrão de 21,17 por conta dos diferentes *folds*), enquanto que, com CNAE_AT100 e AT100, é 5377,6 termos (desvio padrão de 19,45).

5.5 Calibração dos Categorizadores

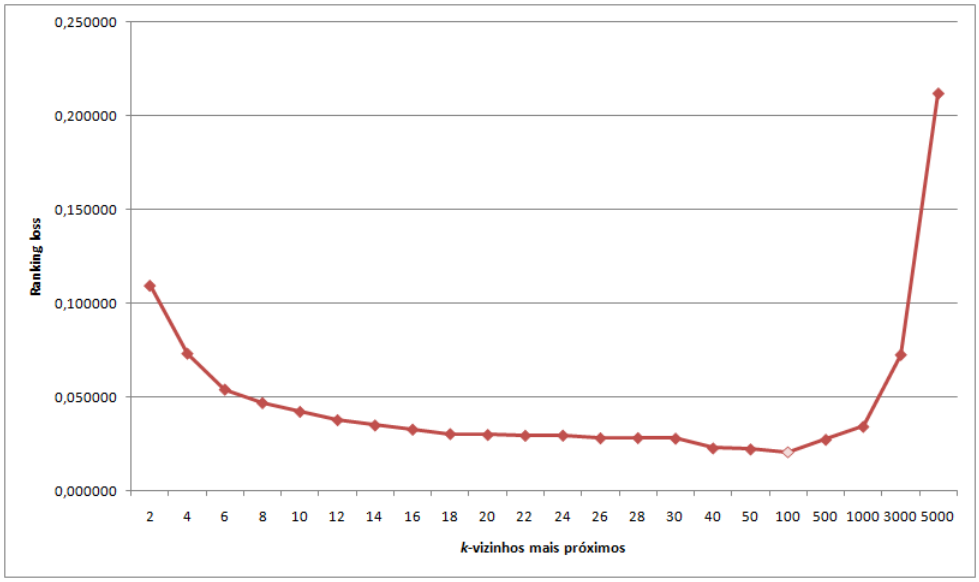
Os categorizadores apresentados no Capítulo 2 possuem parâmetros intrínsecos que devem ser ajustados (calibrados) com o objetivo de conseguir o melhor desempenho para uma determinada base de dados. Tipicamente, antes de realizar os experimentos de 10-*fold cross-validation*, os parâmetros dos categorizadores são calibrados com uma parte dos dados separada especificamente para a calibração, conhecida com dados de validação. Terminada a calibração dos categorizadores, os dados de validação são agregados aos dados de treinamento [Sebastiani02, Witten05].

Para a calibração de cada categorizador precisamos de dados para seu treinamento e teste com o objetivo de ajuste de parâmetros. O ajuste de parâmetros é feito segundo os seguintes passos:

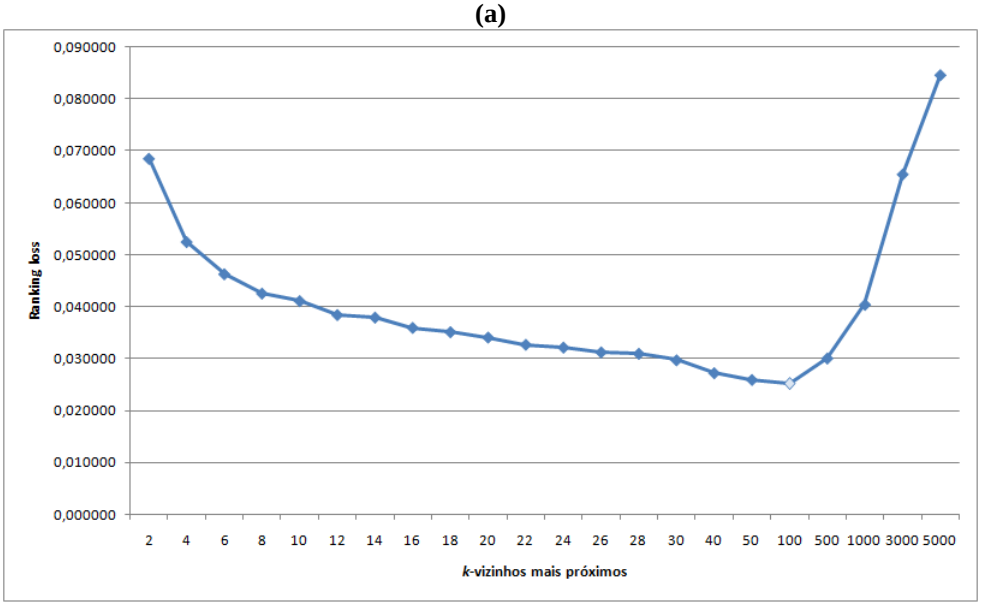
1. os parâmetros do categorizador são ajustados para um conjunto de valores inicial
2. o categorizador é treinado com uma parte dos dados de validação
3. o categorizador é testado com o restante dos dados de validação
4. seu desempenho medido segundo métrica específica e anotado
5. os parâmetros do categorizador são reajustados para um novo conjunto de valores
6. os passos de 2 a 5 são repetidos várias vezes e os parâmetros que produziram o melhor desempenho são escolhidos

Nos nossos experimentos de calibração, escolhemos como conjunto de dados de validação os documentos de treinamento de um dos *folds* das bases de dados empregadas (EX100 ou AT100). Dividimos este conjunto de dados em 10 partes, onde as nove primeiras são utilizadas no treinamento (passo 2, acima) e a décima no teste (passo 3) dos categorizadores; testamos com apenas uma das 10 partes por conta dos custos computacionais envolvidos. A métrica empregada nos experimentos de calibração (passo 4) foi a *ranking loss*. Nos experimentos de calibração, todos os documentos da CNAE_EX100 e CNAE_AT100 são utilizados durante a fase de treinamento.

O categorizador *ML-k NN* possui apenas um parâmetro, isto é, k (ver seções 2.3 e 2.3.2). O categorizador *ML-k NN* foi calibrado examinando seu desempenho para ambas as bases com os seguintes valores de k: 2, 4, 6, 8, 10, 12, 14, 18, 20, 22, 24, 26, 28, 30, 40, 50, 100, 500, 1000 e 5000. A Figura 5-6 mostra os resultados obtidos no passo 4 do procedimento de calibração do *ML-k NN* para as bases de dados EX100 (Figura 5-6(a)) e AT100 (Figura 5-6(b)). Nestas figuras, o eixo vertical representa o valor da métrica *ranking loss* para os diversos valores de k, e eixo horizontal os valores de k.



(a)



(b)

Figura 5-6 – Validação do *ML-* k *NN segundo a métrica ranking loss* para EX100, (a), e AT100, (b).

Conforme a Figura 5-6(a) mostra, para a base de dados EX100, este categorizador apresentou melhor desempenho segundo a métrica escolhida para $k = 100$ (ponto mais claro na Figura 5-6(a)). O mesmo ocorre com a base de dados AT100 (Figura 5-6(b)). Assim, o valor $k = 100$ foi escolhido para todos os demais experimentos com o categorizador *ML-* k *NN*.

O categorizador *VG-RAM WNN-COR* possui dois parâmetros: número de neurônios ($|O|$) e número de sinapses ($|X|$). Para os dois categorizadores a calibração foi realizada com números de neurônio igual a 32, 64, 128, 256, 512 e 1024, e número de sinapses igual 256, 512, 1024 e 2048 para as bases de dados EX100 e AT100.

A Figura 5-7 e a Figura 5-8 apresentam os resultados do processo de validação do *VG-RAM WNN-COR* para as bases EX100 e AT100, respectivamente.

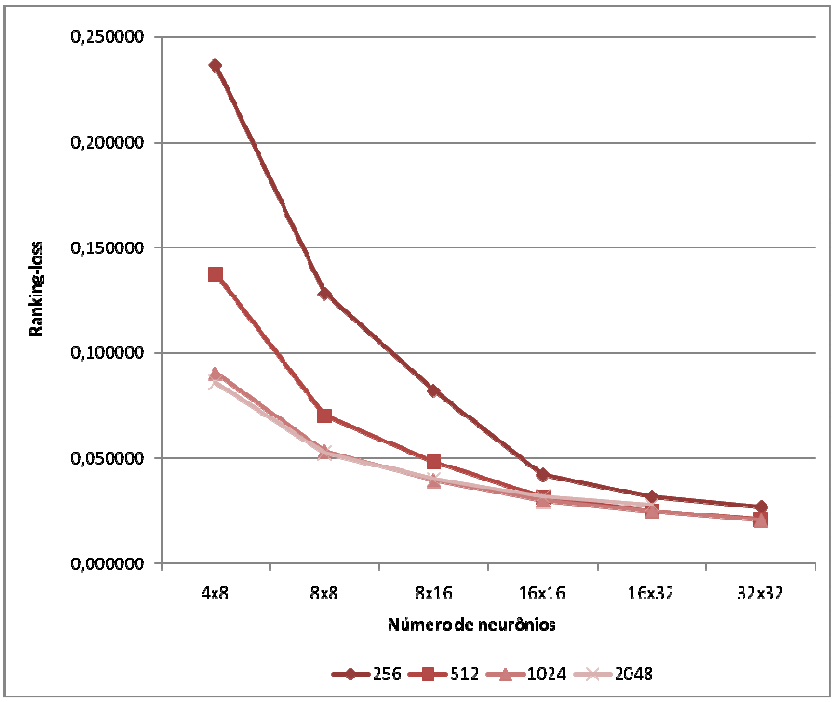


Figura 5-7 – Validação do *VG-RAM WNN-COR* na base EX100.

Conforme mostra a Figura 5-7, este categorizador apresentou melhor desempenho segundo a métrica escolhida para 1024 (32*x*32) neurônios, mas, mais uma vez, não está claro na figura qual o melhor número de sinapses. Como mostra a Tabela 5-1, o melhor número de sinapses é 512. Assim, para a base de dados EX100, os valores $|O| = 32x32$ e $|X| = 512$ foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN-COR*.

Tabela 5-1 – Validação para *VG-RAM WNN-COR* na EX100 para 32x32 neurônios.

Sinapses	***Ranking* loss**
256	0,024758
512	0,020754
1024	0,021162
2048	0,022277

De acordo com a Figura 5-8, o *VG-RAM WNN-COR* apresentou melhor desempenho segundo a métrica *ranking loss* para 1024 (32*x*32) neurônios e 1024 sinapses. Assim, para a

base de dados AT100, os valores $|O|=32x32$ e $|X|=1024$ foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN-COR*.

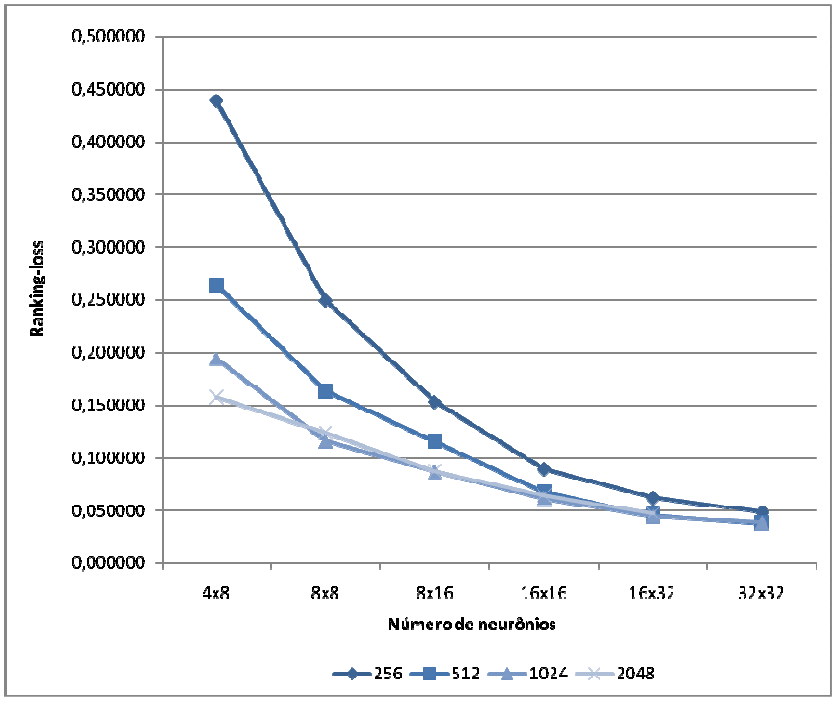


Figura 5-8 – Validação do *VG-RAM WNN-COR* na base AT100.

A Tabela 5-2 sumariza os parâmetros escolhidos para cada categorizador (primeira coluna à esquerda) para as bases de dados EX100 (coluna do meio) e AT100 (última coluna à direita).

Tabela 5-2 – Sumário das escolhas dos parâmetros dos categorizadores na validação para EX100 e AT100.

Categorizador	Bases de dados	
	EX100	**AT100**
ML-k NN	$k=100$	$k=100$
VG-RAM WNN-COR	$\|O\|=32x32$ $\|X\|=512$	$\|O\|=32x32$ $\|X\|=1024$

5.6 Cálculo dos Parâmetros para a Medida de Certeza

Os parâmetros (termos da regra de *Bayes*) do nosso método para mapear graus de crença em medidas de certeza de categorização (Seção 4.1) são específicos de uma dada

técnica de categorização e de uma dada base de documentos. Nos experimentos realizados para obter os parâmetros de nossa modelagem Bayesiana da certeza na categorização, empregamos os categorizadores *ML-kNN* e *VG-RAM WNN-COR* e os conjuntos de dados EX100 e AT100 (Seção 5.1). Particionamos as bases EX100 e AT100 em 10 subconjuntos de 691 e 1049 documentos, respectivamente (o último tem 692 e 1054, respectivamente). Para estimar os termos da regra de *Bayes* (*p(x|k)*, *p(y|k)* e *p(y|x,k)*), para cada uma das bases EX100 e AT100 e para cada um dos categorizadores *ML-kNN* e *VG-RAM WNN-COR*, realizamos uma série de 10 experimentos de calibração. Nestes experimentos, usamos os primeiros 9 dos 10 subconjuntos mencionamos acima. Dividimos esses 9 subconjuntos novamente em 10 subconjuntos, e usamos os primeiros nove para treinamento e o último para calibração. Este processo foi repetido 10 vezes com subconjuntos para treinamento e calibração diferentes. Os valores dos termos da regra de Bayes (*p(x|k)*, *p(y|k)* e *p(y|x,k)*) foram calculados com base nos resultados das categorizações de todos documentos nesses 10 experimentos de calibração.

Para avaliarmos os resultados dos categorizadores *ML-kNN* e *VG-RAM WNN-COR*, utilizamos a métrica *one-error*(*k*), que retorna 0 (zero), se a categoria na posição *k* do *ranking* pertence ao conjunto de categorias pertinentes ao documento de teste, ou 1 (um), caso contrário. Para cada posição *k* do *ranking*, particionamos uniformemente os valores de grau de crença $f(d_j, c_i)$ (Seção 4.1) observados na calibração em 20 intervalos. Dessa forma, para cada posição *k* do *ranking* considerada, *p(y|k)* é praticamente igual para os diferentes intervalos *y*. É válido mencionar que os 20 intervalos são diferentes para cada posição *k* do *ranking* considerada.

```
1  Algoritmo: Cálculos dos Parâmetros para Medida Certeza na Categorização

2  CONST INTER 20
3  CONST POS 6

4  obter_intervalo ( predicao )
5  inicio
6    para i = 0; i < INTER; i = i +1
7      inicio
8        se grau_crenca(predicao) ∈ F[i]
9              return i;
10     fim
11 fim.

12 verificar_classe_pertinete( d_i, predicao )
13  inicio
14    se predicao pertinente d_i
15       retorna 1;
16    returna 0;
17  fim.

18 inicio

   % Treinamento do categorizador
19 CATEGORIZADOR (TV);

   %Acumulando os valores para o caçulo da regra de Bayes por posição do ranking categorias e intervalo y
20   para i = 0|; i <= | Te |; i = i + 1
21    inicio
        % Categorizador retorna ranking de categorias e graus de crença
22      r_i = CATEGORIZADOR (Te [i]);
23      para pos = 0; pos < |C|; pos = pos +1
24       inicio
25         y = obter_intervalo( r_i[ pos ] );
26         count_prediction = count_prediction + 1;
27         a_y_k[ y ][pos] = a_y_k[ y ] [pos] + 1;
28         se verificar_classe_pertinete (Te [i], r_i[ pos ] ) entao
29          inicio
30            a_x_k[pos] = a_x_k [pos] +1;
31            a_y_x_k[ y ][pos] = a_y_x_k[ y ] [pos] + verificar_classe_pertinete (Te [i], r_i[ pos ] );
32          fim
33       fim
34    fim

   % Calculando os termos da regra de Bayes para cada posição do ranking de categorias e intervalo y.
35  para pos = 0; pos < POS; pos = pos +1
36   inicio
37      p_x_k[pos] = a_x_k [pos] / count_prediction;
38      para y = 0; y < INTER; y = y +1
39       inicio
40         p_y_k[ y ][pos] = a_y_k[ y ] [pos] / count_prediction;
41         p_y_x_k[ y ][pos] = a_y_x_k[ y ] [pos] count_prediction;
42       fim
43   fim
44 fim.
```

Figura 5-9 - Pseudocódigo para calcular os parâmetros da medida de certeza

A Figura 5-9 mostra o pseudocódigo para calcular os parâmetros da medida de certeza

Os parâmetros de entrada do algoritmo são TV, Te, INTER, POS. O parâmetro INTER informa em quantos intervalos o conjunto F (veja na seção 4.1) será particionado, formando os intervalos y. O parâmetro POS informa quantas posições do *ranking* de categorias que será avaliado, ou seja, até que posição do *ranking* será computado os parâmetros para o cálculo da medida de certeza do categorizador. De acordo com a Figura 5-9, o passo (1) define o nome do algoritmo, os passos (2) e (3) definem o valor dos parâmetros INTER e POS. Os passos de (4) a (11) definem uma função chamada obter_intervalo que computa em qual intervalo y $\in$F a predição do categorizador se encaixa. Os passos (12) a (17) definem uma função chamada verificar_classe_pertinente que verificar se a predição é correta retornando 1 e 0, caso contrário. Os passos (19) a (34) são repetidos 10 vezes para cada conjunto de treinemento e teste. Desta forma, executamos um treinamento com nove partes e com uma partição não utilizada no treinamento para calibrar, esse procedimento é repetido 10 vezes. Computamos as seguintes ocorrências e acumulamos em variáveis os seguintes resultados:

1. Acumulamos em a_y_k: o número de predições com grau de crença que se encaixam dentro do intervalo y $\in F$, observadas na posição *k* do *ranking* de cateogiras (para as 10 repetições).
2. Acumulamos em a_x_k: o número de predições corretas observadas na posição *k* do *ranking* de categorias (para as 10 repetições).
3. Acumulamos em a_y_x_k: o número de predições com grau de crença que se encaixam dentro do intervalo $y \in F$ do conjunto de predições corretas observadas na posição k do *ranking* de categorias (para as 10 repetições).

Os passos (35) a (41) computam as seguintes probabilidades: $p(y \mid k)$, $p(x \mid k)$ e $p(y \mid x,k)$ (veja seção 4.1) utilizadas para calcular a medida de certeza do categorizador. Como pode ser observado, para cada posição do *ranking* de categorias computamos uma medida de certeza para a predição correspondente.

Os valores dos termos da regra de *Bayes* (*p(x|k), p(y|k)* e *p(y|x,k)*) obtidas através dos experimentos de calibração e o valor da medida de certeza são apresentados na Seção 6.1.

5.7 Validação da Medida de Certeza

Nos experimentos realizados para validar nossa modelagem Bayesiana da certeza na categorização, particionamos as bases EX100 e AT100 em 10 subconjuntos de 691 e 1049 documentos, respectivamente (o último tem 692 e 1054, respectivamente). Para cada uma das bases EX100 e AT100, utilizamos os 9 primeiros subconjuntos para treinar o categorizador e o último subconjunto para testá-lo. O objetivo deste experimento de teste é avaliar se o valor de $p(x|y,k)$ calculado analiticamente (a partir dos termos da Regra de *Bayes* observados nos 10 experimentos de validação) é uma boa estimativa para o valor de $p(x|y,k)$ observado empiricamente. Com esse experimento, verificamos que a nossa abordagem estima corretamente o valor de $p(x|y,k)$, ou seja, o valor de $p(x|y,k)$ calculado analiticamente é similar ao valor de $p(x|y,k)$ observado empiricamente.

A comparação entre os valores de $p(x|y,k)$ calculados analiticamente (por meio da regra de *Bayes* a partir das estimativas de $p(x|k)$, $p(y|k)$ e $p(y|x,k)$ obtidas nos experimentos de calibração) com os valores de $p(x|y,k)$ estimados empiricamente (a partir dos experimentos de teste) são apresentados na Seção 6.1.

5.8 Calibração das Estratégias de Poda

As estratégias de poda apresentadas no Capítulo 3 possuem parâmetros que devem ser ajustados (calibrados) com o objetivo de conseguir o melhor desempenho para uma determinada base de dados. Tipicamente, antes de realizar os experimentos de 10-*fold cross-validation*, os parâmetros das estratégias de poda são calibrados com uma parte dos dados separada especificamente para a calibração, a qual denominamos dados de calibração Os dados de calibração são adicionados aos dados de treinamento nos experimentos de teste [Sebastiani02, Witten05].

Nos nossos experimentos de calibração das estratégias de poda, escolhemos como conjunto de dados de validação os documentos de treinamento de um dos *folds* das bases de dados empregadas (EX100 ou AT100). Dividimos este conjunto de dados em 10 partes, onde as nove primeiras são utilizadas no treinamento e a décima no teste (validação) dos categorizadores; testamos com apenas uma das 10 partes por conta dos custos computacionais

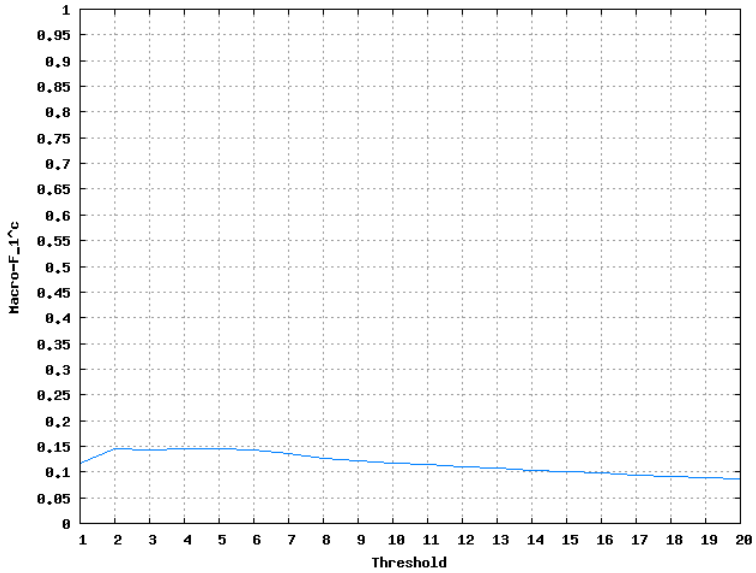
envolvidos. Nos experimentos de calibração, todos os documentos da CNAE_EX100 e CNAE_AT100 são utilizados durante a fase de treinamento. A métrica empregada nos experimentos de calibração foi a $macro - F_1^c$. Escolhemos esta métrica porque as estratégias de poda publicadas na literatura foram avaliadas em termos desta métrica.

A seguir, apresentamos os resultados da calibração dos parâmetros das estratégias de poda avaliadas neste trabalho.

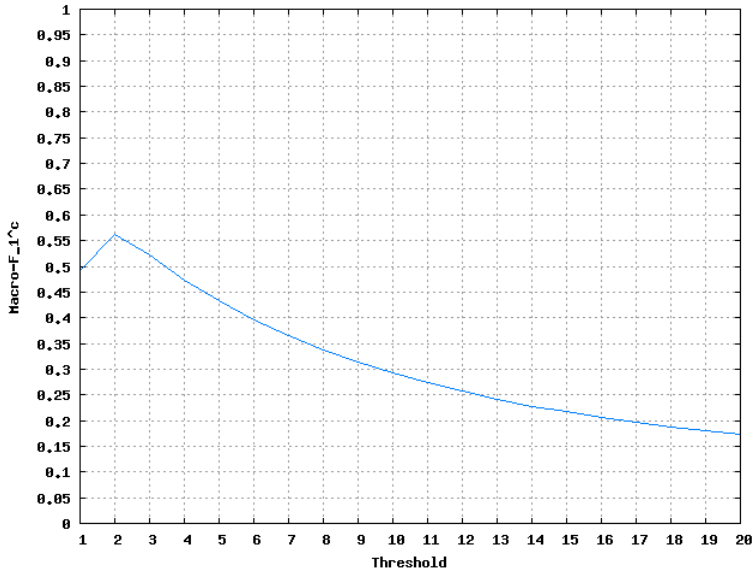
5.8.1 Estratégia RCut

A estratégia de poda RCut possui apenas um parâmetro, isto é, *t* (Seção 3.1). RCut foi calibrada examinando seu desempenho para ambas as bases com os seguintes valores de *t* = 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19 e 20.

A Figura 5-10 mostra os resultados obtidos do procedimento de calibração de RCut para o categorizador *ML-k NN* e para as bases de dados AT100 (Figura 5-10(a)) e EX100 (Figura 5-10(b)). A Figura 5-11 mostra os resultados obtidos do procedimento de calibração de RCut para o categorizador *VG-RAM WNN-COR* e para as bases de dados AT100 (Figura 5-11(a)) e EX100 (Figura 5-11(b)). Nestas figuras, o eixo vertical representa o valor da métrica $macro - F_1^c$ para os diversos valores de *t*, e eixo horizontal os valores de *t*.

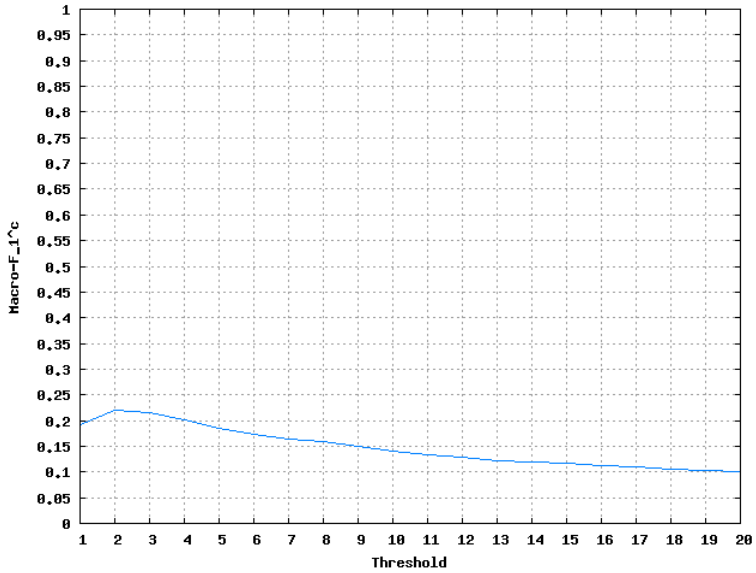


(a)

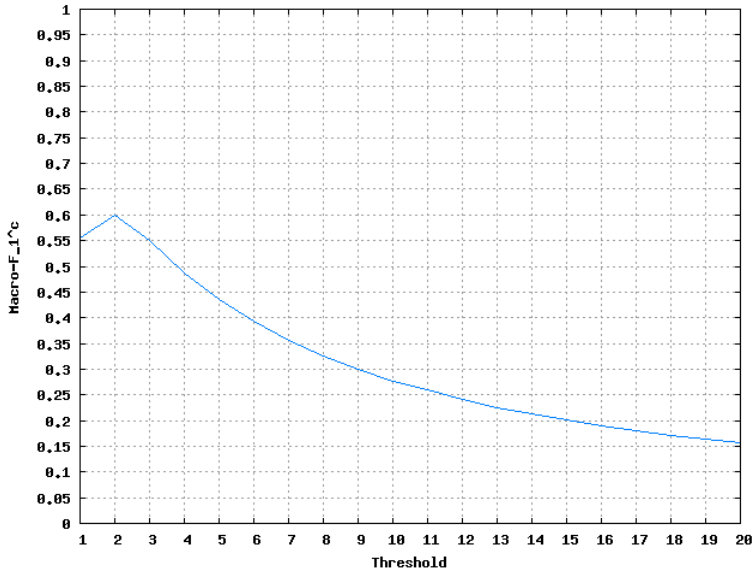


(b)

Figura 5-10 – Calibração de RCut para ML-k NN e para (a) AT100 e (b) EX100.



(a)



(b)

Figura 5-11 – Calibração de RCut para *VG-RAM WNN-COR* para (a) AT100 e (b) EX100.

Conforme a Figura 5-10(a) mostra, para o categorizador *ML-k NN* e para a base de dados AT100, a estratégia RCut apresentou melhor desempenho com $t = 4$. Assim, o valor $t = 4$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base AT100. Já para a base EX100 (Figura 5-10(b)), RCut apresentou melhor desempenho com $t = 2$. Assim, o valor $t = 2$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base EX100.

Conforme a Figura 5-11(a) mostra, para o categorizador *VG-RAM WNN-COR* e para a base de dados AT100, a estratégia RCut apresentou melhor desempenho com $t = 2$. O mesmo ocorre com a base EX100 (Figura 5-11(b)). Assim, o valor $t = 2$ foi escolhido para todos os demais experimentos com o categorizador *VG-RAM WNN-COR* e as bases AT100 e EX100.

A Tabela 5-3 sumariza os valores escolhidos para o parâmetro de RCut para cada categorizador (primeira coluna à esquerda) e para as bases AT100 (coluna do meio) e EX100 (última coluna à direita).

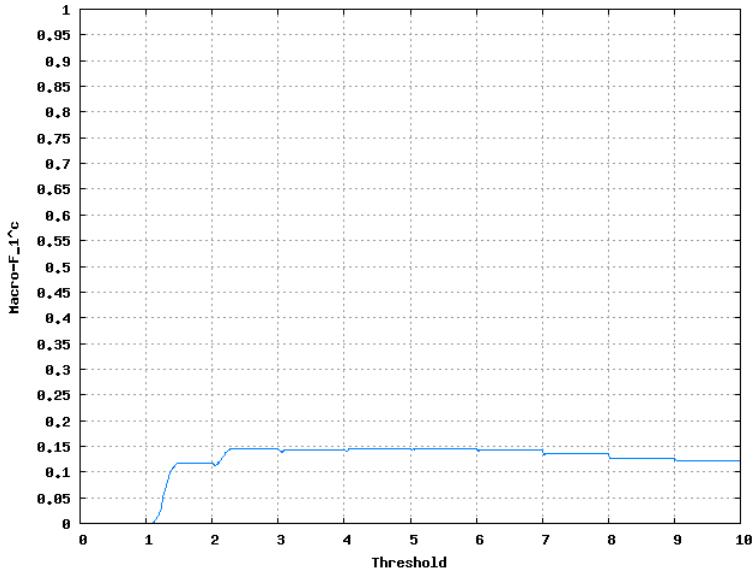
Tabela 5-3 – Sumário dos valores escolhidos para o parâmetro de RCut.

Categorizador	**Bases de dados**	
	AT100	**EX100**
ML-*k* NN	$\tau = 4$	$\tau = 2$
VG-RAM WNN-COR	$\tau = 2$	$\tau = 2$

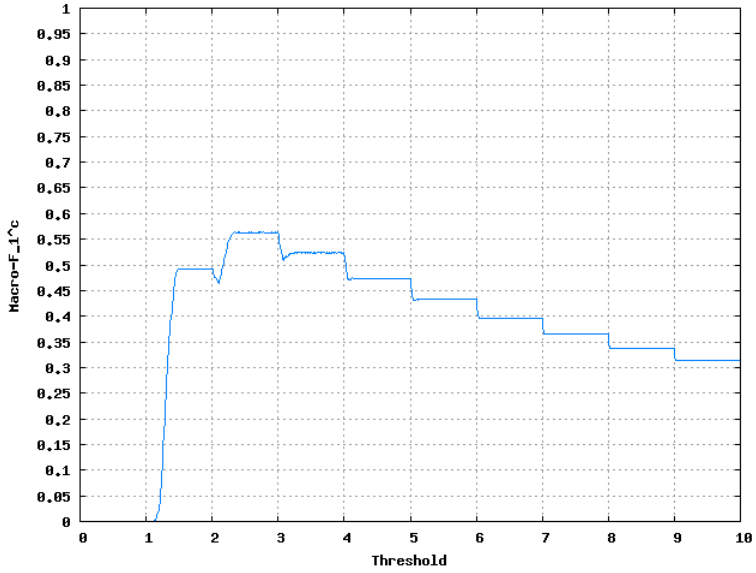
5.8.2 Estratégia RTCut

A estratégia de poda RTCut possui apenas um parâmetro, isto é, τ (ver Seção 3.2). Após transformar o *ranking* de categorias em um *ranking* sintético aplicando a Equação ((3-1), a estratégia de poda RTCut foi calibrada examinando seu desempenho para as ambas as bases com o valor de inicio $\tau = 0.0$, variando em 0.01 até o valor limite 10.0.

A Figura 5-12 mostra os resultados obtidos do procedimento de calibração de RTCut para o categorizador *ML-k NN* e para as bases de dados AT100 (Figura 5-12 (a)) e EX100 (Figura 5-12 (b)). A Figura 5-13 mostra os resultados obtidos do procedimento de calibração de RTCut para o categorizador *VG-RAM WNN-COR* e para as bases de dados AT100 (Figura 5-13 (a)) e EX100 (Figura 5-13 (b)). Nestas figuras, o eixo vertical representa o valor da métrica $macro - F_1^c$ para os diversos valores de τ, e eixo horizontal os valores de τ.

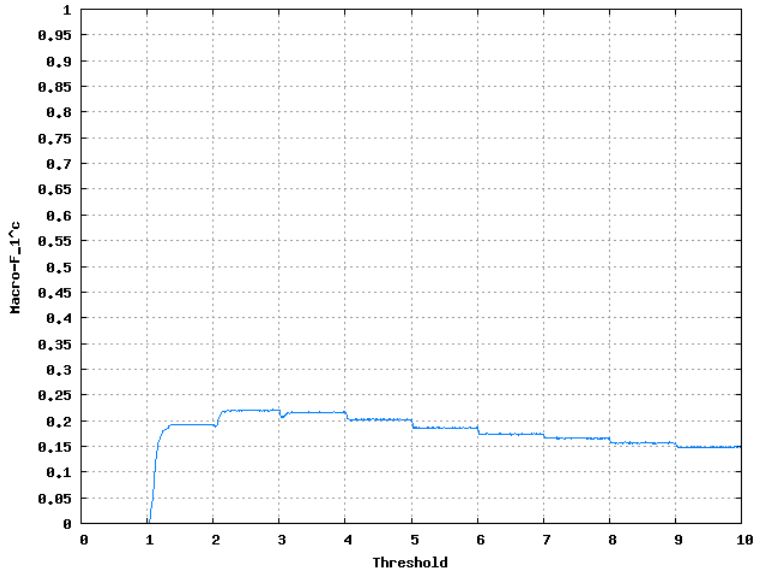


(a)

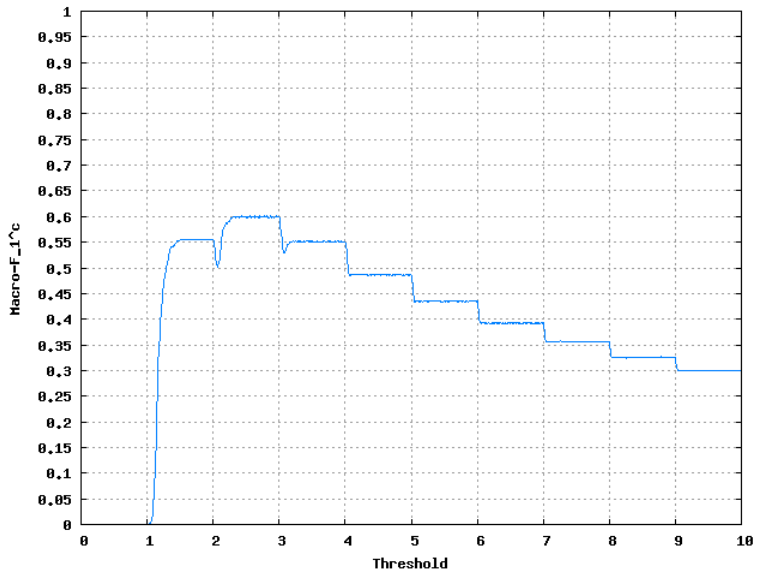


(b)

Figura 5-12 - Calibração de RTCut para *ML-k NN* para (a) AT100 e (b) EX100.



