5
Experimentos

O presente capítulo tem por objetivo apresentar os experimentos computacionais realizados para validar a abordagem de Aprendizagem com Múltiplas Categorias Latentes abordada no capítulo 4.

Primeiro, na seção 5.1 é feita uma avaliação sobre as bases de teste disponíveis que se adequam aos experimentos propostos. Em seguida, na seção 5.2, é feito um resumo explicativo das métricas utilizadas para os experimentos, e na seção 5.3 é delineada a metodologia dos experimentos. Por fim, na seção 5.4 são apresentados os experimentos e a contextualização dos resultados encontrados.

5.1
Base de Teste

Uma base de teste para busca de páginas *Web*, normalmente, é composta por três partes:

1. um conjunto de páginas em que são efetuadas as buscas, chamado de coleção de teste;

2. um conjunto contendo as informações que se deseja buscar, organizada em tópicos;

3. e, conjuntos de páginas para cada tópico, avaliados por sua relevância, referidos como *qrels*.

Comparativamente, a parte que consome mais tempo para ser gerada em uma coleção de teste é, sem dúvida, a dos conjuntos com as páginas avaliadas pela relevância.

Avaliar um conjunto de páginas heterogêneas provenientes de diferentes fontes é uma tarefa hercúlia. O critério de avaliação tem que ser confiável, acurado e reprodutível, e usar a avaliação humana pode ser sempre objeto de discussão e fragilidade do processo de avaliação.

Como é cediço, alguns métodos para reduzir esse problema têm sido propostos, tal como formalmente selecionar um conjunto de páginas promissoras para serem avaliadas e assumir que essas são representativas do todo (CARTERETTE; ALLAN; SITARAMAN, 2006).

Adicionalmente, as coleções de teste atuais se tornaram gigantescas, refletindo a enorme quantidade de informação hodiernamente disponível nos cenários da recuperação da informação (RI). Desta forma, quando se constrói uma base de teste completa, com as três partes antes descritas, é natural que a mesma tenha um tempo de vida útil que permita aproveitar todo o trabalho dispendido.

Durante o tempo de validade da base os modelos de RI desenvolvidos e aplicados à mesma podem ser avaliados sob dois aspectos, quais sejam:

- desempenho em relação a novos tópicos de busca que são gerados;
- e, desempenho em relação a modelos já aplicados anteriormente, permitindo compreender a sua própria evolução e a da base de teste.

Há ponto importante e que merece destaque no tocante ao processo de construção da coleção de teste. Se o objetivo principal é conseguir avaliar o desempenho de modelos de RI aplicados a *Web*, essa coleção, então, tem que ser representativa.

Na seção 2.2.2 verificamos a evolução dessas coleções de teste e resta claro a tentativa de construir tal coleção representativa.

O processo começou em 1995, com a *VLC* (*Very Large Collection Track*) e culminou em 2009 com a coleção *ClueWeb09* (LEMUR, 2010).

A coleção *ClueWeb09* foi gerada pelo *Language Technologies Institute at Carnegie Mellon University* para dar suporte às pesquisas na área de recuperação da informação e tecnologias relacionadas à linguagem humana. A coleção consiste em 1.040.809.705 de páginas *Web*, em dez línguas diferentes, com 25 TB de tamanho descompactada, coletadas entre janeiro e fevereiro de 2009.

Coleção	**Documentos**(Doc) (Doc)	**Hiperlinks**(Hp) (Hp)	**Densidade** $\approx (Hp/Doc)$
WT2g	247.491	1.166.702	5
WT10g	1.692.096	8.062.918	5
.GOV	1.247.753	11.110.985	9
.GOV2	25.205.179	82.711.345	3
ClueWeb09	1.040.809.705	$\approx$ 12B	12

Tabela 5.1: Evolução da densidade de hiperlinks das coleções da *TREC*

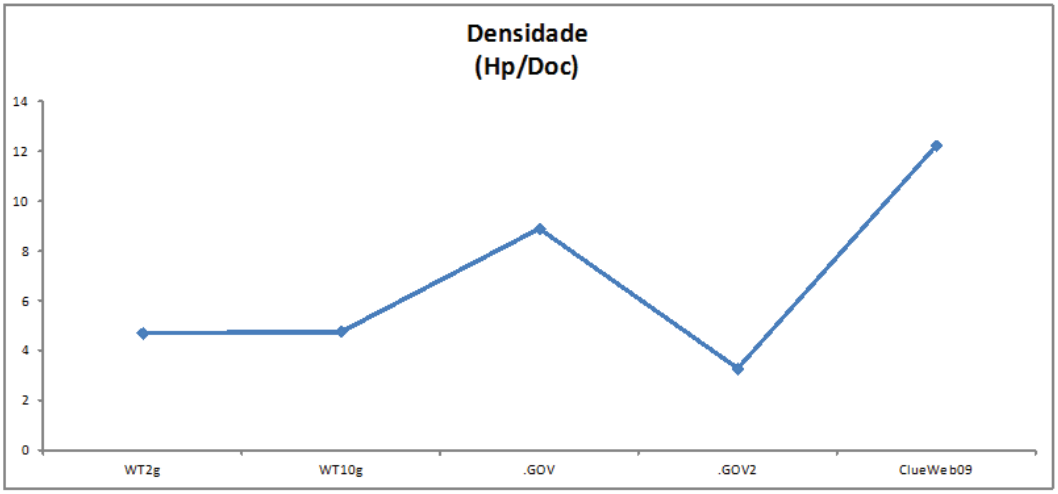

Figura 5.1: Gráfico da evolução da densidade de hiperlinks das coleções da *TREC*

Ainda, duas versões da referida coleção foram publicadas, a categoria A (cat-A) que compreende toda a coleção e a categoria B (cat-B) que corresponde a um subconjunto da cat-A, constituído das primeiras 50 milhões páginas em inglês, com tamanho de 10GB compactada.

