UNIVERSIDADE FEDERAL DO AMAZONAS
PRO-REITORIA DE PESQUISA E PÓS GRADUAÇÃO
DEPARTAMENTO DE APOIO A PESQUISA
PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA

COMPRESSÃO DE ÍNDICES INVERTIDOS DE IMPACTOS UNIFICADOS

Bolsista: Alef Pereira de Nazaré, FAPEAM

MANAUS
2015

UNIVERSIDADE FEDERAL DO AMAZONAS
PRO-REITORIA DE PESQUISA E PÓS GRADUAÇÃO
DEPARTAMENTO DE APOIO A PESQUISA
PROGRAMA INSTITUCIONAL DE BOLSAS DE INICIAÇÃO CIENTÍFICA

RELATÓRIO FINAL
PIB-E/0196/2014

COMPRESSÃO DE ÍNDICES INVERTIDOS DE IMPACTOS UNIFICADOS

Bolsista: Alef Pereira de Nazaré, FAPEAM
Orientador: Prof. Dr. André Luiz da Costa Carvalho

MANAUS
2015

RESUMO

Em máquina de busca, a preocupação de armazenar a mesma informação utilizando a menor quantidade de espaço possível se torna evidente, uma vez que a capacidade de armazenamento dos computadores é limitada, e as bases de dados utilizadas estão em constante crescimento. Para resolver esse problema, é cada vez mais comum a utilização de métodos de compressão nesses sistemas. A ideia deste trabalho consiste em estudar distribuições numéricas que facilitem a compressão de dados nas máquinas de busca, com o objetivo de melhorar o método o *Learn to Precompute Evidence Fusion* (LePrEF). Para a realização dos experimentos, foi utilizada uma implementação do LePrEF usando uma biblioteca *Distributed Evolutionary Algorithms in Python* (DEAP). Essa implementação do LePrEF foi modificada para poder realizar normalização Linear, Exponencial e Logarítmica sobre os dados que a mesma gera. As execução da implementação em DEAP com os parâmetros originais do LePrEF apresentaram resultados semelhantes em relação a qualidade das consultas entre as duas implementações. Os testes com os valores normalizados mostraram que os métodos de normalização Linear e Exponencial obtiveram resultados de qualidade de consulta competitivos com o método base e que utilizando essas normalizações a quantidade de espaço necessário para o armazenamento dos dados pode alcançar valores inferiores a metade da quantidade necessária para armazenas os impactos não normalizados.

Palavras-chave: Compressão; Indexação; Maquina de Busca;

LISTA DE FIGURAS

LISTA DE TABELAS

SUMÁRIO

INTRODUÇÃO

A medida que novos sites são criados e disponibilizados na internet, os mesmos precisam ser adicionados às bases de dados das máquinas de busca para que seja possível localizá-los e acessá-los através das ferramentas de procura na web. Uma vez que o número de novas páginas cresce constantemente, essas bases de dados se tornam cada vez maiores. Tendo como exemplo, o site www.worldwidewebsize.com [1] estimou que, em dezembro de 2014, apenas o Google possuía aproximadamente 45 bilhões de páginas em seu índice.

Para lidar com uma base de dados volumosa como essa, é necessário bastante poder de processamento. Além do mais, este não é o único problema gerado por este crescimento. Uma vez que a capacidade de armazenamento dos computadores é limitada, a preocupação de armazenar a mesma informação utilizando a menor quantidade de espaço possível se torna evidente, e por isso, métodos de compressão são geralmente usados nesses sistemas (Baeza-Yates & Ribeiro-Neto, 1999) [2].

Os índices invertidos, ou *inverted index,* permanecem sendo a principal estrutura de dados utilizadas nestes sistemas (Dean, J. 2009) [3]. Eles são compostos de vários termos, onde, para cada termo, está associada uma lista de ocorrências, ou *posting list*, que contém informações referentes a cada documento onde o termo se encontra. Nos sistemas de busca tradicionais, existe um índice invertido para cada fonte de evidência. Em 2012, foi apresentado o método *Learn to Precompute Evidence Fusion* (LePrEF), que utiliza programação genética para pré-computação de *Unified Term Impacts* (UTIs), valor único que representam o impacto do termo no documento (da Costa Carvalho, A. L. et al., 2012) [5], com base nas várias fontes de evidencia. O objetivo deste trabalho é a compressão desses índices gerados pelo método.

Alguns métodos de compressão, como o Elias gamma, são favorecidos quando os dados a serem comprimidos estão distribuídos de tal forma que a frequência de números menores é maior que a frequência de números maiores (Elias, P. 1975) [4]. A ideia deste trabalho consiste em estudar distribuições numéricas que facilitem a compressão de dados nas máquinas de busca, e propor modificações no método LePrEF (da Costa Carvalho, A. L. et al., 2012) [5], para que seja possível gerar índices em tais distribuições.