(a)



(b)

Figura 5-13 - Calibração de RTCut para *VG-RAM WNN-COR* para (a) AT100 e (b) EX100.

Conforme a Figura 5-12(a) mostra, para o categorizador *ML-k NN* e para a base de dados AT100, a estratégia RTCut apresentou melhor desempenho com $\tau = 4{,}28$. Assim, o valor $t = 4{,}28$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base AT100. Já para a base EX100 (Figura 5-12(b)), RTCut apresentou melhor desempenho com $\tau = 2{,}33$. Assim, o valor $\tau = 2{,}33$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base EX100.

Conforme a Figura 5-13(a) mostra, para o categorizador *VG-RAM WNN-COR* e para a base de dados AT100, a estratégia RTCut apresentou melhor desempenho com $\tau = 2{,}61$. Assim, o valor $\tau = 2{,}61$ foi escolhido para todos os demais experimentos com o categorizador

VG-RAM WNN-COR e a base AT100. Já para a base EX100 (Figura 5-13 (b)), RTCut apresentou melhor desempenho com $\tau = 2{,}63$. Assim, o valor $\tau = 2{,}63$ foi escolhido para todos os demais experimentos com o categorizador *VG-RAM WNN-COR* e a base EX100.

A Tabela 5-4 sumariza os valores escolhidos para o parâmetro de RTCut para cada categorizador (primeira coluna à esquerda) e para as bases AT100 (coluna do meio) e EX100 (última coluna à direita).

Tabela 5-4 – Sumário das escolhas dos parâmetros da estratégia de poda RTCut.

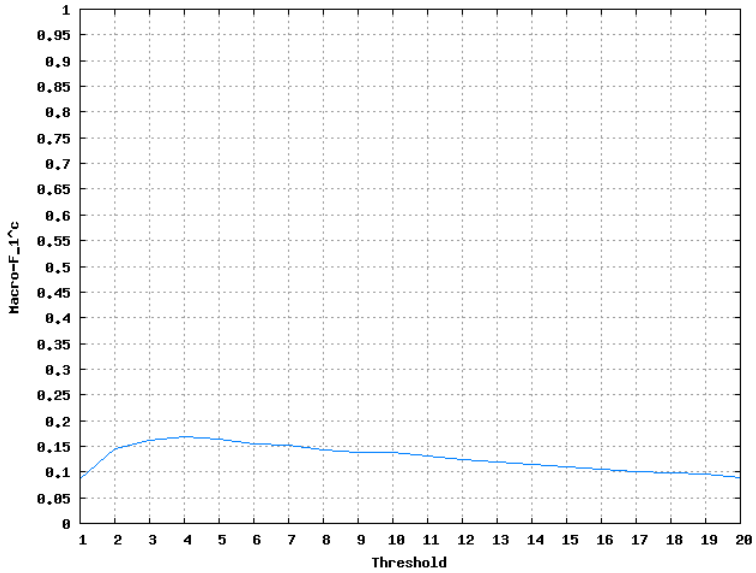
Categorizador	**Bases de dados**	
	AT100	**EX100**
ML-*k* NN	$\tau = 4{,}28$	$\tau = 2{,}33$
VG-RAM WNN-COR	$\tau = 2{,}61$	$\tau = 2{,}63$

5.8.3 Estratégia PCut

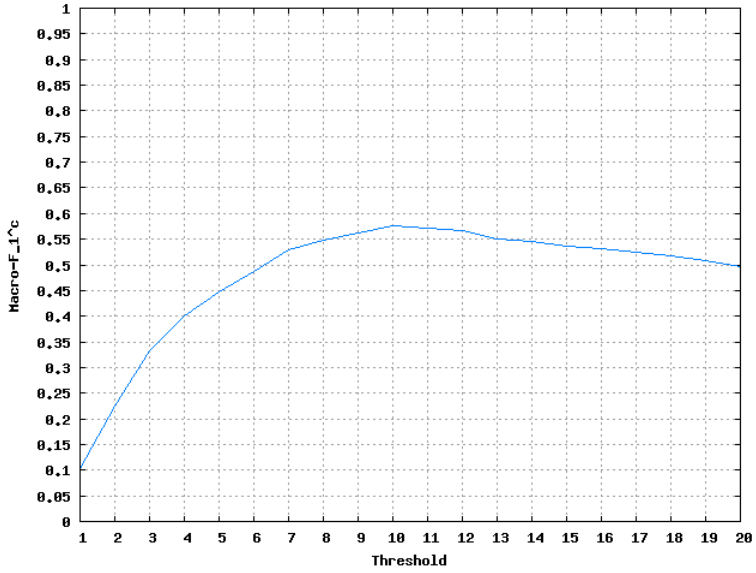
A estratégia de poda PCut possui um parâmetro de ajuste, a saber, *x*, (ver Seção 3.3). PCut foi calibrada examinando seu desempenho para as ambas as bases com os seguintes valores de $x = 1, 2, 3, 4, 5, 6, 7, 8, 9$ e 10.

A Figura 5-14 mostra os resultados obtidos do procedimento de calibração de PCut para o categorizador *ML-k NN* e para as bases de dados AT100 (Figura 5-14(a)) e EX100 (Figura 5-14(b)). A Figura 5-15 mostra os resultados obtidos do procedimento de calibração de PCut para o categorizador *VG-RAM WNN-COR* e para as bases de dados AT100 (Figura 5-15(a)) e EX100 (Figura 5-15(b)). Nestas figuras, o eixo vertical representa o valor da métrica $macro - F_1^c$ para os diversos valores de *x*, e eixo horizontal os valores de *x*.

.

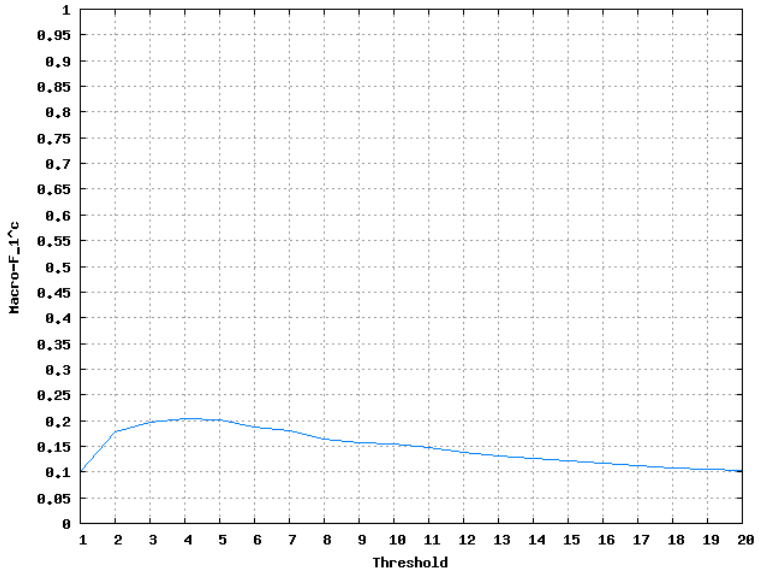


(a)

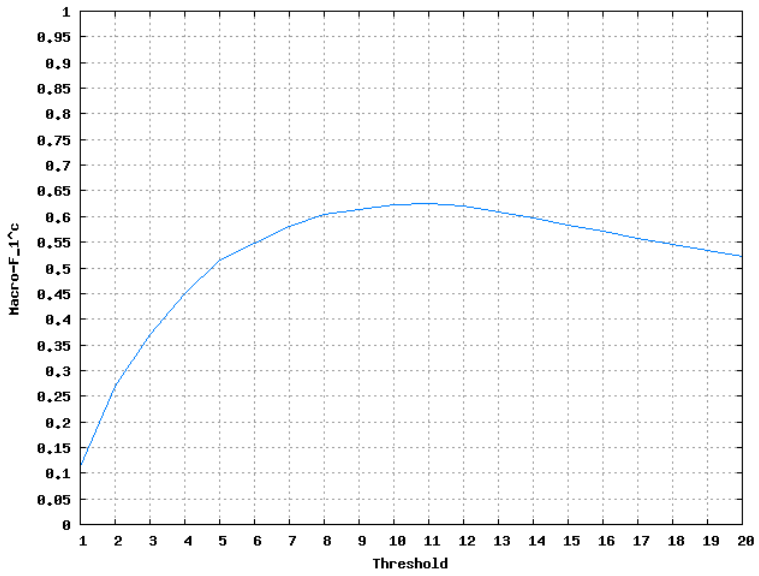


(b)

Figura 5-14 - Calibração de PCut para *ML-*k *NN* e para (a) AT100 e (b) EX100.



(a)



(b)

Figura 5-15 - Calibração de PCut para *VG-RAM WNN-COR* e para (a) AT100 e (b) EX100.

Conforme a Figura 5-14(a) mostra, para o categorizador *ML-k NN* e para a base de dados AT100, a estratégia PCut apresentou melhor desempenho com $x = 4$. Assim, o valor $x = 4$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base AT100. Já para a base EX100 (Figura 5-14(b)), PCut apresentou melhor desempenho com $x = 10$. Assim, o valor $x = 10$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base EX100.

Conforme a Figura 5-15(a) mostra, para o categorizador *VG-RAM WNN-COR* e para a base de dados AT100, a estratégia PCut apresentou melhor desempenho com $x = 4$. Assim, o valor $x = 4$ foi escolhido para todos os demais experimentos com o categorizador *VG-RAM*

WNN-COR e a base AT100. Já para a base EX100 (Figura 5-15(b)), PCut apresentou melhor desempenho com x = 11. Assim, o valor x = 11 foi escolhido para todos os demais experimentos com o categorizador *VG-RAM WNN-COR* e a base EX100.

A Tabela 5-5 sumariza os valores escolhidos para o parâmetro de PCut para cada categorizador (primeira coluna à esquerda) e para as bases AT100 (coluna do meio) e EX100 (última coluna à direita).

Tabela 5-5 – Sumário das escolhas dos parâmetros da estratégia de poda PCut.

Categorizador	**Bases de dados**	
	AT100	**EX100**
ML-*k* NN	$\tau = 4$	$\tau = 10$
VG-RAM WNN-COR	$\tau = 4$	$\tau = 11$

5.8.4 Estratégia SCut

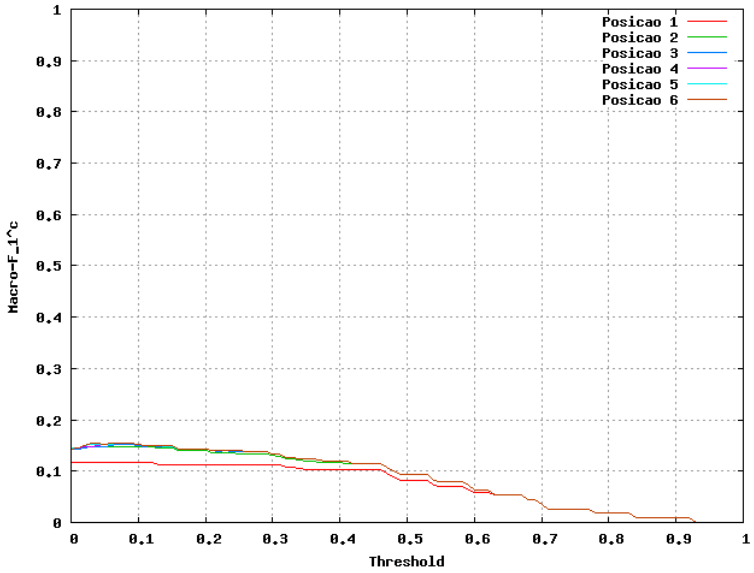
A estratégia de poda SCut possui um parâmetro de ajuste para cada categoria c_j, a saber, τ_j, (ver Seção 3.4). A estratégia de poda SCut foi calibrada examinando seu desempenho para as ambas as bases com o valor de inicio τ_j = 0.0, variando em 0.01 até o valor limite 1.0. para cada categoria c_j.

O APÊNDICE A apresenta os parâmetros obtidos no procedimento de calibração de SCut para os categorizadores *ML-k NN* e *VG-RAM WNN-COR* e para as bases de dados AT100 e EX100. A Tabela 9-1, Tabela 9-2, Tabela 9-3, Tabela 9-4 e Tabela 9-5 mostram os resultados obtidos do procedimento de calibração do SCut para o categorizador *ML-k NN* e para as bases de dados EX100 e AT100. Tabela 9-6, Tabela 9-7, Tabela 9-8, Tabela 9-9 e Tabela 9-10 mostram os resultados obtidos do procedimento de calibração do SCut para o categorizador *VG-RAM WNN-COR* e para as bases de dados EX100 e AT100.

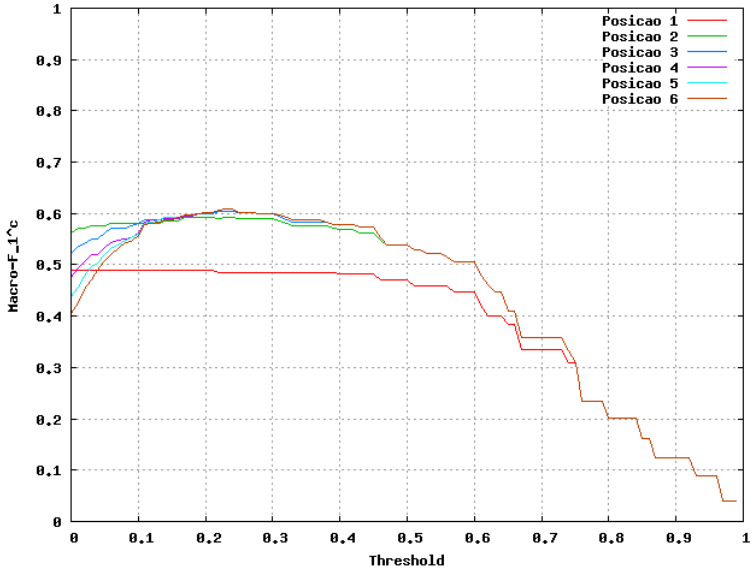
5.8.5 Estratégia BCut

A estratégia de poda BCut possui apenas um parâmetro, isto é, τ (ver Seção 4.2.1). Após transformar o *ranking* original em um *ranking* probabilístico, a estratégia de poda BCut foi calibrada examinando seu desempenho para ambas as bases com o valor de inicio $\tau = 0.0$, variando em *0.01* até o valor limite *1.00*.

A Figura 5-16 mostra os resultados obtidos do procedimento de calibração de BCut para o categorizador *ML-k NN* e para as bases de dados AT100 (Figura 5-16(a)) e EX100 (Figura 5-16(b)). A Figura 5-17 mostra os resultados obtidos do procedimento de calibração de BCut para o categorizador *VG-RAM WNN-COR* e para as bases de dados AT100 (Figura 5-17(a)) e EX100 (Figura 5-17(b)). Nestas figuras, o eixo vertical representa o valor da métrica $macro - F_1^c$ para os diversos valores de τ, e eixo horizontal os valores de τ.

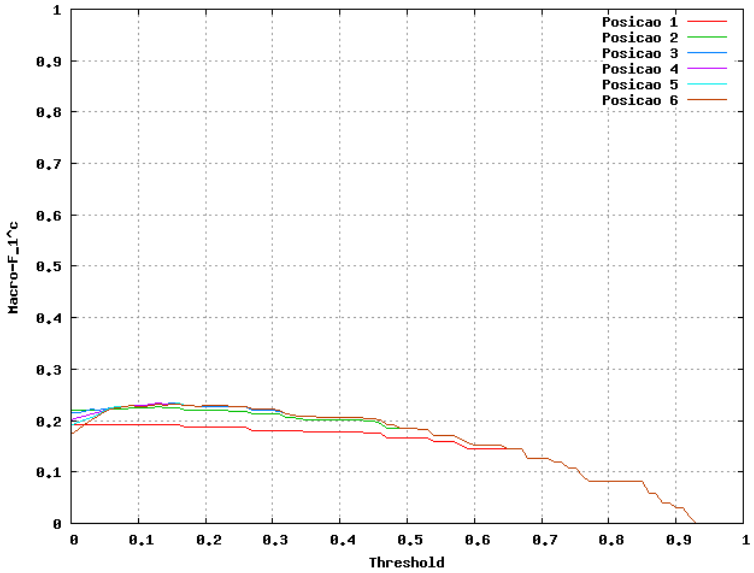


(a)

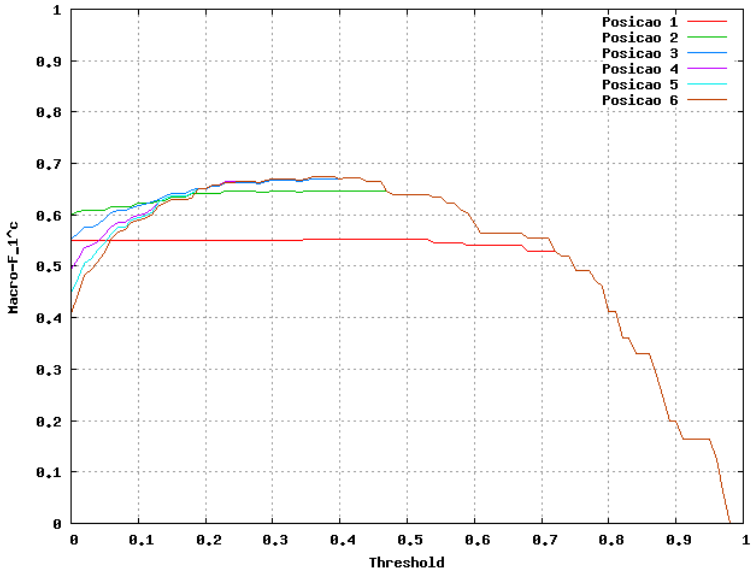


(b)

Figura 5-16 - Calibração de BCut para *ML-k NN* e para (a) AT100 e (b) EX100.



(a)



(b)

Figura 5-17 - Calibração de BCut para *VG-RAM WNN-COR* e para (a) AT100 e (b) EX100.

Conforme a Figura 5-16(a) mostra, para o categorizador *ML-k NN* e para a base de dados AT100, a estratégia BCut apresentou melhor desempenho com $\tau = 0,03$. Assim, o valor $\tau = 0,03$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base AT100. Já para a base EX100 (Figura 5-16(b)), BCut apresentou melhor desempenho com $\tau = 0,22$. Assim, o valor $\tau = 0,22$ foi escolhido para todos os demais experimentos com o categorizador *ML-k NN* e a base EX100.

Conforme a Figura 5-17(a) mostra, para o categorizador *VG-RAM WNN-COR* e para a base de dados AT100, a estratégia BCut apresentou melhor desempenho com $\tau = 0,13$. Assim, o valor $\tau = 0,13$ foi escolhido para todos os demais experimentos com o categorizador *VG-RAM WNN-COR* e a base AT100. Já para a base EX100 (Figura 5-17(b)), BCut apresentou

melhor desempenho com $\tau = 0{,}36$. Assim, o valor $\tau = 0{,}36$ foi escolhido para todos os demais experimentos com o categorizador *VG-RAM WNN-COR* e a base EX100.

A Tabela 5-6 sumariza os valores escolhidos para o parâmetro de BCut para cada categorizador (primeira coluna à esquerda) e para as bases AT100 (coluna do meio) e EX100 (última coluna à direita).

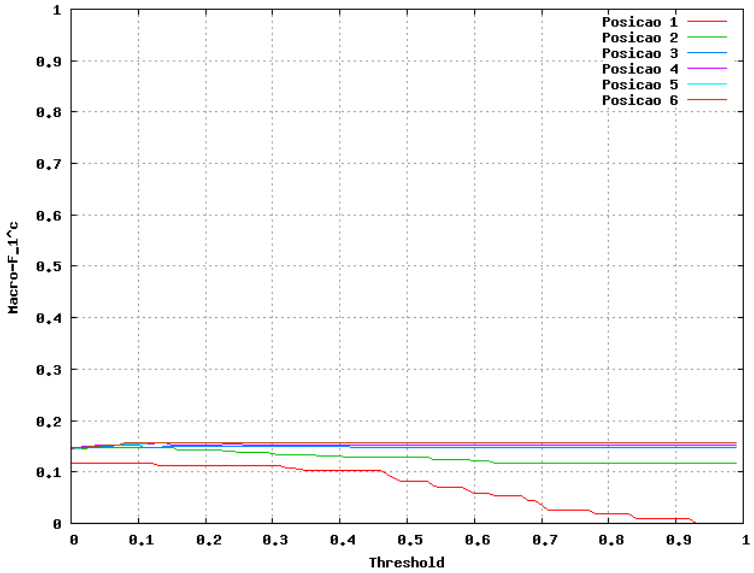
Tabela 5-6 – Sumário das escolhas dos parâmetros da estratégia de poda BCut.

Catetegorizador	Bases de dados	
	AT100	**EX100**
ML-*k* NN	$\tau = 0{,}03$	$\tau = 0{,}22$
VG-RAM WNN-COR	$\tau = 0{,}13$	$\tau = 0{,}36$

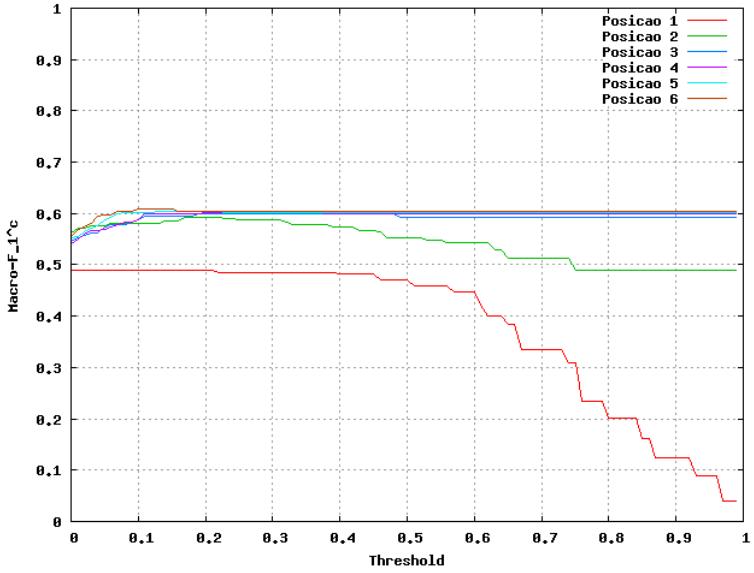
5.8.6 Estratégia PBCut

A estratégia de poda PBCut possui um parâmetro para cada posição do *ranking* avaliada, isto é, τ_i (ver Seção 4.2.2). A estratégia de poda PBCut foi calibrada examinando seu desempenho para as ambas bases com o valor de inicio $\tau_i = 0.0$, variando em *0.01* até o valor limite *1.00* para cada posição do *ranking*.

A Figura 5-18 mostra os resultados obtidos do procedimento de calibração de PBCut para o categorizador *ML-k NN* e para as bases de dados AT100 (Figura 5-18(a)) e EX100 (Figura 5-18(b)). A Figura 5-19 mostra os resultados obtidos do procedimento de calibração de PBCut para o categorizador *VG-RAM WNN-COR* e para as bases de dados AT100 (Figura 5-19(a)) e EX100 (Figura 5-19(b)). Nestas figuras, o eixo vertical representa o valor da métrica $macro - F_1^c$ para os diversos valores de τ_i, e eixo horizontal os valores de τ_i.

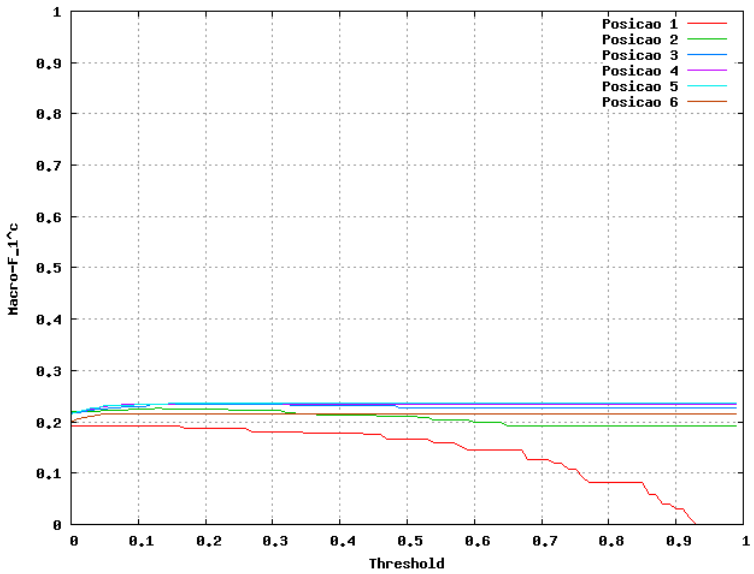


(a)

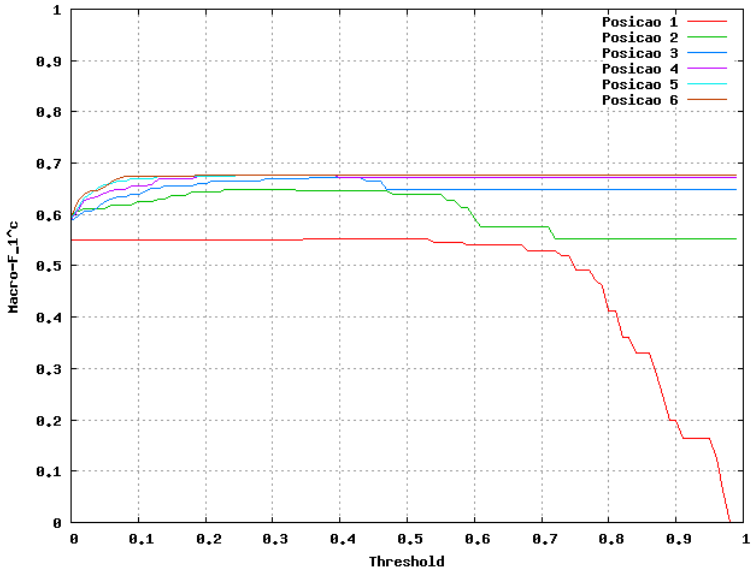


(b)

Figura 5-18 - Calibração de PBCut para *ML-k NN* e para (a) AT100 e (b) EX100.



(a)



(b)

Figura 5-19 - Calibração de PBCut para *VG-RAM WNN-COR* e para (a) AT100 e (b) EX100.

Conforme a Figura 5-18(a) mostra, para o categorizador *ML-k NN* e para a base de dados AT100, a estratégia PBCut apresentou melhor desempenho com τ_i igual aos valores mostrados na Tabela 5-7 (coluna AT100). Assim, os valores dos parâmetros da Tabela 5-7 (coluna AT100) foram escolhidos para todos os demais experimentos com o categorizador *ML-k NN* e a base AT100. Já para a base EX100 (Figura 5-18(b)), PBCut apresentou melhor desempenho com τ_i igual aos valores mostrados na Tabela 5-7 (coluna EX100). Assim, os valores dos parâmetros da Tabela 5-7 (coluna EX100) foram escolhidos para todos os demais experimentos com o categorizador *ML-k NN* e a base EX100.

Tabela 5-7 – Parâmetro obtidos na calibração da estratégia de poda PBCut segundo *ML-k NN* para AT100 e EX100.

AT100	EX100
τ_1 = 0,00	τ_1 = 0,00
τ_2 = 0,12	τ_2 = 0,17
τ_3 = 0,09	τ_3 = 0,25
τ_4 = 0,13	τ_4 = 0,22
τ_5 = 0,16	τ_5 = 0,13
τ_6 = -1,00	τ_6 = 0,10

Conforme a Figura 5-19 (a) mostra, para o categorizador *VG-RAM WNN-COR* e para a base de dados AT100, a estratégia PBCut apresentou melhor desempenho com τ_i igual aos valores mostrados na Tabela 5-8 (coluna AT100). Assim, os valores dos parâmetros da Tabela 5-8 (coluna AT100) foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN-COR* e a base AT100. Já para a base EX100 (Figura 5-19 (b)), PBCut apresentou melhor desempenho com τ_i igual aos valores mostrados na Tabela 5-8(coluna EX100). Assim, os valores dos parâmetros da Tabela 5-8(coluna EX100) foram escolhidos para todos os demais experimentos com o categorizador *VG-RAM WNN-COR* e a base EX100.

Tabela 5-8 - Parâmetro obtidos na calibração da estratégia de poda PBCut segundo *VG-RAM WNN-COR* para AT100 e EX100.

AT100	EX100
τ_1 = 0,00	τ_1 = 0,35
τ_2 = 0,13	τ_2 = 0,23
τ_3 = 0,15	τ_3 = 0,36
τ_4 = 0,20	τ_4 = 0,23
τ_5 = 0,00	τ_5 = -1,00
τ_6 = -1,00	τ_6 = -1,00

A Tabela 5-9 sumariza os parâmetros escolhidos para a estratégia de poda PBCut sob os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para as bases de dados AT100 (coluna do meio) e EX100 (última coluna à direita).

Tabela 5-9 – Sumário das escolhas dos parâmetros da estratégia de poda PBCut.

Categorizador	Bases de dados	
	AT100	**EX100**
ML- kNN	τ_1 = 0,00	τ_1 = 0,00
	τ_2 = 0,12	τ_2 = 0,17
	τ_3 = 0,09	τ_3 = 0,25
	τ_4 = 0,13	τ_4 = 0,22
	τ_5 = 0,16	τ_5 = 0,13
	τ_6 = -1,00	τ_6 = 0,10
VG-RAM WNN-COR	τ_1 = 0,00	τ_1 = 0,35
	τ_2 = 0,13	τ_2 = 0,23
	τ_3 = 0,15	τ_3 = 0,36
	τ_4 = 0,20	τ_4 = 0,23
	τ_5 = 0,00	τ_5 = -1,00
	τ_6 = -1,00	τ_6 = -1,00

6 RESULTADOS EXPERIMENTAIS

Neste capítulo, apresentamos nossos resultados experimentais. Avaliamos se o valor da medida de certeza de categorização, calculado analiticamente pelo nosso método por meio da regra de *Bayes*, é uma boa estimativa para o valor da medida de certeza de categorização observada empiricamente. Analisamos também o impacto da nossa estratégia de poda de *ranking* de categorias, BCut, e sua variante, PBCut, no desempenho das técnicas de categorização multi-rótulo de texto, k-vizinhos mais próximos multi-rótulo (multi-label k-nearest neighbors - *ML-k NN*) [Zhang07] e rede neural sem peso do tipo *VG-RAM* com correlação de dados (*data correlated virtual generalizing random access memory weightless neural networks – VG-RAM WNN-COR*) [Aleksander98, Badue08, DeSouza08, DeSouza09a, DeSouza09b], no contexto da categorização de descrições de atividades econômicas de empresas brasileiras segundo a Classificação Nacional de Atividades Econômicas (CNAE) [CNAE03]. Ademais, analisamos o impacto no desempenho do categorizadores *ML-k NN* e *VG-RAM WNN-COR* de três métodos de poda comumente usados na literatura de RI [Yang01, Lee02, Fan07]: (i) RCut, baseada na posição das categorias no *ranking*; (ii) PCut, baseada na popularidade das categorias no conjunto de treinamento; (iii) SCut, baseada no grau de crença com que o sistema atribui as categorias aos documentos; e (iv) suas variantes RTCut, proposta por Yang (2001), e PCut* e SCut*, propostas neste trabalho. O desempenho dos categorizadores foi medido em termos das métricas de avaliação de desempenho de categorização multi-rótulo de texto: *exact match* [Kazawa05], precision [Sebastiani02, Manning08], recall [Sebastiani02, Manning08], e F_1 [Sebastiani02, Manning08].

6.1 Validação da Medida de Certeza

A comparação entre os valores de $p(x|y,k)$ calculados analiticamente (por meio da regra de *Bayes* a partir das estimativas de $p(x|k)$, $p(y|k)$ e $p(y|x,k)$ obtidas nos experimentos de calibração) com os valores de $p(x|y,k)$ estimados empiricamente são apresentados nesta seção para o categorizador *VG-RAM WNN-COR* e a base de dados EX100. Os valores da medida de

certeza para *VG-RAM WNN-COR* e a base de dados AT100 e *ML-k NN* e as bases AT100 e EX100 são apresentados no APÊNDICE B.

A Tabela 6-1, Tabela 6-2, Tabela 6-3, Tabela 6-4 e Tabela 6-5 apresentam a comparação entre os valores de $p(x|y,k)$ calculados analiticamente (por meio da regra de *Bayes* a partir das estimativas de $p(x|k)$, $p(y|k)$ e $p(y|x,k)$ obtidas nos experimentos de validação) com os valores de $p(x|y,k)$ estimados empiricamente para o categorizador *VG-RAM WNN-COR* e a base EX100 a partir dos experimentos de teste para $k = 1, 2, 3, 4$ e 5, i.e., primeira, segunda, ..., e quinta posição do *ranking*, respectivamente. Nessas tabelas, a coluna Intervalo mostra cada um dos 20 intervalos de valores de f observados nos experimentos de validação, a coluna Validação mostra os valores de $p(x|y,k)$ calculados analiticamente por meio da regra de *Bayes* com os resultados dos experimentos de validação, e a coluna Teste mostra os valores de $p(x|y,k)$ estimados empiricamente a partir dos experimentos de teste.

Como pode ser observado na Tabela 6-1, Tabela 6-2, Tabela 6-3, Tabela 6-4 e Tabela 6-5, os valores de $p(x|y,k)$ calculados analiticamente por meio da regra de *Bayes* são muito próximos aos valores de $p(x|y,k)$ estimados empiricamente, o que demonstra que, usando nossa metodologia, é possível prever no teste com o último *fold* (não visto pelo *VG-RAM WNN-COR* durante o treinamento) o quão certo está o *VG-RAM WNN-COR* quanto à primeira categoria no seu *ranking* de saída ser pertinente para um dado documento. É importante destacar que esta medida de certeza vai de 0% a 100% – uma medida facilmente compreensível para um operador do SCAE humano.

O sistema SCAE usa a Tabela 6-1, Tabela 6-2, Tabela 6-3, Tabela 6-4 e Tabela 6-5 da seguinte forma. Se o *VG-RAM WNN-COR* predisse a categoria c_i para o documento d_j com grau de crença $f(d_j, c_i)$ dentro de um intervalo y (dentre os 20 intervalos observados na validação), e posicionou a categoria c_i na posição $r(d_j, c_i)$ do *ranking*, então a medida de certeza para essa predição pode ser expressa por $p(x|y,k)$, onde $y \subset f(d_j, c_i)$ e $k = r(d_j, c_i)$.

A tabela Tabela 6-1, Tabela 6-2, Tabela 6-3, Tabela 6-4 e Tabela 6-5 mostram os resultados do uso de nossa metodologia para valores de k iguais a 1, 2, 3, 4 e 5 respectivamente. Como pode ser visto nestas tabelas, também para estes valores de k é possível prever no teste com o último *fold* o quão certo está o *VG-RAM WNN-COR* quanto à categoria na posição k no seu *ranking* de saída ser pertinente para um dado documento. Note que, quanto maior o k (quanto mais abaixo no *ranking* de saída do categorizador), menos provável que a categoria atribuída pelo categorizador seja pertinente ao documento (ver

última linha das tabelas). Isso é esperado, já que, para a base de dados empregada no treinamento (EX100), é incomum existirem mais que dois códigos pertinentes a um dado documento.