Atendida a representatividade, resta saber se consegue atender aos requisitos para que algoritmos baseados apenas na estrutura de hiperlinks consigam um bom desempenho.

As coleções anteriores a *ClueWeb09* eram pouco informativas para esses tipos de algoritmos conforme descrito na seção 2.2.2. Porém, fazendo um estudo preliminar da densidade de hiperlinks das coleções, podemos supor que existe uma considerável melhora na coleção *ClueWeb09* em relação às demais, como pode ser observado na tabela 5.1 e no gráfico 5.1.

O gráfico 5.1 reflete também uma evolução temporal das coleções, uma vez que essas estão ordenadas da esquerda para a direita pelo seu ano de lançamento.

Destarte, analisando o processo evolutivo, a coleção *WT2g* possui uma densidade média de quatro hiperlinks por página, chegando a quase nove na *.GOV* e, mesmo com a inserção de mais aproximadamente 24 milhões de páginas na *.GOV*, a densidade caiu para aproximadamente três.

Por ultimo, destacamos que até o momento do presente trabalho, a densidade chega a aproximadamente doze com a coleção *ClueWeb09*, o que indica um cenário mais favorável para as técnicas de RI baseadas na estrutura de hiperlinks.

Dentro desse contexto, a base *ClueWeb09* foi escolhida para compor a primeira parte da base de teste dos experimentos. Apenas foi utilizada a parte da língua inglesa da coleção, que corresponde a 503.903.810 páginas Web.

Uma vez definida a primeira parte, foi necessário definir quais as informações que se deseja buscar na coleção, ou seja, o conjunto de tópicos.

Depois de 2009, a *Text Retrieval Conference* (TREC) começou a utilizar a coleção *ClueWeb09* em sua competição anual denominada *TREC Web Track*.

Nessa competição, existem diversos tipos de busca, conforme descrito na seção 2.2.2. Para cada tipo de busca, os responsáveis pela competição disponibilizam um conjunto de tópicos para que os participantes busquem informações na *ClueWeb09* e submetam os seus resultados à Comissão de Avaliação da Competição. Após a etapa de submissões, a Comissão disponibiliza os conjuntos de páginas relevantes para cada tópico apresentado e calcula o desempenho dos participantes.

Desta forma, as partes dois e três da base de teste estão definidas. Para os experimentos, foram utilizados os conjuntos de tópicos e seus respectivos conjuntos de avaliação para a busca do tipo *ad hoc*, que é composto por termos curtos de consulta e totalmente automatizados, sem interferência humana na análise dos tópicos. Apenas é permitida a manipulação dos tópicos através de algoritmos computacionais, tal como, retirada de *stopwords*.

Desde 2009, apenas duas competições ocorreram utilizando a coleção *ClueWeb09*, a *TREC Web Track 2009* (*WT09*) e a *TREC Web Track 2010* (*WT10*), e em cada uma das competições, foi gerado um conjunto com cinquenta tópicos para a busca do tipo *ad hoc* com as suas respectivas avaliações de relevância.

Em relação ao número de participantes, em 2010, vinte diferentes grupos participaram da busca do tipo *ad hoc*, uma redução de 20% em relação a

Grupo
Carnegie Mellon University
University of Massachusetts Amhers
Chinese Academy of Sciences
Microsoft Research
The University Of Melbourne
Tsinghua University
University of Glasgow
University of Maryland
University of Twente
Hungarian Academy of Science
Multimodal Computing and Interaction

Tabela 5.2: Resumo dos participantes da categoria A da *Web TREC Track 2010*

2009, quando vinte e cinco grupos participaram. Como os experimentos foram feitos levando em consideração a categoria A da coleção, a tabela 5.2 resume os participantes da *WT10* na modalidade de busca *ad hoc* que submeteram resultados para a mesma categoria (STANDARTS; TECHNOLOGY, 2000).

Em resumo, a base de teste dos experimentos é composta de:

1. coleção de teste: *ClueWeb09*;
2. conjunto de tópicos: cem tópicos, em que cinquenta provém da *WT09* e cinquenta provém da *WT10*;
3. e, *qrels*: cem conjuntos para cada tópico, em que cinquenta provém da *WT09* e cinquenta provém da *WT10*.

Na próxima seção são apresentadas as medidas de avaliação utilizadas para comparar os resultados.