A quantidade de bits utilizados por um número comprimido com Elias gamma pode ser obtida sem a necessidade da compressão, este trabalho irá usar isto para calcular o espaço necessário para o armazenamento de impactos que serão computadas através do LePrEF

modificado para realizar as normalizações Linear, Exponencial e Logarítmica sobre esses impactos.

Este trabalho está estruturado da seguinte forma: A seção 1 apresenta os estudos base que deram a direção para este trabalho. A seção 2 descreve a metodologia utilizada para o desenvolvimento do mesmo. Na seção 3 são expostos e discutidos os resultados parciais obtidos até o momento. Enfim, são mostradas as conclusões e sugestões para futuros estudos.

1. REVISÃO BIBLIOGRÁFICA

No processamento de uma consulta, a etapa de fusão de evidencia se baseia em obter os resultados individuais, ou *rankings* individuais, da consulta, que é extraída de cada fonte de evidencia, e então os resultados individuais são fundidos em um único resultado, o *ranking* final, através de algum método de fusão. Buscando estratégias para tornar esta etapa mais crítica, vários estudos apresentaram métodos para combinação de diferentes fontes de evidencia para a geração dos resultados finais como combinação Linear simples, como por exemplo (T. Westerveld et al., 2001; I. Silva et al., 2000; P. Calado et al., 2003) [6, 7, 8]. Posteriormente, surgiu a ideia de utilizar aprendizagem de máquina para criação de funções de ordenação de resultados, ou funções de *ranking*, únicas (W. Fan et al., 2004) [9], que não mais calculavam os *rankings* individuais, e sim o *ranking* final da consulta, tendo como entrada as diferentes fontes de evidências. Entretanto, como a etapa de *ranking* nessas abordagens é realizado no momento da consulta, isso acaba deixando o processo mais complexo.

Para resolver esse problema, em um estudo de 2012, foi proposta a combinação de várias fontes de evidência através do calculo, em tempo de indexação, de um valor que representa a importância de um termo para um documento (da Costa Carvalho et al., 2012) [5]. Estes valores pré-computados são chamados de *Unified Term Impacts* (UTIs) [5]. No mesmo estudo, foi apresentado também um método para a computação desses índices, chamado *Learn to Precompute Evidence Fusion* (LePrEF) [5] que utiliza programação genética (William F. Punch et al., 1996) [10] para aprender padrões utilizados para geração dos UTIs. Como agora o cálculo do impacto dos termos é efetuado em tempo de indexação, o processo de consulta se torna muito mais eficiente.

Além da melhoria no processamento de consultas, foi apontado também que utilização de UTIs reduz o espaço necessário para armazenamento dos índices, já que não é necessário manter os índices originais das características dos termos [5]. Outro ponto importante a se observar é que o método proposto para a computação desses índices, o LePrEF, possibilita a geração de valores inteiros, assim facilitando a compressão dos mesmos, em comparação com a compressão de índices reais, que requerem maior carga de processamento (Baeza-Yates & Ribeiro-Neto, 1999; da Costa Carvalho, A. L. et al., 2012) [2, 5], o que pode resultar em ganhos de espaço ainda maiores, se utilizados métodos de compressão. Entretanto, o estudo da compressão dos índices inteiros não era o objetivo do trabalho.

Um recente trabalho na área de compressão de índices, faz uma comparação de desempenho entre diferentes métodos de compressão, para diferentes partes da estrutura de dados das máquinas de busca (Catena, M. et al., 2014) [11]. O estudo não somente se limita ao uso da taxa de compressão dos índices como forma de avaliação de qualidade dos métodos, mas também utiliza parâmetros, como por exemplo, tempo médio de respostas das consulta, para fornecer resultados comparativos de melhor qualidade no contexto de sistemas de máquina de busca. Este estudo revelou que a utilização de compressão levou a ganhos significativos no tempo médio de resposta das consultas, e são esses ganhos, um dos principais motivadores para o estudo dos valores de UTIs gerados pelo LePrEF, com o objetivo de melhorar ainda mais o método.

2. MÉTODOS UTILIZADOS

Algoritmo para geração dos índices. Neste trabalho é empregada uma implementação do LePrEF utilizando a biblioteca de programação genética *Distributed Evolutionary Algorithms in Python* (DEAP) (Félix-Antoine Fortin et al., 2012) [12] para a computação dos índices a serem comprimidos. DEAP é um sistema evolucionário recentemente desenvolvido, que tem como o objetivo rápida prototipação e testes de conceitos. Programação genética é uma meta-heurística que simula a evolução de indivíduos durante gerações. Ao passar das gerações, os indivíduos com maior grau de adaptabilidade, ou *fitness*, são mantidos enquanto os de grau inferior são cruzados com outros indivíduos ou descartados, então os sobreviventes dessa nova geração sofrem mutações, para assim serem testados novamente. Cada indivíduo é representado por uma árvore contendo nós finais ou funções. No caso do LePrEF, os finais, ou nós folhas, são compostos pelos dados das fontes de evidência, enquanto que os nós de funções, ou nós intermediários, representam as operações de soma, multiplicação ou divisão que serão efetuadas sobre os dados para a obtenção dos valores dos impactos. Os parâmetros do método seguem a mesma configuração do LePrEF, e são apresentados na tabela a seguir:

Parâmetro	**Valor**
N. de gerações	40
Tamanho da população	1000
Profundidade da árvore	17
Tamanho do torneio	6
Taxa de cruzamento	0.85
Taxa de mutação	0.05

***Tabela 1**: Configuração dos parâmetros do LePrEF DEAP.*

Métodos de normalização dos impactos. Para estes experimentos, o LePrEF foi modificado para aplicar diferentes transformações sobre os valores gerados. Nos testes utilizando algum dos método de normalização implementados, os impactos são gerados e armazenados na lista de índices invertidos normalmente, em seguida o método de normalização escolhido é aplicado sobre as listas de ocorrências de cada termos de forma independente, assim os valores da lista de ocorrências de um termo não influenciam nos valores da lista de outro termo. Com a aplicação do método é obtida uma nova lista de índices invertidos com valores

normalizados, restando apenas a realização da discretização no processo de geração dos índices. A realização da normalização antes da avaliação da qualidade do indivíduo, permite que a programação genética leve em consideração os novos valores de impactos durante a evolução de seus indivíduos, identificando assim indivíduos com bom nível de adaptabidade mesmo depois da mudança em sua distribuição numérica. O objetivo do processo de normalização é a possibilidade de definir limites aos valores dos dados, isso permite a geração de listas com valores de impactos em intervalos numéricos reduzidos o que aumenta a taxa de compressão. Os métodos utilizados para normalização dos dados foram: Linear, Exponencial e Logarítmica. Todos os métodos tomam como parâmetro um valor do impacto (*uti*), o valor máximo de impacto da lista de ocorrências (*maximo*) e o valor de parâmetro que determina o limite máximo dos novos impactos no intervalo ao qual serão mapeados (*M*), e retornam o valor do impacto calculado no novo intervalo (uti_{novo}). Todas as funções consideram $uti \geq 0$ e estão no domínio $0 \leq uti_{novo} \leq M-1$. Nos experimentos foi utilizado o valor 8 como parâmetro M, assim, os valores de UTIs obtidos na normalização estão 0 e 7. As formulas dos mesmos são descritos a seguir.

- **Linear:**

$$uti_{novo} = linear(uti, maximo, M) = uti \times \frac{M-1}{maximo}$$

- **Exponencial**:

$$uti_{novo} = \exp(uti, maximo, M) = 2^{uti \times \frac{\log_2(M-1)}{maximo}}$$

- **Logarítmica:**

$$uti_{novo} = lgrt(uti, maximo, M) = \log_2(uti \times \frac{2^{M-1}}{maximo})$$

Discretização dos valores. Foram levadas em consideração duas estratégias de discretização: truncamento e arredondamento. Caso o experimento seja executado com algum método de normalização, a discretização é realizada logo após a aplicação do método, caso contrário, o processo é executado assim que o impacto é criado. Em ambos os casos a discretização é efetuada antes da avaliação de qualidade do indivíduo.

Base de dados. A base de referência utilizada neste trabalho para a avaliação da qualidade do LePrEF é a LETOR (Liu et al., 2007) [13]. O conjunto de dados adotado é o MQ2007, e o mesmo possui 1700 consultas e é composta por 46 características de 1000 documentos. Das 46 características do conjunto de dados MQ2007, foram selecionadas 21 para serem utilizadas

no método LePrEF, por serem as únicas que estão disponíveis durante a indexação.

Modo de validação. LETOR4 é dividida em cinco partes, permitindo validação cruzada de cinco folds para avaliação dos resultados. Na validação cruzada de cinco, três folds são utilizados para treino, um para validação e um para teste.

Métrica de avaliação das consultas. O método de avaliação adotada neste trabalho para verificar a qualidade das consultas, é a NDCG@N para consultas individuais, e a MEAN NDCG para o conjunto de consultas total em treino, validação e teste.

Avaliação da compressão. O método de compressão avaliado nos experimentos foi o Elias gamma. Não foi realizado a compressão em si dos valores de impacto, em vez disso, foi utilizado a função que calcula a quantidade de bits necessários para armazenar determinado valor.

- Formula:

$$bits = f(x) = 2 \times \lfloor \log_2(x) \rfloor + 1$$

Para o objetivo do trabalho esta informação é suficiente, pois através da formula é possível calcular diretamente a quantidade de bits utilizados para os métodos comprimidos com Elias gamma e obter assim a taxa de compressão sem a necessidade de comprimir os dados.