Tabela 6-1 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=1 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,048906)	***p(x|y,k)***	0,346155	***p(x|y,k)***	0,291667
2	(0,048906 - 0,059188)	**p(x|y,k)**	0,530551	***p(x|y,k)***	0,400000
3	(0,059188 - 0,066578)	**p(x|y,k)**	0,581998	***p(x|y,k)***	0,625000
4	(0,066578 - 0,073284)	**p(x|y,k)**	0,672018	***p(x|y,k)***	0,500000
5	(0,073284 - 0,079183)	**p(x|y,k)**	0,742759	***p(x|y,k)***	0,434783
6	(0,079183 - 0,084823)	**p(x|y,k)**	0,742759	***p(x|y,k)***	0,692308
7	(0,084823 - 0,090226)	**p(x|y,k)**	0,723465	***p(x|y,k)***	0,578947
8	(0,090226 - 0,095975)	**p(x|y,k)**	0,784559	***p(x|y,k)***	0,480000
9	(0,095975 - 0,101643)	**p(x|y,k)**	0,819937	***p(x|y,k)***	0,814815
10	(0,101643 - 0,108434)	**p(x|y,k)**	0,778138	***p(x|y,k)***	0,774194
11	(0,108434 - 0,115315)	**p(x|y,k)**	0,816727	***p(x|y,k)***	0,838710
12	(0,115315 - 0,122085)	**p(x|y,k)**	0,836006	***p(x|y,k)***	0,880952
13	(0,122085 - 0,130695)	**p(x|y,k)**	0,791657	***p(x|y,k)***	0,750000
14	(0,130695 - 0,141219)	**p(x|y,k)**	0,861300	***p(x|y,k)***	0,894737
15	(0,141219 - 0,153803)	**p(x|y,k)**	0,874595	***p(x|y,k)***	0,837838
16	(0,153803 - 0,171683)	**p(x|y,k)**	0,900326	***p(x|y,k)***	0,878049
17	(0,171683 - 0,198192)	**p(x|y,k)**	0,887452	***p(x|y,k)***	0,882353
18	(0,198192 - 0,232168)	**p(x|y,k)**	0,958194	***p(x|y,k)***	0,783784
19	(0,232168 - 0,319712)	**p(x|y,k)**	0,967841	***p(x|y,k)***	0,943396
20	(0,319712 - 1,000000)	**p(x|y,k)**	0,970967	***p(x|y,k)***	0,933333

Tabela 6-2 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=2 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,022611)	**p(x|y,k)**	0,009616	***p(x|y,k)***	0,031250
2	(0,022611 - 0,027632)	**p(x|y,k)**	0,019293	***p(x|y,k)***	0,050000
3	(0,027632 - 0,032089)	**p(x|y,k)**	0,054660	***p(x|y,k)***	0,037037
4	(0,032089 - 0,036620)	**p(x|y,k)**	0,090033	***p(x|y,k)***	0,031250
5	(0,036620 - 0,040404)	**p(x|y,k)**	0,121794	***p(x|y,k)***	0,080000
6	(0,040404 - 0,044269)	**p(x|y,k)**	0,145161	***p(x|y,k)***	0,115385
7	(0,044269 - 0,047990)	**p(x|y,k)**	0,170420	***p(x|y,k)***	0,000000
8	(0,047990 - 0,052008)	**p(x|y,k)**	0,225080	***p(x|y,k)***	0,250000
9	(0,052008 - 0,055889)	**p(x|y,k)**	0,292606	***p(x|y,k)***	0,172414
10	(0,055889 - 0,060299)	**p(x|y,k)**	0,270093	***p(x|y,k)***	0,250000
11	(0,060299 - 0,064851)	**p(x|y,k)**	0,331186	***p(x|y,k)***	0,270270
12	(0,064851 - 0,069875)	**p(x|y,k)**	0,385853	***p(x|y,k)***	0,400000
13	(0,069875 - 0,074904)	**p(x|y,k)**	0,401926	***p(x|y,k)***	0,281250
14	(0,074904 - 0,081272)	**p(x|y,k)**	0,472666	***p(x|y,k)***	0,420000
15	(0,081272 - 0,087618)	**p(x|y,k)**	0,479100	***p(x|y,k)***	0,375000
16	(0,087618 - 0,095580)	**p(x|y,k)**	0,572346	***p(x|y,k)***	0,500000
17	(0,095580 - 0,105363)	**p(x|y,k)**	0,591639	***p(x|y,k)***	0,718750
18	(0,105363 - 0,120024)	**p(x|y,k)**	0,601286	***p(x|y,k)***	0,589744
19	(0,120024 - 0,152786)	**p(x|y,k)**	0,559486	***p(x|y,k)***	0,564103
20	(0,152786 - 1,000000)	**p(x|y,k)**	0,712908	***p(x|y,k)***	0,750000

Tabela 6-3 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=3 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,017595)	**p(x|y,k)**	0,006411	***p(x|y,k)***	0,142857
2	(0,017595 - 0,020573)	**p(x|y,k)**	0,009648	***p(x|y,k)***	0,038462
3	(0,020573 - 0,023336)	**p(x|y,k)**	0,016076	***p(x|y,k)***	0,000000
4	(0,023336 - 0,025939)	**p(x|y,k)**	0,016076	***p(x|y,k)***	0,035714
5	(0,025939 - 0,028364)	**p(x|y,k)**	0,032155	***p(x|y,k)***	0,034483
6	(0,028364 - 0,030728)	**p(x|y,k)**	0,045016	***p(x|y,k)***	0,035714
7	(0,030728 - 0,033007)	**p(x|y,k)**	0,051448	***p(x|y,k)***	0,000000
8	(0,033007 - 0,035250)	**p(x|y,k)**	0,045016	***p(x|y,k)***	0,045455
9	(0,035250 - 0,037608)	**p(x|y,k)**	0,061092	***p(x|y,k)***	0,000000
10	(0,037608 - 0,040024)	**p(x|y,k)**	0,083600	***p(x|y,k)***	0,038462
11	(0,040024 - 0,042752)	**p(x|y,k)**	0,115756	***p(x|y,k)***	0,030303
12	(0,042752 - 0,045650)	**p(x|y,k)**	0,131832	***p(x|y,k)***	0,000000
13	(0,045650 - 0,048971)	**p(x|y,k)**	0,109324	***p(x|y,k)***	0,162162
14	(0,048971 - 0,052439)	**p(x|y,k)**	0,183280	***p(x|y,k)***	0,111111
15	(0,052439 - 0,057078)	**p(x|y,k)**	0,205788	***p(x|y,k)***	0,230769
16	(0,057078 - 0,061834)	**p(x|y,k)**	0,205788	***p(x|y,k)***	0,189189
17	(0,061834 - 0,068206)	**p(x|y,k)**	0,285254	***p(x|y,k)***	0,266667
18	(0,068206 - 0,077441)	**p(x|y,k)**	0,351615	***p(x|y,k)***	0,156250
19	(0,077441 - 0,092551)	**p(x|y,k)**	0,437299	***p(x|y,k)***	0,357143
20	(0,092551 - 1,000000)	**p(x|y,k)**	0,467745	***p(x|y,k)***	0,562500

Tabela 6-4 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=4 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,014855)	**p(x|y,k)**	0,000000	***p(x|y,k)***	0,000000
2	(0,014855 - 0,017219)	**p(x|y,k)**	0,009647	***p(x|y,k)***	0,000000
3	(0,017219 - 0,019100)	**p(x|y,k)**	0,003205	***p(x|y,k)***	0,000000
4	(0,019100 - 0,020833)	**p(x|y,k)**	0,006411	***p(x|y,k)***	0,000000
5	(0,020833 - 0,022457)	**p(x|y,k)**	0,012944	***p(x|y,k)***	0,000000
6	(0,022457 - 0,024081)	**p(x|y,k)**	0,009616	***p(x|y,k)***	0,000000
7	(0,024081 - 0,025773)	**p(x|y,k)**	0,012861	***p(x|y,k)***	0,041667
8	(0,025773 - 0,027458)	**p(x|y,k)**	0,019355	***p(x|y,k)***	0,000000
9	(0,027458 - 0,029207)	**p(x|y,k)**	0,028846	***p(x|y,k)***	0,037037
10	(0,029207 - 0,030888)	**p(x|y,k)**	0,051447	***p(x|y,k)***	0,034483
11	(0,030888 - 0,032725)	**p(x|y,k)**	0,061290	***p(x|y,k)***	0,085714
12	(0,032725 - 0,034578)	**p(x|y,k)**	0,038585	***p(x|y,k)***	0,033333
13	(0,034578 - 0,036846)	**p(x|y,k)**	0,048232	***p(x|y,k)***	0,064516
14	(0,036846 - 0,039273)	**p(x|y,k)**	0,115015	***p(x|y,k)***	0,093750
15	(0,039273 - 0,041846)	**p(x|y,k)**	0,087379	***p(x|y,k)***	0,062500
16	(0,041846 - 0,044872)	**p(x|y,k)**	0,128204	***p(x|y,k)***	0,096774
17	(0,044872 - 0,048928)	**p(x|y,k)**	0,125806	***p(x|y,k)***	0,131579
18	(0,048928 - 0,054723)	**p(x|y,k)**	0,183279	***p(x|y,k)***	0,062500
19	(0,054723 - 0,064407)	**p(x|y,k)**	0,227563	***p(x|y,k)***	0,210526
20	(0,064407 - 1,000000)	**p(x|y,k)**	0,391589	***p(x|y,k)***	0,440000

Tabela 6-5 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=5 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,013276)	**p(x|y,k)**	0,003205	***p(x|y,k)***	0,027027
2	(0,013276 - 0,015484)	**p(x|y,k)**	0,000000	***p(x|y,k)***	0,000000
3	(0,015484 - 0,016903)	**p(x|y,k)**	0,012862	***p(x|y,k)***	0,000000
4	(0,016903 - 0,018129)	**p(x|y,k)**	0,003216	***p(x|y,k)***	0,000000
5	(0,018129 - 0,019425)	**p(x|y,k)**	0,006431	***p(x|y,k)***	0,000000
6	(0,019425 - 0,020612)	**p(x|y,k)**	0,012862	***p(x|y,k)***	0,034483
7	(0,020612 - 0,021800)	**p(x|y,k)**	0,016077	***p(x|y,k)***	0,000000
8	(0,021800 - 0,023065)	**p(x|y,k)**	0,000000	***p(x|y,k)***	0,030303
9	(0,023065 - 0,024476)	**p(x|y,k)**	0,009616	***p(x|y,k)***	0,000000
10	(0,024476 - 0,025764)	**p(x|y,k)**	0,012862	***p(x|y,k)***	0,000000
11	(0,025764 - 0,027174)	**p(x|y,k)**	0,038586	***p(x|y,k)***	0,035714
12	(0,027174 - 0,028626)	**p(x|y,k)**	0,022581	***p(x|y,k)***	0,068966
13	(0,028626 - 0,030175)	**p(x|y,k)**	0,028939	***p(x|y,k)***	0,034483
14	(0,030175 - 0,031882)	**p(x|y,k)**	0,035370	***p(x|y,k)***	0,125000
15	(0,031882 - 0,033888)	**p(x|y,k)**	0,051282	***p(x|y,k)***	0,045455
16	(0,033888 - 0,036267)	**p(x|y,k)**	0,048388	***p(x|y,k)***	0,060606
17	(0,036267 - 0,039041)	**p(x|y,k)**	0,067308	***p(x|y,k)***	0,142857
18	(0,039041 - 0,042965)	**p(x|y,k)**	0,080646	***p(x|y,k)***	0,103448
19	(0,042965 - 0,050193)	**p(x|y,k)**	0,122187	***p(x|y,k)***	0,075472
20	(0,050193 - 1,000000)	**p(x|y,k)**	0,248390	***p(x|y,k)***	0,294118

6.2 Comparação entre as Estratégias de Poda

Nesta seção, apresentamos a formulação das métricas de avaliação de desempenho de categorização multi-rótulo de texto empregadas neste trabalho: *exact match* [Kazawa05], *precision* [Sebastiani02, Manning08], *recall* [Sebastiani02, Manning08] e F_β [Sebastiani02, Manning08]. Estas métricas avaliam o conjunto exato de categorias $\hat{C}_j$, predito para o documento de teste d_j.

6.2.1 Exact Match

A métrica ***exact match*** ($exact - match_j$) avalia o quão freqüente todas e somente todas as categorias pertinentes estão presentes no conjunto de categorias preditas de d_j. A formulação original de Kazawa et al. [Kazawa05] é apresentada na Equação (**(6.3)**.

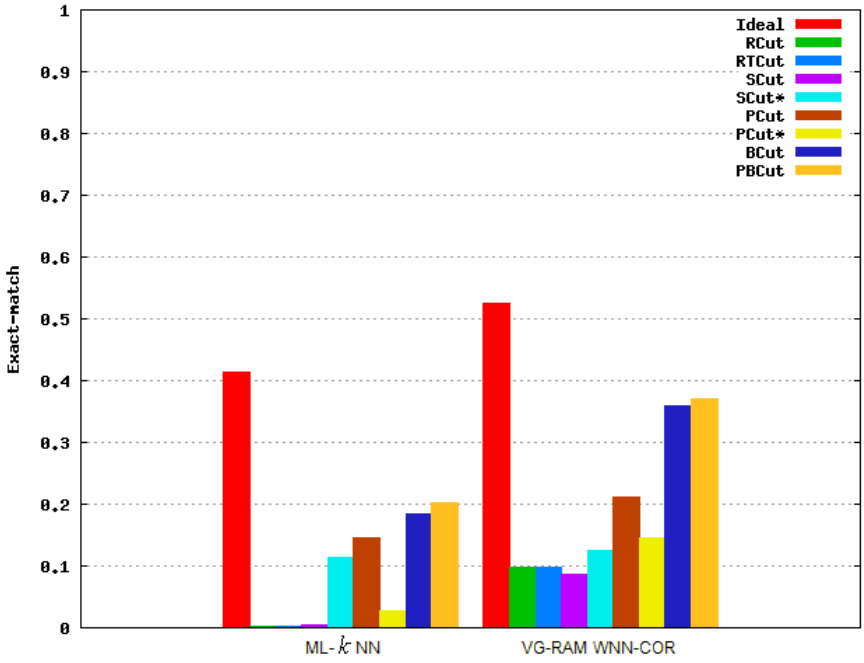
$$exact-macth_j = \begin{cases} 1 \;\; se \;\; \hat{C}_j^{|C_j|} = C_j \\ 0 \;\; caso \;\; contrário \end{cases} \tag{6.3}$$

Se o conjunto $\hat{C}_j^{|C_j|}$ é igual ao conjunto C_j, $exact-macth_j = 1$, caso contrário, $exact-macth_j = 0$. O desempenho global é obtido conforme Equação ((**6.1)**. Quanto maior o valor de *exact match*, melhor o desempenho do categorizador. O desempenho é perfeito quando $exact-match = 1$.

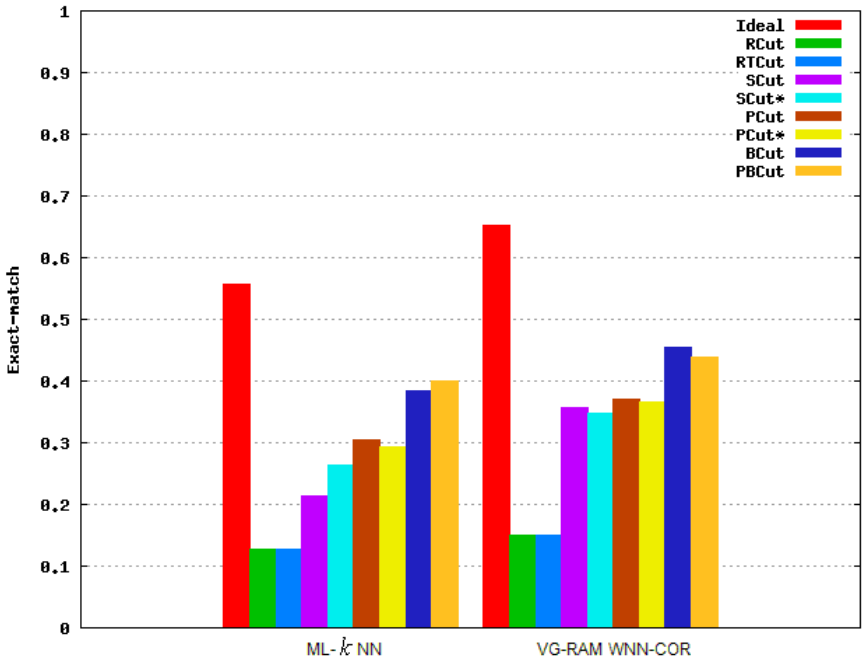
$$exact-match = \frac{1}{|Te|}\sum_{j=1}^{|Te|} exact-match_j \tag{6.1}$$

A Figura 6-1 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica *exact match* segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para as bases AT100 e EX100. No gráfico apresentado na Figura 6-1, existem nove conjuntos de barras, um para cada categorizador empregado, onde a amplitude de cada barra indica o valor de *exact match* (média dos 10 *folds*) para cada uma das diversas estratégias de poda de *ranking* de categorias. Em cada conjunto de barras, da esquerda para a direita, a primeira barra indica a poda Ideal[1] do *ranking*, a segunda a estratégia de poda RCut, a terceira a estratégia de poda RTCut, a quarta a estratégia de poda SCut, a quinta a estratégia de poda SCut*, a sexta a estratégia de poda PCut, a sétima a estratégia de poda PCut*, a oitava a estratégia de poda BCut, a nona a estratégia de poda PBCut (ver legenda nos gráficos da figura).

[1] A estratégia de poda de *ranking* Ideal retorna para um dado documento d_j as $|C_j|$ categorias do topo do *ranking*, ou seja, a quantidade de categorias retornadas $|\hat{C}_j|$ é igual a quantidade de categorias realmente pertinente $|C_j|$.



(a)



(b)

Figura 6-1 - Resultado da métrica *exact-match* para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-1 mostram, o valor de *exact match* do categorizador *ML-k NN* com a base AT100 (Figura 6-1(a)) é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de *exact match* ao usar a estratégia de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut* SCut e SCut*. O mesmo ocorre com o

categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-1(a)). O valor de *exact match* com a estratégia de poda SCut* é significativamente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-1(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-1(a)). O valor de *exact match* com a estratégia de poda PCut* é significativamente menor do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-1(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-1(a)).

Como apresentado na Figura 6-1, as barras do gráfico mostram o valor de *exact match* do categorizador *ML-k NN* com a base de dados EX100 (Figura 6-1(b)) é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de *exact match* ao usar as estratégias de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut* SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-1(b)). O valor de *exact match* com a estratégia de poda SCut* é significativamente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-1(b)). O valor de *exact match* com a estratégia de poda SCut* é menor do que com a estratégia de poda SCut (tradicional) com o categorizador *VG-RAM WNN-COR* com a base EX100 (Figura 6-1(b)). O valor de *exact match* com a estratégia de poda PCut* é menor que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-1(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica *exact match* mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para ambas bases, como visto na Figura 6-1(a)(b). Isso acontece porque a estratégia BCut poda o *ranking* com base na probabilidade da categorização estar correta e a estratégia PBCut poda o *ranking* com base na probabilidade da categorização estar correta em cada posição do *ranking*, observando o decremento da medida de certeza a medida que a posição da categoria no *ranking* aumenta. Desta forma, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o desempenho do sistema em termos da *exact match* do que as estratégias RCut, RTCut, PCut, PCut* SCut e SCut*, pois conseguem podar o *ranking* de maneira a retornar um conjunto de categorias idênticos ao conjunto de categorias pertinente.

6.2.2 Precisão (*precision*) Orientada à Categoria

A métrica ***precisão (precision)* orientada à categoria** ($precision_i^c$) avalia a fração de documentos de teste categorizados sob a categoria c_i que são verdadeiramente associados a c_i . A formulação é apresentada na Equação (6.2).

$$precision_i^c = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|\hat{C}_j^{|C_j|}|} \qquad 6.2)$$

A métrica *precision* orientada à categoria também pode ser computada utilizando a tabela de contingência da categoria c_i (Tabela 6-6), de acordo com a Equação (6.3).

$$precision_i^c = \frac{TP_i}{TP_i + FP_i} \qquad 6.3)$$

onde FP_i (falsos positivos para c_i) é o número de documentos de teste que foram incorretamente categorizados sob c_i ; TN_i (verdadeiros negativos para c_i) é o número de documentos de teste que foram corretamente não categorizados sob c_i ; TP_i (verdadeiros positivos para c_i) é o número de documentos de teste que foram corretamente categorizados sob c_i ; e FN_i (falsos negativos para c_i) é o número de documentos de teste que foram incorretamente não categorizados sob c_i .

Tabela 6-6 – Tabela de contingência da categoria c_i .

Categoria c_i		**Julgamentos do especialista**	
		SIM	NÃO
Julgamentos do categorizador	SIM	TP_i	FP_i
	NÃO	FN_i	TN_i

O desempenho global de *precision* orientada à categoria pode ser computado pelo método *macroaveraging* ($macro-precision^c$) e *microaveraging* ($micro-precision^c$), Equação (6.4) e Equação (6.5), respectivamente [Sebatiani2002]. O método *macroaveraging* reporta o desempenho global sobre a soma dos resultados de $precision_i^c$ (Equação (6.4)), e o *microaveraging* sobre a soma das decisões individuais em termos da tabela de contingência, $TP_i \big/ (TP_i + FP_i)$ (Equação (6.5)), para cada categoria c_i.

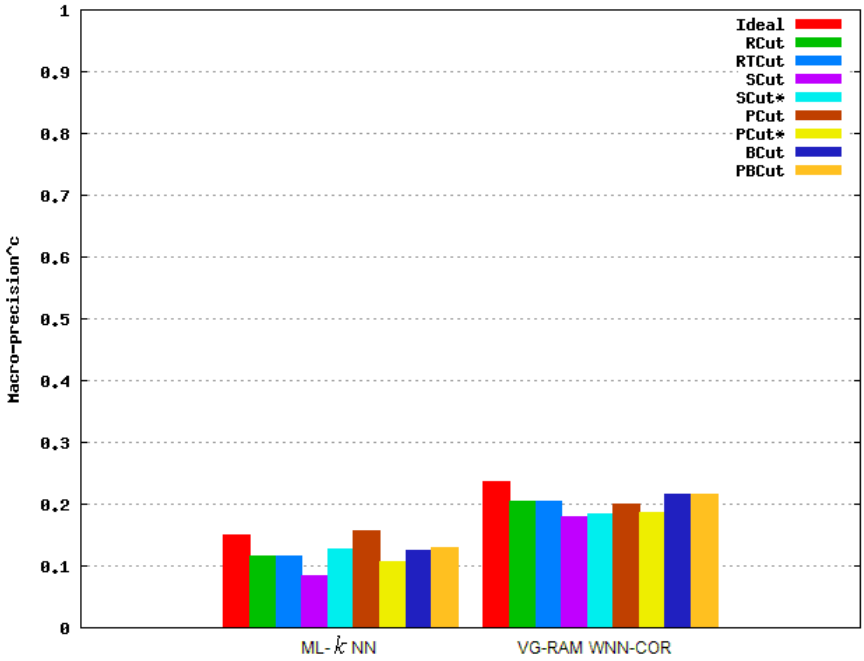
$$macro-precision^c = \frac{\sum_{i=1}^{|C|} precision_i^c}{|C|} \qquad 6.4)$$

$$micro-precision^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FP_i)} \qquad 6.5)$$

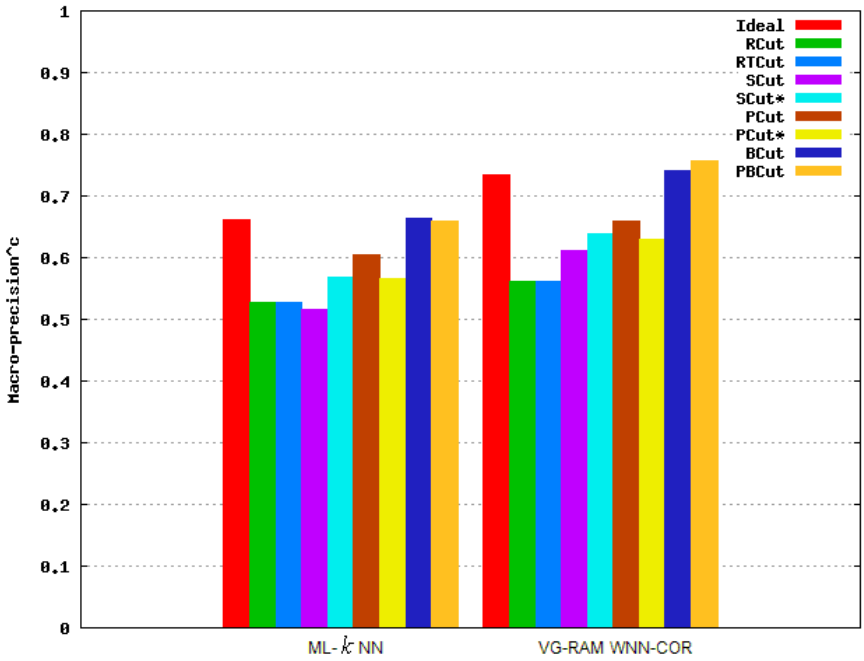
Os métodos *macroaveraging* e *microaveraging* podem dar resultados bastante diferentes, especialmente se as generalidades das categorias são desiguais [Manning08; Sebastiani02]. A habilidade de um categorizador de se comportar bem mediante categorias com baixa generalidade é evidenciada muito mais por *macroaveraging* e do que por *microaveraging*. O método *macroaveraging* dá peso igual para cada categoria, enquanto *microaveraging* dá peso igual para cada decisão de categorização [Manning08]. Desta forma, categorias com alta generalidade dominam aquelas com baixa generalidade em *microaveraging*.

Quanto maior o valor de $macro-precision^c$ e $micro-precision^c$ melhor o desempenho do categorizador. O desempenho é perfeito quando $macro-precision^c = 1$ e $micro-precision^c = 1$.

A Figura 6-2 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $macro-precision^c$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-2(a)) e EX100 (Figura 6-2(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-2 - Resultado da métrica $macro-precision^c$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-2 mostram, o valor de $macro-precision^c$ do categorizador *ML-k NN* com a base AT100 (Figura 6-2(a)) é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor da $macro-precision^c$ ao usar a estratégia de poda de *ranking* PCut é significativamente maior do que os resultados obtidos ao

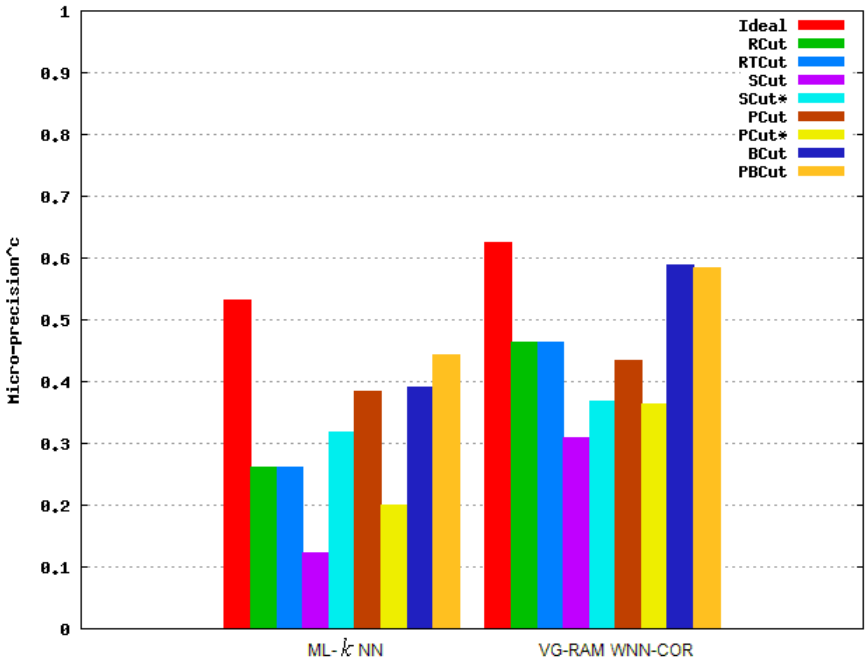
usar as estratégias RCut, RTCut, PCut*, SCut, SCut*, BCut e PBCut. O resultados obtidos com o categorizador *VG-RAM WNN* com a base de dados AT100 (Figura 6-2(a)) mostram um equilíbrio parcial entre as estratégias de poda RCut, RTCut, BCut e PBCut. O valor de $macro-precision^c$ com a estratégia de poda SCut* é significativamente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-2(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-2(a)). O valor de $macro-precision^c$ com a estratégia de poda PCut* é significativamente menor do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-2(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-2(a)).

Como o apresentado na Figura 6-2, as barras do gráfico mostram o valor de $macro-precision^c$ do categorizador ML-*k* NN com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro-precision^c$ ao usar as estratégias de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-2(a)). O valor de $macro-precision^c$ com a estratégia de poda SCut* é significativamente maior que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-2 (b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-2(b)). O valor de $macro-precision^c$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-2(B)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-2 (b)).

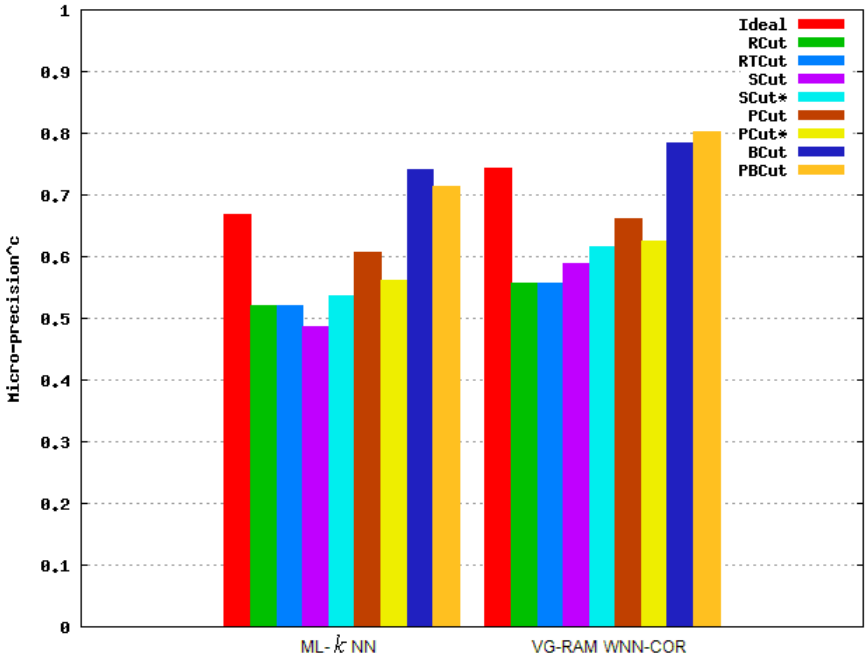
A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $macro-precision^c$ mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para a base de dados EX100, como visto na, Figura 6-2(b). Isso acontece porque a estratégia BCut poda o *ranking* com base na probabilidade da categorização estar correta e a estratégia PBCut poda o *ranking* com base na probabilidade da categorização estar correta em cada posição do *ranking*, observando o decremento da medida de certeza a medida que a posição da categoria no *ranking* aumenta. Desta forma, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o

desempenho do sistema em termos da métrica $macro-precision^c$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut* para estes categorizadores e para a base EX100.

A Figura 6-3 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $micro-precision^c$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-3(a)) e EX100 (Figura 6-3(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-3 - Resultado da métrica $micro-precision^c$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-3 mostram, o valor de $micro-precision^c$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. Os valores de $micro-precision^c$ ao usar a estratégia de poda de *ranking* BCut e PBCut são maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN* com a base de dados AT100 (Figura 6-3(a)). O valor de $micro-precision^c$ com a estratégia de poda SCut* é significativamente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-3(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-3(a)). O valor de $micro-precision^c$ com a estratégia de poda PCut* é significativamente menor do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-3(a)).

Com apresentado na Figura 6-3, as barras do gráfico mostram o valor de $micro-precision^c$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-precision^c$ ao usar as estratégias de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-3(b)). O valor de $micro-precision^c$ com a estratégia de poda SCut* é significativamente maior di que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-3(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-3(b)). O valor de $micro-precision^c$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-3(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $micro-precision^c$ mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para ambas bases, como visto na Figura 6-3(a)(b). Isso acontece porque a estratégia BCut poda o *ranking* com base na probabilidade da categorização estar correta e a estratégia PBCut poda o *ranking* com base na

probabilidade da categorização estar correta em cada posição do *ranking*, observando o decremento da medida de certeza a medida que a posição da categoria no *ranking* aumenta. Desta forma, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o desempenho do sistema em termos da métrica $micro - precision^c$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*.

6.2.3 Revocação (*recall*) Orientada à Categoria

A métrica **revocação *(recall)* orientada à categoria (**$recall_i^c$**)** avalia a fração de documentos de teste verdadeiramente associados com a categoria c_i que são categorizados sob c_i. A formulação original é apresentada na Equação (6.6).

$$recall_i^c = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|C_j|} \tag{6.6}$$

O valor de $recall_i^c$ também pode ser computado em termos da tabela de contingência da categoria c_i, Tabela 6-6, conforme Equação (6.7).

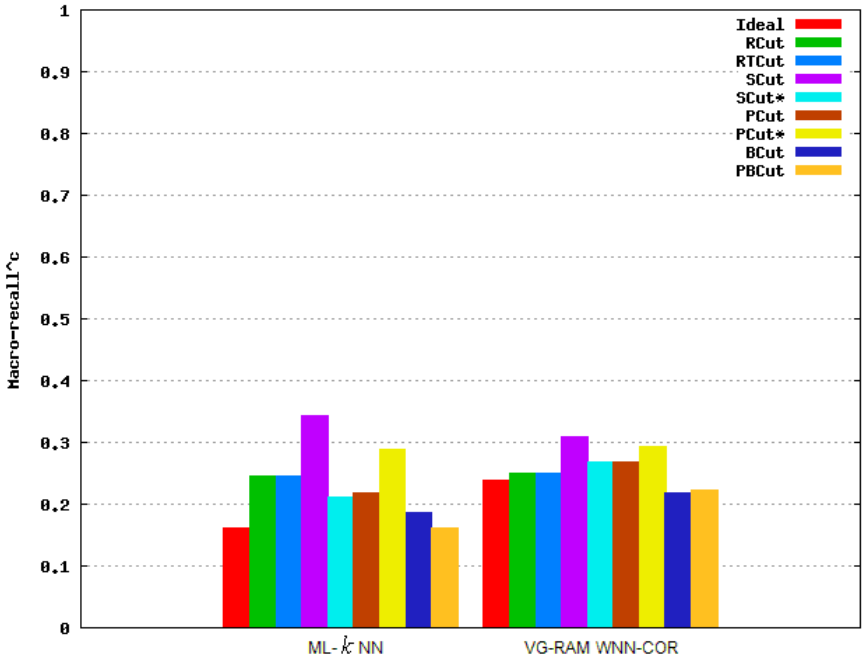
$$recall_i^c = \frac{TP_i}{TP_i + FN_i} \tag{6.7}$$

O desempenho global de recall orientada à categoria é calculado pelos métodos $macro - recall^c$ e $micro - recall^c$, Equação (6.8) e Equação (6.9), respectivamente. Quanto maior o valor de $macro - recall^c$ e $micro - recall^c$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - recall^c = 1$ e $micro - recall^c = 1$.

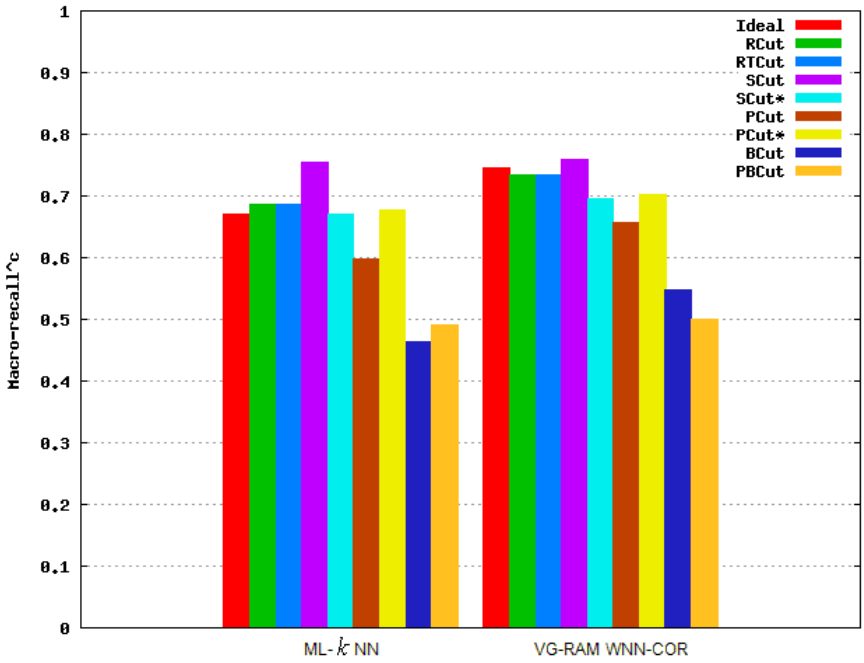
$$macro - recall^c = \frac{\sum_{i=1}^{|C|} recall_i^c}{|C|} \tag{6.8}$$

$$micro - recall^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|} (TP_i + FN_i)} \quad (6.9)$$

A Figura 6-4 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $macro - recall^c$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-4(a)) e EX100 (Figura 6-4(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-4 - Resultado da métrica $macro-recall^c$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

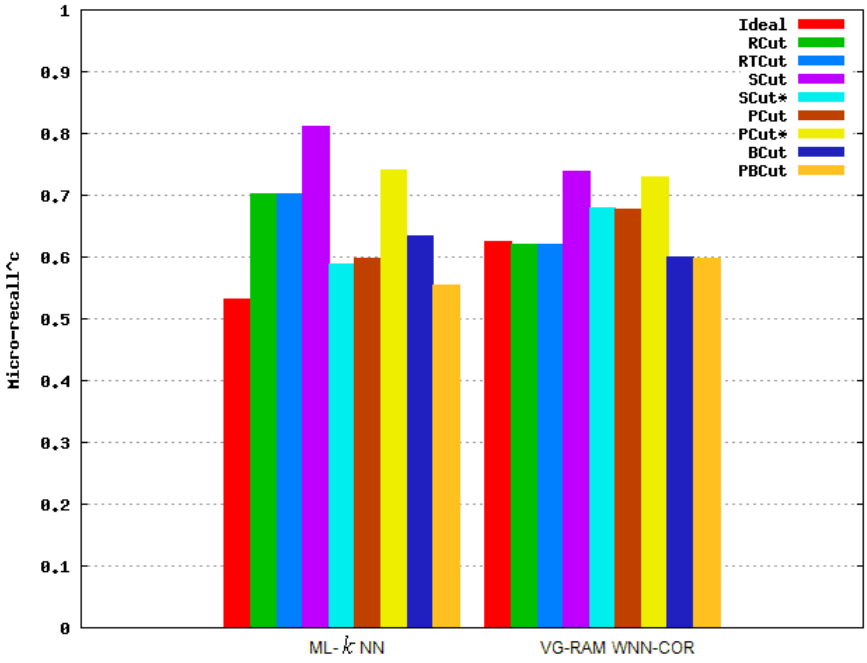
Conforme as barras do gráfico da Figura 6-4 mostram, o valor de $macro-recall^c$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro-recall^c$ ao usar a estratégia de poda de *ranking* SCut é significativamente maior do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut. O mesmo ocorre com o categorizador *VG-RAM*

WNN com a base de dados AT100 (Figura 6-4(a)). O valor de $macro-recall^c$ com a estratégia de poda SCut* é significativamente menor do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-4(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-4(a)). O valor de $macro-recall^c$ com a estratégia de poda PCut* é significativamente maior do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-4(a)).