5.2
Medidas de Avaliação dos Resultados

De forma a compatibilizar os resultados dos experimentos com os resultados publicados pela *TREC*, foram adotadas as mesmas métricas de avaliação da competição *Web TREC Track 2010*, quais sejam, *Precision at 20* (P@20), *Expected Reciprocal Rank* (ERR@20) (CHAPELLE et al., 2009), *Normalized Discounted Cumulated Gain* ($nDCG$@20) (JÄRVELIN; KEKÄLÄINEN, 2002)

e *Mean Average Precision*(MAP) (CARTERETTE; ALLAN; SITARAMAN, 2006).

A métrica ERR é calculada levando em consideração a escala de valores definidas para a avaliação das páginas. Cada página i é avaliada com base em uma escala de relevância que vai de 0 a 4 (e_i), em que o 0 representa não avaliada ou lixo e 4 a mais alta relevância.

De acordo com Chapelle et al. (2009), o ponto chave da métrica é definir uma função $R(e_i)$, que mapeie a escala de relevância em probabilidade de relevância. A *TREC 2010 Web Track* definiu a função $R(e_i)$ como $R(e_i) = \frac{2^{e_i}-1}{16}$.

Então, quando uma página i não é relevante ($e_i = 0$), a probabilidade de um usuário achá-la relevante é 0, entretanto, quando uma página i é extremamente relevante ($e_i = 4$), então a probabilidade da relevância é perto de 1.

Assim, a partir da definição da ERR (CHAPELLE et al., 2009), para as k primeiras páginas da lista de classificação, o $ERR@k$ é calculado como se segue

$$ERR@k = \sum_{i=1}^{k} \frac{R(e_i)}{i} \prod_{j=1}^{i-1}(1 - R(e_i)).$$

A partir da mesma ideia de relevância, a métrica *Cumulated Gain* (CG) utiliza a escala de relevância e a posição da classificação da página para converter a lista de classificação em uma lista de ganho de páginas.

Assim, a escala de relevância de cada página é utilizada como uma medida de ganho para a sua posição na classificação resultante e o ganho é somado progressivamente da primeira posição até a n-ésima.

De acordo com Järvelin e Kekäläinen (2002), a função CG é calculada como

$$CG[i] \;=\; \left\{ \begin{array}{cc} G[1], & se\ i = 1 \\ CG[i-1] + G[i], & \text{caso contrário} \end{array} \right\} \qquad (5\text{-}1)$$

Para melhor compreensão, suponha que $G = (3, 2, 3, 0, 0, 1, 2, 2, 3, 0)$ é uma lista de classificação de páginas com as valores das avaliações de relevância, ou seja, $G[1]$ é o valor da avaliação da página que ocupa a primeira posição da lista. Aplicando a equação 5-1 em G o resultado é uma lista de ganho

cumulativo $CG = (3, 5, 8, 8, 8, 9, 11, 13, 16, 16)$.

A métrica *Discounted Cumulated Gain* (DCG) reflete a ideia de que quanto maior é a posição de um documento relevante em uma classificação de páginas, menos relevante ele é para o usuário da classificação, porque é pouco provável que o usuário examine páginas em posições altas devido ao tempo, esforço e informação acumulada a partir das páginas já visitadas.

Assim, é necessário aplicar uma função que desconte progressivamente o valor da avaliação da página conforme a sua posição na classificação aumenta.

De acordo com Järvelin e Kekäläinen (2002), a função DCG pode ser calculada como se segue

$$DCG[i] = \left\{ \begin{array}{cc} G[1], & se\ i < b \\ DCG[i-1] + \frac{G[i]}{log_b i}, & \text{caso contrário} \end{array} \right\} \tag{5-2}$$

onde b é a base logarítmica.

Dando continuidade ao exemplo acima, aplicando a equação 5-2 em $CG = (3, 5, 8, 8, 8, 9, 11, 13, 16, 16)$ o resultado é uma lista de ganho cumulativo descontado $DCG = (3, 5, 6.89, 6.89, 6.89, 7.28, 7.99, 8.66, 9.61, 9.61)$.

Assim, adaptando a equação 5-2 para os parâmetros da *TREC 2010 Web Track* a função $DCG@k$ das primeiras k paginas passa a ser

$$DCG@k = \sum_{i=1}^{k} \frac{2^{e_i} - 1}{log_2(1+i)} \tag{5-3}$$

A métrica *Precision at k* ($P@k$) é a que possui a ideia mais simples. A escala de avaliação da página é binária, ou seja, 1 se a página é relevante e 0 caso contrário. Então, é somada a pontuação das páginas relevantes entre a primeira posição e a k-ésima posição e a soma total dividida por k. De forma a compatibilizar a escala g_i a escala binária $B(G_i)$ trata os valores de 1-4 as relevantes e 0 como não relevante.

A métrica $P@k$ pode, então, ser definida como se segue

$$P@k = \frac{\sum_{i=1}^{k} B(G_i)}{k} \tag{5-4}$$

Exemplificando a $P@k$, a partir da lista G, aplicando a definição de $B(G_i)$ em G é gerada uma G' igual a $(1, 1, 1, 0, 0, 1, 1, 1, 1, 0)$. Logo as $P@5$ e $P@10$

de $G^{'}$ são respectivamente 0.6 e 0.7.