3. RESULTADOS E DISCUSSÕES

3.1. Reprodução Dos Resultados Base

Neste trabalho, foi decidido como primeira atividade experimental, reproduzir os resultados obtidos pelo método LePrEF original em uma escala menor. Para isso, foram realizadas várias execuções com diferentes sementes. Inicialmente foram executadas cinco sementes aleatórias para cada fold, e o indivíduo da semente com o melhor resultado de MEAN NDCG@N no conjunto de validação foi selecionado como o melhor. A necessidade da execução do método com várias sementes aleatórias se dá pelo fato da população inicial ser gerada aleatoriamente, assim o resultado da GP pode ser afetado pela inicialização da população. Os resultados obtidos com a implementação DEAP de programação genética são similares aos obtidos no trabalho original apresentado por da Costa Carvalho, et al. Os valores de MEAN NDCG@N para as cinco sementes e a melhor delas em cada fold, utilizando UTIs reais, são mostrados na figura 1:

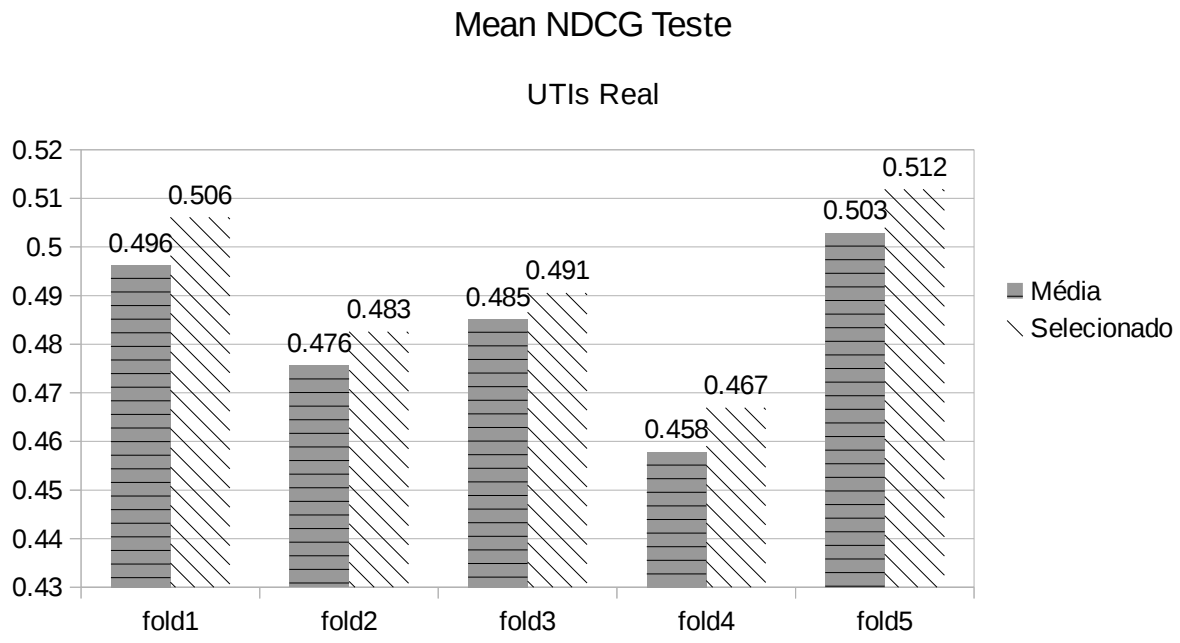

***Figura 1**: Média dos MEAN NDCG@N das cinco execuções e o valor da semente selecionada paro o teste usando impactos reais.*

Foram realizados também experimentos utilizando UTIs inteiros como forma de comparação com o método original. Geração de UTIs inteiros é o alicerce deste trabalho, uma

vez que o objetivo do mesmo é a compressão dos impactos, trabalhar com números reais pode não ser viável, como o processo de compressão de números em ponto flutuante é mais complexa que compressão de números inteiros podendo interferir no desempenho das máquinas de busca. Para a discretização dos impactos no LePrEF original, foi usada a estratégia de truncamento logo em seguida da geração dos UTIs, após isso é realizado o cálculo de adaptabilidade do indivíduo. O calculo de *fitness* efetuado posteriormente aos truncamentos permite que a GP capture as características produzidas por essas alterações, assim, gerando indivíduos adaptados a essas mudanças. Os valores de MEAN NDCG para as cinco sementes e a melhor delas em cada fold, utilizando UTIs truncados, são mostrados na figura 2:

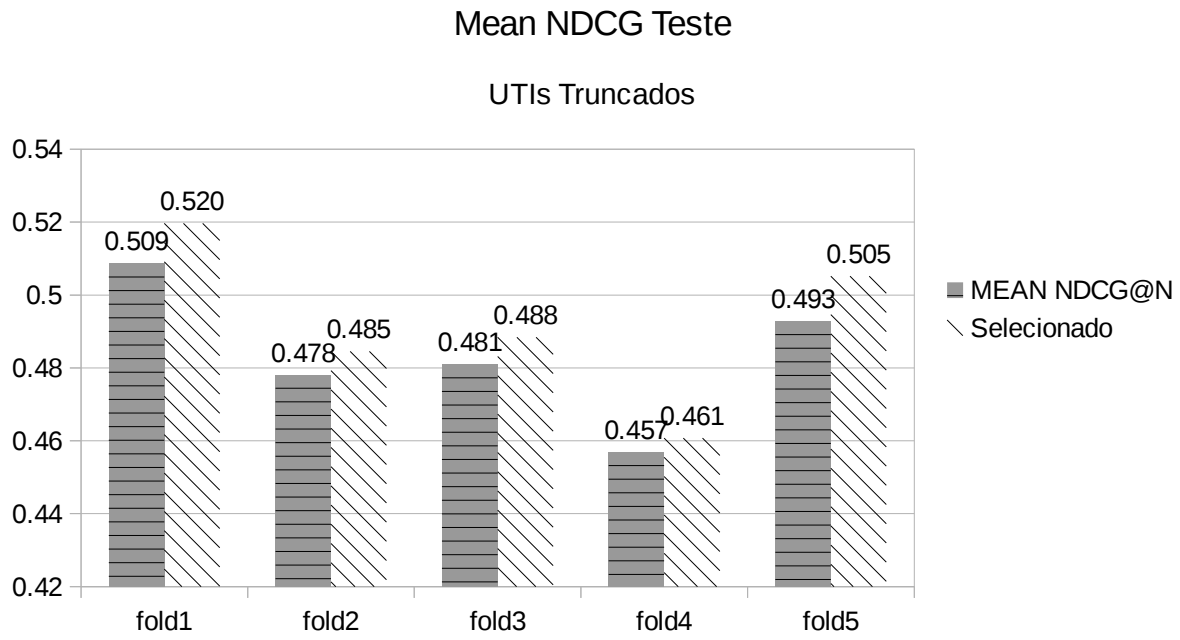

***Figura 2**: Média dos MEAN NDCG@N das cinco execuções e o valor da semente selecionada paro o teste usando impactos truncados.*

Abaixo está a tabela com a média dos resultados dos MEAN NDCG dos métodos para cada fold. Com os resultados obtidos por este experimento, vemos que ambos os métodos tiveram desempenho semelhante, com o LePrEF Inteiro obtendo uma média ligeiramente inferior ao LePrEF Real, o que é de se esperar. Essa característica foi apontada no trabalho original, nos experimento realizado por da Costa Carvalho porém, a diferença entre os métodos foi um pouco mais evidente, isso pode ser explicado pela quantidade de execuções com sementes aleatórias realizadas, já que no experimento de 2012 foram executadas 10 sementes aleatórias e nestes experimentos foi escolhido a execução de apenas cinco sementes.

Os resultados pode ser visualizado na tabela a seguir:

	LePrEF Real	LePrEF Inteiro (Truncados)
Fold1	0,4962826547	0,5086452199
Fold2	0,4756109352	0,4780561735
Fold3	0,485188618	0,4809642648
Fold4	0,4578906641	0,4569012898
Fold5	0,5029263971	0,4927660972
Média	0,4835798538	0,483466609

***Tabela 2**: Média dos resultados dos folds para LePrEF Real e LePrEF Inteiro.*

Nos resultado mostrados a seguir serão levados em consideração apenas os resultados para do experimento para UTIs inteiros.

3.2. Experimento Com Normalização

A metodologia de normalização empregada nos experimentos tem como objetivo reduzir e limitar o intervalo numérico dos UTIs gerados. No processo de cálculo dos UTIs sem normalização, os impactos podem assumir valores bastante variados dependendo da semente aleatória. Todos os métodos de normalização experimentados neste trabalho recebem um valor *M* como parâmetro, este valor representa o limite máximo ao valor que um UTI pode obter, independente do seu valor originado da programação genética. No caso destes experimentos, foi decidido utilizar o valor 8 para a normalização dos impactos. Com o valor desse parâmetro e com uma função para mapear os números, foi possível transformar os dados da distribuição gerada pela GP, em dados de uma outra distribuição, com valor máximo pré-definido.

As funções utilizadas nesse experimento foram a Exponencial, a Linear e a Logarítmica. Para cada uma delas, os valores reais gerados pela GP foram transformados resultando em um outro número real na nova distribuição. Para transformá-los em inteiro, foi aplicado um dos métodos de discretização utilizados, truncamento ou arredondamento, obtendo assim um número inteiro entre 0 e *M-1*. Para o cálculo do fitness, os UTIs normalizados sofrem o processo inverso da transformação, porém, não retornando ao valor real, e sim um valor aproximado, já que os UTIs utilizados para os cálculos são inteiro e

perderam informação. Após a realização desse cálculo de retorno, a indivíduo era avaliado.