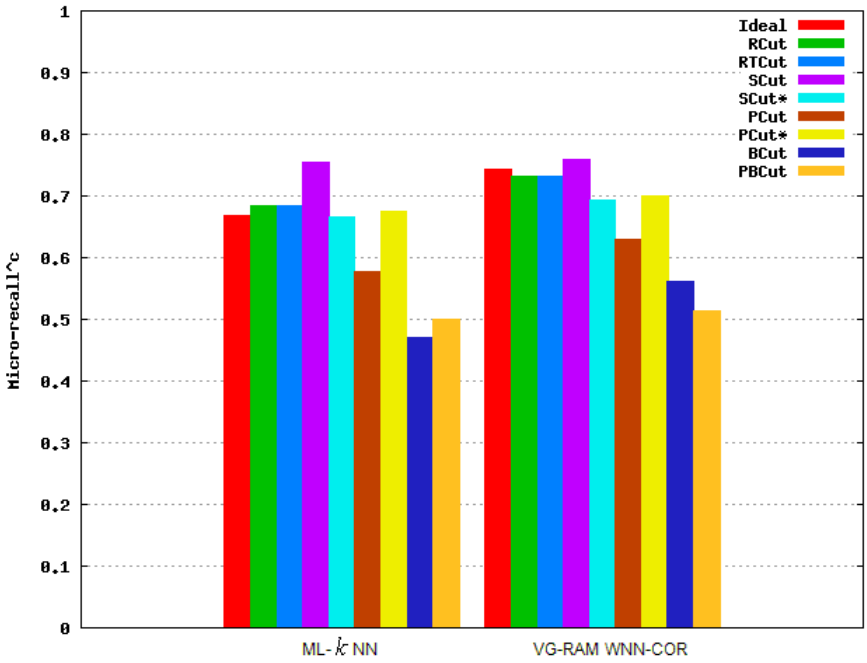
Com o apresentado na Figura 6-4, as barras do gráfico mostram o valor de $macro-recall^c$ do categorizador ML-*k* NN com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro-recall^c$ ao usar a estratégia de poda de *ranking* SCut é significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut. O valor de $macro-recall^c$ com a estratégia de poda SCut* é significativamente menor do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-4(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-4(b)). O valor de $macro-recall^c$ com a estratégia de poda PCut* é maior mdo que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-4(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $macro-recall^c$ mostram que a estratégia de poda SCut melhora o desempenho desses categorizadores para ambas bases, como visto na Figura 6-4 (a)(b). Isso acontece porque SCut otimiza o desempenho do categorizador por categoria, tornando-a particularmente efetiva quando o desempenho do sistema em categorias raras é a função alvo a ser otimizada. Desta forma, a estratégia de poda SCut é mais efetiva para otimizar o desempenho do sistema em termos da métrica $macro-recall^c$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut.

A Figura 6-5 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $micro-recall^c$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-5(a)) e EX100 (Figura 6-5(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-5 - Resultado da $micro-recall^c$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-5 mostram, o valor de $micro-recall^c$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-recall^c$ ao usar a estratégia de poda de *ranking* SCut é significativamente maior do que os resultados obtidos ao usar as estratégias RCut,

RTCut, PCut, PCut*, SCut*, BCut e PBCut. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-5(a)). O valor de $micro-recall^c$ com a estratégia de poda SCut* é significativamente menor do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-4(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-4(a)). O valor de $micro-recall^c$ com a estratégia de poda PCut* com o categorizador *ML-k NN* com a base AT100 (Figura 6-4(a)) é significativamente maior do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-4(a)).

Com o apresentado na Figura 6-5, as barras do gráfico mostram o valor de $micro-recall^c$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-recall^c$ ao usar a estratégia de poda de *ranking* SCut é significativamente maior do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-5(b)). O valor de $micro-recall^c$ com a estratégia de poda SCut* é significativamente menor do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* e com a base EX100 (Figura 6-4(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-4(a)). O valor de $micro-recall^c$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-4(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-4(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $micro-recall^c$ mostram que a estratégia de poda SCut melhora o desempenho desses categorizadores para ambas bases, como visto na, Figura 6-5 (a)(b). Desta forma, a estratégia de poda SCut é mais efetiva para otimizar o desempenho do sistema em termos da métrica $micro-recall^c$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut.

.

6.2.4 F_β Orientada à Categoria

A métrica F_β **orientada à categoria (** $F_{\beta_i}^{c}$ **)** avalia a média harmônica ponderada de $precision_i^c$ e $recall_i^c$**.** A formulação original de Rijsbergen [Rijsbergen79] é mostrada na Equação (6.10).

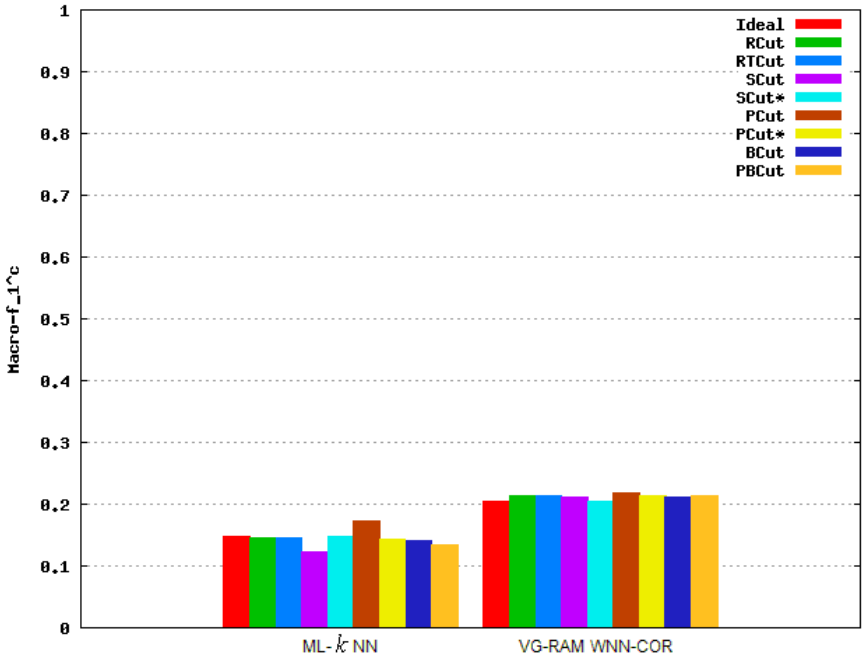
$$F_{\beta_i}^{c} = \frac{(\beta^2 + 1) * precision_i^c * recall_i^c}{\beta^2 * precision_i^c + recall_i^c} \tag{6.10}$$

Na Equação (6.10), β pode ser visto como o grau relativo de importância atribuído para $precision_i^c$ e $recall_i^c$ [Sebastiani02]. Se $\beta = 0$, $F_{\beta_i}^{c}$ coincide com $precision_i^c$; $\beta = +\infty$, $F_{\beta_i}^{c}$ coincide com $recall_i^c$. Neste trabalho um valor de $\beta = 1$ é utilizado, atribuindo importância igual para $precision_i^c$ e $recall_i^c$. O desempenho global de F_1^c pode ser computado tanto por $macro - F_1^c$ (Equação (6.11)) quanto $micro - F_1^c$ (Equação (6.12)). Quanto maior o valor de $macro - F_1^c$ e $micro - F_1^c$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - F_1^c = 1$ e $micro - F_1^c = 1$.

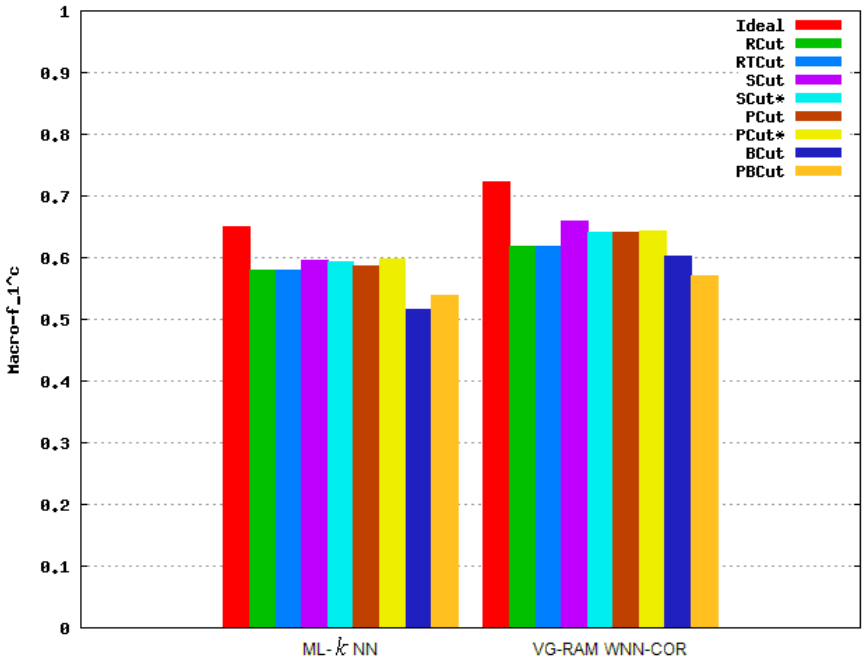
$$macro - F_1^c = \frac{1}{|C|}\sum_{i=1}^{C} F_{1i}^{c} \tag{6.11}$$

$$micro - F_1^c = \frac{2 * micro - precision_i^c * micro - recall_i^c}{micro - precision_i^c + micro - recall_i^c} \tag{6.12}$$

A Figura 6-6 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $macro - F_1^c$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-6(a)) e EX100 (Figura 6-6(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-6 - Resultado da métrica $macro - F_1^c$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-6 mostram, o valor de $macro - F_1^c$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro - F_1^c$ do categorizador *ML-k NN* com a base AT100 ao usar a estratégia de poda PCut é levemente maior do que os resultados obtidos ao usar as

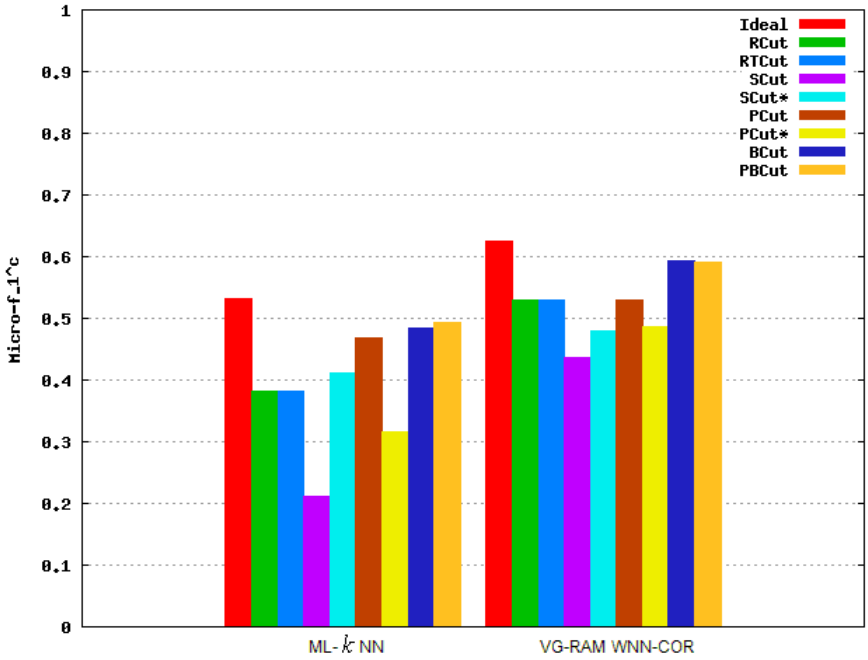
estratégias RCut, RTCut, PCut*, SCut, SCut*, BCut e PBCut. O categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-6(a)) apresenta o valor de $macro-F_1^c$ equilibrado ao usar as estratégias de poda de *ranking* RCut, RTCut, SCut, SCut*, PCut, PCut*, BCut e PBCut. O valor de $macro-F_1^c$ com a estratégia de poda SCut* é maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-6(a)). O valor de $macro-F_1^c$ com a estratégia de poda SCut* é menor do que com a estratégia de poda SCut (tradicional) com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-6(a)). O valor de $macro-F_1^c$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-6(a)).

Com o apresentado na Figura 6-6, as barras do gráfico mostram o valor de $macro-F_1^c$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O categorizador *ML-k NN* com a base de dados EX100 (Figura 6-6(b)) apresenta o valor de $macro-F_1^c$ levemente equilibrado ao usar as estratégias de poda de *ranking* RCut, RTCut, SCut, SCut*, PCut e PCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-6(b)) para as estratégias de poda SCut, SCut*, PCut e PCut*, apesar da estratégia SCut ser parcialmente melhor que as demais estratégias. O valor de $macro-F_1^c$ com a estratégia de poda SCut* é menor do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-6(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-6(b)). O valor de $macro-F_1^c$ com a estratégia de poda PCut* é parcialmente maior do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-6(b)).

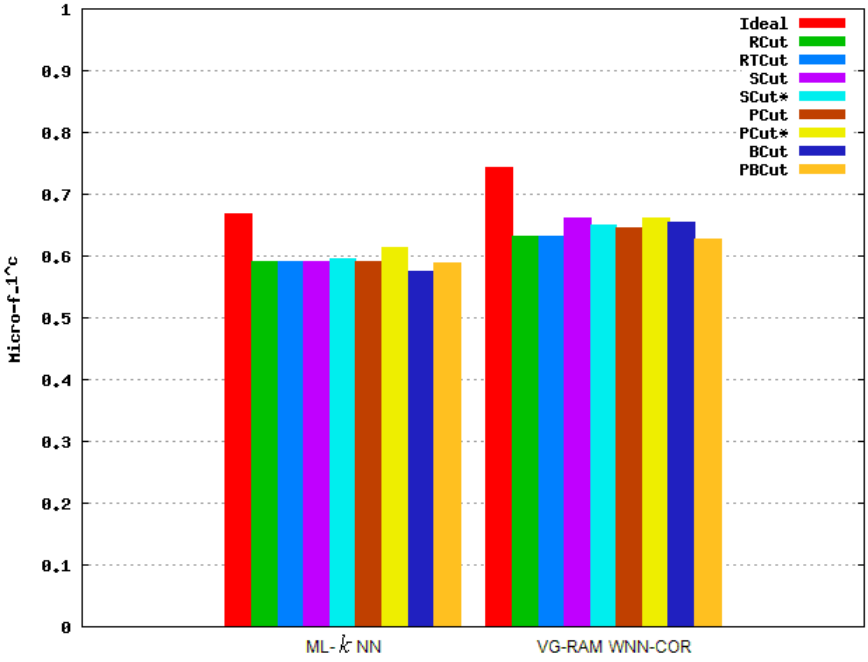
A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $macro-F_1^c$ mostra que não existe uma estratégia de poda que apresente um desempenho mais efetiva para estes categorizadores e para as bases de dados AT100 e EX100 de forma geral, como visto na, Figura 6-6(a)(b). Apesar disso, a estratégia de poda SCut mostrasse mais regular ao ser avaliada sob a métrica $macro-F_1^c$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut.

A Figura 6-7 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $micro-F_1^c$ segundo os categorizadores *ML-*

k NN e *VG-RAM WNN-COR* para a base AT100 (Figura 6-7(a)) e EX100 (Figura 6-7(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-7 - Resultado da métrica $micro - F_1^c$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-7 mostram, o valor de $micro - F_1^c$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no

ranking de categorias. O valor de $micro - F_1^c$ ao usar as estratégias de poda de *ranking* BCut e PBCut são maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-7(a)). O valor de $micro - F_1^c$ com a estratégia de poda SCut* é maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-7(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-7(a)). O valor de $micro - F_1^c$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-7(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-7(a)).

Com o apresentado na Figura 6-7, as barras do gráfico mostram o valor de $micro - F_1^c$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O categorizador *ML-k NN* com a base de dados EX100 (Figura 6-7(b)) apresenta o valor de $micro - F_1^c$ equilibrado ao usar as estratégias de poda de *ranking* RCut, RTCut, SCut, SCut*, PCut e PCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-7(b)). O valor de $micro - F_1^c$ com a estratégia de poda SCut* é parcialmente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-7(b)). O valor de $micro - F_1^c$ com a estratégia de poda SCut* é menor que com a estratégia de poda SCut (tradicional) com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-7 (b)). O valor de $micro - F_1^c$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-7(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-7(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $micro - F_1^c$ mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para a base de dados AT100, como visto na Figura 6-7(a). Desta forma, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o desempenho do sistema em termos da métrica $micro - F_1^c$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut* para a base AT100. A análise de desempenho dos resultados obtidos para estes categorizados e para a base EX100 apresenta equilíbrio ao empregar as estratégias de poda.

6.2.5 Precisão (*precision*) Orientada a Documento

A métrica **precisão *(precision)* orientada a documento (** $precision_j^d$ **)** avalia a fração de categorias preditas que são pertinentes ao documento de teste d_j. A formulação é mostrada na Equação (6.13).

$$precision_j^d = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|\hat{C}_j^{|C_j|}|} \tag{6.13}$$

O valor de $precision_j^d$ também pode ser computado usando a tabela de contingência do documento d_j (Tabela 6-7), conforme Equação (6.14).

$$precision_j^d = \frac{TP_j}{TP_j + FP_j} \tag{6.14}$$

onde FP_j (falsos positivos para d_j) é o número de categorias que foram incorretamente preditas para d_j, TN_j (verdadeiros negativos para d_j) é o número de categorias que foram corretamente não preditas para d_j; TP_j (verdadeiros positivos para d_j) é o número de categorias que foram corretamente preditas para d_j; e FN_j (falsos negativos para d_j) é o número de categorias que foram incorretamente não preditas para d_j.

Tabela 6-7 – Tabela de contingência do documento d_j.

Documento d_j		**Julgamentos do especialista**	
		SIM	NÃO
Julgamentos do categorizador	SIM	TP_j	FP_j
	NÃO	FN_j	TN_j

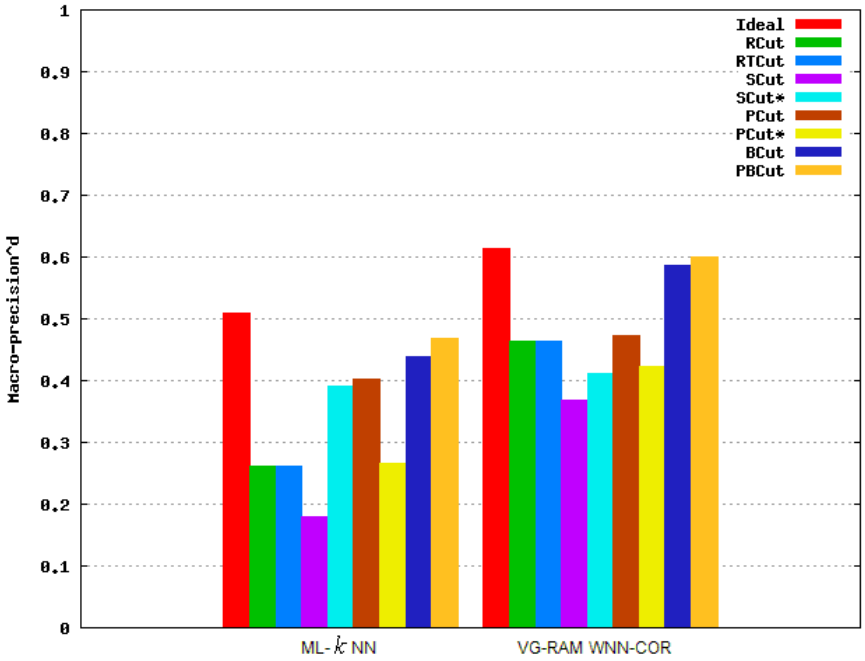
O desempenho global de *precision* orientada a documento é calculado pelos métodos $macro - precision^d$ e $micro - precision^d$, Equação (6.15) e Equação (6.16), respectivamente. Quanto maior o valor de $macro - precision^d$ e $micro - precision^d$, melhor

o desempenho do categorizador. O desempenho é perfeito quando $macro-precision^d=1$ e $micro-precision^d=1$.

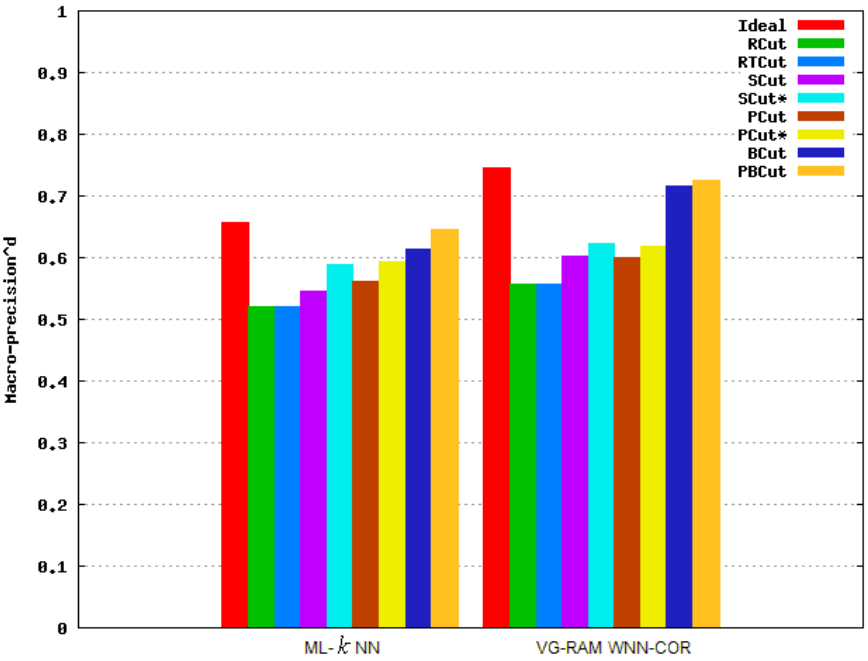
$$macro-precision^d=\frac{\sum_{j=1}^{|Te|} precision_j^d}{|Te|} \tag{6.15}$$

$$micro-precision^d=\frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|}\left(TP_j+FP_j\right)} \tag{6.16}$$

A Figura 6-8 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $macro-precision^d$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-8(a)) e EX100 (Figura 6-8(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-8 - Resultado da métrica $macro - precision^d$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-8 mostram, o valor de $macro - precision^d$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro - precision^d$ ao usar as estratégias de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as

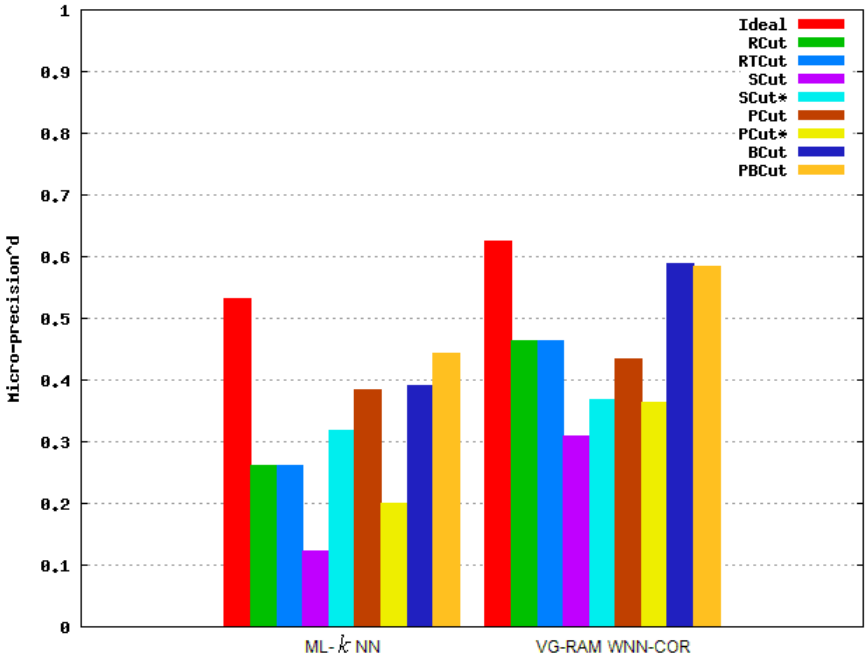
estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN* com a base de dados AT100 (Figura 6-8(a)). O valor de $macro-precision^d$ com a estratégia de poda SCut* é maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-8(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-8(a)). O valor de $macro-precision^d$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-8(a)).

Como apresentado na Figura 6-8, as barras do gráfico mostram o valor de $macro-precision^d$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro-precision^d$ ao usar a estratégia de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-8(b)). O valor de $macro-precision^d$ com a estratégia de poda SCut* é maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-8(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-8(b)). O valor de $macro-precision^d$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-8(b)).

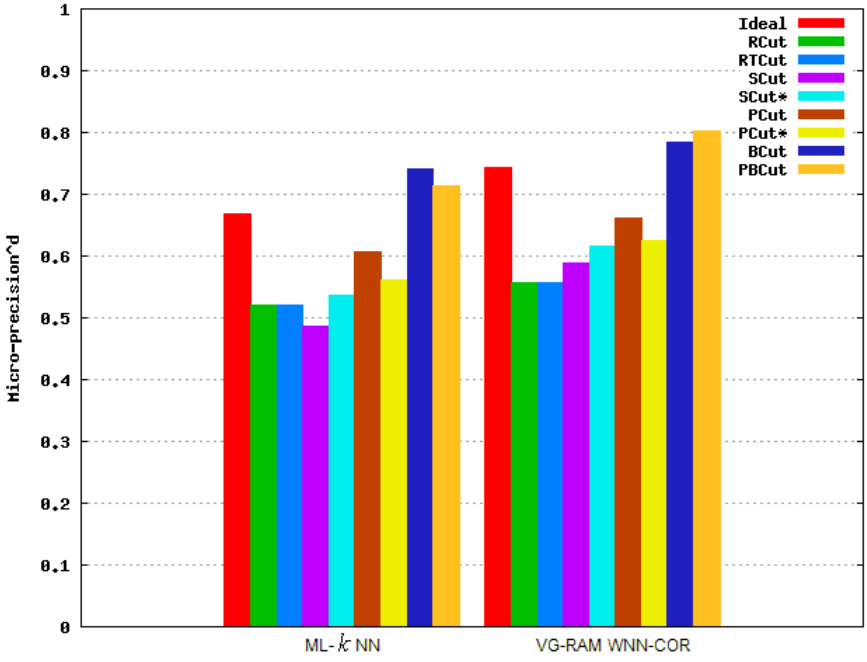
A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $macro-precision^d$ mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para ambas bases, como visto na Figura 6-8(a)(b). Isso acontece porque a estratégia BCut poda o *ranking* com base na probabilidade da categorização estar correta e a estratégia PBCut poda o *ranking* com base na probabilidade da categorização estar correta em cada posição do *ranking*, observando o decremento da medida de certeza a medida que a posição da categoria no *ranking* aumenta. Desta forma, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o desempenho do sistema em termos da métrica $macro-precision^d$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*.

A Figura 6-9 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $micro-precision^d$ segundo os categorizadores

ML-k NN e *VG-RAM WNN-COR* para a base AT100 (Figura 6-9(a)) e EX100 (Figura 6-9(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-9- Resultado da métrica $micro-precision^d$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-9 mostram, o valor de $micro-precision^d$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-precision^d$ ao usar as estratégias de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-9(a)). O valor de $macro-precision^d$ com a estratégia de poda SCut* é maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-9(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-9(a)). O valor de $macro-precision^d$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-9(a)).

Como apresentado na Figura 6-9, as barras do gráfico mostram o valor de $micro-precision^d$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-precision^d$ ao usar as estratégias de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-9(b)). O valor de $micro-precision^d$ com a estratégia de poda SCut* é maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-9 (b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-9(b)). O valor de $micro-precision^d$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-9(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $micro-precision^d$ mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para ambas bases, como visto na Figura 6-9(a)(b). Isso acontece porque a estratégia BCut poda o *ranking* com base na probabilidade da categorização estar correta e a estratégia PBCut poda o *ranking* com base na probabilidade da categorização estar correta em cada posição do *ranking*, observando o decremento da medida de certeza a medida que a posição da categoria no *ranking* aumenta. Desta forma, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o

desempenho do sistema em termos da métrica $micro - precision^d$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*.

6.2.6 Revocação (*recall*) Orientada a Documento

A métrica **revocação (*recall*) orientada a documento (**$recall_j^d$**)** avalia a fração de categorias pertinentes que são preditas para o documento de teste d_j. A formulação é apresentada na Equação (6.17).

$$recall_j^d = \frac{\left|\hat{C}_j^{|C_j|} \cap C_j\right|}{|C_j|} \tag{6.17}$$

O valor de $recall_j^d$ pode também ser obtido em termos da tabela de contingência do documento d_j (Tabela 6-7) conforme a Equação (6.18).

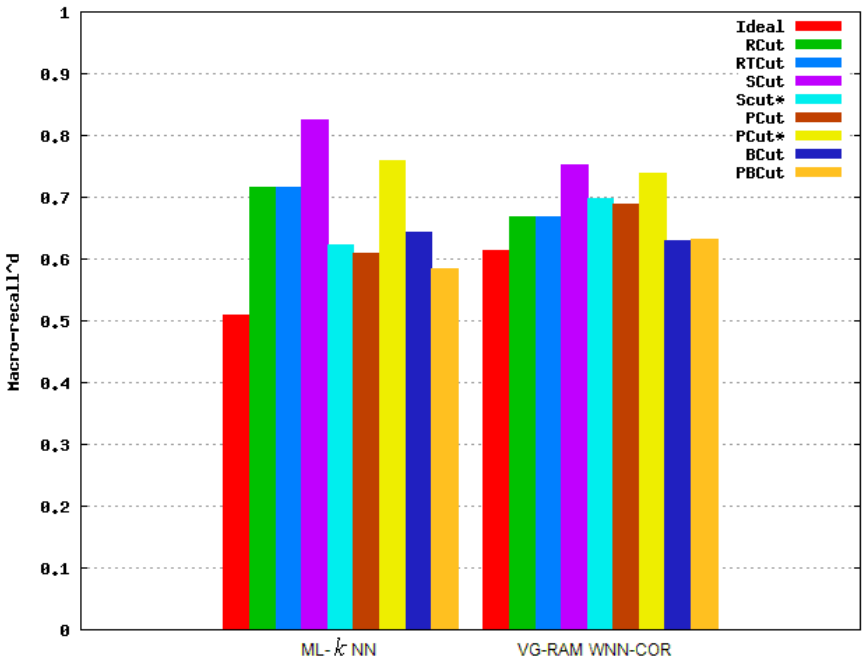
$$recall_j^d = \frac{TP_j}{TP_j + FN_j} \tag{6.18}$$

O desempenho global de *recall* orientado a documento é calculado pelos métodos $macro - recall^d$ e $micro - recall^d$, Equação (6.19) e Equação (6.20), respectivamente. Quanto maior o valor de $macro - recall^d$ e $micro - recall^d$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - recall^d = 1$ e $micro - recall^d = 1$.

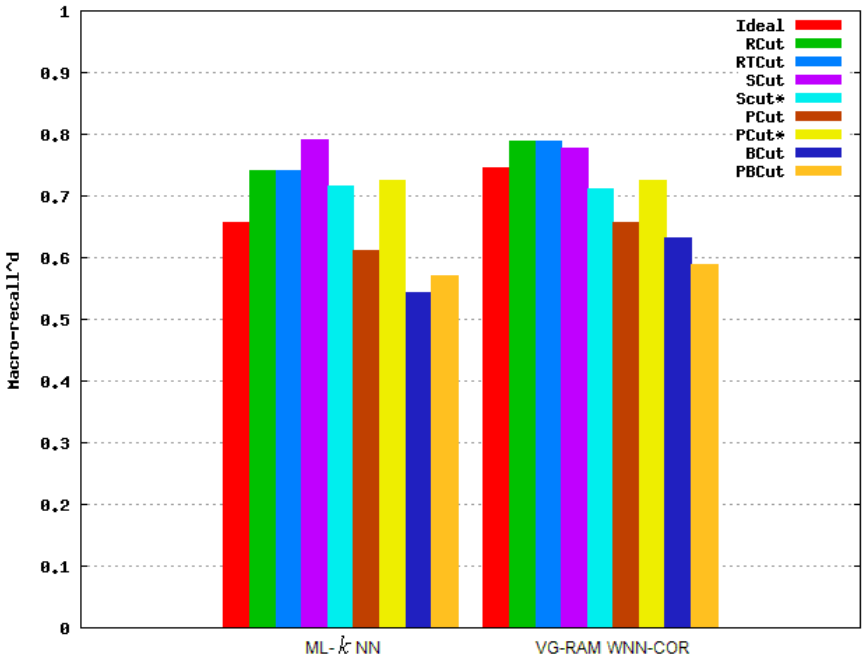
$$macro - recall^d = \frac{\sum_{j=1}^{|Te|} recall_j^d}{|Te|} \tag{6.19}$$

$$micro - recall^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|} \left(TP_j + FN_j\right)} \tag{6.20}$$

A Figura 6-10 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $macro-recall^d$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-10(a)) e EX100 (Figura 6-10(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

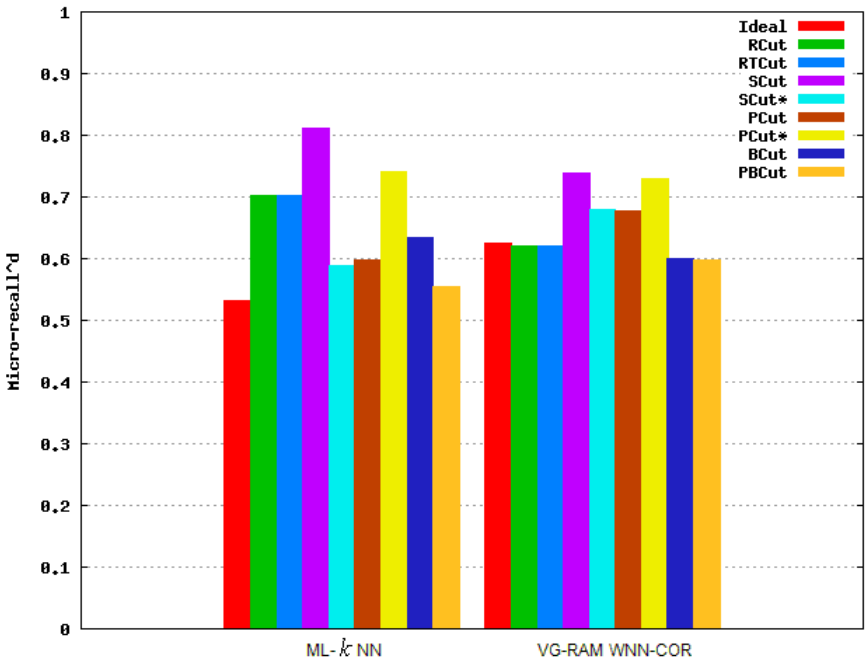
Figura 6-10 - Resultado da métrica $macro-recall^d$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-10 mostram, o valor de $macro-recall^d$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro-recall^d$ ao usar as estratégias de poda de *ranking* SCut é significativamente maior do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com esta base, e com a base de dados AT100 (Figura 6-10). O valor de $macro-recall^d$ com a estratégia de poda SCut* é menor do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-10 (a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-10 (a)). O valor de $macro-recall^d$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-10(a)).