A *Average Precision* (AP) para uma única consulta σ é calculada pela média simples dos valores das precisões em cada posição que contenha uma página relevante, ou seja

$$AP_{\sigma}@k = \frac{\sum_{i=1}^{k} B(G_i) * P@i}{\sum_{i=1}^{k} B(G_i)}. \tag{5-5}$$

Porém, a MAP é a média aritmética dos valores da AP para um conjunto de consultas, $1 \leq \sigma \leq q$.

Então, formalmente

$$MAP@k = \frac{1}{q} \sum_{\sigma=1}^{q} \sum_{i=1}^{k} AP_{\sigma}@i. \tag{5-6}$$

Aplicando a equação 5-5 em $G^{'} = (1, 1, 1, 0, 0, 1, 1, 1, 1, 0)$ para calcular AP@10, então

$$AP@10 = \frac{(P@1) + (P@2) + (P@3) + (P@6) + (P@7) + (P@8) + (P@9)}{7} =$$

$$= \frac{1 + 1 + 1 + (4/6) + (5/7) + (6/8) + (7/9)}{7} \approx 0.845.$$

Como última análise, observamos que a MAP é uma métrica sensível às mudanças da classificação de uma página.

Por exemplo, considere uma classificação de páginas em que apenas um página é relevante e ocupa a posição 10. Se por algum motivo essa página for reclassificada para a primeira posição, a P@10 se manteve inalterada, com valor igual a 0.1. Porém, a MAP vai aumentar de 0.1 para 1.

Na próxima seção é apresentada a metodologia utilizada nos experimentos.

5.3 Metodologia

O principal objetivo da aprendizagem explicitada é encontrar a matriz F, que minimiza a função de erro E (equação 4-3).

Desta forma, na presente seção é definida a metodologia que foi aplicada aos experimentos realizados neste trabalho.

Inicialmente, assim como no modelo de Kleinberg, é necessário utilizar uma máquina de busca baseada em texto para construir os grafos iniciais para cada consulta σ, R_σ. Para tal, foi utilizada a máquina de busca *Indri* (INDRI, 2009) para indexar e consultar a coleção *Clueweb09*. Para a indexação, as palavras com pouca significação, *stopwords*, foram removidas e todos os demais termos foram normalizados na metodologia de Krovetz (1993). Desta forma, o índice principal da máquina é totalmente baseado em texto.

Adicionalmente, foi utilizado o grafo da coleção fornecido pela *Carnegie Mellon University* e o mapeamento dos identificadores das páginas da coleção fornecidos pelo *National Institute of Standards and Technology* (NIST), ambos são dados derivados da *ClueWeb09* (LEMUR, 2010).

Com a indexação finalizada, foram construídos os grafos induzidos para cada umas das 100 consultas σ provenientes da *WT09* e da *WT10*, como descrito na seção 5.1.

Os conjuntos inicias R_σ foram construídos com as mil primeiras páginas retornadas pela *Indri*. De forma a estender os R_σ conforme descrito na seção 2.3.1, foram inseridas todas as páginas que são apontadas pelas páginas contidas em R_σ, e até cinquenta páginas que apontam para cada uma das páginas do R_σ, d=50.

Depois desse passo, foram obtidos conjuntos S_σ com 15.000 páginas em média. Nas competições *WT09* e *WT10*, cada participante submeteu um lista ordenada de 10.000 páginas para cada tópico, ou seja, o tamanho dos conjuntos S_σ gerados são mais do que suficientes.

Como próximo passo, para treinar e testar o *XHITS*, dois conjuntos de grafos foram gerados a partir de S_σ: um de treinamento e outro para teste.

O conjunto de treinamento L foi gerado, por duas regras de formação diferentes, evitando sobreposição:

1. selecionando os 50 grafos de S_σ relativos as consultas σ da *WT09*;
2. e, selecionando os 50 grafos de S_σ relativos as consultas σ da *WT10*.

A primeira forma de montagem do conjunto de treinamento tenta reproduzir o que se teria de informação disponível para treinar o modelo em estudo, no momento da submissão em 2010, para a *TREC 2010 Web Track*. Ou seja,