Cada método de normalização foi testado com um dos tipos de discretização, assim gerando seis conjuntos de experimentos: Exponencial usando arredondamento (ExpRound), Exponencial usando truncamento (ExpTrunc), Linear usando arredondamento (LinearRound), Linear usando truncamento (LinearTrunc), Logarítmica usando arredondamento (LogRound), Logarítmica usando truncamento (LogTrunc). Os resultados obtidos estão na tabela a seguir:

	ExpRound	ExpTrunc	LinearRound	LinearTrunc	LogRound	LogTrunc
Fold1	0,4986851	0,5010096	0,5010157	0,5006127	0,4621190	0,4652815
Fold2	0,4672193	0,4725188	0,4701902	0,4709779	0,4468561	0,4384385
Fold3	0,4764583	0,4724744	0,4754251	0,4727208	0,4552860	0,4495370
Fold4	0,4489213	0,4479707	0,4489292	0,4479383	0,4186383	0,4188574
Fold5	0,4947497	0,4929439	0,4945809	0,4907245	0,4628423	0,4582845
Média	0,4772067	0,4773835	0,4780282	0,4765948	0,4491484	0,4460798

***Tabela 3**: Resultados para os experimentos com normalização.*

As médias de NDCG@N pra cada conjunto de experimentos segue no gráfico. Os resultados foram dispostos em ordem decrescente em relação ao valor obtido:

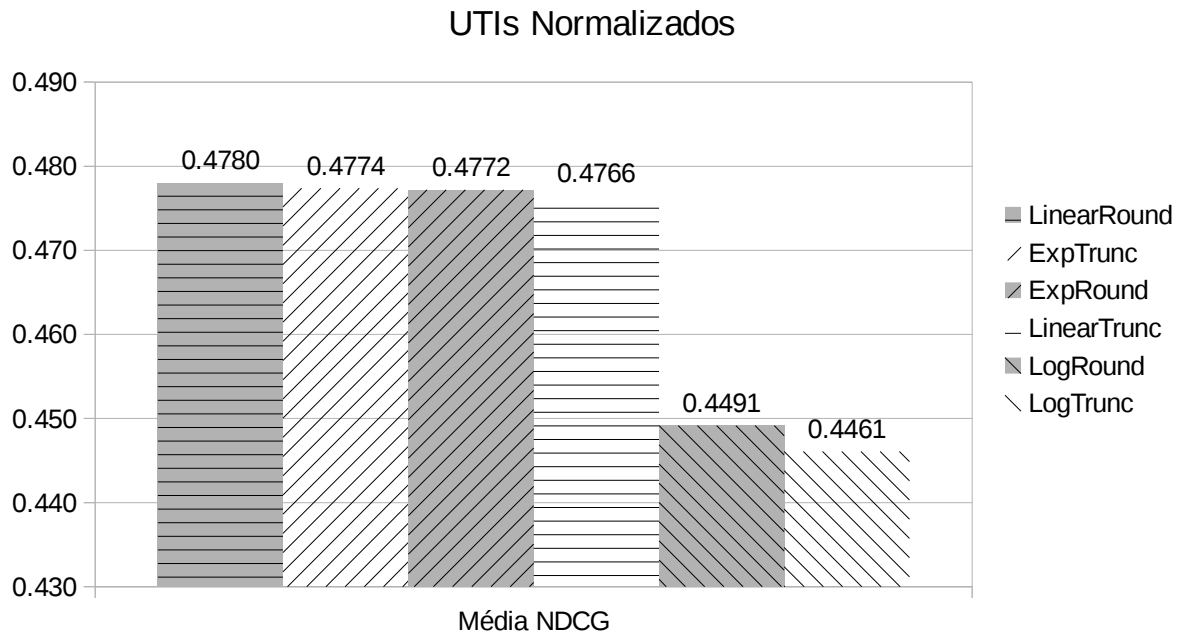

***Figura 3**: Média NDCG para UTIs Normalizados.*

Como é possível observar, o LePrEF utilizando a normalização Linear e a Exponencial tiveram resultados bem próximos um dos outros tanto com truncamento quando com

arredondamento. Nos experimentos utilizando a função Linear, o método de discretização que teve melhor resultado na média foi o arredondamento, enquanto que para a função Exponencial foi a estratégia de truncamento que se mostrou superior ao arredondamento. As execuções utilizando a função Logarítmica no entanto, obtiveram resultados significativamente inferiores aos outros métodos tanto para truncamento quanto para arredondamento. A diferença entre o resultado do Linear Truncado – conjunto de experimento que obteve o pior resultado dentre os Lineares e Exponenciais – e o resultado da Logarítmica arredondada – que obteve o melhor resultado das Logarítmicas – é de 0.275, o que significa que a LogRound foi 2.75% inferior ao LinearTrunc. Uma porcentagem de 2.75% parece mínima, mas num sistema de máquina de busca pode fazer uma grande diferença.