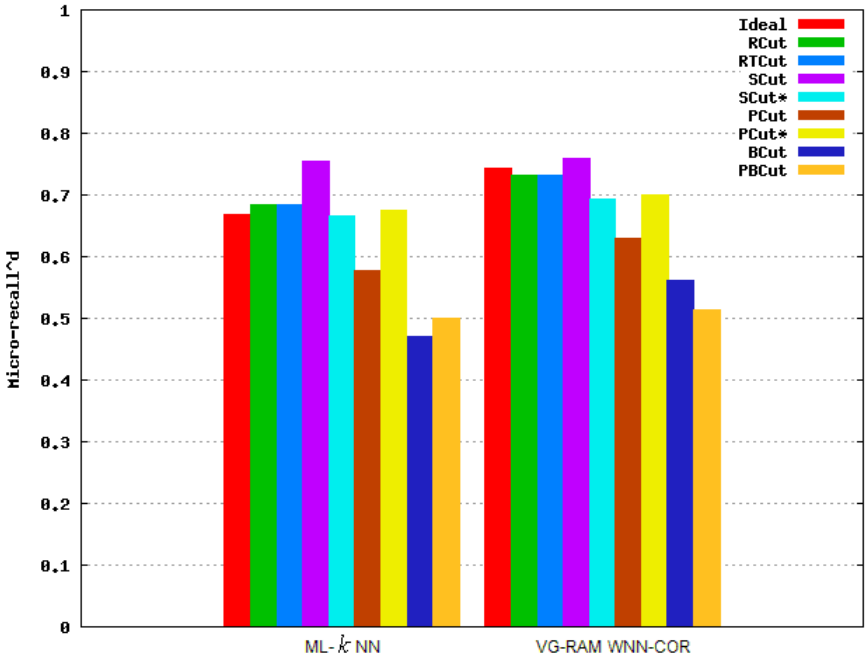
Como apresentado na Figura 6-10, as barras do gráfico mostram o valor de $macro-recall^d$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro-recall^d$ ao usar as estratégias de poda de *ranking* SCut são significativamente maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut. O valor de $macro-recall^d$ com a estratégia de poda SCut* é menor que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-10(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-10(b)). O valor de $macro-recall^d$ com a estratégia de poda PCut* é maior que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-10(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-10(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $macro-recall^d$ mostram que a estratégia de poda SCut melhor o desempenho desses categorizadores para ambas bases, como visto na Figura 6-10(a)(b). Isso acontece porque SCut otimiza o desempenho do categorizador por categoria, tornando-a particularmente efetiva quando o desempenho do sistema em categorias raras é a função alvo a ser otimizada. Desta forma, a estratégia de poda SCut é mais efetiva para otimizar o desempenho do sistema em termos da métrica $macro-recall^d$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut.

A Figura 6-11 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $micro - recall^d$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-11(a)) e EX100 (Figura 6-11(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-11 - Resultado da métrica $micro - recall^d$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-11 mostram, o valor de $micro-recall^d$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-recall^d$ ao usar a estratégia de poda de *ranking* SCut é significativamente maior do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut*, PCut, SCut*, BCut e PBCut. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com esta base, e com a base de dados AT100 (Figura 6-11(a)). O valor de $micro-recall^d$ com a estratégia de poda SCut* é segnificamente menor que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-11(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-11(a)). O valor de $micro-recall^d$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-11(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-11(a)).

Como apresentado na Figura 6-11, as barras do gráfico mostram o valor de $micro-recall^d$ do categorizador *ML-k NN* com a base EX100 (Figura 6-11(b)) é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-recall^d$ ao usar a estratégia de poda de *ranking* SCut é significativamente maior do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut. O valor de $micro-recall^d$ com a estratégia de poda SCut* é menor que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-11(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-11(b)). O valor de $micro-recall^d$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-11(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $micro-recall^d$ mostram que a estratégia de poda SCut melhor o desempenho desses categorizadores para ambas bases, como visto na Figura 6-11 (a)(b). Desta forma, a estratégia de poda SCut é mais efetiva para otimizar o desempenho do sistema em termos da métrica $micro-recall^d$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut*, BCut e PBCut.

6.2.7 F_β Orientada a Documento

A métrica F_β **orientada a documento (** $F_{\beta_j}^{\,d}$ **)** avalia a média harmônica ponderada de $precision_j^d$ e $recall_j^d$. A formulação original de Rijsbergen [Rijsbergen79] é mostrada na Equação (6.21).

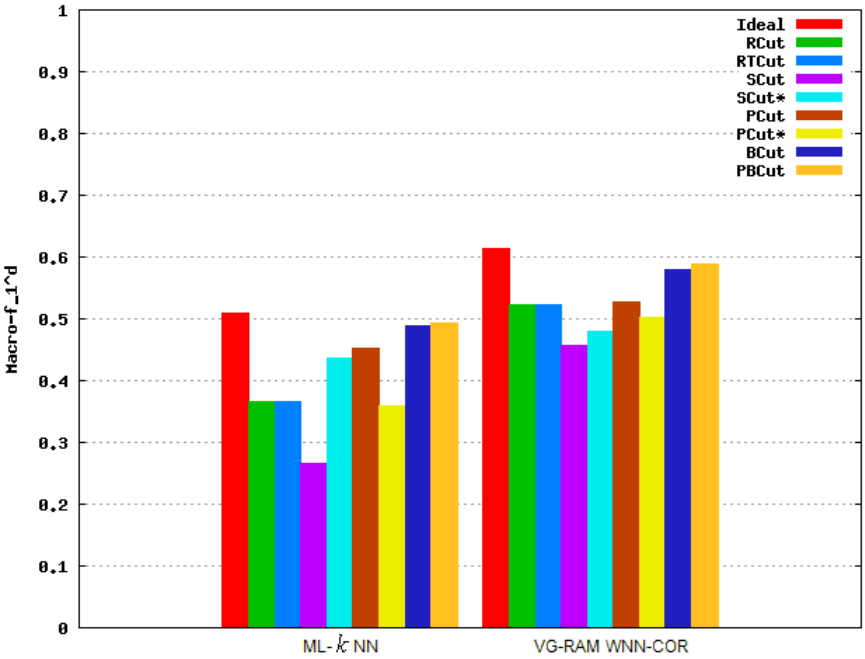
$$F_{\beta_j}^{\,d} = \frac{(\beta^2+1) * precision_j^d * recall_j^d}{\beta^2 * precision_j^d + recall_j^d} \tag{6.21}$$

Como na métrica F_β orientada à categoria, $\beta = 1$ é utilizado, atribuindo importância igual para $precision_j^d$ e $recall_j^d$. O desempenho global de $F_1^{\,d}$ é computado pelos métodos $macro - F_1^{\,d}$ (Equação (6.22)) e $micro - F_1^{\,d}$ (Equação (6.23)). Quanto maior o valor de $macro - F_1^{\,d}$ e $micro - F_1^{\,d}$, melhor o desempenho do categorizador. O desempenho é perfeito quando $macro - F_1^{\,d} = 1$ e $micro - F_1^{\,d} = 1$.

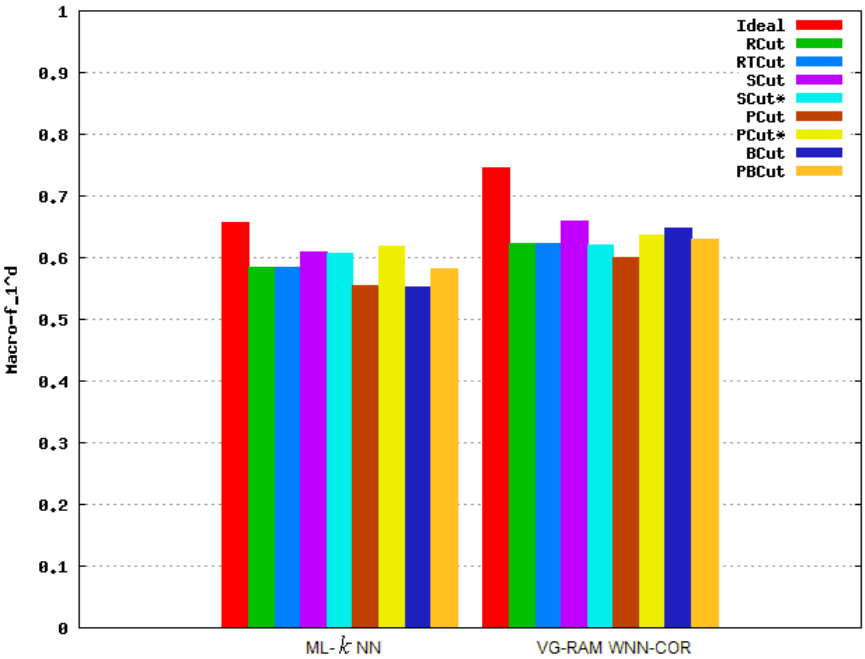
$$macro - F_1^{\,d} = \frac{1}{|Te|}\sum_{j=1}^{|Te|} F_{1j}^{\,d} \tag{6.22}$$

$$micro - F_1^{\,d} = \frac{2 * micro - precision_j^d * micro - recall_j^d}{micro - precision_j^d + micro - recall_j^d} \tag{6.23}$$

A Figura 6-12 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $macro - F_1^{\,d}$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-12(a)) e EX100 (Figura 6-12(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-12 - Resultado da métrica $macro - F_1^d$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

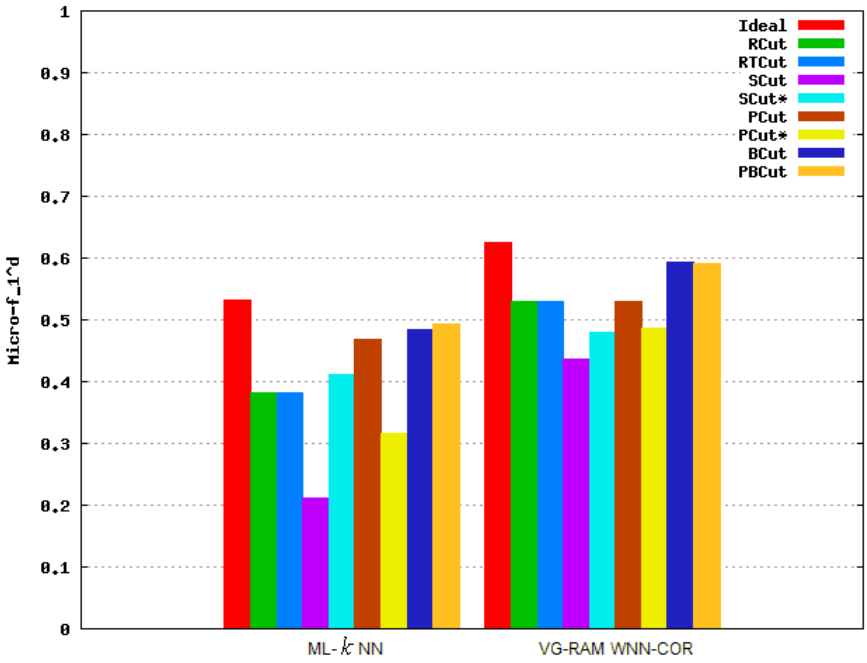
Conforme as barras do gráfico da Figura 6-12 mostram, o valor de $macro - F_1^d$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro - F_1^d$ ao usar a estratégia de poda de *ranking* BCut e PBCut são significativamente maiores do que os resultados obtidos ao usar as estratégias

RCut, RTCut, PCut, PCut* SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-12(a)). O valor de $macro - F_1^d$ com a estratégia de poda SCut* é significamente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-12 (a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-12(a)). O valor de $macro - F_1^d$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-12 (a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-12 (a)).

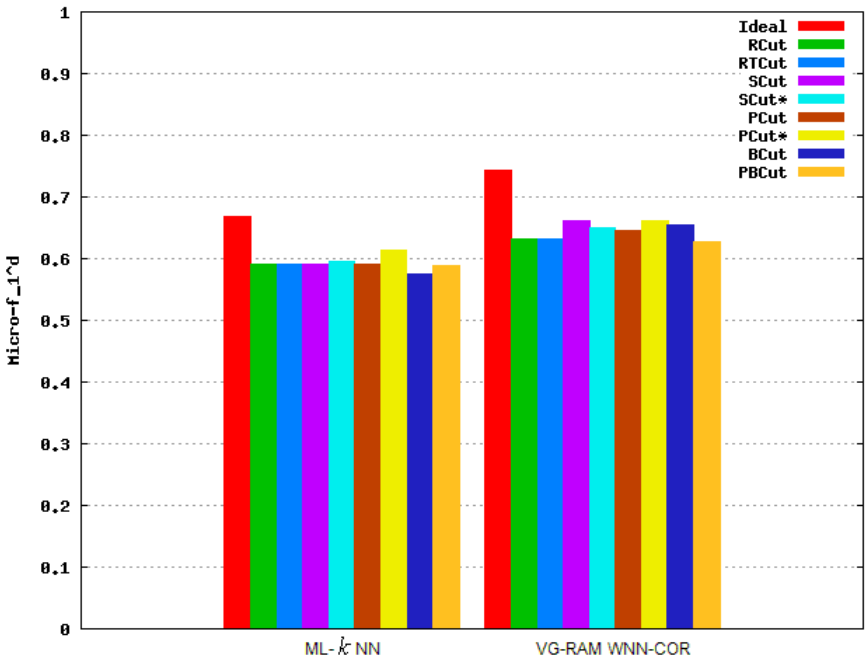
Como apresentado na Figura 6-12, as barras do gráfico mostram o valor de $macro - F_1^d$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $macro - F_1^d$ ao usar a estratégia de poda PCut é parcialmente maior do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut*, SCut, SCut*, BCut e PBCut. Já com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-12(b)) o valor de $macro - F_1^d$ ao usar a estratégia de poda SCut é significamente maior do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut, BCut e PBCut. O valor de $macro - F_1^d$ com a estratégia de poda SCut* é menor do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-12(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-12(b)). O valor de $macro - F_1^d$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-12(b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-12(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $macro - F_1^d$ mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para a base de dados AT100, como visto na Figura 6-12(a). O desempenho obtido com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $macro - F_1^d$ e para a base EX100 mostra um equilíbrio nos valores das estratégias. Apesar disso, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o desempenho do sistema em termos da métrica $macro - F_1^d$ do que as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut* para a base AT100.

A Figura 6-13 mostra de forma gráfica o impacto do uso de estratégias de poda no *ranking* de categorias avaliadas pela métrica $micro - F_1^d$ segundo os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a base AT100 (Figura 6-13(a)) e EX100 (Figura 6-13(b)) respectivamente. Esta figura segue o mesmo formato da Figura 6-1.



(a)



(b)

Figura 6-13 - Resultado da métrica $micro - F_1^d$ para as bases (a) AT100 e (b) EX100. Quanto maior, melhor.

Conforme as barras do gráfico da Figura 6-13 mostram, o valor de $micro-F_1^d$ do categorizador *ML-k NN* com a base AT100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-F_1^d$ ao usar a estratégia de poda de *ranking* BCut e PBCut são maiores do que os resultados obtidos ao usar as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-13(a)). O valor de $micro-F_1^d$ com a estratégia de poda SCut* é segnificamente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-13(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-13(a)). O valor de $micro-F_1^d$ com a estratégia de poda PCut* é menor do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base AT100 (Figura 6-13(a)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados AT100 (Figura 6-13(a)).

Como apresentado na Figura 6-13, as barras do gráfico mostram o valor de $micro-F_1^d$ do categorizador *ML-k NN* com a base EX100 é impactado pelo uso de estratégias de poda no *ranking* de categorias. O valor de $micro-F_1^d$ ao usar a estratégia de poda de *ranking* mostram um equilíbrio nos resultados das estratégias RCut, RTCut, PCut, PCut*, SCut, SCut*, BCut e PBCut. O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-13(b)). O valor de $micro-F_1^d$ com a estratégia de poda SCut* é parcialmente maior do que com a estratégia de poda SCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-13 (b)). O valor de $micro-F_1^d$ com a estratégia de poda SCut* é menor do que com a estratégia de poda SCut (tradicional) com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-13(b)). O valor de $micro-F_1^d$ com a estratégia de poda PCut* é maior do que com a estratégia de poda PCut (tradicional) com o categorizador *ML-k NN* com a base EX100 (Figura 6-13 (b)). O mesmo ocorre com o categorizador *VG-RAM WNN-COR* com a base de dados EX100 (Figura 6-13(b)).

A análise do desempenho dos resultados obtidos com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $micro-F_1^d$ mostram que as estratégias de poda BCut e PBCut melhoram o desempenho desses categorizadores para a base de dados AT100, como visto na Figura 6-13(a). O desempenho obtido com os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para a métrica $micro-F_1^d$ e para a base EX100 mostra um equilíbrio nos valores

das estratégias. Apesar disso, as estratégias de poda BCut e PBCut são mais efetivas para otimizar o desempenho do sistema em termos da métrica $micro - F_1^{\,d}$ para base de dados AT100 do que as estratégias RCut, RTCut, PCut, PCut*, SCut e SCut*.

Note que o desempenho dos categorizadores pelo método *microaveraging* dá resultado igual, independente de ser definida orientada à categoria ou a documento. A expansão das formulações de $micro - precision^c$ e $micro - precision^d$ é mostrada na Equação (6.24) e Equação (6.25), respectivamente.

$$micro - precision^c = \frac{\sum_{i=1}^{|C|} TP_i}{\sum_{i=1}^{|C|}(TP_i + FP_i)} = \frac{\sum_{i=1}^{|C|}\sum_{j=1}^{|Te|} TP_{ij}}{\sum_{i=1}^{|C|}\left(\sum_{j=1}^{|Te|} TP_{ij} + \sum_{j=1}^{|Te|} FP_{ij}\right)} \tag{6.24}$$

$$micro - precision^d = \frac{\sum_{j=1}^{|Te|} TP_j}{\sum_{j=1}^{|Te|}(TP_j + FP_j)} = \frac{\sum_{j=1}^{|Te|}\sum_{i=1}^{|C|} TP_{ij}}{\sum_{j=1}^{|Te|}\left(\sum_{i=1}^{|C|} TP_{ij} + \sum_{i=1}^{|C|} FP_{ij}\right)} \tag{6.25}$$

Observa-se pela Equação (6.24) e Equação (6.25), $micro - precision^c$ é igual a $micro - precision^d$. Analogamente, $micro - recall^c$ e $micro - F_1^{\,c}$ são iguais a $micro - recall^d$ e $micro - F_1^{\,d}$, respectivamente.

6.2.8 *Test-T* Estatístico

Para apresentar uma visão mais clara do desempenho relativo das estratégias de poda de *ranking* propostas neste trabalho com as estratégias encontradas na literatura, uma ordem parcial $\succ$ é definida para compararmos as estratégias de poda BCut e PBCut (propostas) com as estratégias RCut, RTCut, SCut e PCut (encontradas na literatura) e também com as variantes PCut* e SCut para cada métrica de avaliação de desempenho, onde E1 $\succ$ E2 significa que o desempenho da estratégia de poda E1 é significativamente melhor do que a estratégia E2 para a métrica específicada (teste t pareado ao nível de 5% de significância). Se o desempenho não é significativamente diferente, a ordem parcial E1 ≡ E2 é utilizada.

A

Tabela 6-8 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *ML-k NN* e para a base de dados AT100 utilizando a representação de

ordem parcial. Este sumário apresenta, a comparação (*test-t* pareado) entre a estratégia BCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-8 - Resultado do *Test-t* para o categorizador *ML-k NN* e para a base AT100. Comparação com BCut.

Métricas	BCut x Ideal	BCut x RCut	BCut x RTCut	BCut x SCut	BCut x SCut*	BCut x PCut	BCut x PCut*	BCut x PBCut
exact match	BCut ≻ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
macro-precision-c	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≡ SCut*	BCut ≺ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-precision-c	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
macro-recall-c	BCut ≻ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≻ PBCut
micro-recall-c	BCut ≻ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≺ PCut*	BCut ≻ PBCut
macro-f1-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≻ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≡ PCut*	BCut ≻ PBCut
micro-f1-c	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
macro-precision-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-precision-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
macro-recall-d	BCut ≻ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≺ PCut*	BCut ≻ PBCut
micro-recall-d	BCut ≻ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≺ PCut*	BCut ≻ PBCut
macro-f1-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-f1-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
Ordem Geral	BCut(5) < Ideal(8)	BCut(8) > RCut(5)	BCut(8) > RTCut(5)	BCut(9) > SCut(4)	BCut(10) > SCut(3)	BCut(10) > PCut(3)	BCut(8) > PCut*(4)	BCut(6) < PBCut(7)

Como a Tabela 6-8 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *ML-k NN* para a base de dados AT100. De acordo com a Tabela 6-8, a ordem geral mostra que a estratégia de poda BCut é mais apropriada para esta base de dados, categorizador e métricas.

A Tabela 6-9 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *ML-k NN* e para a base de dados AT100 utilizando a representação de ordem parcial. Este sumário apresenta, a comparação (*test-t* pareado) entre a estratégia PBCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-9 - Resultado do *Test-t* para o categorizador *ML-k NN* e para a base AT100. Comparação com PBCut.

Métricas	PBCut x Ideal	PBCut x RCut	PBCut x RTCut	PBCut x SCut	PBCut x SCut*	PBCut x PCut	PBCut x PCut*	PBCut x PBCut
exact match	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
macro-precision-c	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≡ SCut*	PBCut ≺ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-precision-c	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
macro-recall-c	PBCut ≡ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
micro-recall-c	PBCut ≻ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
macro-f1-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≻ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
micro-f1-c	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
macro-precision-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-precision-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
macro-recall-d	PBCut ≻ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
micro-recall-d	PBCut ≻ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
macro-f1-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-f1-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
Ordem Geral	PBCut(3) < Ideal(9)	PBCut(8) > RCut(5)	PBCut(8) > RTCut(5)	PBCut(9) > SCut(4)	PBCut(7) > SCut(5)	PBCut(7) > PCut(6)	PBCut(8) > PCut*(4)	PBCut(7) > BCut(6)

Como a Tabela 6-9 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *ML-k NN* para a base de dados AT100. De acordo com a Tabela 6-9, a ordem geral mostra que a estratégia de poda PBCut é mais apropriada para esta base de dados, categorizador e métricas. Note que a estratégia de poda PBCut é mais apropriada para estas características do que a estratégia BCut.

A Tabela 6-10 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *ML-k NN* e para a base de dados EX100 utilizando a representação de ordem parcial. Este sumário apresenta, a comparação entre a estratégia BCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-10 - Resultado do *Test-t* para o categorizador *ML-k NN* e para a base EX100. Comparação com BCut.

Métricas	BCut x Ideal	BCut x RCut	BCut x RTCut	BCut x SCut	BCut x SCut*	BCut x PCut	BCut x PCut*	BCut x PBCut
exact match	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
macro-precision-c	BCut ≡ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
micro-precision-c	BCut ≻ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
macro-recall-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≺ PBCut
micro-recall-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≺ PBCut
macro-f1-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≺ PBCut
micro-f1-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≺ PBCut
macro-precision-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-precision-d	BCut ≻ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
macro-recall-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-recall-d	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≺ PBCut
macro-f1-d	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≡ PCut	BCut ≺ PCut*	BCut ≺ PBCut
micro-f1-d	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≺ PBCut
Ordem Geral	Ideal(10) > BCut(2)	BCut(6) < RCut(7)	BCut(6) < RTCut(7)	BCut(6) < SCut(7)	BCut(6) < SCut(7)	BCut(6) = PCut(6)	BCut(6) < PCut*(7)	BCut(3) < PBCut(10)

Como a Tabela 6-10 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *ML-k NN* para a base de dados EX100. De acordo com a Tabela 6-10, a ordem geral mostra que a estratégia de poda BCut não é mais apropriada para ser avaliada sob todas esssas estratégias.

A Tabela 6-11 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *ML-k NN* e para a base de dados EX100 utilizando a representação de ordem parcial. Este sumário apresenta, a comparação entre a estratégia PBCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-11 - Resultado do *Test-t* para o categorizador *ML- k NN* e para a base EX100. Comparação com PBCut.

Métricas	PBCut x Ideal	PBCut x RCut	PBCut x RTCut	PBCut x SCut	PBCut x SCut*	PBCut x PCut	PBCut x PCut*	PBCut x PBCut
exact match	PBCut ≻ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
macro-precision-c	PBCut ≡ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
micro-precision-c	PBCut ≻ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
macro-recall-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
micro-recall-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
macro-f1-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
micro-f1-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≡ SCut	PBCut ≡ SCut*	PBCut ≡ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
macro-precision-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-precision-d	PBCut ≻ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
macro-recall-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
micro-recall-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
macro-f1-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≻ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
micro-f1-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≡ SCut	PBCut ≡ SCut*	PBCut ≡ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
Ordem Geral	PBCut(3) < Ideal(9)	PBCut(5) < RCut(8)	PBCut(5) < RTCut(8)	PBCut(5) < SCut(6)	PBCut(5) < SCut(6)	PBCut(6) > PCut(5)	PBCut(5) < PCut*(8)	BCut(3) < PBCut(10)

Como a Tabela 6-11 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *ML- k NN* para a base de dados EX100. De acordo com a Tabela 6-11, a ordem geral mostra que a estratégia de poda PBCut não é mais apropriada para ser avaliada sob todas esssas métricas.

A Tabela 6-12 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *VG-RAM WNN-COR* e para a base de dados AT100 utilizando a representação de ordem parcial. Este sumário apresenta, a comparação entre a estratégia BCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-12 – Resultado do *Test-t* para o categorizador *VG-RAM WNN-COR* e para a base AT100. Comparação com BCut.

Métricas	BCut x Ideal	BCut x RCut	BCut x RTCut	BCut x SCut	BCut x SCut*	BCut x PCut	BCut x PCut*	BCut x PBCut
exact match	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
macro-precision-c	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-precision-c	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
macro-recall-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≺ PBCut
micro-recall-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≡ PBCut
macro-f1-c	BCut ≺ Ideal	BCut ≡ RCut	BCut ≡ RTCut	BCut ≡ SCut	BCut ≻ SCut*	BCut ≡ PCut	BCut ≡ PCut*	BCut ≺ PBCut
micro-f1-c	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
macro-precision-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≺ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-precision-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
macro-recall-d	BCut ≻ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≻ PCut*	BCut ≡ PBCut
micro-recall-d	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≡ PBCut
macro-f1-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-f1-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
Ordem Geral	BCut(1) < Ideal(12)	BCut(8) > RCut(4)	BCut(8) > RTCut(4)	BCut(8) > SCut(4)	BCut(8) > SCut(5)	BCut(8) > PCut(4)	BCut(9) > PCut*(3)	BCut(4) < PBCut(6)

Como a Tabela 6-12 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *VG-RAM WNN-COR* para a base de dados AT100. De acordo com a Tabela 6-12, a ordem geral mostra que a estratégia de poda BCut é mais apropriada para esta base de dados, categorizador e métricas.

A Tabela 6-13 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *VG-RAM WNN-COR* e para a base de dados AT100 utilizando a representação de ordem parcial. Este sumário apresenta, a comparação entre a estratégia PBCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-13 - Resultado do *Test-t* para o categorizador *VG-RAM WNN-COR* e para a base AT100. Comparação com PBCut.

Métricas	PBCut x Ideal	PBCut x RCut	PBCut x RTCut	PBCut x SCut	PBCut x SCut*	PBCut x PCut	PBCut x PCut*	PBCut x BCut
exact match	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
macro-precision-c	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-precision-c	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
macro-recall-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≻ BCut
micro-recall-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≡ BCut
macro-f1-c	PBCut ≺ Ideal	PBCut ≡ RCut	PBCut ≡ RTCut	PBCut ≡ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≡ PCut*	PBCut ≻ BCut
micro-f1-c	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
macro-precision-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-precision-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
macro-recall-d	PBCut ≻ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≡ BCut
micro-recall-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≡ BCut
macro-f1-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-f1-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
Ordem Geral	PBCut(1) < Ideal(12)	PBCut(8) > RCut(4)	PBCut(8) > RTCut(4)	PBCut(8) > SCut(4)	PBCut(9) > SCut(4)	PBCut(9) > PCut(4)	PBCut(8) > PCut*(4)	PBCut(6) > BCut(4)

Como a Tabela 6-13 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *VG-RAM WNN-COR* para a base de dados AT100. De acordo com a Tabela 6-13, a ordem geral mostra que a estratégia de poda PBCut é mais apropriada para esta base de dados, categorizador e métricas. Note que a estratégia de poda PBCut é mais apropriada para estas características do que a estratégia BCut.

A Tabela 6-14 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *VG-RAM WNN-COR* e para a base de dados EX100 utilizando a representação de ordem parcial. Este sumário apresenta, a comparação entre a estratégia BCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-14 - Resultado do *Test-t* para o categorizador *VG-RAM WNN-COR* e para a base EX100. Comparação com BCut

Métricas	BCut x Ideal	BCut x RCut	BCut x RTCut	BCut x SCut	BCut x SCut*	BCut x PCut	BCut x PCut*	BCut x PBCut
exact match	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
macro-precision-c	BCut ≡ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-precision-c	BCut ≻ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
macro-recall-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≻ PBCut
micro-recall-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≻ PBCut
macro-f1-c	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≻ PBCut
micro-f1-c	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≺ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≡ PCut*	BCut ≻ PBCut
macro-precision-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
micro-precision-d	BCut ≻ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≺ PBCut
macro-recall-d	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≻ PBCut
micro-recall-d	BCut ≺ Ideal	BCut ≺ RCut	BCut ≺ RTCut	BCut ≺ SCut	BCut ≺ SCut*	BCut ≺ PCut	BCut ≺ PCut*	BCut ≻ PBCut
macro-f1-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≻ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≻ PCut*	BCut ≻ PBCut
micro-f1-d	BCut ≺ Ideal	BCut ≻ RCut	BCut ≻ RTCut	BCut ≺ SCut	BCut ≻ SCut*	BCut ≻ PCut	BCut ≡ PCut*	BCut ≻ PBCut
Ordem Geral	BCut(2) < Ideal(10)	BCut(8) > RCut(5)	BCut(8) > RTCut(5)	BCut(6) < SCut(7)	BCut(8) > SCut(5)	BCut(8) > PCut(5)	BCut(6) > PCut*(5)	BCut(9) > PBCut(4)

Como a Tabela 6-14 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *VG-RAM WNN-COR* para a base de dados EX100. De acordo com a Tabela 6-14, a ordem geral mostra que a estratégia de poda BCut é mais apropriada para esta base de dados, categorizador e métricas.

A Tabela 6-15 mostra um sumário dos resultados obtidos da Seção 6.2.1 à Seção 6.2.7 para o categorizador *VG-RAM WNN-COR* e para a base de dados EX100 utilizando a representação de ordem parcial. Este sumário apresenta, a comparação entre a estratégia PBCut e as demais estratégias de poda apresentadas neste trabalho.

Tabela 6-15 - Resultado do *Test-t* para o categorizador *VG-RAM WNN-COR* e para a base EX100. Comparação com PBCut.

Métricas	PBCut x Ideal	PBCut x RCut	PBCut x RTCut	PBCut x SCut	PBCut x SCut*	PBCut x PCut	PBCut x PCut*	PBCut x PBCut
exact match	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≺ BCut
macro-precision-c	PBCut ≻ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-precision-c	PBCut ≻ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
macro-recall-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
micro-recall-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
macro-f1-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
micro-f1-c	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
macro-precision-d	PBCut ≺ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
micro-precision-d	PBCut ≻ Ideal	PBCut ≻ RCut	PBCut ≻ RTCut	PBCut ≻ SCut	PBCut ≻ SCut*	PBCut ≻ PCut	PBCut ≻ PCut*	PBCut ≻ BCut
macro-recall-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
micro-recall-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
macro-f1-d	PBCut ≺ Ideal	PBCut ≡ RCut	PBCut ≡ RTCut	PBCut ≺ SCut	PBCut ≡ SCut*	PBCut ≻ PCut	PBCut ≡ PCut*	PBCut ≺ BCut
micro-f1-d	PBCut ≺ Ideal	PBCut ≺ RCut	PBCut ≺ RTCut	PBCut ≺ SCut	PBCut ≺ SCut*	PBCut ≺ PCut	PBCut ≺ PCut*	PBCut ≺ BCut
Ordem Geral	PBCut(3) < Ideal(10)	PBCut(5) < RCut(7)	PBCut(5) < RTCut(7)	PBCut(5) < SCut(8)	PBCut(5) < SCut(7)	PBCut(6) < PCut(7)	PBCut(5) < PCut*(7)	PBCut(4) < BCut(9)

Como a Tabela 6-15 mostra, as estratégias de poda de *ranking* impactam a maioria das métricas utilizadas para avaliar o desempenho do categorizador *VG-RAM WNN-COR* para a base de dados EX100. De acordo com a Tabela 6-15, a ordem geral mostra que a estratégia de poda PBCut não é mais apropriada para esta base de dados, categorizador e métricas.

Os resultados obtidos com as métricas que avaliam precisão mostram que, as estratégias de poda de *ranking* BCut e PBCut melhoram o desempenho dos categorizadores para as bases AT100 e EX100. Além disso, as estratégias de poda de *ranking* BCut e PBCut melhoram o desempenho para a métrica *exact-match* o desempenho dos categorizadores *ML- k NN e VG-RAM WNN-COR* para as bases AT100 e EX100.

7 DISCUSSÃO

Neste capítulo, apresentamos os trabalhos correlatos ao estudo sobre estratégias de poda de *ranking* na categorização multi-rótulo de texto. Além disso, apresentamos uma análise crítica sobre este trabalho.

7.1 Trabalhos Correlatos

O estudo sobre estratégias de poda de *ranking* tem sido pouco explorado na área de categorização automática de texto. Algoritmos de categorização tem sido o principal foco de pesquisa nessa área enquanto que estratégias de poda de *ranking* são apenas mencionadas como uma etapa sem importância na categorização de texto [Yang01].

Yang, em 2001[Yang01] apresentou um estudo sobre as estratégias de poda. Este estudo mostra uma análise do impacto de desempenho da estratégia de poda de *ranking* de um categorizador em condições diversas. Yang usou o categorizador *k*NN, cinco coleções de documentos, três estratégias de poda incluindo a estratégia baseada no *ranking* (RCut), baseada em probabilidade (PCut) e baseada no grau de crença (SCut). Além destas, Yang propôs uma nova variante da estratégia RCut, denominada RTCut. Os resultatos experimentais obtidos por Yang, mostram que a escolha de uma estratégia de poda pode influenciar significamente no desempenho do categorizador *k*NN e que o resultado da melhor estratégia poda pode variar de acordo com aplicação. Yang apresentou a estratégia SCut como a melhor estratégia para ajuste fino, mas, com o risco de *overfitting*. A estratégia de poda PCut mostrou um desempenho mais estável devido ao uso de informações sobre a distribuição das categorizas no conjunto de treinamento, com a desvantagens de não ser capaz de tomar decisões *online*. RCut é adequada para respostas *online* mas sua decisão de corte é independente da decisão do categorizador. RTCut elaborada para atenuar o compromisso entre revocação e precisão, superar RCut.

Lee, em 2002 [Lee02], propôs a estratégia de poda de *ranking* de categorias RinScut que usa o melhor das estratégias de poda SCut e RCut. A estratégia RinScut utiliza a estratégia SCut para tratar o problema de categorização multi-rótulo de texto em que um

documento pertence a várias categorias. A estratégia de poda SCut encontra diferentes τ_i para cada categoria otimizando a performance local (por categoria) enquanto que o RCut retorna a mesma quantidade de categorias para todos os documentos otimizando a performance global. Lee definiu a estratégia Rank-in-score para usar a força dessas duas estratégias. O RinScut encontra dois pontos de corte para cada categoria (S_{top} *e* S_{botton}) a partir da estratégia SCut. Os graus de crença dos documentos que estiverem acima do S_{top} são retorndos diretamente como resposta para cada categoria c_i. Os graus de crença que estiverem entre o S_{top} e S_{botton} são considerados como zona de ambigüidades. Para resolver a ambiguide é utilizado o estratégia de poda RCut para tomar a decisão final. Nos experimentos de Lee a estratégia de poda RinScut apresenta melhor desempenho para *micro* e *macro-averaged* F_1 do que os valores encontrados pela estratégia de poda SCut.