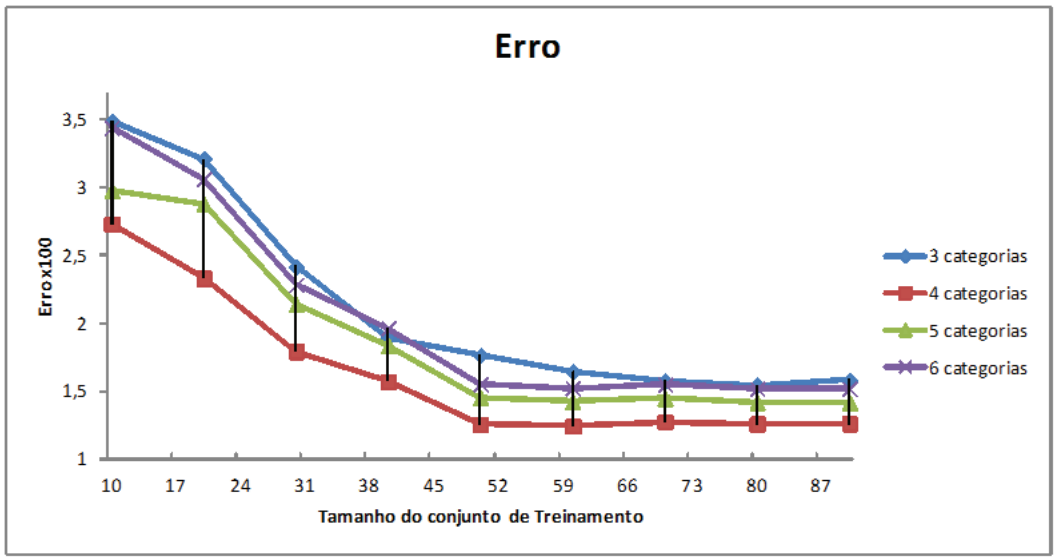

Figura 5.2: Gráfico da evolução da erro com o aumento do conjunto de treinamento. O valor do erro está multiplicado por 100.

utiliza os resultados anteriores a $WT10$ para treinar o modelo e assim tentar atingir um melhor desempenho. A segunda forma serve como estudo comparativo e apenas inverte a relação temporal das competições, isto é, os resultados de 2010 estavam disponíveis para treinar o modelo e as consultas ainda não divulgadas seriam as de 2009.

Ainda em relação ao tamanho do conjunto de treinamento foi feito um estudo empírico para saber o comportamento do aprendizado com a variação do tamanho do conjunto de treinamento.

Como pode ser visto na figura 5.2, conforme o conjunto de treinamento foi aumentando, por consequência, a função erro (E) foi decaindo, mostrando que o modelo de AMCL proposto foi aprendendo com as consultas inseridas, chegando a uma região de saturação em torno do tamanho 60 do conjunto. Podemos verificar que o mesmo ocorreu quando o número de categorias foi aumentando, crescendo a confiabilidade da metodologia de aprendizagem.

Em princípio, pelo fato de a região de saturação ter ocorrido com o tamanho do conjunto de treinamento em torno de 60, a abordagem de separar os conjuntos de treinamento e teste com metade dos grafos, ou seja, conjuntos de 50 grafos, não deve prejudicar os resultados experimentais sobre as capacidades do *XHITS*.

Após a montagem do conjunto de treinamento, os restantes dos grafos formaram o conjunto de testes.

As regras de formação dos conjuntos de treinamento acabaram definindo dois tipos de experimentos diferentes:

– Experimento com Conjunto de Treinamento *WT09* (ECTWT09);
– e, Experimento com Conjunto de Treinamento *WT10* (ECTWT10).

Em cada um dos experimentos acima, o número de categorias variou entre duas e seis, pois o limite inferior reduz o modelo ao *HITS* e o superior foi delimitado pelos próprios resultados empíricos. O objetivo principal de variar o número de categorias é iniciar uma avaliação exploratória do modelo generalizado.

A estrutura dos experimentos ECTWT09 e ECTWT10 é a mesma, diferindo, apenas, em relação ao conjunto de treinamento e teste.

Assim, independente do experimento, ambos seguiram passos representados no algoritmo 5.1. Esse recebe os seguintes parâmetros de entrada:

– o conjunto de treinamento L, que pode ser proveniente de uma das duas regras de formação;
– o conjunto O contendo as classificações de referência para cada consulta σ;
– as consultas σ;
– o tempo total que se deseja rodar o experimento, $TExec$, que serve como parâmetro de parada do treinamento;
– o maior número de categorias, $NumCat$, que se deseja utilizar, sabendo que o limite inferior é 3;
– e, o número máximo de iterações do algoritmo de AMCL, $AMCLMaxIt$.

Os parâmetros de saída são duas matrizes F_{min} e E_{min}, que armazenam os menores erros encontrados por cada categoria e as matrizes F que geram os erros mínimos por categoria.

Primeiro, são feitas as inicializações dos valores da matriz E_{min} com o maior positivo que o tipo numérico da matriz pode conter servindo como valor de teto para o erro, conforme pode ser visto nas linhas 2 a 4.

Em seguida, para cada categoria, linhas 5 a 13, o algoritmo diminui a precisão do erro para o algoritmo AMCL (seção 4.2) toda vez que encontra um

Algoritmo 5.1: Algoritmo dos Experimentos

Entrada:
L:Conjunto contendo os grafos de treinamento
O:Conjunto contendo as classificações de referência para cada consulta σ
σ:As consultas
$TExec$:Tempo total de execução
$NumCat$: Número de categorias
$AMCLMaxIt$:Número máximo de iterações do AMCL
Saída: A matriz F definida, o menor erro E_{min} reportado

```
1  início
2      para i:=3,..,NumCat faça
3          E_min[i]:=MaiorPositivo() ; // E_min[i] armazena o menor
           erro corrente reportado para a categoria i
4      fim
5      para i:=3,..,NumCat faça
6          enquanto TempoExecucAtual() ≤ TExec faça
7              TempErro,TempF ← AMCL(L,O,AMCLMaxIt,i);
8              se TempErro ≤ E_min[i] então
9                  E_min[i] ← TempErro;
10                 F[i] ← TempF ; // F[i] armazena a matriz F
                   relativo ao E_min[i] para a categoria i
11             fim
12         fim
13     fim
14 fim
```

erro menor que o anterior, linhas 8 a 11, tentando a cada iteração melhorar os resultados do treinamento.