Todos os métodos de normalizados testados obtiveram resultados inferiores ao método LePrEF Inteiro que usa truncamento para discretização dos impactos, isso já podia ser esperado, pois a normalização interfere bem mais no intervalo numérico dos UTIs do que um simples trucamento dos valores reais calculados pela GP, desta forma há perda de informação um pouco maior nos métodos normalizados em comparação com o LePrEF Inteiro. A comparação entre o melhor de cada normalização e o LePrEF Inteiro está na figura a seguir:

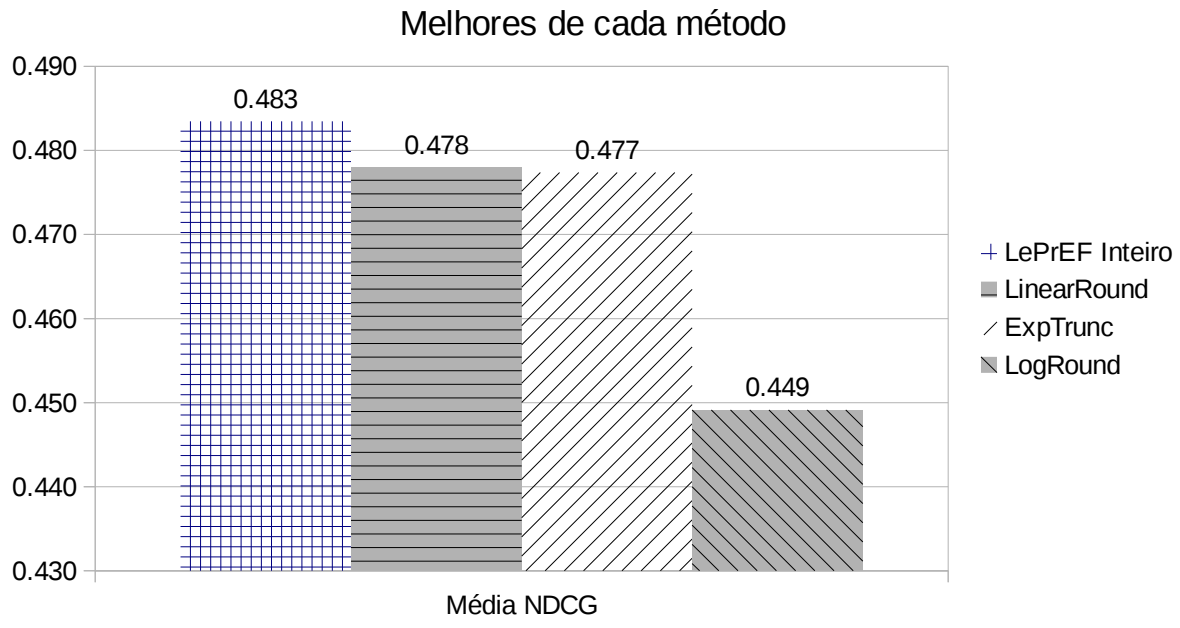

***Figura 4**: Comparação entre os métodos de normalização e o LePrEF Inteiro.*

Tomando como base o LePrEF Inteiro e comparando com os outros métodos, o LinearRound ficou com uma diferença de 0.005, ou seja, meio por cento abaixo da base. Para o método ExpTrunc, essa porcentagem sobe para apenas 0.6%. Já a discrepância maior está na

comparação com o LogRound, que é de 0.034, o que significa que o método LogRound foi 3.4% inferior ao método base nos testes realizados. Ainda que com resultados inferiores, as diferenças dos métodos Linear e Exponencial relativas ao método base podem ser consideradas pequenas em relação as diferenças obtida pelos métodos Logarítmicos, assim as estratégias Linear e Exponencial se mostram bem mais competitivas com o LePrEF Inteiro que a estratégia Logarítmica no que diz respeito a qualidade de consultas.

Na subseção a seguir serão mostradas as influencias que a normalização pode causar na compressão dos impactos unificados.

3.3. Compressão Dos Impactos Utilizando Elias Gamma

O número de impactos gerados para cada fold segue na tabela abaixo, este número é usado para calcular a média dos bits utilizados em cada método.

	Fold1	Fold2	Fold3	Fold4	Fold5
n	49989	54083	55437	55246	53509

***Tabela 4**: Número de impactos gerados para cada fold.*

A seguir serão apresentadas as tabelas das médias dos bits utilizados por cada conjunto de execuções, e por ultimo uma tabela comparativa entre os métodos. Os bits foram contados para cada usando a formula descrita na seção anterior e divididos pelo número de UTIs do fold, apresentados na tabela 4, assim as tabelas representam a quantidade de bits médios que é necessário para armazenar um valor de UTI após a compressão utilizando Elias gamma.