Fan, em 2007 [Fan07], apresentou uma análise sobre a estratégia de poda SCutFBR, uma heurística proposta por Yang, para o categorizador SVM (*Support Vector Machines*) empregando as bases de dados RCV1-V2, Scene, Yeast, Ohsumed e Yahoo! Directories. Os experimentos realizados por Fan foram otimizados segundo as métricas de avaliação: *macro-average F-measure*, *micro-average F-measure* e *exact match ratio*. Fan observou os seguintes aspectos em seus experimentos: limiares de poda de *ranking* muito alto/baixo ocorrem quando a base de dados está desiquilibrada. Fan também descobriu que a *micro-average F-measure* é menos sensível para a otimização do que outras métricas, como por exemplo, a *macro-average*. Além disso, verificou que a otimização circular para a estratégia de poda SCutFBR deveria melhorar o desempenho da calibração, os seus resultados mostram que houve uma melhor apenas marginal, portanto, calibrar os limiares de corte para cada categoria sem rotacionar é suficiente na prática.

Este trabalho propõe um método para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto um problema ainda pouco explorado na área de RI e propomos também duas estratégias de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto, a qual denominamos *bayes cut* (BCut) e *position bayes cut* (PBCut). Além disso, avalimos o efeito da nossa estratégia de poda sobre as métricas mais populares de avaliação de desempenho de categorização multi-rótulo de texto empregadas pela comunidade de RI.

7.2 Análise Crítica deste Trabalho

Uma das limitações deste trabalho é a falta de uma visão geral do desempenho das estratégias de poda de *ranking*, ou seja, uma análise do *test-t* para todas as possíveis combinações de estratégias de poda. Outra possível limitação é o emprego de base de dados dentro de um único domínio de problemas de categorização. Isso poderia ser resolvido pela validação dos nossos experimentos com *benchmarks* utilizados nas principais literaturas sobre categorização de texto. Isso poderia ser resolvido pela validação dos nossos experimentos com benchmarks utilizados nas principais literaturas sobre categorização de texto.

8 CONCLUSÃO E TRABALHOS FUTUROS

Neste capítulo apresentamos um sumário do trabalho, nossas conclusões e propostas de trabalhos futuros.

8.1 Sumário

Neste trabalho, propomos um método para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto. Nosso método é baseado na regra de *Bayes*, que permite alterar as probabilidades *a priori* tendo em conta novas evidências de forma a obter as probabilidades *a posteriori*. Propomos também, uma estratégia para determinar limiares de poda para o *ranking* de categorias baseada na medida de certeza de categorização multi-rótulo de texto descrita acima, a qual denominamos *bayesian cut* (BCut). Na estratégia de poda BCut, um único limiar de poda, τ, para todas as categorias c_i é escolhido de modo a maximizar o desempenho de categorização, i.e., sua habilidade de atribuir todas e apenas as categorias pertinentes a um dado documento. Além desta estratégia, propomos uma variante para BCut que utiliza diferentes limiares de poda τ_p para diferentes posições p do *ranking*, a qual denominamos *position based bayesian cut* (PBCut). A estratégia de poda PBCut pode produzir um desempenho superior ao de BCut, porque a medida de certeza de categorização em uma dada categoria diminui à medida que a posição da categoria no *ranking* aumenta.

Além disso, investigamos o impacto no desempenho de categorização multi-rótulo de texto de três métodos de poda comumente usados na literatura de RI [Yang01, Lee02, Fan07]: RCut, PCut, SCut e uma variante de RCut - RTCut [Yang01], também, propomos novas variantes para PCut e SCut – PCut* e SCut*, respectivamente – para tratar problemas existentes nestas abordagens. Em nossa análise experimental, utilizamos os categorizadores *ML-kNN* e *VG-RAM WNN-COR* segundo as métricas de avaliação: *exact match*, *precision*, *recall* e F_1. Os experimentos foram realizados com duas bases de dados, contendo documentos textuais descrevendo atividades econômicas de empresas brasileiras, com características diferenciadas em termos de freqüência de ocorrência das categorias: AT100 e

EX100. A base de dados EX100 contém documentos categorizados dentro de 105 categorias, onde cada categoria ocorre exatamente em 100 diferentes documentos; e a base de dados AT100 contém documentos categorizados dentro de 692 categorias, onde cada categoria ocorre em até 100 diferentes documentos.

Nossos resultados experimentais mostraram que, os valores das medidas de certeza calculados analiticamente são próximos dos valores encontrados empiricamente, além disso, as estratégias de poda baseadas na medida de certeza afetam significativamente o desempenho dos categorizadores.

8.2 Conclusões

Os resultados experimentais apresentados no Capítulo 6 - na Seção 6.1 – mostram o quão os valores das medidas de certeza (valores de $p(x|y,k)$) do categorizador calculados analiticamente por meio da regra de *Bayes* são próximos (semelhantes) aos valores de $p(x|y,k)$ estimados empiricamente, demonstrando que é possível prever o quão certo está o categorizador quanto uma categoria pertencente ao *ranking* de saída ser pertinente para um dado documento em uma posição. Após compararmos os resultados das medidas de certeza para os categorizadores *ML-k NN* e *VG-RAM WNN-COR* para as bases de dados AT100 e EX100 concluimos que nosso modelo de mapeamento de grau de crença em probabilidade pode ser utilizado para tal definição, visto que, os resultados calculados análiticamente estão muito próximos dos resultados obtidos empiricamente.

Os resultados experimentais apresentados no Capítulo 6 – na Seção 6.2 - mostram que o desempenho de um categorizador segundo uma determinada métrica é significativamente diferente (teste t pareado bicaudal com nível de significância 5%) dependendo da estratégia de poda de *ranking* empregada. Os experimentos realizados com a base de dados AT100 mostram que, o desempenho dos categorizadores *ML-k NN* e *VG-RAM WNN-COR* ao empregar as estratégias de poda de *ranking* BCut e PBCut é otimizado ao ser avaliados por métricas de precisão. Este comportamento é repetido ao realizar experimentos com a base de dados EX100.

Nas métricas que avaliam precisão, as estratégias de poda BCut e PBCut são mais apropriadas para este parâmetro de avaliação, pois utilizam a medida de certeza de categorização que nada mais é que a probabilidade da predição estar correta. As estratégias de

poda de *ranking* Scut e PCut * são mais apropriadas para serem empregas às métricas que avaliam a revocação da categorização. Então, este trabalho demostra que é possível mapear os valores de graus de crenças em valores probabilísticos na categorização automática de texto e que na definição de estratégias de poda, uma das informações que devem ser consideradas é o tipo de avaliação que o sistema de categorização será submetido.

8.3 Trabalhos Futuros

Os resultados satisfatórios obtidos neste trabalho motivam continuar as pesquisas sobre modelos para mapear graus de crença em medidas de certeza de categorização multi-rótulo de texto e sobre estratégias para determinar limiares de poda de *ranking* de categorias baseada na medida de certeza de categorização.

Uma direção para trabalho futuro seria correlacionar a estratégia de poda de *ranking* de categorias com as características das bases de dados empregadas, o que poderia levar a uma abordagem mais genérica, mais independente da base de dados. Outra direção para pesquisas futuras seria utilizar *benchmarks* de problemas de categorização em domínios diferentes daquele utilizado neste trabalho. Finalmente, outra direção para pesquisa seria utilizar outras técnicas de categorização de texto multi-rótulo, o que permitiria verificar o nível de generalização do nosso modelo de medida de certeza e das estratégias de poda de *ranking* nele baseadas.

9 REFERÊNCIAS BIBLIOGRÁFICAS

[Aiolli08] F. Aiolli, R. Cardin, F. Sebastiani, and A. Sperduti. Preferential Text Classification: Learning Algorithms and Evaluation Measures. Information Retrieval Journal, pages 1386-4564, 2008.

[Aleksander98] I. Aleksander. RAM-Based Neural Networks, chapter From WISARD to MAGNUS: a Family of Weightless Virtual Neural Machines, pages 18–30. World Scientific, 1998.

[Antiqueira05] L. Antiqueira. Obtenção e Associação de Termos na Construção de uma Ontologia para a Área de Nanotecnologia. São Carlos: USP, 2005. 40 p. Monografia de Graduação – Instituto de Ciências Matemáticas e de Computação, USP, São Carlos, 2005.

[Aspell08] ASPELL. GNU Aspell. Disponível: http://aspell.net/ . Último acesso em: 20 de Agosto de 2008.

[Badue08] C. Badue, F. Pedroni, and A. F. De Souza. Multi-Label Text Categorization using VG-RAM Weightless Neural Networks. Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08), pp. 105-110, Salvador, Bahia, Brazil, October 2008.

[Baeza99] R. Baeza-Yates, and B. Ribeiro-Neto. Modern Information Retrieval. 1. ed. New York: Addison-Wesley, 1999.

[Baoli03] L. Baoli, Y. Shiwen, and L. Qin. An Improved k-Nearest Neighbor Algorithm for Text Categorization. In Proceedings of the 20th International Conference on Computer Processing of Oriental Languages, Shen Yang, China, pages 469-475, 2003.

[Boutell04] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning Multi-Label Scene Classification. Pattern Recognition, 37(9): pages 1757–1771, 2004.

[Cherman07] E. Cherman, H. de Lee, D. Honorato, C. Coy, J. Fagundes, J. Góes, F. Wu. Metodologia de Mapeamento Automático de Laudos Colonoscópicos. XVI EAIC, 2007.

[Ciarelli08] P. M. Ciarelli. Rede Neural Probabilístca para a Classicação de Atividades Econômicas. Vitória: UFES, 2008. 82 p. Dissertação – Programa de Pós-Graduação em Engenharia Elétrica, Universidade Federal do Espírito Santo, Vitória, 2008.

[Ciarelli09] P. M. Ciarelli, E. Oliveira, and C. Badue. Multi-Label Text Categorization Using a Probabilistic Neural Network. International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM), July 2009 (accepted for publication).

[Clare01] A. Clare and R. D. King. Knowledge Discovery in Multi-Label Phenotype Data. In Lecture Notes in Computer Science, volume 2168, pages 42–53, 2001.

[CNAE03] CNAE. Classificação Nacional de Atividades Econômicas – Fiscal (CNAE-Fiscal) 1.1. Instituto Brasileiro de Geografia e Estatística (IBGE), Rio de Janeiro, RJ, 2003.

[Cohen96] W. W. Cohen and Y. Singer. Context-sensitive Learning Methods for Text Categorization. In SIGIR'96: Proceedings of the 19th Annual International ACM SIGIR Conference on Research and Development in Information Retrievel, 1996. 307-315.

[Comité03] F. D. Comité, R. Gilleron, and M. Tommasi. Learning multi-label alternating decision tree from texts and data. In Lecture Notes in Computer Science, volume 2734, pages 35–49. Springer, 2003.

[Cooper68] W.S Cooper. Expected search length: A Single Measure of Retrieval Effectiveness Based on Weak Ordering Action of Retrieval Systems. Journal of the American Society for Information Science, 19(1), pages 30 – 41, 1968.

[Crowell03] J. Crowell, Q.T. Zeng, S.Kogan. A Technique to Improve the Spelling Suggestion Rank in Medical Queries. AMIA 2003 Symposium Proceedings, page 823, 2003.

[DeSouza07] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, and L. Veronese. Automated Free Text Classification of Economic Activities using VG-RAM Weightless Neural Networks. In 7th IEEE International Conference on Intelligent Systems Design and Applications, pages 782–787. IEEE Computer Society, 2007.

[DeSouza08] A. F. De Souza, C. Badue, B. Z. Melotti, F. T. Pedroni, and F. L. L. Almeida. Improving VG-RAM WNN Multi-Label Text Categorization via Label Correlation. In 2nd Workshop on Intelligent Text Categorization and Clustering (WITCC'08), 8th IEEE International Conference on Intelligent Systems Design and Applications (ISDA'08), volume 01, pages 437–442. IEEE Computer Society, 2008.

[DeSouza09a] A. F. De Souza, F. Pedroni, E. Oliveira, P. M. Ciarelli, W. F. Henrique, L. Veronese, and C. Badue. Automated Multi-label Text Categorization with VG-RAM Weightless Neural Networks. Neurocomputing, vol. 72, no. 10-12, pp. 2209-2217, June 2009.

[DeSouza09b] A. F. De Souza, B. Z. Melotti, and C. Badue. Multi-Label Text Categorization with a Data Correlated VG-RAM Weightless Neural Network. International Journal of Computer Information Systems and Industrial Management Applications (IJCISIM), July 2009 (accepted for publication).

[Dunlop97] M. D Dunlop. Time Relevance and Interaction Modeling for Information Retrieval, in Proc. ACM SIGIR, pages 206-213, 1997.

[Elisseeff02] A. Elisseeff and J. Weston. A Kernel Method for Multi-Labelled Classification. In Advances in Neural Information Processing Systems, volume 14, pages 681–687. MIT Press, 2002.

[Fagin03] R. Fagin, R. Kumar, and D. Sivakumar. Comparing Top k Lists. Proceedings of the fourteenth annual ACM-SIAM symposium on Discrete algorithms, pages 28–36, Philadelphia, USA, 2003.

[Fagin04] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, and E. Vee. Comparing and Aggregating Rankings with Ties. Proceedings of the twenty-third ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems, pages 47-58, France, 2004.

[Fagin06] R. Fagin, R. Kumar, M. Mahdian, D. Sivakumar, & E. Vee. Comparing partial rankings. SIAM Journal on Discrete Mathematics, 20(3), pages 628–648, 2006.

[Fan07] R.-E. Fan and C.-J. Lin. A Study on Threshold Selection for Multi-Label Classification. Technical Report, National Taiwan University, 2007.

[Gao04] S. Gao, W. Wu, C.-H. Lee, and T.-S. Chua. A MfoM Learning Approach to Robust Multiclass Multi-Label Text Categorization. In Proceedings of the 21st International Conference on Machine Learning, pages 329–336, 2004.

[Hair05] J. F. Hair, R. E. Anderson, R. L. Tatham e W. C. Black. Análise Multivariada de Dados. Tradução por Adonai Schlup Sant'Ana e Anselmo Chavese Neto. Quinta Edição. US, 2005.

[Hao07] X. Hao, X. Tao, C. Zhang. Yunfa Hu, An Effective Method To Improve kNN Text Classifier. Software Engineering, Artificial Intelligence, Networking, and Parallel/Distributed Computing, 2007. SNPD 2007. Eighth ACIS International Conference, vol.1, no., pages 379-384, July 30 2007-Aug. 1 2007.

[Haykin99] S. Haykin. Redes Neurais Princípios e práticas. 2ª Edição. São Paulo, 1999.

[Hull93] D. Hull. Using Statistical Testing in the Evaluation of Retrieval Experiments. Proceedings of the 16th annual international ACM SIGIR conference on Research and development in information retrieval, pages 329-338, USA, 1993.

[Joachims98] Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. Proceedings of the Tenth European Conference on Machine Learning (ECML'98), Springer Verlag, pages 137-142, 1998.

[Kazawa05] H. Kazawa, T. Izumitani, H. Taira, and E. Maeda. Maximal Margin Labeling for Multi-Topic Text Categorization. In Advances in Neural Information Processing Systems 17, pages 649–656. MIT Press, 2005.

[Lee02] K. H. Lee, J. Kay, and B. H. Kang. Lazy Linear Classifier and Rank-in-Score Threshold in Similarity-Based Text Categorization. International Conference on Machine Learning Workshop on Text Learning TextML'2002), Sydney, Australia, pages 36-43, July 8, 2002.

[Lewis92] D. Lewis. An Evaluation of Phrasal and Clustered Representations on a Text Categorization task. In 15th Ann Int ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'92), pages 37 50, 1992.

[Lewis94] D. Lewis and M. Ringuette. Comparison of two Learning Algorithms for Text Categorization. In Proceedings of the Third Annual Symposium on Document Analysis and Information Retrieval (SDAIR'94), Nevada, Las Vegas, 1994. University of Nevada, Las Vegas.

[Lewis96] D.D. Lewis et al. Training Algorithms Linear Text Classifier. In Proc. Of the 19 th annual international ACM SIGIR Conference on Research and development in information retrieval SIGIR'96), pages 298 306, 1996.

[Ludermir99] T. B. Ludermir, A. C. P. L. F. Carvalho, A. P. Braga, and M. D. Souto. Weightless Neural Models: A Review of Current and Past Works. Neural Computing Surveys, 2: pages 41–61, 1999.

[Manning08] C. D. Manning, P. Raghavan, and H. Schútze. An Introduction to Information Retrieval. Cambridge University Press, Cambridge, England, 2008.

[Martins04] D. Martins, and M. J Silva. Spelling Correction for Search Engine Queries. In Book Series of Lecture Notes in Computer Science, Vol. 3230, pages 372-383, 2004.

[McCallum99] A. McCallum. Multi-Label Text Classification with a Mixture Model Trained by EM. In Working Notes of the AAAI'99 Workshop on Text Learning, pages 1–7, 1999.

[Melotti09] Z. B. Melotti. Efeito do *Ranking* sobre Métricas de Categorização Multi-Rótulo de texto. Vitória: UFES, Dissertação - Programa de Pós-Graduação em Informática, Universidade Federal do Espiríto Santos, Vitória, 2009.

[Mitchell97] T. M. Mitchell. Machine learning. McGraw Hill, New York, US, 1997.

[Mitchell98] R. J. Mitchell, J. M. Bishop, S. K. Box, and J. F. Hawker. RAM-Based Neural Networks, chapter Comparison of Some Methods for Processing Grey Level Data in Weightless Networks, pages 61–70. World Scientific, 1998.

[Monard03] M. C. Monard & J. A. Baranauskas. Conceitos sobre Aprendizado de Máquina. In Sistemas Inteligentes – Fundamentos e Aplicações, S.O Rezende, Editora Manole, pages 89-114, 2003.

[Oliveira08a] E. Oliveira, P. M. Ciarelli, A. F. De Souza, and C. Badue. Using a Probabilistic Neural Network for a Large Multi-Label Problem. Proceedings of the 10th Brazilian Symposium on Neural Networks (SBRN'08), pp. 195-200, Salvador, Bahia, Brazil, October 2008.

[Oliveira08b] E. Oliveira, P. M. Ciarelli, and C. Badue. A Comparison Between a kNN based Aproach and a PNN Algorithm for a Multi-Label Classification Problem. Proceedings of the 2nd Workshop on Intelligent Text Categorization and Clustering of the 8th International Conference on Inteligent System Design and Applications (ISDA'08), pp. 628-633, Kaohsiung City, Taiwan, November 2008.

[Picard84] R. R. Picard, and R. D. Cook. Cross-Validation of Regression Models. Journal of the American Statistical Association, 79(387), pages 575–583, 1984.

[Rijsbergen79] V. Rijsbergen, C. J. Information Retrieval (Second ed.). Butterworths, London, UK, 1979. Available at http://www.dcs.gla.ac.uk/Keith.

[Romero04] E. Romero, L. Màrquez, and X. Carreras. Margin Maximization with Feed-Forward Neural Networks: A Comparative Study with SVM and Adaboost. Neurocomputing, 57: pages 313–344, 2004.

[Salton75] G. Salton, A. Wong, and C. Yang. A vector space model for automatic indexing. Communications of the ACM 18, 11, 613–620. Also reprinted in [Sparck Jones and Willett 1997], pages 273–280, 1975.

[Sbc09] Sociedade Brasileira de Computação, Grandes desafios da pesquisa em computação no Brasil 2006-2016. Último acesso em 12/08/2009.

[SCAE08] Sistema Computacional de Codificação Automática de Atividades Econômicas (SCAE), Projeto de Classificação Automática em CNAE-Subclasses – Relato de Cumprimento de Metas No. 4. Universidade Federal do Espírito Santo, Vitória, 2008.

[Schapire99] R. E. Schapire and Y. Singer. Improved Boosting Algorithms Using Confidence-Rated Predictions. Machine Learning, 27(3): pages 297–336, 1999.

[Schapire00] R. E. Schapire and Y. Singer. BoosTexter: A boosting-Based System for Text Categorization. Machine Learning, 39(2/3): pages 135–168, 2000.

[Sebastiani02] F. Sebastiani. Machine learning in automated text categorization. ACM

Computing Surveys, 34(1): pages 1–47, 2002.

[Sparacino00] G. Sparacino, C. Tombolato, C. Cobelli. Maximum-Likelihood versus Maximum a Posteriori Parameter Estimation of Physiological System Models: the c-peptide impulse response case study. Biomedical Engineering, IEEE Transactions on Volume 47, Issue 6, pages 801 – 811, June 2000.

[Student08] Student. The Probable Error of a Mean. Biometrika on Volume 6, pages 1 – 25, 1908.

[Ueda03] N. Ueda and K. Saito. Parametric Mixture Models for Multi-Label Text. In Advances in Neural Information Processing Systems, volume 15, pages 721–728. MIT Press, 2003.

[Witten05] Ian H. Witten & E. Frank. Data Mining Practical Machine Learning Tools and Techniques. Second Edition. US, 2005.

[Yang99] Y. Yang. An Evaluation of Statistical Approaches to Text Categorization. In Information Retrieval, Volume 1, pages 69-90, Hingham, US, 1999.

[Yang01] Y. Yang. A Study of Thresholding Strategies for Text Categorization. In Proceedings of the 24th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01), pages 137–145, New Orleans, Louisiana, United States, 2001.

[Yavuz98] T. Yavuz and H. Altay Guvenir. Application of k-nearest Neighbor on Feature Projections Classifier to Text Categorization. Proceedings of ISCIS, 13th International Symposium on Computer and Information Sciences, pages 135-142, 1998.

[Zhang06] M.-L. Zhang, Z.-H. Zhou,. Multi-label Neural Networks with Applications to Functional Genomics and Text Categorization. IEEE Transactions on Knowledge and Data Engineering 18(10), pages 1338–1351, 2006.

[Zhang07] M.-L. Zhang and Z.-H. Zhou. ML-KNN: A Lazy Learning Approach to Multi-Label Learning. Pattern Recognition, 40(7): pages 2038–2048, 2007.

APÊNDICE A – PARÂMETROS OBTIDOS NO PROCEDIMENTO DE CALIBRAÇÃO DE SCUT

A.1 Parâmetros obtidos no procedimento de calibração de SCut para o categorizador ML-*k* NN e para a base AT100

O apêndice A apresenta os parâmetros obtidos no procedimento de calibração de SCut para os categorizadores ML-*k* NN e VG-RAM WNN-COR e para as bases de dados AT100 e EX100. A Tabela 9-1, Tabela 9-2, Tabela 9-3, Tabela 9-4 e

Tabela 9-5 mostram os parâmetros obtidos do procedimento de calibração do SCut aplicado ao categorizador ML-*k* NN para as bases de dados EX100 e AT100.

APÊNDICE A

Tabela 9-1 – Parâmetros obtidos no procedimento de calibração de SCut para *ML-k NN* e para a base AT100.

τ1	0,0005	***τ21***	0,0000	***τ41***	0,0013	***τ61***	0,0100	***τ81***	0,0010	***τ101***	0,0003	***τ121***	0,0059	***τ141***	0,0003	***τ161***	0,0280	***τ181***	0,0080
τ2	0,0003	***τ22***	0,0148	***τ42***	0,0010	***τ62***	0,0100	***τ82***	0,0010	***τ102***	0,0080	***τ122***	0,0012	***τ142***	0,0066	***τ162***	0,0290	***τ182***	0,0010
τ3	0,0073	***τ23***	0,0003	***τ43***	0,0056	***τ63***	0,0473	***τ83***	0,0010	***τ103***	0,0160	***τ123***	0,0005	***τ143***	0,0012	***τ163***	0,0000	***τ183***	0,1390
τ4	0,0003	***τ24***	0,0020	***τ44***	0,0190	***τ64***	0,0037	***τ84***	0,0001	***τ104***	0,0191	***τ124***	0,0016	***τ144***	0,0020	***τ164***	0,0040	***τ184***	0,0320
τ5	0,0039	***τ25***	0,0697	***τ45***	0,0010	***τ65***	0,0700	***τ85***	0,0235	***τ105***	0,0281	***τ125***	0,0002	***τ145***	0,0390	***τ165***	0,0022	***τ185***	0,0008
τ6	0,0286	***τ26***	0,0977	***τ46***	0,0010	***τ66***	0,0550	***τ86***	0,0031	***τ106***	0,0001	***τ126***	0,0005	***τ146***	0,0014	***τ166***	0,0015	***τ186***	0,0005
τ7	0,0002	***τ27***	0,0013	***τ47***	0,0012	***τ67***	0,0005	***τ87***	0,0320	***τ107***	0,0001	***τ127***	0,0003	***τ147***	0,0020	***τ167***	0,0002	***τ187***	0,0002
τ8	0,0010	***τ28***	0,0005	***τ48***	0,0170	***τ68***	0,0003	***τ88***	0,2710	***τ108***	0,0001	***τ128***	0,0006	***τ148***	0,0005	***τ168***	0,0043	***τ188***	0,0023
τ9	0,0049	***τ29***	0,0025	***τ49***	0,0006	***τ69***	0,0006	***τ89***	0,0480	***τ109***	0,0005	***τ129***	0,0013	***τ149***	0,0007	***τ169***	0,0023	***τ189***	0,0052
τ10	0,0317	***τ30***	0,0009	***τ50***	0,0009	***τ70***	0,0010	***τ90***	0,0193	***τ110***	0,0023	***τ130***	0,0009	***τ150***	0,0050	***τ170***	0,0081	***τ190***	0,0027
τ11	0,0042	***τ31***	0,0017	***τ51***	0,0001	***τ71***	0,0005	***τ91***	0,0180	***τ111***	0,0032	***τ131***	0,0127	***τ151***	0,0160	***τ171***	0,0006	***τ191***	0,0001
τ12	0,0012	***τ32***	0,0004	***τ52***	0,0011	***τ72***	0,0005	***τ92***	0,0034	***τ112***	0,0066	***τ132***	0,0001	***τ152***	0,0003	***τ172***	0,0151	***τ192***	0,0532
τ13	0,0009	***τ33***	0,0096	***τ53***	0,0252	***τ73***	0,0008	***τ93***	0,0120	***τ113***	0,0019	***τ133***	0,0011	***τ153***	0,0030	***τ173***	0,0128	***τ193***	0,0147
τ14	0,0034	***τ34***	0,0019	***τ54***	0,0080	***τ74***	0,0013	***τ94***	0,0320	***τ114***	0,0015	***τ134***	0,0003	***τ154***	0,0011	***τ174***	0,0070	***τ194***	0,0007
τ15	0,0004	***τ35***	0,0004	***τ55***	0,0286	***τ75***	0,0010	***τ95***	0,0380	***τ115***	0,0041	***τ135***	0,0013	***τ155***	0,0018	***τ175***	0,0900	***τ195***	0,0060
τ16	0,0001	***τ36***	0,0033	***τ56***	0,0004	***τ76***	0,0023	***τ96***	0,0026	***τ116***	0,0002	***τ136***	0,0006	***τ156***	0,0109	***τ176***	0,0014	***τ196***	0,0050
τ17	0,0060	***τ37***	0,0119	***τ57***	0,0095	***τ77***	0,0040	***τ97***	0,0004	***τ117***	0,0177	***τ137***	0,0048	***τ157***	0,0227	***τ177***	0,0004	***τ197***	0,0022
τ18	0,0013	***τ38***	0,0105	***τ58***	0,0235	***τ78***	0,0024	***τ98***	0,0076	***τ118***	0,2180	***τ138***	0,0070	***τ158***	0,0048	***τ178***	0,0320	***τ198***	0,0001
τ19	0,0030	***τ39***	0,0073	***τ59***	0,0015	***τ79***	0,0120	***τ99***	0,0050	***τ119***	0,0167	***τ139***	0,0141	***τ159***	0,0030	***τ179***	0,1270	***τ199***	0,0013
τ20	0,0118	***τ40***	0,0004	***τ60***	0,0021	***τ80***	0,0140	***τ100***	0,0001	***τ120***	0,2090	***τ140***	0,0016	***τ160***	0,0095	***τ180***	0,0235	***τ200***	0,0012

APÊNDICE A

Tabela 9-2 - Parâmetros obtidos no procedimento de calibração de SCut para *ML-k NN* e para a base AT100.

τ201	0,0094	***τ221***	0,1410	***τ241***	0,0512	***τ261***	0,0027	***τ281***	0,0030	***τ301***	0,0590	***τ321***	0,1360	***τ341***	0,0037	***τ361***	0,0010	***τ381***	0,0010
τ202	0,0764	***τ222***	0,0020	***τ242***	0,0200	***τ262***	0,0074	***τ282***	0,0530	***τ302***	0,0790	***τ322***	0,0023	***τ342***	0,0113	***τ362***	0,0030	***τ382***	0,0023
τ203	0,0164	***τ223***	0,0087	***τ243***	0,0004	***τ263***	0,0493	***τ283***	0,0060	***τ303***	0,0043	***τ323***	0,0589	***τ343***	0,0103	***τ363***	0,1306	***τ383***	0,0209
τ204	0,0023	***τ224***	0,0001	***τ244***	0,0040	***τ264***	0,0090	***τ284***	0,0440	***τ304***	0,0380	***τ324***	0,0230	***τ344***	0,0050	***τ364***	0,0060	***τ384***	0,0280
τ205	0,0060	***τ225***	0,0101	***τ245***	0,0012	***τ265***	0,0028	***τ285***	0,0172	***τ305***	0,0020	***τ325***	0,0023	***τ345***	0,0050	***τ365***	0,0024	***τ385***	0,0224
τ206	0,0035	***τ226***	0,0001	***τ246***	0,0163	***τ266***	0,0374	***τ286***	0,0026	***τ306***	0,0330	***τ326***	0,0720	***τ346***	0,0685	***τ366***	0,0014	***τ386***	0,0345
τ207	0,0330	***τ227***	0,0069	***τ247***	0,0121	***τ267***	0,0013	***τ287***	0,0017	***τ307***	0,0710	***τ327***	0,1070	***τ347***	0,0010	***τ367***	0,0050	***τ387***	0,0060
τ208	0,0024	***τ228***	0,0003	***τ248***	0,0009	***τ268***	0,0001	***τ288***	0,0020	***τ308***	0,0285	***τ328***	0,0020	***τ348***	0,0002	***τ368***	0,0320	***τ388***	0,0005
τ209	0,0348	***τ229***	0,0032	***τ249***	0,0017	***τ269***	0,0026	***τ289***	0,0271	***τ309***	0,0118	***τ329***	0,0160	***τ349***	0,0160	***τ369***	0,0057	***τ389***	0,1770
τ210	0,0021	***τ230***	0,0218	***τ250***	0,0362	***τ270***	0,0040	***τ290***	0,0082	***τ310***	0,0910	***τ330***	0,1110	***τ350***	0,0075	***τ370***	0,0158	***τ390***	0,0280
τ211	0,0094	***τ231***	0,0010	***τ251***	0,0003	***τ271***	0,0120	***τ291***	0,0060	***τ311***	0,0119	***τ331***	0,1690	***τ351***	0,0090	***τ371***	0,0020	***τ391***	0,0010
τ212	0,0190	***τ232***	0,0080	***τ252***	0,0002	***τ272***	0,0012	***τ292***	0,0510	***τ312***	0,0002	***τ332***	0,2120	***τ352***	0,0114	***τ372***	0,0005	***τ392***	0,0002
τ213	0,0109	***τ233***	0,0007	***τ253***	0,0024	***τ273***	0,0438	***τ293***	0,0024	***τ313***	0,1850	***τ333***	0,0410	***τ353***	0,2503	***τ373***	0,0443	***τ393***	0,0060
τ214	0,0069	***τ234***	0,0001	***τ254***	0,0022	***τ274***	0,0010	***τ294***	0,0030	***τ314***	0,1680	***τ334***	0,1500	***τ354***	0,0216	***τ374***	0,0022	***τ394***	0,0018
τ215	0,0013	***τ235***	0,0030	***τ255***	0,0026	***τ275***	0,0029	***τ295***	0,0005	***τ315***	0,1220	***τ335***	0,0094	***τ355***	0,0057	***τ375***	0,0177	***τ395***	0,0005
τ216	0,0014	***τ236***	0,0029	***τ256***	0,0005	***τ276***	0,0043	***τ296***	0,0052	***τ316***	0,0990	***τ336***	0,0010	***τ356***	0,0110	***τ376***	0,0029	***τ396***	0,0048
τ217	0,0012	***τ237***	0,0010	***τ257***	0,0006	***τ277***	0,0010	***τ297***	0,0025	***τ317***	0,2830	***τ337***	0,0150	***τ357***	0,0010	***τ377***	0,0445	***τ397***	0,0207
τ218	0,0006	***τ238***	0,0003	***τ258***	0,0167	***τ278***	0,0037	***τ298***	0,0140	***τ318***	0,0270	***τ338***	0,0003	***τ358***	0,0007	***τ378***	0,0003	***τ398***	0,0230
τ219	0,0001	***τ239***	0,0001	***τ259***	0,0012	***τ279***	0,0010	***τ299***	0,0190	***τ319***	0,3470	***τ339***	0,0014	***τ359***	0,0170	***τ379***	0,0004	***τ399***	0,0020
τ220	0,0013	***τ240***	0,0015	***τ260***	0,0013	***τ280***	0,0010	***τ300***	0,1590	***τ320***	0,0044	***τ340***	0,0297	***τ360***	0,0030	***τ380***	0,0025	***τ400***	0,0560

APÊNDICE A

Tabela 9-3 - Parâmetros obtidos no procedimento de calibração de SCut para *ML-k NN* e para a base AT100.

τ401	0,0010	***τ421***	0,1660	***τ441***	0,0570	***τ461***	0,0300	***τ481***	0,0037	***τ501***	0,0010	***τ521***	0,0002	***τ541***	0,0045	***τ561***	0,0046	***τ581***	0,0510
τ402	0,0780	***τ422***	0,1360	***τ442***	0,0070	***τ462***	0,0123	***τ482***	0,0450	***τ502***	0,0002	***τ522***	0,0031	***τ542***	0,0011	***τ562***	0,1680	***τ582***	0,2587
τ403	0,0320	***τ423***	0,1900	***τ443***	0,0626	***τ463***	0,0330	***τ483***	0,0430	***τ503***	0,0055	***τ523***	0,0386	***τ543***	0,0130	***τ563***	0,0280	***τ583***	0,1340
τ404	0,1000	***τ424***	0,2210	***τ444***	0,0980	***τ464***	0,0619	***τ484***	0,0459	***τ504***	0,0260	***τ524***	0,0360	***τ544***	0,3810	***τ564***	0,3220	***τ584***	0,1620
τ405	0,0020	***τ425***	0,0480	***τ445***	0,0370	***τ465***	0,0010	***τ485***	0,0077	***τ505***	0,1470	***τ525***	0,0020	***τ545***	0,0330	***τ565***	0,0035	***τ585***	0,0790
τ406	0,0064	***τ426***	0,0130	***τ446***	0,0420	***τ466***	0,0130	***τ486***	0,0030	***τ506***	0,0180	***τ526***	0,1570	***τ546***	0,0343	***τ566***	0,1130	***τ586***	0,0680
τ407	0,2370	***τ427***	0,0460	***τ447***	0,0131	***τ467***	0,0280	***τ487***	0,0620	***τ507***	0,0110	***τ527***	0,0067	***τ547***	0,0018	***τ567***	0,1020	***τ587***	0,0006
τ408	0,0610	***τ428***	0,1650	***τ448***	0,0140	***τ468***	0,0019	***τ488***	0,1300	***τ508***	0,0006	***τ528***	0,0140	***τ548***	0,2300	***τ568***	0,1020	***τ588***	0,0048
τ409	0,0630	***τ429***	0,1010	***τ449***	0,1180	***τ469***	0,1880	***τ489***	0,0007	***τ509***	0,0078	***τ529***	0,0790	***τ549***	0,1140	***τ569***	0,1320	***τ589***	0,1460
τ410	0,2330	***τ430***	0,2510	***τ450***	0,0820	***τ470***	0,1530	***τ490***	0,1690	***τ510***	0,0099	***τ530***	0,0740	***τ550***	0,1760	***τ570***	0,1320	***τ590***	0,0096
τ411	0,0600	***τ431***	0,0220	***τ451***	0,1560	***τ471***	0,0320	***τ491***	0,0550	***τ511***	0,0004	***τ531***	0,0040	***τ551***	0,2710	***τ571***	0,1050	***τ591***	0,0320
τ412	0,0850	***τ432***	0,0250	***τ452***	0,0010	***τ472***	0,0035	***τ492***	0,1330	***τ512***	0,0034	***τ532***	0,0060	***τ552***	0,2280	***τ572***	0,1970	***τ592***	0,0007
τ413	0,0600	***τ433***	0,0710	***τ453***	0,0002	***τ473***	0,0150	***τ493***	0,2150	***τ513***	0,0006	***τ533***	0,1360	***τ553***	0,3010	***τ573***	0,0000	***τ593***	0,0014
τ414	0,1170	***τ434***	0,1800	***τ454***	0,0640	***τ474***	0,0050	***τ494***	0,0620	***τ514***	0,0080	***τ534***	0,0010	***τ554***	0,3520	***τ574***	0,0233	***τ594***	0,1400
τ415	0,0920	***τ435***	0,0820	***τ455***	0,0080	***τ475***	0,0150	***τ495***	0,0004	***τ515***	0,0050	***τ535***	0,0028	***τ555***	0,0323	***τ575***	0,0060	***τ595***	0,0720
τ416	0,1800	***τ436***	0,0200	***τ456***	0,0410	***τ476***	0,0204	***τ496***	0,0018	***τ516***	0,0140	***τ536***	0,0911	***τ556***	0,0020	***τ576***	0,0850	***τ596***	0,2140
τ417	0,0860	***τ437***	0,0010	***τ457***	0,0030	***τ477***	0,0180	***τ497***	0,0010	***τ517***	0,0477	***τ537***	0,0100	***τ557***	0,0010	***τ577***	0,1480	***τ597***	0,0002
τ418	0,1590	***τ438***	0,0016	***τ458***	0,0008	***τ478***	0,0590	***τ498***	0,0055	***τ518***	0,1520	***τ538***	0,0037	***τ558***	0,2220	***τ578***	0,0040	***τ598***	0,0096
τ419	0,2330	***τ439***	0,0230	***τ459***	0,0000	***τ479***	0,0032	***τ499***	0,0034	***τ519***	0,0034	***τ539***	0,0015	***τ559***	0,0190	***τ579***	0,0650	***τ599***	0,0003
τ420	0,0230	***τ440***	0,0044	***τ460***	0,0043	***τ480***	0,0645	***τ500***	0,0010	***τ520***	0,0048	***τ540***	0,0097	***τ560***	0,0027	***τ580***	0,2520	***τ600***	0,0720

Tabela 9-4 - Parâmetros obtidos no procedimento de calibração de SCut para *ML-k NN* e para a base AT100.