Com o objetivo de conferir maior confiabilidade aos resultados durante o processo de aprendizado do algoritmo AMCL, esse efetua uma validação cruzada *10-fold*. O valor da taxa de aprendizado μ das equações 4-6, 4-7 e 4-8 é igual a 1 na primeira iteração e reduz em 10% do seu valor em cada iteração.

5.4 Experimentos e Resultados

À luz de tudo o que foi exposto na seção precedente, os resultados obtidos com os experimentos foram subdivididos em duas subseções com o escopo, apenas, de proporcionar melhor compreensão.

A primeira trata do experimento ECTWT09 e a segunda, por sua vez,

refere-se ao experimento ECTWT10. Importante salientar, entretanto, que somente o melhor resultado de cada experimento foi aqui reportado.

5.4.1
Experimento com Conjunto de Treinamento WT09

No presente experimento, os seguintes parâmetros de entrada forma definidos para o algoritmo 5.1, tal como descrito na seção anterior:

- o conjunto de treinamento L, formado pelos grafos gerados a partir dos tópicos da WT09;
- o conjunto O contendo as classificações de referência para os tópicos da WT09;
- as consultas σ, formada pelos tópicos da WT09;
- o tempo total do experimento, $TExec$ com 168h;
- o maior número de categorias, $NumCat$ como 6;
- e, o número máximo de iterações do algoritmo de AMCL, $AMCLMaxIt$ como 100.

Finalizado o treinamento, a matriz F encontrada foi utilizada para classificar as páginas relativas as consultas σ do conjunto de teste, formado pelos grafos gerados a partir dos tópicos da *WT10*.

A partir das classificações, foi gerada uma lista com as 10.000 primeiras páginas por cada consulta σ e calculada as métricas definidas na seção 5.2, conforme os resultados publicados na competição *WT10*.

Dentro das 168 horas disponíveis por categoria para minimizar a função E, o instante que o menor valor de E foi encontrado, reflete como o algoritmo AMCL escolhe os seus pontos iniciais no seu espaço de soluções.

Verificamos que os mínimos foram encontrados em instantes bem diferentes dentro da mesma categoria, como se pode observar na tabela 5.3 e no gráfico 5.3, motivo pelo qual não se pode aferir se existe um tempo mais ajustado para o treinamento do que 168 horas. Os experimentos foram executados em um computador com um processador Intel(R) Core(TM) i7 950, 3.07GHz, memória principal com tamanho de 12 GB, e discos rígidos de 2 a 3 TB, SATA, 3 Gb/s.

A variação da qualidade dos resultados em relação ao número de categorias é reportada no gráfico 5.4, no qual se pode verificar que existe um máximo

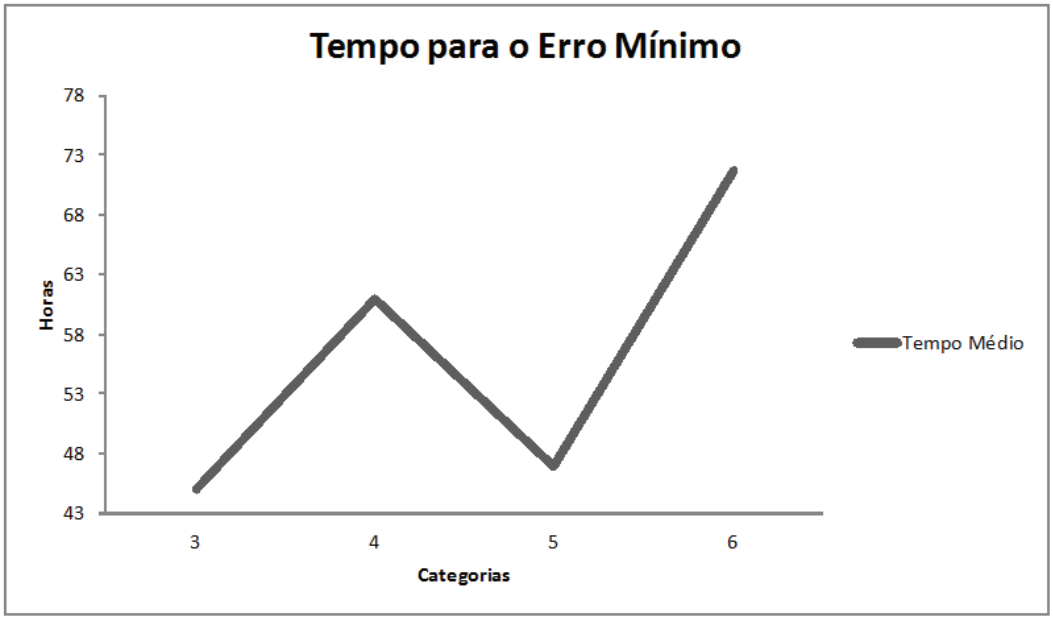

Figura 5.3: Gráfico da evolução da tempo médio para encontrar o erro mínimo por categoria.

		Execução				Tempo Médio(H)
		1	2	3	4	
Categorias	3	16,87	34,78	22,84	105,45	44,985
	4	53,21	10,56	94,68	85,4	60,9625
	5	20,15	80,24	10,14	77,25	46,945
	6	105,89	35,68	50,12	95,23	71,73

Tabela 5.3: Tempo médio de execução do experimento por categoria

quando se chega a quatro categorias, exatamente onde o melhor resultado foi encontrado.

Decorre desse gráfico, porém, duas possíveis situações:

- o tempo de treinamento foi insuficiente, logo o aprendizado foi prejudicado;
- ou, a partir de quatro começam a ser inseridas informações desnecessárias, atrapalhando o desempenho do modelo.

Tais questões ainda permanecem em aberto e serão objeto de trabalhos futuros.

Os melhores resultados desse experimento estão resumidos na tabela 5.4.

Com o objetivo, apenas, de posicionar em perspectiva o presente trabalho, comparamos os resultados aqui alcançados após o processo referido com os resultados das melhores submissões dos participantes da *WT10*.

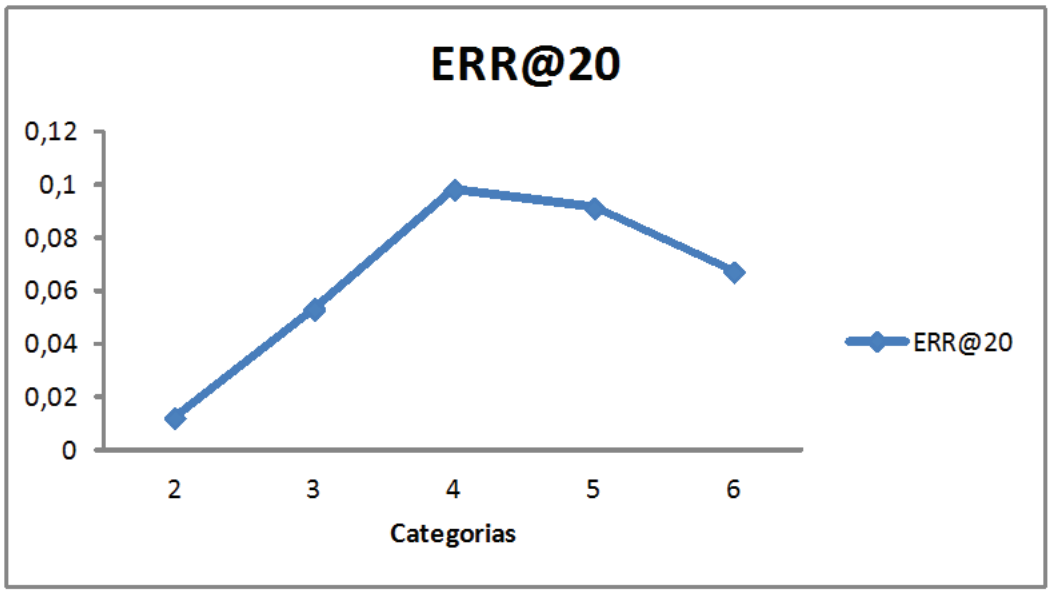

Figura 5.4: Gráfico da evolução da métrica ERR@20 por categoria.

O desempenho do *XHITS* com a AMCL mostra que o mesmo possui uma abordagem competitiva quando comparado aos resultados das submissões realizadas. Observamos, nesse sentido, que ordenando pela métrica ERR@20 o *XHITS* atingiu a sexta posição dentre vinte grupos. Ordenando por $nDCG$@20 e P@20, por sua vez, o *XHITS* atingiu a quarta posição.

O resultado baseado na ERR@20 indica que as posições da maior parte das páginas mais relevantes são mais significativas entre a décima e a vigésima posições. Assim, com 40% de páginas relevantes entre as vinte primeiras páginas, temos oito páginas relevantes em vinte. A P@10 é 0.4, o que significa que quatro páginas são relevantes dentre as dez primeiras. Então, descontando a P@10 da P@20, podemos concluir que o *XHITS* atingiu uma distribuição uniforme de páginas relevantes entre as primeiras dez e vinte páginas.

5.4.2
Experimento com Conjunto de Treinamento WT10

No presente experimento, os seguintes parâmetros de entrada forma definidos para o algoritmo 5.1, tal como descrito na seção anterior:

- o conjunto de treinamento L, formado pelos grafos gerados a partir dos tópicos da WT10;
- o conjunto O contendo as classificações de referência para os tópicos da WT10;

Grupo	**ERR@20**	**nDCG@20**	**P@20**	**MAP**
Microsoft Research	0.166	0.237	0.344	0.082
baseline	0.164	0.241	0.374	0.069
University of Massachusetts Amhers	0.138	0.293	0.484	0.148
Tsinghua University	0.128	0.201	0.331	0.112
University of Glasgow	0.127	0.245	0.411	0.127
XHITS	0.125	0.242	0.403	0.124
The University Of Melbourne	0.119	0.181	0.293	0.080
Carnegie Mellon University	0.112	0.212	0.400	0.157

Tabela 5.4: Os melhores resultados da busca *ad hoc* ordenado pela métrica ERR@20

- as consultas σ, formada pelos tópicos da WT10;
- o tempo total do experimento, $TExec$ com 168h;
- o maior número de categorias, $NumCat$ como 6;
- e, o número máximo de iterações do algoritmo de AMCL, $AMCLMaxIt$ como 100.