τ601	0,0180	***τ621***	0,1550	***τ641***	0,0299	***τ661***	0,1860	***τ681***	0,0080
τ602	0,1550	***τ622***	0,1600	***τ642***	0,0009	***τ662***	0,0035	***τ682***	0,0010
τ603	0,0130	***τ623***	0,0361	***τ643***	0,0380	***τ663***	0,0130	***τ683***	0,0004
τ604	0,0001	***τ624***	0,0480	***τ644***	0,0002	***τ664***	0,0039	***τ684***	0,1850
τ605	0,0005	***τ625***	0,0000	***τ645***	0,0091	***τ665***	0,0619	***τ685***	0,0830
τ606	0,0584	***τ626***	0,0004	***τ646***	0,0044	***τ666***	0,0361	***τ686***	0,0302
τ607	0,0012	***τ627***	0,0060	***τ647***	0,1680	***τ667***	0,0590	***τ687***	0,0060
τ608	0,0038	***τ628***	0,0010	***τ648***	0,0950	***τ668***	0,0030	***τ688***	0,0020
τ609	0,0019	***τ629***	0,1830	***τ649***	0,0200	***τ669***	0,0010	***τ689***	0,0060
τ610	0,0004	***τ630***	0,0330	***τ650***	0,1600	***τ670***	0,0020	***τ690***	0,0005
τ611	0,1900	***τ631***	0,0010	***τ651***	0,0541	***τ671***	0,0144	***τ691***	0,0407
τ612	0,0040	***τ632***	0,1900	***τ652***	0,0750	***τ672***	0,0057	***τ692***	0,0001
τ613	0,0100	***τ633***	0,0693	***τ653***	0,0030	***τ673***	0,0060		
τ614	0,0030	***τ634***	0,0037	***τ654***	0,0450	***τ674***	0,2500		
τ615	0,0422	***τ635***	0,0006	***τ655***	0,0088	***τ675***	0,0020		
τ616	0,0010	***τ636***	0,0010	***τ656***	0,0050	***τ676***	0,0050		
τ617	0,0013	***τ637***	0,0840	***τ657***	0,0070	***τ677***	0,0115		
τ618	0,0800	***τ638***	0,0018	***τ658***	0,0010	***τ678***	0,0004		
τ619	0,1040	***τ639***	0,0235	***τ659***	0,0170	***τ679***	0,0009		
τ620	0,0010	***τ640***	0,0037	***τ660***	0,0140	***τ680***	0,0020		

A.2 Parâmetros obtidos no procedimento de calibração de SCut para o categorizador ML-*k* NN e para a base EX100

Tabela 9-5 - Parâmetros obtidos no procedimento de calibração de SCut para *ML-k NN* e para a base EX100.

τ1	0,0870	***τ21***	0,0410	***τ41***	0,1570	***τ61***	0,1910	***τ81***	0,1870	***τ101***	0,0910
τ2	0,1390	***τ22***	0,1530	***τ42***	0,1060	***τ62***	0,3180	***τ82***	0,1750	***τ102***	0,1580
τ3	0,0940	***τ23***	0,1610	***τ43***	0,2370	***τ63***	0,0610	***τ83***	0,1880	***τ103***	0,4740
τ4	0,1320	***τ24***	0,1880	***τ44***	0,2590	***τ64***	0,0930	***τ84***	0,1340	***τ104***	0,1870
τ5	0,0960	***τ25***	0,2640	***τ45***	0,2440	***τ65***	0,2200	***τ85***	0,1120	***τ105***	0,1870
τ6	0,2550	***τ26***	0,1010	***τ46***	0,2330	***τ66***	0,1320	***τ86***	0,1390		
τ7	0,0870	***τ27***	0,0940	***τ47***	0,0170	***τ67***	0,1010	***τ87***	0,1580		
τ8	0,1140	***τ28***	0,1360	***τ48***	0,0800	***τ68***	0,2010	***τ88***	0,0980		
τ9	0,1010	***τ29***	0,0840	***τ49***	0,0890	***τ69***	0,1060	***τ89***	0,1220		
τ10	0,2320	***τ30***	0,1210	***τ50***	0,1350	***τ70***	0,1350	***τ90***	0,2390		
τ11	0,0610	***τ31***	0,1140	***τ51***	0,0590	***τ71***	0,0460	***τ91***	0,1320		
τ12	0,2600	***τ32***	0,1940	***τ52***	0,3230	***τ72***	0,0740	***τ92***	0,0360		
τ13	0,1520	***τ33***	0,1880	***τ53***	0,1610	***τ73***	0,0970	***τ93***	0,1070		
τ14	0,1490	***τ34***	0,4050	***τ54***	0,0630	***τ74***	0,3160	***τ94***	0,4390		
τ15	0,0670	***τ35***	0,1140	***τ55***	0,2380	***τ75***	0,1660	***τ95***	0,1410		
τ16	0,1930	***τ36***	0,1940	***τ56***	0,0960	***τ76***	0,0950	***τ96***	0,1190		
τ17	0,2110	***τ37***	0,2510	***τ57***	0,0950	***τ77***	0,0530	***τ97***	0,1320		
τ18	0,2080	***τ38***	0,1850	***τ58***	0,3190	***τ78***	0,0550	***τ98***	0,1670		
τ19	0,1050	***τ39***	0,1060	***τ59***	0,3190	***τ79***	0,1310	***τ99***	0,2320		
τ20	0,0720	***τ40***	0,0830	***τ60***	0,1010	***τ80***	0,0410	***τ100***	0,0860		

A.3 Parâmetros obtidos no procedimento de calibração de SCut para o categorizador VG-RAM WNN-COR e para a base AT100

A Tabela 9-6, Tabela 9-7, Tabela 9-8, Tabela 9-9 e Tabela 9-10 mostram os parâmetros obtidos do procedimento de calibração do SCut aplicado ao categorizador *VG-RAM WNN-COR* para as bases de dados EX100 e AT100.

APÊNDICE A

Tabela 9-6 - Parâmetros obtidos no procedimento de calibração de SCut para *VG-RAM WNN-COR* e para a base AT100.

τ1	0,0057	***τ21***	0,0016	***τ41***	0,0093	***τ61***	0,0460	***τ81***	0,0150	***τ101***	0,0050	***τ121***	0,0203	***τ141***	0,0261	***τ161***	0,0230	***τ181***	0,0390
τ2	0,0078	***τ22***	0,0219	***τ42***	0,0050	***τ62***	0,0210	***τ82***	0,0041	***τ102***	0,0080	***τ122***	0,0167	***τ142***	0,0117	***τ162***	0,0420	***τ182***	0,0120
τ3	0,0138	***τ23***	0,0126	***τ43***	0,0227	***τ63***	0,0223	***τ83***	0,0037	***τ103***	0,0210	***τ123***	0,0023	***τ143***	0,0071	***τ163***	0,0037	***τ183***	0,0431
τ4	0,0056	***τ24***	0,0067	***τ44***	0,0110	***τ64***	0,0134	***τ84***	0,0095	***τ104***	0,0369	***τ124***	0,0249	***τ144***	0,0070	***τ164***	0,0209	***τ184***	0,0250
τ5	0,0333	***τ25***	0,0350	***τ45***	0,0230	***τ65***	0,0370	***τ85***	0,0252	***τ105***	0,0430	***τ125***	0,0143	***τ145***	0,0390	***τ165***	0,0196	***τ185***	0,0027
τ6	0,0084	***τ26***	0,0266	***τ46***	0,0090	***τ66***	0,0414	***τ86***	0,0140	***τ106***	0,0101	***τ126***	0,0032	***τ146***	0,0074	***τ166***	0,0118	***τ186***	0,0169
τ7	0,0062	***τ27***	0,0073	***τ47***	0,0266	***τ67***	0,0268	***τ87***	0,1145	***τ107***	0,0256	***τ127***	0,0035	***τ147***	0,0130	***τ167***	0,0150	***τ187***	0,0069
τ8	0,0110	***τ28***	0,0085	***τ48***	0,0320	***τ68***	0,0068	***τ88***	0,0550	***τ108***	0,0043	***τ128***	0,0114	***τ148***	0,0568	***τ168***	0,0233	***τ188***	0,0181
τ9	0,0192	***τ29***	0,0123	***τ49***	0,0756	***τ69***	0,0152	***τ89***	0,0280	***τ109***	0,0048	***τ129***	0,0073	***τ149***	0,0090	***τ169***	0,0208	***τ189***	0,0238
τ10	0,0284	***τ30***	0,0130	***τ50***	0,0090	***τ70***	0,0155	***τ90***	0,0137	***τ110***	0,0153	***τ130***	0,0144	***τ150***	0,0154	***τ170***	0,0385	***τ190***	0,0076
τ11	0,0111	***τ31***	0,0129	***τ51***	0,0141	***τ71***	0,0072	***τ91***	0,0268	***τ111***	0,0185	***τ131***	0,1887	***τ151***	0,0130	***τ171***	0,0060	***τ191***	0,0080
τ12	0,0056	***τ32***	0,0077	***τ52***	0,0201	***τ72***	0,0161	***τ92***	0,0177	***τ112***	0,0228	***τ132***	0,0099	***τ152***	0,0134	***τ172***	0,0357	***τ192***	0,0276
τ13	0,0099	***τ33***	0,0348	***τ53***	0,0319	***τ73***	0,0143	***τ93***	0,0210	***τ113***	0,0069	***τ133***	0,0181	***τ153***	0,0080	***τ173***	0,0372	***τ193***	0,0080
τ14	0,0284	***τ34***	0,0712	***τ54***	0,0200	***τ74***	0,0055	***τ94***	0,0250	***τ114***	0,0114	***τ134***	0,0126	***τ154***	0,0120	***τ174***	0,0240	***τ194***	0,0122
τ15	0,0600	***τ35***	0,0020	***τ55***	0,0071	***τ75***	0,0140	***τ95***	0,0350	***τ115***	0,0358	***τ135***	0,0046	***τ155***	0,0074	***τ175***	0,0280	***τ195***	0,0060
τ16	0,0043	***τ36***	0,0172	***τ56***	0,0270	***τ76***	0,0072	***τ96***	0,0333	***τ116***	0,0063	***τ136***	0,0114	***τ156***	0,0272	***τ176***	0,0132	***τ196***	0,0160
τ17	0,0345	***τ37***	0,0461	***τ57***	0,0279	***τ77***	0,0080	***τ97***	0,0053	***τ117***	0,0349	***τ137***	0,0059	***τ157***	0,0499	***τ177***	0,0036	***τ197***	0,0152
τ18	0,0134	***τ38***	0,0574	***τ58***	0,0090	***τ78***	0,1210	***τ98***	0,0692	***τ118***	0,0870	***τ138***	0,0190	***τ158***	0,0074	***τ178***	0,0274	***τ198***	0,0029
τ19	0,0190	***τ39***	0,0032	***τ59***	0,0094	***τ79***	0,0310	***τ99***	0,0135	***τ119***	0,0233	***τ139***	0,0591	***τ159***	0,0275	***τ179***	0,0360	***τ199***	0,0032
τ20	0,0375	***τ40***	0,0070	***τ60***	0,0091	***τ80***	0,0260	***τ100***	0,0022	***τ120***	0,0440	***τ140***	0,0048	***τ160***	0,0357	***τ180***	0,0136	***τ200***	0,0195

Tabela 9-7 - Parâmetros obtidos no procedimento de calibração de SCut para *VG-RAM WNN-COR* e para a base AT100.

τ201	0,0085	***τ221***	0,0470	***τ241***	0,0534	***τ261***	0,0171	***τ281***	0,0699	***τ301***	0,0720	***τ321***	0,0560	***τ341***	0,0334	***τ361***	0,0030	***τ381***	0,0010
τ202	0,0220	***τ222***	0,0080	***τ242***	0,1070	***τ262***	0,0271	***τ282***	0,0380	***τ302***	0,0765	***τ322***	0,0140	***τ342***	0,0359	***τ362***	0,0170	***τ382***	0,0561
τ203	0,0076	***τ223***	0,0084	***τ243***	0,0219	***τ263***	0,0219	***τ283***	0,1960	***τ303***	0,0218	***τ323***	0,0548	***τ343***	0,0339	***τ363***	0,0436	***τ383***	0,0663
τ204	0,0188	***τ224***	0,0033	***τ244***	0,0180	***τ264***	0,0470	***τ284***	0,0070	***τ304***	0,0260	***τ324***	0,0220	***τ344***	0,0120	***τ364***	0,0219	***τ384***	0,0400
τ205	0,0242	***τ225***	0,0090	***τ245***	0,0036	***τ265***	0,0330	***τ285***	0,1014	***τ305***	0,0230	***τ325***	0,0096	***τ345***	0,0160	***τ365***	0,0846	***τ385***	0,0249
τ206	0,0118	***τ226***	0,0030	***τ246***	0,0171	***τ266***	0,0065	***τ286***	0,0194	***τ306***	0,0340	***τ326***	0,0290	***τ346***	0,0437	***τ366***	0,0296	***τ386***	0,0164
τ207	0,0510	***τ227***	0,0106	***τ247***	0,0139	***τ267***	0,0120	***τ287***	0,0402	***τ307***	0,0340	***τ327***	0,0580	***τ347***	0,0320	***τ367***	0,0120	***τ387***	0,0200
τ208	0,0080	***τ228***	0,0016	***τ248***	0,0075	***τ268***	0,0037	***τ288***	0,0334	***τ308***	0,0228	***τ328***	0,0060	***τ348***	0,0218	***τ368***	0,0483	***τ388***	0,0029
τ209	0,0440	***τ229***	0,0075	***τ249***	0,0073	***τ269***	0,0109	***τ289***	0,0149	***τ309***	0,0240	***τ329***	0,0280	***τ349***	0,0530	***τ369***	0,0198	***τ389***	0,0680
τ210	0,0157	***τ230***	0,0150	***τ250***	0,1533	***τ270***	0,0620	***τ290***	0,0258	***τ310***	0,0350	***τ330***	0,0670	***τ350***	0,0421	***τ370***	0,0119	***τ390***	0,0250
τ211	0,0154	***τ231***	0,0030	***τ251***	0,0048	***τ271***	0,0270	***τ291***	0,0130	***τ311***	0,0128	***τ331***	0,0920	***τ351***	0,0140	***τ371***	0,0426	***τ391***	0,0410
τ212	0,0340	***τ232***	0,0200	***τ252***	0,0064	***τ272***	0,0115	***τ292***	0,0330	***τ312***	0,0099	***τ332***	0,1330	***τ352***	0,0020	***τ372***	0,0094	***τ392***	0,0136
τ213	0,0301	***τ233***	0,0294	***τ253***	0,0030	***τ273***	0,0601	***τ293***	0,0104	***τ313***	0,0410	***τ333***	0,0840	***τ353***	0,0589	***τ373***	0,0373	***τ393***	0,0230
τ214	0,0165	***τ234***	0,0029	***τ254***	0,0016	***τ274***	0,0171	***τ294***	0,0080	***τ314***	0,0690	***τ334***	0,0860	***τ354***	0,0195	***τ374***	0,0070	***τ394***	0,0020
τ215	0,0113	***τ235***	0,0187	***τ255***	0,0051	***τ275***	0,0191	***τ295***	0,0214	***τ315***	0,0520	***τ335***	0,0218	***τ355***	0,0084	***τ375***	0,0743	***τ395***	0,0080
τ216	0,0119	***τ236***	0,0111	***τ256***	0,0093	***τ276***	0,0212	***τ296***	0,0058	***τ316***	0,0690	***τ336***	0,0410	***τ356***	0,0192	***τ376***	0,0052	***τ396***	0,0187
τ217	0,0153	***τ237***	0,0120	***τ257***	0,0052	***τ277***	0,0040	***τ297***	0,0136	***τ317***	0,0610	***τ337***	0,0370	***τ357***	0,0120	***τ377***	0,0565	***τ397***	0,0181
τ218	0,0202	***τ238***	0,0077	***τ258***	0,0390	***τ278***	0,0146	***τ298***	0,0190	***τ318***	0,0400	***τ338***	0,0079	***τ358***	0,0115	***τ378***	0,0094	***τ398***	0,0190
τ219	0,0072	***τ239***	0,0040	***τ259***	0,0065	***τ279***	0,0250	***τ299***	0,0390	***τ319***	0,0680	***τ339***	0,0060	***τ359***	0,0180	***τ379***	0,0215	***τ399***	0,0230
τ220	0,0081	***τ240***	0,0094	***τ260***	0,0049	***τ280***	0,0250	***τ300***	0,0570	***τ320***	0,0128	***τ340***	0,0010	***τ360***	0,0110	***τ380***	0,0010	***τ400***	0,0560

Tabela 9-8 - Parâmetros obtidos no procedimento de calibração de SCut para *VG-RAM WNN-COR* e para a base AT100.

τ401	0,0030	***τ421***	0,0850	***τ441***	0,0420	***τ461***	0,0370	***τ481***	0,0107	***τ501***	0,0250	***τ521***	0,0083	***τ541***	0,0226	***τ561***	0,0165	***τ581***	0,0540
τ402	0,0470	***τ422***	0,0410	***τ442***	0,0240	***τ462***	0,1023	***τ482***	0,0590	***τ502***	0,0167	***τ522***	0,0399	***τ542***	0,0306	***τ562***	0,0470	***τ582***	0,1498
τ403	0,0290	***τ423***	0,0690	***τ443***	0,0698	***τ463***	0,0130	***τ483***	0,0288	***τ503***	0,0360	***τ523***	0,0252	***τ543***	0,0400	***τ563***	0,0040	***τ583***	0,0730
τ404	0,0640	***τ424***	0,0840	***τ444***	0,0640	***τ464***	0,0502	***τ484***	0,0416	***τ504***	0,0330	***τ524***	0,0390	***τ544***	0,0940	***τ564***	0,1130	***τ584***	0,0680
τ405	0,0130	***τ425***	0,0400	***τ445***	0,0310	***τ465***	0,0200	***τ485***	0,0111	***τ505***	0,0950	***τ525***	0,0340	***τ545***	0,0450	***τ565***	0,0167	***τ585***	0,0660
τ406	0,0091	***τ426***	0,0170	***τ446***	0,0960	***τ466***	0,0070	***τ486***	0,0250	***τ506***	0,0540	***τ526***	0,1050	***τ546***	0,0266	***τ566***	0,0640	***τ586***	0,0720
τ407	0,1240	***τ427***	0,0570	***τ447***	0,0226	***τ467***	0,0640	***τ487***	0,0200	***τ507***	0,0620	***τ527***	0,0136	***τ547***	0,0057	***τ567***	0,0610	***τ587***	0,0173
τ408	0,0580	***τ428***	0,0570	***τ448***	0,0120	***τ468***	0,0208	***τ488***	0,0720	***τ508***	0,0267	***τ528***	0,0330	***τ548***	0,0590	***τ568***	0,0610	***τ588***	0,0098
τ409	0,0950	***τ429***	0,0640	***τ449***	0,1090	***τ469***	0,0890	***τ489***	0,0287	***τ509***	0,1492	***τ529***	0,0520	***τ549***	0,0520	***τ569***	0,0580	***τ589***	0,0430
τ410	0,0630	***τ430***	0,0730	***τ450***	0,0660	***τ470***	0,0730	***τ490***	0,1200	***τ510***	0,0341	***τ530***	0,1230	***τ550***	0,0810	***τ570***	0,0550	***τ590***	0,0299
τ411	0,0400	***τ431***	0,1030	***τ451***	0,0940	***τ471***	0,0760	***τ491***	0,1640	***τ511***	0,0124	***τ531***	0,0200	***τ551***	0,1000	***τ571***	0,0410	***τ591***	0,1170
τ412	0,0570	***τ432***	0,0490	***τ452***	0,0020	***τ472***	0,0193	***τ492***	0,0900	***τ512***	0,0120	***τ532***	0,0830	***τ552***	0,1180	***τ572***	0,0600	***τ592***	0,0042
τ413	0,0900	***τ433***	0,0560	***τ453***	0,0259	***τ473***	0,0490	***τ493***	0,0690	***τ513***	0,0080	***τ533***	0,0920	***τ553***	0,0550	***τ573***	0,0031	***τ593***	0,0133
τ414	0,0360	***τ434***	0,0730	***τ454***	0,0420	***τ474***	0,0900	***τ494***	0,0491	***τ514***	0,0540	***τ534***	0,0173	***τ554***	0,0320	***τ574***	0,0244	***τ594***	0,0930
τ415	0,0500	***τ435***	0,0330	***τ455***	0,0170	***τ475***	0,0230	***τ495***	0,0209	***τ515***	0,0040	***τ535***	0,0090	***τ555***	0,0301	***τ575***	0,0110	***τ595***	0,0610
τ416	0,0950	***τ436***	0,0330	***τ456***	0,0640	***τ476***	0,0512	***τ496***	0,0419	***τ516***	0,0470	***τ536***	0,1237	***τ556***	0,0085	***τ576***	0,0720	***τ596***	0,0840
τ417	0,0690	***τ437***	0,0030	***τ457***	0,0078	***τ477***	0,0520	***τ497***	0,0070	***τ517***	0,0267	***τ537***	0,0470	***τ557***	0,0100	***τ577***	0,1120	***τ597***	0,1200
τ418	0,0610	***τ438***	0,0086	***τ458***	0,0151	***τ478***	0,0790	***τ498***	0,0837	***τ518***	0,0360	***τ538***	0,0171	***τ558***	0,0570	***τ578***	0,0220	***τ598***	0,0160
τ419	0,0710	***τ439***	0,0470	***τ459***	0,0023	***τ479***	0,0155	***τ499***	0,0180	***τ519***	0,0079	***τ539***	0,0239	***τ559***	0,0300	***τ579***	0,0450	***τ599***	0,0059
τ420	0,3890	***τ440***	0,0050	***τ460***	0,0173	***τ480***	0,0096	***τ500***	0,0157	***τ520***	0,0381	***τ540***	0,2422	***τ560***	0,0113	***τ580***	0,0850	***τ600***	0,0370

Tabela 9-9 - Parâmetros obtidos no procedimento de calibração de SCut para *VG-RAM WNN-COR* e para a base AT100.

τ601	0,0246	***τ621***	0,0930	***τ641***	0,0619	***τ661***	0,0780	***τ681***	0,1247
τ602	0,0610	***τ622***	0,1080	***τ642***	0,0209	***τ662***	0,0128	***τ682***	0,0020
τ603	0,0590	***τ623***	0,0240	***τ643***	0,0360	***τ663***	0,0100	***τ683***	0,0050
τ604	0,0270	***τ624***	0,0500	***τ644***	0,0249	***τ664***	0,0115	***τ684***	0,1570
τ605	0,0652	***τ625***	0,0023	***τ645***	0,0147	***τ665***	0,0586	***τ685***	0,0750
τ606	0,0740	***τ626***	0,0062	***τ646***	0,0131	***τ666***	0,0286	***τ686***	0,0103
τ607	0,0116	***τ627***	0,0440	***τ647***	0,0440	***τ667***	0,0230	***τ687***	0,0103
τ608	0,0190	***τ628***	0,0490	***τ648***	0,0440	***τ668***	0,0250	***τ688***	0,0200
τ609	0,0100	***τ629***	0,0930	***τ649***	0,0470	***τ669***	0,0170	***τ689***	0,0222
τ610	0,0255	***τ630***	0,0500	***τ650***	0,0730	***τ670***	0,0030	***τ690***	0,0113
τ611	0,1160	***τ631***	0,0100	***τ651***	0,1465	***τ671***	0,0195	***τ691***	0,0020
τ612	0,0290	***τ632***	0,0500	***τ652***	0,0300	***τ672***	0,0269	***τ692***	0,0035
τ613	0,0150	***τ633***	0,0429	***τ653***	0,0100	***τ673***	0,0150		
τ614	0,0120	***τ634***	0,0123	***τ654***	0,0450	***τ674***	0,0630		
τ615	0,0589	***τ635***	0,0084	***τ655***	0,0605	***τ675***	0,0200		
τ616	0,0240	***τ636***	0,0030	***τ656***	0,0160	***τ676***	0,0090		
τ617	0,0176	***τ637***	0,0730	***τ657***	0,0170	***τ677***	0,2038		
τ618	0,1240	***τ638***	0,0274	***τ658***	0,0210	***τ678***	0,0079		
τ619	0,0530	***τ639***	0,0264	***τ659***	0,0460	***τ679***	0,0228		
τ620	0,0110	***τ640***	0,0132	***τ660***	0,0370	***τ680***	0,0130		

A.4 Parâmetros obtidos no procedimento de calibração de SCut para o categorizador VG-RAM WNN-COR e para a base EX100

Tabela 9-10 - Parâmetros obtidos no procedimento de calibração de SCut para *VG-RAM WNN-COR* e para a base EX100.

τ1	0,0660	***τ21***	0,0390	***τ41***	0,0560	***τ61***	0,0740	***τ81***	0,0860	***τ101***	0,0590
τ2	0,0600	***τ22***	0,0500	***τ42***	0,0880	***τ62***	0,0660	***τ82***	0,0800	***τ102***	0,1070
τ3	0,0370	***τ23***	0,0460	***τ43***	0,0750	***τ63***	0,0770	***τ83***	0,0890	***τ103***	0,1010
τ4	0,0750	***τ24***	0,0690	***τ44***	0,0660	***τ64***	0,0620	***τ84***	0,0390	***τ104***	0,0550
τ5	0,0700	***τ25***	0,0710	***τ45***	0,0830	***τ65***	0,0780	***τ85***	0,0520	***τ105***	0,0770
τ6	0,0550	***τ26***	0,0760	***τ46***	0,0840	***τ66***	0,0690	***τ86***	0,0850		
τ7	0,0860	***τ27***	0,0450	***τ47***	0,0210	***τ67***	0,0620	***τ87***	0,0790		
τ8	0,0820	***τ28***	0,0720	***τ48***	0,0960	***τ68***	0,0400	***τ88***	0,0610		
τ9	0,0750	***τ29***	0,0500	***τ49***	0,0470	***τ69***	0,0640	***τ89***	0,0730		
τ10	0,0510	***τ30***	0,0540	***τ50***	0,0570	***τ70***	0,0720	***τ90***	0,0930		
τ11	0,0490	***τ31***	0,0650	***τ51***	0,0630	***τ71***	0,0380	***τ91***	0,0930		
τ12	0,0590	***τ32***	0,0710	***τ52***	0,1140	***τ72***	0,0720	***τ92***	0,0350		
τ13	0,0670	***τ33***	0,0700	***τ53***	0,0590	***τ73***	0,0340	***τ93***	0,1120		
τ14	0,0680	***τ34***	0,1500	***τ54***	0,0410	***τ74***	0,0580	***τ94***	0,0650		
τ15	0,0790	***τ35***	0,0530	***τ55***	0,0770	***τ75***	0,0520	***τ95***	0,0550		
τ16	0,0770	***τ36***	0,0520	***τ56***	0,0450	***τ76***	0,0620	***τ96***	0,0460		
τ17	0,0760	***τ37***	0,0920	***τ57***	0,0610	***τ77***	0,0330	***τ97***	0,0710		
τ18	0,0800	***τ38***	0,0700	***τ58***	0,0620	***τ78***	0,0570	***τ98***	0,0980		
τ19	0,0830	***τ39***	0,0670	***τ59***	0,0630	***τ79***	0,0600	***τ99***	0,1020		
τ20	0,0320	***τ40***	0,0520	***τ60***	0,0440	***τ80***	0,0320	***τ100***	0,0520		

APÊNDICE B – PROBABILIDADES *P(X|Y,K)* DE VALIDAÇÃO VERSUS *P(X|Y,K)* DE TESTE

A Tabela 9-11, Tabela 9-12, Tabela 9-13, Tabela 9-14 e Tabela 9-15 apresentam a comparação entre os valores de $p(x|y,k)$ calculados analiticamente (por meio da regra de Bayes a partir das estimativas de $p(x|k)$, $p(y|k)$ *e* $p(y|x,k)$ obtidas nos experimentos de validação) com os valores de $p(x|y,k)$ estimados empiricamente (a partir dos experimentos de teste para k = { 1, 2, 3, 4 e 5 }, referentes as cincro primeiras posições do *ranking*) para o categorizador ML-*k* NN empregando a base AT100. Nas tabelas citadas acima, a coluna Intervalo mostra cada um dos 20 intervalos de valores de f observados nos experimentos de validação, a coluna Validação mostra os valores de $p(x|y,k)$ calculados analiticamente por meio da regra de Bayes com os resultados dos experimentos de validação, e a coluna Teste mostra os valores de $p(x|y,k)$ estimados empiricamente a partir dos experimentos de teste.

Como pode ser observado na Tabela 9-11, Tabela 9-12, Tabela 9-13, Tabela 9-14 e Tabela 9-15, os valores de $p(x|y,k)$ calculados analiticamente por meio da regra de Bayes são muito próximos aos valores de $p(x|y,k)$ estimados empiricamente, o que demonstra que, usando nossa metodologia, é possível prever no teste com o último *fold* (não visto pelo ML-*k* NN durante o treinamento) o quão certo está o ML-*k* NN quanto à primeira categoria no seu *ranking* de saída ser pertinente para um dado documento. É importante destacar que esta medida de certeza vai de 0% a 100% – uma medida facilmente compreensível para um operador do SCAE humano.

O sistema SCAE usa as tabelas Tabela 9-11, Tabela 9-12, Tabela 9-13, Tabela 9-14 e Tabela 9-15 da seguinte forma. Se o ML-*k* NN predisse a categoria c_i para o documento d_j com grau de crença $f(d_j, c_i)$ dentro de um intervalo y (dentre os 20 intervalos observados na validação), e posicionou a categoria c_i na posição $r(d_j, c_i)$ do *ranking*, então a medida de certeza para essa predição pode ser expressa por $p(x|y,k)$, onde $y \subset f(d_j, c_i)$ e $k = r(d_j, c_i)$.

A Tabela 9-11, Tabela 9-12, Tabela 9-13, Tabela 9-14 e Tabela 9-15 mostram os resultados do uso de nossa metodologia para valores de k iguais a 1, 2, 3, 4 e 5 respectivamente. Como pode ser visto nestas tabelas, também para estes valores de k é possível prever no teste com o último *fold* o quão certo está o *ML-k NN* quanto à categoria na posição k no seu *ranking* de saída ser pertinente para um dado documento. Note que, quanto

maior o *k* (quanto mais abaixo no *ranking* de saída do categorizador), menos provável que a categoria atribuída pelo categorizador seja pertinente ao documento (ver última linha das tabelas). Isso é esperado, já que, para a base de dados empregada no treinamento (AT100), é incomum existirem mais que dois códigos pertinentes a um dado documento.

Tabela 9-11 – Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=1 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,155871)	**p(x|y,k)**	0,122622	***p(x|y,k)***	0,204082
2	(0,155871 - 0,198844)	**p(x|y,k)**	0,203393	***p(x|y,k)***	0,200000
3	(0,198844 - 0,229916)	**p(x|y,k)**	0,317798	***p(x|y,k)***	0,363636
4	(0,229916 - 0,256375)	**p(x|y,k)**	0,330508	***p(x|y,k)***	0,230769
5	(0,256375 - 0,280119)	**p(x|y,k)**	0,341106	***p(x|y,k)***	0,431818
6	(0,280119 - 0,303212)	**p(x|y,k)**	0,461866	***p(x|y,k)***	0,369565
7	(0,303212 - 0,326125)	**p(x|y,k)**	0,466099	***p(x|y,k)***	0,538462
8	(0,326125 - 0,346033)	**p(x|y,k)**	0,476698	***p(x|y,k)***	0,625000
9	(0,346033 - 0,366272)	**p(x|y,k)**	0,489407	***p(x|y,k)***	0,533333
10	(0,366272 - 0,386598)	**p(x|y,k)**	0,536022	***p(x|y,k)***	0,659091
11	(0,386598 - 0,409530)	**p(x|y,k)**	0,546610	***p(x|y,k)***	0,468085
12	(0,409530 - 0,434036)	**p(x|y,k)**	0,586870	***p(x|y,k)***	0,568627
13	(0,434036 - 0,457209)	**p(x|y,k)**	0,593225	***p(x|y,k)***	0,627451
14	(0,457209 - 0,486895)	**p(x|y,k)**	0,620766	***p(x|y,k)***	0,666667
15	(0,486895 - 0,518835)	**p(x|y,k)**	0,678651	***p(x|y,k)***	0,649123
16	(0,518835 - 0,559086)	**p(x|y,k)**	0,690031	***p(x|y,k)***	0,600000
17	(0,559086 - 0,607647)	**p(x|y,k)**	0,707631	***p(x|y,k)***	0,711111
18	(0,607647 - 0,669649)	**p(x|y,k)**	0,775432	***p(x|y,k)***	0,816327
19	(0,669649 - 0,759295)	**p(x|y,k)**	0,832635	***p(x|y,k)***	0,914286
20	(0,759295 - 1,000000)	**p(x|y,k)**	0,928713	***p(x|y,k)***	0,961538

APÊNDICE B

Tabela 9-12 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=2 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,046800)	**p(x|y,k)**	0,029599	***p(x|y,k)***	0,020833
2	(0,046800 - 0,067131)	**p(x|y,k)**	0,067797	***p(x|y,k)***	0,062500
3	(0,067131 - 0,081792)	**p(x|y,k)**	0,069913	***p(x|y,k)***	0,122449
4	(0,081792 - 0,094919)	**p(x|y,k)**	0,112287	***p(x|y,k)***	0,127660
5	(0,094919 - 0,107926)	**p(x|y,k)**	0,112287	***p(x|y,k)***	0,043478
6	(0,107926 - 0,120012)	**p(x|y,k)**	0,156781	***p(x|y,k)***	0,155556
7	(0,120012 - 0,132169)	**p(x|y,k)**	0,133474	***p(x|y,k)***	0,058824
8	(0,132169 - 0,144638)	**p(x|y,k)**	0,133474	***p(x|y,k)***	0,176471
9	(0,144638 - 0,155147)	**p(x|y,k)**	0,220342	***p(x|y,k)***	0,191489
10	(0,155147 - 0,167371)	**p(x|y,k)**	0,211867	***p(x|y,k)***	0,285714
11	(0,167371 - 0,179440)	**p(x|y,k)**	0,241528	***p(x|y,k)***	0,266667
12	(0,179440 - 0,190638)	**p(x|y,k)**	0,266952	***p(x|y,k)***	0,256410
13	(0,190638 - 0,204448)	**p(x|y,k)**	0,300851	***p(x|y,k)***	0,259259
14	(0,204448 - 0,219379)	**p(x|y,k)**	0,290255	***p(x|y,k)***	0,315789
15	(0,219379 - 0,233891)	**p(x|y,k)**	0,292376	***p(x|y,k)***	0,347826
16	(0,233891 - 0,249914)	**p(x|y,k)**	0,364411	***p(x|y,k)***	0,456522
17	(0,249914 - 0,271343)	**p(x|y,k)**	0,404663	***p(x|y,k)***	0,574468
18	(0,271343 - 0,295373)	**p(x|y,k)**	0,538137	***p(x|y,k)***	0,518519
19	(0,295373 - 0,333566)	**p(x|y,k)**	0,591106	***p(x|y,k)***	0,604651
20	(0,333566 - 1,000000)	**p(x|y,k)**	0,624732	***p(x|y,k)***	0,711864

Tabela 9-13 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=3 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,021835)	**p(x|y,k)**	0,012685	***p(x|y,k)***	0,018182
2	(0,021835 - 0,031798)	**p(x|y,k)**	0,029662	***p(x|y,k)***	0,023810
3	(0,031798 - 0,040027)	**p(x|y,k)**	0,044493	***p(x|y,k)***	0,021739
4	(0,040027 - 0,047837)	**p(x|y,k)**	0,055085	***p(x|y,k)***	0,040816
5	(0,047837 - 0,054814)	**p(x|y,k)**	0,105932	***p(x|y,k)***	0,113636
6	(0,054814 - 0,060665)	**p(x|y,k)**	0,063559	***p(x|y,k)***	0,092593
7	(0,060665 - 0,066884)	**p(x|y,k)**	0,088985	***p(x|y,k)***	0,021739
8	(0,066884 - 0,073148)	**p(x|y,k)**	0,105932	***p(x|y,k)***	0,057692
9	(0,073148 - 0,079471)	**p(x|y,k)**	0,097459	***p(x|y,k)***	0,166667
10	(0,079471 - 0,086391)	**p(x|y,k)**	0,095340	***p(x|y,k)***	0,029412
11	(0,086391 - 0,093646)	**p(x|y,k)**	0,103813	***p(x|y,k)***	0,192982
12	(0,093646 - 0,100427)	**p(x|y,k)**	0,135594	***p(x|y,k)***	0,090909
13	(0,100427 - 0,108284)	**p(x|y,k)**	0,116528	***p(x|y,k)***	0,085714
14	(0,108284 - 0,116739)	**p(x|y,k)**	0,175848	***p(x|y,k)***	0,145833
15	(0,116739 - 0,125955)	**p(x|y,k)**	0,203391	***p(x|y,k)***	0,214286
16	(0,125955 - 0,136692)	**p(x|y,k)**	0,199156	***p(x|y,k)***	0,255319
17	(0,136692 - 0,148551)	**p(x|y,k)**	0,235172	***p(x|y,k)***	0,170732
18	(0,148551 - 0,165724)	**p(x|y,k)**	0,292376	***p(x|y,k)***	0,195122
19	(0,165724 - 0,190991)	**p(x|y,k)**	0,379239	***p(x|y,k)***	0,351852
20	(0,190991 - 1,000000)	**p(x|y,k)**	0,410900	***p(x|y,k)***	0,379310

APÊNDICE B

Tabela 9-14 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=4 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,012437)	**p(x|y,k)**	0,014800	***p(x|y,k)***	0,000000
2	(0,012437 - 0,017597)	**p(x|y,k)**	0,014831	***p(x|y,k)***	0,037037
3	(0,017597 - 0,022350)	**p(x|y,k)**	0,023304	***p(x|y,k)***	0,032787
4	(0,022350 - 0,026812)	**p(x|y,k)**	0,019067	***p(x|y,k)***	0,025641
5	(0,026812 - 0,031215)	**p(x|y,k)**	0,042373	***p(x|y,k)***	0,085106
6	(0,031215 - 0,035319)	**p(x|y,k)**	0,038136	***p(x|y,k)***	0,000000
7	(0,035319 - 0,039399)	**p(x|y,k)**	0,048729	***p(x|y,k)***	0,062500
8	(0,039399 - 0,043048)	**p(x|y,k)**	0,067796	***p(x|y,k)***	0,075472
9	(0,043048 - 0,047127)	**p(x|y,k)**	0,074153	***p(x|y,k)***	0,055556
10	(0,047127 - 0,051496)	**p(x|y,k)**	0,078390	***p(x|y,k)***	0,134615
11	(0,051496 - 0,055322)	**p(x|y,k)**	0,065678	***p(x|y,k)***	0,050000
12	(0,055322 - 0,060445)	**p(x|y,k)**	0,091103	***p(x|y,k)***	0,068182
13	(0,060445 - 0,065048)	**p(x|y,k)**	0,125001	***p(x|y,k)***	0,129630
14	(0,065048 - 0,070333)	**p(x|y,k)**	0,114407	***p(x|y,k)***	0,111111
15	(0,070333 - 0,076525)	**p(x|y,k)**	0,120763	***p(x|y,k)***	0,166667
16	(0,076525 - 0,083404)	**p(x|y,k)**	0,148305	***p(x|y,k)***	0,236364
17	(0,083404 - 0,091928)	**p(x|y,k)**	0,156780	***p(x|y,k)***	0,279070
18	(0,091928 - 0,103028)	**p(x|y,k)**	0,148305	***p(x|y,k)***	0,045455
19	(0,103028 - 0,119972)	**p(x|y,k)**	0,222458	***p(x|y,k)***	0,163265
20	(0,119972 - 1,000000)	**p(x|y,k)**	0,257858	***p(x|y,k)***	0,354167

Tabela 9-15 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=5 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,008109)	**p(x|y,k)**	0,012685	***p(x|y,k)***	0,000000
2	(0,008109 - 0,011586)	**p(x|y,k)**	0,016950	***p(x|y,k)***	0,037736
3	(0,011586 - 0,014541)	**p(x|y,k)**	0,014831	***p(x|y,k)***	0,000000
4	(0,014541 - 0,017468)	**p(x|y,k)**	0,019027	***p(x|y,k)***	0,057143
5	(0,017468 - 0,020293)	**p(x|y,k)**	0,021187	***p(x|y,k)***	0,000000
6	(0,020293 - 0,023035)	**p(x|y,k)**	0,029724	***p(x|y,k)***	0,037037
7	(0,023035 - 0,026077)	**p(x|y,k)**	0,029661	***p(x|y,k)***	0,000000
8	(0,026077 - 0,028816)	**p(x|y,k)**	0,048729	***p(x|y,k)***	0,137255
9	(0,028816 - 0,031337)	**p(x|y,k)**	0,044398	***p(x|y,k)***	0,078947
10	(0,031337 - 0,034265)	**p(x|y,k)**	0,063695	***p(x|y,k)***	0,088889
11	(0,034265 - 0,037388)	**p(x|y,k)**	0,050848	***p(x|y,k)***	0,019608
12	(0,037388 - 0,040463)	**p(x|y,k)**	0,040255	***p(x|y,k)***	0,000000
13	(0,040463 - 0,043415)	**p(x|y,k)**	0,067797	***p(x|y,k)***	0,020833
14	(0,043415 - 0,046828)	**p(x|y,k)**	0,076271	***p(x|y,k)***	0,080000
15	(0,046828 - 0,051191)	**p(x|y,k)**	0,052966	***p(x|y,k)***	0,063830
16	(0,051191 - 0,055682)	**p(x|y,k)**	0,091101	***p(x|y,k)***	0,062500
17	(0,055682 - 0,061202)	**p(x|y,k)**	0,067797	***p(x|y,k)***	0,076923
18	(0,061202 - 0,068732)	**p(x|y,k)**	0,116525	***p(x|y,k)***	0,051282
19	(0,068732 - 0,079801)	**p(x|y,k)**	0,154661	***p(x|y,k)***	0,145833
20	(0,079801 - 1,000000)	**p(x|y,k)**	0,182388	***p(x|y,k)***	0,180000

A Tabela 9-16, Tabela 9-17, Tabela 9-18, Tabela 9-19 e Tabela 9-20 apresentam a comparação entre os valores de *p(x|y,k)* calculados analiticamente (por meio da regra de Bayes a partir das estimativas de *p(x|k)*, *p(y|k)* e *p(y|x,k)* obtidas nos experimentos de validação) com os valores de *p(x|y,k)* estimados empiricamente (a partir dos experimentos de teste para k = { 1, 2, 3, 4 e 5 }, referentes as 4 primeiras posições do *ranking*) para o

categorizador *ML-k NN* empregando a base EX100. Nas tabelas citadas acima, a coluna Intervalo mostra cada um dos 20 intervalos de valores de *f* observados nos experimentos de validação, a coluna Validação mostra os valores de *p(x|y,k)* calculados analiticamente por meio da regra de Bayes com os resultados dos experimentos de validação, e a coluna Teste mostra os valores de *p(x|y,k)* estimados empiricamente a partir dos experimentos de teste.

Como pode ser observado na Tabela 9-16, Tabela 9-17, Tabela 9-18, Tabela 9-19 e Tabela 9-20, os valores de *p(x|y,k)* calculados analiticamente por meio da regra de *Bayes* são muito próximos aos valores de *p(x|y,k)* estimados empiricamente, o que demonstra que, usando nossa metodologia, é possível prever no teste com o último *fold* (não visto pelo *ML-k NN* durante o treinamento) o quão certo está o *ML-k NN* quanto à primeira categoria no seu *ranking* de saída ser pertinente para um dado documento. É importante destacar que esta medida de certeza vai de 0% a 100% – uma medida facilmente compreensível para um operador do SCAE humano.

A Tabela 9-16, Tabela 9-17, Tabela 9-18, Tabela 9-19 e Tabela 9-20 mostram os resultados do uso de nossa metodologia para valores de k iguais a 1, 2, 3, 4 e 5 respectivamente. Como pode ser visto nestas tabelas, também para estes valores de k é possível prever no teste com o último *fold* o quão certo está o *ML-k NN* quanto à categoria na posição *k* no seu *ranking* de saída ser pertinente para um dado documento. Note que, quanto maior o *k* (quanto mais abaixo no *ranking* de saída do categorizador), menos provável que a categoria atribuída pelo categorizador seja pertinente ao documento (ver última linha das tabelas). Isso é esperado, já que, para a base de dados empregada no treinamento (EX100), é incomum existirem mais que dois códigos pertinentes a um dado documento.

APÊNDICE B

Tabela 9-16 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=1 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,213654)	**p(x|y,k)**	0,217953	***p(x|y,k)***	0,310345
2	(0,213654 - 0,253865)	**p(x|y,k)**	0,398711	***p(x|y,k)***	0,270270
3	(0,253865 - 0,283170)	**p(x|y,k)**	0,450167	***p(x|y,k)***	0,457143
4	(0,283170 - 0,307869)	**p(x|y,k)**	0,565923	***p(x|y,k)***	0,451613
5	(0,307869 - 0,328687)	**p(x|y,k)**	0,504824	***p(x|y,k)***	0,560000
6	(0,328687 - 0,347477)	**p(x|y,k)**	0,614152	***p(x|y,k)***	0,600000
7	(0,347477 - 0,366800)	**p(x|y,k)**	0,607723	***p(x|y,k)***	0,666667
8	(0,366800 - 0,385357)	**p(x|y,k)**	0,668809	***p(x|y,k)***	0,769231
9	(0,385357 - 0,405425)	**p(x|y,k)**	0,665594	***p(x|y,k)***	0,536585
10	(0,405425 - 0,424138)	**p(x|y,k)**	0,649523	***p(x|y,k)***	0,681818
11	(0,424138 - 0,443853)	**p(x|y,k)**	0,755622	***p(x|y,k)***	0,769231
12	(0,443853 - 0,467713)	**p(x|y,k)**	0,736337	***p(x|y,k)***	0,676471
13	(0,467713 - 0,492551)	**p(x|y,k)**	0,752408	***p(x|y,k)***	0,692308
14	(0,492551 - 0,521014)	**p(x|y,k)**	0,752408	***p(x|y,k)***	0,620690
15	(0,521014 - 0,551039)	**p(x|y,k)**	0,797422	***p(x|y,k)***	0,880000
16	(0,551039 - 0,592087)	**p(x|y,k)**	0,848878	***p(x|y,k)***	0,756757
17	(0,592087 - 0,638092)	**p(x|y,k)**	0,868164	***p(x|y,k)***	0,694444
18	(0,638092 - 0,695906)	**p(x|y,k)**	0,929263	***p(x|y,k)***	0,964286
19	(0,695906 - 0,780403)	**p(x|y,k)**	0,964634	***p(x|y,k)***	0,976744
20	(0,780403 - 1,000000)	**p(x|y,k)**	0,990323	***p(x|y,k)***	1,000000

Tabela 9-17 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=2 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,044119)	**p(x|y,k)**	0,006407	***p(x|y,k)***	0,025641
2	(0,044119 - 0,069430)	**p(x|y,k)**	0,025725	***p(x|y,k)***	0,000000
3	(0,069430 - 0,094407)	**p(x|y,k)**	0,057877	***p(x|y,k)***	0,062500
4	(0,094407 - 0,116874)	**p(x|y,k)**	0,138262	***p(x|y,k)***	0,147059
5	(0,116874 - 0,133240)	**p(x|y,k)**	0,163987	***p(x|y,k)***	0,076923
6	(0,133240 - 0,149573)	**p(x|y,k)**	0,225081	***p(x|y,k)***	0,242424
7	(0,149573 - 0,162703)	**p(x|y,k)**	0,247589	***p(x|y,k)***	0,333333
8	(0,162703 - 0,175137)	**p(x|y,k)**	0,244371	***p(x|y,k)***	0,303030
9	(0,175137 - 0,187067)	**p(x|y,k)**	0,305466	***p(x|y,k)***	0,290323
10	(0,187067 - 0,200746)	**p(x|y,k)**	0,311901	***p(x|y,k)***	0,350000
11	(0,200746 - 0,211643)	**p(x|y,k)**	0,327973	***p(x|y,k)***	0,419355
12	(0,211643 - 0,223880)	**p(x|y,k)**	0,421220	***p(x|y,k)***	0,384615
13	(0,223880 - 0,236515)	**p(x|y,k)**	0,382633	***p(x|y,k)***	0,375000
14	(0,236515 - 0,248824)	**p(x|y,k)**	0,463025	***p(x|y,k)***	0,531250
15	(0,248824 - 0,264757)	**p(x|y,k)**	0,456590	***p(x|y,k)***	0,454545
16	(0,264757 - 0,282097)	**p(x|y,k)**	0,524113	***p(x|y,k)***	0,342857
17	(0,282097 - 0,303480)	**p(x|y,k)**	0,559482	***p(x|y,k)***	0,466667
18	(0,303480 - 0,328872)	**p(x|y,k)**	0,646302	***p(x|y,k)***	0,617647
19	(0,328872 - 0,364351)	**p(x|y,k)**	0,627012	***p(x|y,k)***	0,750000
20	(0,364351 - 1,000000)	**p(x|y,k)**	0,741938	***p(x|y,k)***	0,870968

APÊNDICE B

Tabela 9-18 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=3 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,018310)	**p(x|y,k)**	0,009616	***p(x|y,k)***	0,000000
2	(0,018310 - 0,027769)	**p(x|y,k)**	0,016077	***p(x|y,k)***	0,029412
3	(0,027769 - 0,037083)	**p(x|y,k)**	0,025724	***p(x|y,k)***	0,000000
4	(0,037083 - 0,044837)	**p(x|y,k)**	0,041801	***p(x|y,k)***	0,000000
5	(0,044837 - 0,053478)	**p(x|y,k)**	0,057877	***p(x|y,k)***	0,033333
6	(0,053478 - 0,060969)	**p(x|y,k)**	0,045017	***p(x|y,k)***	0,071429
7	(0,060969 - 0,068917)	**p(x|y,k)**	0,080384	***p(x|y,k)***	0,103448
8	(0,068917 - 0,076836)	**p(x|y,k)**	0,090031	***p(x|y,k)***	0,038462
9	(0,076836 - 0,085796)	**p(x|y,k)**	0,106108	***p(x|y,k)***	0,093750
10	(0,085796 - 0,094922)	**p(x|y,k)**	0,109325	***p(x|y,k)***	0,129032
11	(0,094922 - 0,104509)	**p(x|y,k)**	0,189709	***p(x|y,k)***	0,175000
12	(0,104509 - 0,112968)	**p(x|y,k)**	0,183279	***p(x|y,k)***	0,142857
13	(0,112968 - 0,122218)	**p(x|y,k)**	0,192926	***p(x|y,k)***	0,322581
14	(0,122218 - 0,131435)	**p(x|y,k)**	0,189709	***p(x|y,k)***	0,214286
15	(0,131435 - 0,141726)	**p(x|y,k)**	0,244373	***p(x|y,k)***	0,290323
16	(0,141726 - 0,153663)	**p(x|y,k)**	0,247587	***p(x|y,k)***	0,225000
17	(0,153663 - 0,166652)	**p(x|y,k)**	0,276527	***p(x|y,k)***	0,333333
18	(0,166652 - 0,182104)	**p(x|y,k)**	0,385852	***p(x|y,k)***	0,333333
19	(0,182104 - 0,204839)	**p(x|y,k)**	0,372989	***p(x|y,k)***	0,290323
20	(0,204839 - 1,000000)	**p(x|y,k)**	0,483871	***p(x|y,k)***	0,550000

Tabela 9-19 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=4 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,010632)	**p(x|y,k)**	0,000000	***p(x|y,k)***	0,000000
2	(0,010632 - 0,015101)	**p(x|y,k)**	0,012862	***p(x|y,k)***	0,025641
3	(0,015101 - 0,020026)	**p(x|y,k)**	0,009647	***p(x|y,k)***	0,024390
4	(0,020026 - 0,024620)	**p(x|y,k)**	0,019292	***p(x|y,k)***	0,031250
5	(0,024620 - 0,028993)	**p(x|y,k)**	0,009647	***p(x|y,k)***	0,029412
6	(0,028993 - 0,033145)	**p(x|y,k)**	0,028939	***p(x|y,k)***	0,000000
7	(0,033145 - 0,037407)	**p(x|y,k)**	0,028939	***p(x|y,k)***	0,142857
8	(0,037407 - 0,041924)	**p(x|y,k)**	0,045017	***p(x|y,k)***	0,088235
9	(0,041924 - 0,046791)	**p(x|y,k)**	0,061093	***p(x|y,k)***	0,100000
10	(0,046791 - 0,051331)	**p(x|y,k)**	0,054661	***p(x|y,k)***	0,000000
11	(0,051331 - 0,056289)	**p(x|y,k)**	0,073955	***p(x|y,k)***	0,080000
12	(0,056289 - 0,062003)	**p(x|y,k)**	0,093247	***p(x|y,k)***	0,121212
13	(0,062003 - 0,067422)	**p(x|y,k)**	0,109325	***p(x|y,k)***	0,078947
14	(0,067422 - 0,074505)	**p(x|y,k)**	0,106109	***p(x|y,k)***	0,178571
15	(0,074505 - 0,082368)	**p(x|y,k)**	0,112540	***p(x|y,k)***	0,111111
16	(0,082368 - 0,090437)	**p(x|y,k)**	0,102894	***p(x|y,k)***	0,151515
17	(0,090437 - 0,099790)	**p(x|y,k)**	0,160772	***p(x|y,k)***	0,166667
18	(0,099790 - 0,111837)	**p(x|y,k)**	0,199358	***p(x|y,k)***	0,300000
19	(0,111837 - 0,128731)	**p(x|y,k)**	0,218650	***p(x|y,k)***	0,193548
20	(0,128731 - 1,000000)	**p(x|y,k)**	0,370970	***p(x|y,k)***	0,419355

APÊNDICE B

Tabela 9-20 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=5 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,006329)	**p(x|y,k)**	0,000000	***p(x|y,k)***	0,000000
2	(0,006329 - 0,009125)	**p(x|y,k)**	0,000000	***p(x|y,k)***	0,000000
3	(0,009125 - 0,011836)	**p(x|y,k)**	0,006431	***p(x|y,k)***	0,000000
4	(0,011836 - 0,014649)	**p(x|y,k)**	0,012862	***p(x|y,k)***	0,000000
5	(0,014649 - 0,017253)	**p(x|y,k)**	0,019293	***p(x|y,k)***	0,027027
6	(0,017253 - 0,020112)	**p(x|y,k)**	0,019293	***p(x|y,k)***	0,000000
7	(0,020112 - 0,022601)	**p(x|y,k)**	0,032154	***p(x|y,k)***	0,000000
8	(0,022601 - 0,025437)	**p(x|y,k)**	0,028939	***p(x|y,k)***	0,000000
9	(0,025437 - 0,028076)	**p(x|y,k)**	0,041800	***p(x|y,k)***	0,050000
10	(0,028076 - 0,031041)	**p(x|y,k)**	0,028939	***p(x|y,k)***	0,071429
11	(0,031041 - 0,034298)	**p(x|y,k)**	0,035370	***p(x|y,k)***	0,096774
12	(0,034298 - 0,037931)	**p(x|y,k)**	0,054662	***p(x|y,k)***	0,125000
13	(0,037931 - 0,041624)	**p(x|y,k)**	0,045016	***p(x|y,k)***	0,095238
14	(0,041624 - 0,046093)	**p(x|y,k)**	0,070739	***p(x|y,k)***	0,032258
15	(0,046093 - 0,050630)	**p(x|y,k)**	0,061093	***p(x|y,k)***	0,090909
16	(0,050630 - 0,056356)	**p(x|y,k)**	0,048231	***p(x|y,k)***	0,068966
17	(0,056356 - 0,062987)	**p(x|y,k)**	0,077170	***p(x|y,k)***	0,075000
18	(0,062987 - 0,072475)	**p(x|y,k)**	0,109324	***p(x|y,k)***	0,170732
19	(0,072475 - 0,086527)	**p(x|y,k)**	0,128618	***p(x|y,k)***	0,259259
20	(0,086527 - 1,000000)	**p(x|y,k)**	0,222582	***p(x|y,k)***	0,289474

A Tabela 9-21, Tabela 9-22, Tabela 9-23, Tabela 9-24 e Tabela 9-25 apresentam a comparação entre os valores de *p(x|y,k)* calculados analiticamente (por meio da regra de Bayes a partir das estimativas de *p(x|k)*, *p(y|k)* e *p(y|x,k)* obtidas nos experimentos de validação) com os valores de *p(x|y,k)* estimados empiricamente (a partir dos experimentos de teste para $k = \{ 1, 2, 3, 4 \text{ e } 5 \}$, referentes as 4 primeiras posições do *ranking*) para o categorizador *VG-RAM WNN-COR* empregando a base AT100. Nas tabelas citadas acima, a coluna Intervalo mostra cada um dos 20 intervalos de valores de *f* observados nos experimentos de validação, a coluna Validação mostra os valores de *p(x|y,k)* calculados analiticamente por meio da regra de Bayes com os resultados dos experimentos de validação, e a coluna Teste mostra os valores de *p(x|y,k)* estimados empiricamente a partir dos experimentos de teste.

Como pode ser observado na . Tabela 9-21, Tabela 9-22, Tabela 9-23, Tabela 9-24 e Tabela 9-25, os valores de *p(x|y,k)* calculados analiticamente por meio da regra de Bayes são muito próximos aos valores de *p(x|y,k)* estimados empiricamente, o que demonstra que, usando nossa metodologia, é possível prever no teste com o último *fold* (não visto pelo *VG-RAM WNN-COR* durante o treinamento) o quão certo está o *VG-RAM WNN-COR* quanto à primeira categoria no seu *ranking* de saída ser pertinente para um dado documento. É importante destacar que esta medida de certeza vai de 0% a 100% – uma medida facilmente compreensível para um operador do SCAE humano.

APÊNDICE B

A Tabela 9-21, Tabela 9-22, Tabela 9-23, Tabela 9-24 e Tabela 9-25 mostram os resultados do uso de nossa metodologia para valores de k iguais a 1, 2, 3, 4 e 5 respectivamente. Como pode ser visto nestas tabelas, também para estes valores de *k* é possível prever no teste com o último *fold* o quão certo está o *VG-RAM WNN-COR* quanto à categoria na posição *k* no seu *ranking* de saída ser pertinente para um dado documento. Note que, quanto maior o *k* (quanto mais abaixo no *ranking* de saída do categorizador), menos provável que a categoria atribuída pelo categorizador seja pertinente ao documento (ver última linha das tabelas). Isso é esperado, já que, para a base de dados empregada no treinamento (AT100), é incomum existirem mais que dois códigos pertinentes a um dado documento.

Tabela 9-21 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=1 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,043683)	**p(x|y,k)**	0,162793	***p(x|y,k)***	0,156863
2	(0,043683 - 0,054221)	**p(x|y,k)**	0,266383	***p(x|y,k)***	0,269231
3	(0,054221 - 0,063678)	**p(x|y,k)**	0,348196	***p(x|y,k)***	0,454545
4	(0,063678 - 0,072280)	**p(x|y,k)**	0,434325	***p(x|y,k)***	0,383333
5	(0,072280 - 0,080387)	**p(x|y,k)**	0,463983	***p(x|y,k)***	0,550000
6	(0,080387 - 0,088538)	**p(x|y,k)**	0,533896	***p(x|y,k)***	0,642857
7	(0,088538 - 0,096833)	**p(x|y,k)**	0,574150	***p(x|y,k)***	0,607843
8	(0,096833 - 0,105029)	**p(x|y,k)**	0,584747	***p(x|y,k)***	0,571429
9	(0,105029 - 0,113738)	**p(x|y,k)**	0,677973	***p(x|y,k)***	0,705882
10	(0,113738 - 0,121858)	**p(x|y,k)**	0,673734	***p(x|y,k)***	0,795918
11	(0,121858 - 0,130501)	**p(x|y,k)**	0,760602	***p(x|y,k)***	0,764706
12	(0,130501 - 0,140014)	**p(x|y,k)**	0,730931	***p(x|y,k)***	0,750000
13	(0,140014 - 0,151198)	**p(x|y,k)**	0,716108	***p(x|y,k)***	0,654545
14	(0,151198 - 0,164800)	**p(x|y,k)**	0,750005	***p(x|y,k)***	0,769231
15	(0,164800 - 0,181700)	**p(x|y,k)**	0,855934	***p(x|y,k)***	0,843137
16	(0,181700 - 0,204285)	**p(x|y,k)**	0,851695	***p(x|y,k)***	0,883721
17	(0,204285 - 0,233108)	**p(x|y,k)**	0,877127	***p(x|y,k)***	1,000000
18	(0,233108 - 0,279846)	**p(x|y,k)**	0,898308	***p(x|y,k)***	0,914894
19	(0,279846 - 0,370054)	**p(x|y,k)**	0,911024	***p(x|y,k)***	0,880000
20	(0,370054 - 1,000000)	**p(x|y,k)**	0,920323	***p(x|y,k)***	0,886364

APÊNDICE B

Tabela 9-22 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=2 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,021463)	**p(x|y,k)**	0,016915	***p(x|y,k)***	0,000000
2	(0,021463 - 0,027431)	**p(x|y,k)**	0,048730	***p(x|y,k)***	0,023810
3	(0,027431 - 0,031852)	**p(x|y,k)**	0,080510	***p(x|y,k)***	0,088889
4	(0,031852 - 0,036090)	**p(x|y,k)**	0,110170	***p(x|y,k)***	0,115385
5	(0,036090 - 0,039831)	**p(x|y,k)**	0,114406	***p(x|y,k)***	0,062500
6	(0,039831 - 0,043366)	**p(x|y,k)**	0,120763	***p(x|y,k)***	0,152174
7	(0,043366 - 0,046864)	**p(x|y,k)**	0,152543	***p(x|y,k)***	0,220000
8	(0,046864 - 0,050741)	**p(x|y,k)**	0,139829	***p(x|y,k)***	0,108696
9	(0,050741 - 0,055118)	**p(x|y,k)**	0,188559	***p(x|y,k)***	0,275000
10	(0,055118 - 0,059809)	**p(x|y,k)**	0,235169	***p(x|y,k)***	0,258621
11	(0,059809 - 0,064968)	**p(x|y,k)**	0,237290	***p(x|y,k)***	0,320755
12	(0,064968 - 0,070270)	**p(x|y,k)**	0,247883	***p(x|y,k)***	0,162162
13	(0,070270 - 0,076008)	**p(x|y,k)**	0,315679	***p(x|y,k)***	0,357143
14	(0,076008 - 0,083098)	**p(x|y,k)**	0,336866	***p(x|y,k)***	0,395349
15	(0,083098 - 0,091246)	**p(x|y,k)**	0,368646	***p(x|y,k)***	0,470588
16	(0,091246 - 0,100592)	**p(x|y,k)**	0,457629	***p(x|y,k)***	0,382979
17	(0,100592 - 0,113269)	**p(x|y,k)**	0,510596	***p(x|y,k)***	0,560000
18	(0,113269 - 0,130818)	**p(x|y,k)**	0,534884	***p(x|y,k)***	0,428571
19	(0,130818 - 0,161165)	**p(x|y,k)**	0,596608	***p(x|y,k)***	0,673469
20	(0,161165 - 1,000000)	**p(x|y,k)**	0,645696	***p(x|y,k)***	0,607843

Tabela 9-23 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=3 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,015482)	**p(x|y,k)**	0,010571	***p(x|y,k)***	0,043478
2	(0,015482 - 0,019578)	**p(x|y,k)**	0,021187	***p(x|y,k)***	0,000000
3	(0,019578 - 0,022814)	**p(x|y,k)**	0,023207	***p(x|y,k)***	0,017857
4	(0,022814 - 0,025325)	**p(x|y,k)**	0,033971	***p(x|y,k)***	0,000000
5	(0,025325 - 0,027816)	**p(x|y,k)**	0,031647	***p(x|y,k)***	0,040000
6	(0,027816 - 0,029958)	**p(x|y,k)**	0,044777	***p(x|y,k)***	0,038462
7	(0,029958 - 0,032214)	**p(x|y,k)**	0,059322	***p(x|y,k)***	0,102041
8	(0,032214 - 0,034456)	**p(x|y,k)**	0,069915	***p(x|y,k)***	0,045455
9	(0,034456 - 0,036983)	**p(x|y,k)**	0,080509	***p(x|y,k)***	0,018519
10	(0,036983 - 0,039548)	**p(x|y,k)**	0,073996	***p(x|y,k)***	0,044444
11	(0,039548 - 0,042098)	**p(x|y,k)**	0,110404	***p(x|y,k)***	0,140000
12	(0,042098 - 0,045143)	**p(x|y,k)**	0,110170	***p(x|y,k)***	0,159091
13	(0,045143 - 0,048649)	**p(x|y,k)**	0,114408	***p(x|y,k)***	0,085106
14	(0,048649 - 0,052854)	**p(x|y,k)**	0,127119	***p(x|y,k)***	0,187500
15	(0,052854 - 0,057227)	**p(x|y,k)**	0,144069	***p(x|y,k)***	0,244444
16	(0,057227 - 0,062706)	**p(x|y,k)**	0,180085	***p(x|y,k)***	0,200000
17	(0,062706 - 0,069953)	**p(x|y,k)**	0,220340	***p(x|y,k)***	0,277778
18	(0,069953 - 0,080629)	**p(x|y,k)**	0,302969	***p(x|y,k)***	0,294118
19	(0,080629 - 0,098341)	**p(x|y,k)**	0,324154	***p(x|y,k)***	0,512821
20	(0,098341 - 1,000000)	**p(x|y,k)**	0,488467	***p(x|y,k)***	0,527273

APÊNDICE B

Tabela 9-24 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=4 do *ranking* em cada um dos 20 intervalos observados de *f*.

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,012561)	**p(x|y,k)**	0,004229	***p(x|y,k)***	0,000000
2	(0,012561 - 0,015410)	**p(x|y,k)**	0,006342	***p(x|y,k)***	0,021739
3	(0,015410 - 0,017877)	**p(x|y,k)**	0,008493	***p(x|y,k)***	0,016393
4	(0,017877 - 0,019972)	**p(x|y,k)**	0,027543	***p(x|y,k)***	0,017241
5	(0,019972 - 0,021753)	**p(x|y,k)**	0,016949	***p(x|y,k)***	0,026316
6	(0,021753 - 0,023256)	**p(x|y,k)**	0,033756	***p(x|y,k)***	0,057143
7	(0,023256 - 0,024883)	**p(x|y,k)**	0,038217	***p(x|y,k)***	0,038462
8	(0,024883 - 0,026599)	**p(x|y,k)**	0,040340	***p(x|y,k)***	0,020000
9	(0,026599 - 0,028345)	**p(x|y,k)**	0,050847	***p(x|y,k)***	0,050847
10	(0,028345 - 0,030178)	**p(x|y,k)**	0,046511	***p(x|y,k)***	0,052632
11	(0,030178 - 0,032051)	**p(x|y,k)**	0,063695	***p(x|y,k)***	0,046512
12	(0,032051 - 0,034091)	**p(x|y,k)**	0,044211	***p(x|y,k)***	0,068182
13	(0,034091 - 0,036354)	**p(x|y,k)**	0,059701	***p(x|y,k)***	0,000000
14	(0,036354 - 0,038855)	**p(x|y,k)**	0,076273	***p(x|y,k)***	0,065217
15	(0,038855 - 0,041879)	**p(x|y,k)**	0,093221	***p(x|y,k)***	0,097561
16	(0,041879 - 0,045718)	**p(x|y,k)**	0,086866	***p(x|y,k)***	0,069767
17	(0,045718 - 0,050535)	**p(x|y,k)**	0,129238	***p(x|y,k)***	0,081967
18	(0,050535 - 0,057143)	**p(x|y,k)**	0,156781	***p(x|y,k)***	0,090909
19	(0,057143 - 0,069693)	**p(x|y,k)**	0,194918	***p(x|y,k)***	0,122449
20	(0,069693 - 1,000000)	**p(x|y,k)**	0,314465	***p(x|y,k)***	0,302326

Tabela 9-25 - Probabilidades *p(x|y,k)* de validação versus *p(x|y,k)* de teste para a posição *k*=5 do *ranking* em cada um dos 20 intervalos observados de *f*

Ordem Intervalo	Intervalo	Validação		Teste	
1	(0,000000 - 0,010737)	**p(x|y,k)**	0,008457	***p(x|y,k)***	0,000000
2	(0,010737 - 0,013037)	**p(x|y,k)**	0,002118	***p(x|y,k)***	0,000000
3	(0,013037 - 0,014770)	**p(x|y,k)**	0,012712	***p(x|y,k)***	0,000000
4	(0,014770 - 0,016522)	**p(x|y,k)**	0,004228	***p(x|y,k)***	0,000000
5	(0,016522 - 0,017941)	**p(x|y,k)**	0,008493	***p(x|y,k)***	0,023256
6	(0,017941 - 0,019231)	**p(x|y,k)**	0,029662	***p(x|y,k)***	0,020000
7	(0,019231 - 0,020573)	**p(x|y,k)**	0,025370	***p(x|y,k)***	0,018868
8	(0,020573 - 0,021877)	**p(x|y,k)**	0,021187	***p(x|y,k)***	0,000000
9	(0,021877 - 0,023211)	**p(x|y,k)**	0,014862	***p(x|y,k)***	0,000000
10	(0,023211 - 0,024648)	**p(x|y,k)**	0,033898	***p(x|y,k)***	0,000000
11	(0,024648 - 0,025945)	**p(x|y,k)**	0,036017	***p(x|y,k)***	0,000000
12	(0,025945 - 0,027491)	**p(x|y,k)**	0,050848	***p(x|y,k)***	0,071429
13	(0,027491 - 0,029170)	**p(x|y,k)**	0,042373	***p(x|y,k)***	0,025641
14	(0,029170 - 0,031072)	**p(x|y,k)**	0,042373	***p(x|y,k)***	0,022727
15	(0,031072 - 0,033122)	**p(x|y,k)**	0,044398	***p(x|y,k)***	0,023810
16	(0,033122 - 0,035670)	**p(x|y,k)**	0,048832	***p(x|y,k)***	0,044444
17	(0,035670 - 0,038920)	**p(x|y,k)**	0,065679	***p(x|y,k)***	0,043478
18	(0,038920 - 0,043607)	**p(x|y,k)**	0,091102	***p(x|y,k)***	0,063830
19	(0,043607 - 0,051114)	**p(x|y,k)**	0,144069	***p(x|y,k)***	0,061224
20	(0,051114 - 1,000000)	**p(x|y,k)**	0,213835	***p(x|y,k)***	0,127660