Finalizado o treinamento, a matriz F encontrada foi utilizada para classificar as páginas relativas as consultas σ do conjunto de teste, formado pelos grafos gerados a partir dos tópicos da *WT09*. A partir das classificações, foi gerada uma lista com as 10.000 primeiras páginas por cada consulta σ e calculadas as métricas definidas na seção 5.2, conforme os resultados publicados na competição *WT09*.

Dentro das 168 horas disponíveis por categoria para minimizar a função E, o instante em que o menor valor de E foi encontrado remete às mesmas conclusões do experimento da subseção 5.4.1 e por esse motivo não será reportado.

A variação da qualidade dos resultados em relação ao número de categorias é reportada no gráfico 5.5 corroborando com as conclusões da subseção 5.4.1.

Os melhores resultados desse experimento estão resumidos nas tabelas 5.5 e 5.6.

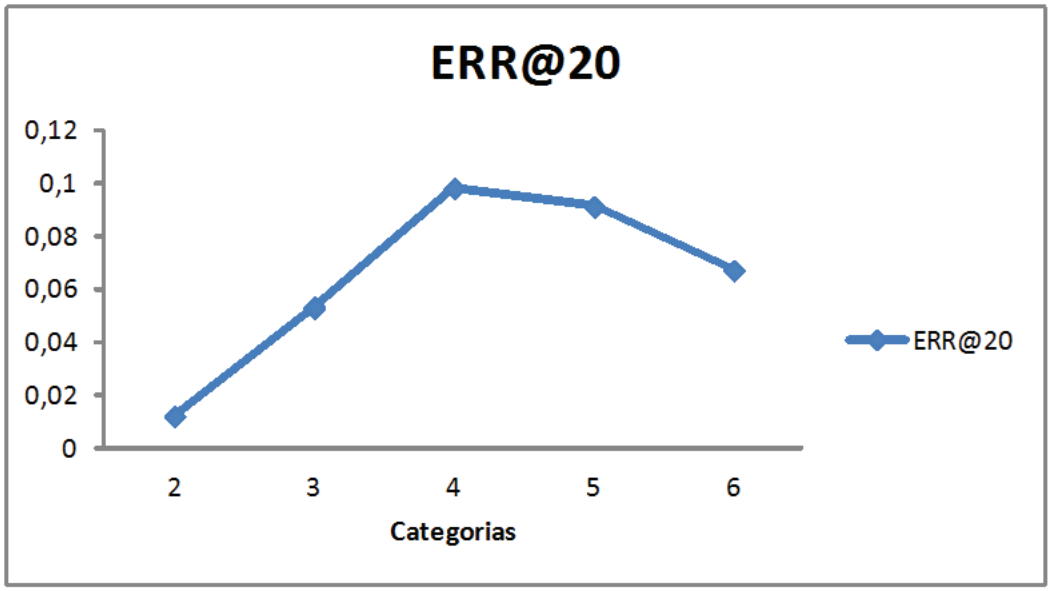

Figura 5.5: Gráfico da evolução da métrica $ERR@20$ por categoria.

Grupo	**P@20**
Microsoft Research Cambridge	0.390479
University of Amsterdam	0.382783
Microsoft Research Asia	0.377338
University of Waterloo	0.366025
University of Melbourn	0.297297
XHITS	0.219257
University of Waterloo(Smuker)	0.177072

Tabela 5.5: Os melhores resultados da busca *ad hoc* ordenado pela métrica $P@20$

Grupo	**MAP**
University of Waterloo	0.043362
University of Waterloo(Smuker)	0.034555
University of Melbourn	0.033712
Microsoft Research Asia	0.033240
University of Amsterdam	0.033225
XHITS	0.031236
Microsoft Research Cambridge	0.029981

Tabela 5.6: Os melhores resultados da busca *ad hoc* ordenado pela métrica MAP

Porém, o critério de ordenação utilizado para montar o resultado final da competição foi baseado na métrica MAP. Assim, comparamos os resultados alcançados pelo *XHITS* com os resultados das melhores submissões, tal como antes exposto, e o desempenho do *XHITS* com a AMCL se confirma como uma abordagem competitiva, reforçando o experimento anterior.

Ordenando tanto pela métrica $P@20$, conforme a tabela 5.5, tanto pela métrica MAP conforme a tabela 5.6, o *XHITS* atingiu a sexta posição dentre treze grupos que submeteram para a busca *ad hoc* cat-A.

Finalmente, é importante gizar que a principal característica do modelo do *XHITS* é que o mesmo somente utiliza a informação derivada dos hiperlinks para gerar as suas classificações das páginas, diferentemente das demais abordagens que utilizam frequência de texto e texto âncora, como empreendido pelos participantes das competições *WT09* e *WT10*, cujos resultados serviram, neste trabalho, como parâmetro de comparação com os obtidos através do experimentos realizados.