Para LePrEF Inteiro, observamos que os valores médios de bits utilizados para cada folds é muito variado dependendo da semente, esse resultado ocorre devido a aleatoriedade com a quais os impactos são gerados no método, por isso é possível observar valores médios como na seed1 para o fold5 onde cada impacto ocupa em média 12.25 bits, representado que para essa semente a distribuição numérica obtida possuem muitos valores altos ocupando mais espaço de armazenamento, e a semente 4 no mesmo fold, onde cada impacto ocupa em média 2.98 bits, representado que a distribuição numérica é composta por números menores, ocupando menos espaço para armazenamento que os UTIs obtidos pela semente um. Os valores médios de bits utilizados por UTIs para o LePrEF Inteiro podem ser observados na tabela a seguir:

Média LePrEF Inteiro	fold1	fold2	fold3	fold4	fold5
Seed1	6.9278041169	12.585932733	5.3802153796	7.6427976686	12.25765759
Seed2	7.4741243073	3.088364181	6.5964247705	13.311443362	3.929058663
Seed3	4.3726219768	6.1967531387	13.31444703	6.7920392427	4.2389504569
Seed4	2.9978595291	4.0289554943	4.5464401032	5.6996705644	2.9861518623
Seed5	10.223389146	12.839136143	10.950682757	6.9298410745	11.54151638
Média	6.3991598152	7.7478283379	8.157642008	8.0751583825	6.9906669906

Tabela 5*: Média para cada semente e a média dos folds do método LePrEF Inteiro.*

Como no método LePrEF Inteiro não há normalização, é normal que os valores variem bastante, já nos métodos com normalização, o esperado é que os mesmos possuam uma variação menor e também uma média de bits por UTIs inferior inferior ao LePrEF Inteiro. Isso é explicado pelo fato do parâmetro M escolhido ser igual a 8, um número relativamente pequeno, devido isso, enquanto os valores de impactos do método base podem sofrer maior oscilação, os UTIs gerados pelos métodos normalizados estão sempre no intervalo de 0 a 7. Os resultados para o método Linear utilizando truncamento e arredondamento seguem nas tabelas abaixo:

Média LinearTrunc	fold1	fold2	fold3	fold4	fold5
Seed1	3.8846946328	3.2006360594	3.8347493551	3.8138326757	3.6554972061
Seed2	3.6421012623	3.8799068099	3.7906993524	3.5745212323	3.4094451401
Seed3	3.9044789854	3.889114879	3.7508703573	3.9936103971	3.7945953017
Seed4	3.7913140891	3.5654641939	3.2621895124	3.9513629946	3.9707712721
Seed5	3.6340794975	3.7611264168	3.7371611018	3.8178510661	4.0565886113
Média	3.7713336934	3.6592496718	3.6751339358	3.8302356732	3.7773795063

Tabela 6*: Média para cada semente e a média dos folds do método LinearTrunc.*

Média linearRound	fold1	fold2	fold3	fold4	fold5
Seed1	3.9309848167	3.9325296304	3.9453072857	3.9085725663	4.0363116485
Seed2	3.4957490648	3.7484791894	3.8780778181	3.8120406907	3.6849875722
Seed3	3.9473484167	3.3234842742	3.973681837	3.965210151	4.0053262068
Seed4	3.7729700534	3.5586043674	3.8529321572	3.8816746914	3.9897961091
Seed5	3.8819940387	3.9622062386	3.6184136948	3.7661731166	3.7275972266
Média	3.805809278	3.70506074	3.8536825586	3.8667342432	3.8888037526

Tabela 7*: Média para cada semente e a média dos folds do método LinearRound.*

O comportamento esperado pelo método Exponencial não é tão diferente do Linear já que ambos são normalizados. Os resultados para Exponencial com impactos trucados e arredondados podem ser visualizados nas tabelas abaixo:

Média expTrunc	fold1	fold2	fold3	fold4	fold5
Seed1	3.5624237332	3.9306806205	3.9243826325	3.5377040872	3.6925563924
Seed2	3.7021744784	3.7174158238	3.9357829608	3.6224161025	3.7152815414
Seed3	3.66872712	3.4446868702	3.5627108249	3.2008109184	3.8609019044
Seed4	3.9654323951	3.5949004308	3.8415498674	3.438438982	3.8322338298
Seed5	3.5317769909	3.598117708	3.4140014792	3.7470224089	3.7748976808
Média	3.6861069435	3.6571602907	3.735685553	3.5092784998	3.7751742697

***Tabela 8**: Média para cada semente e a média dos folds do método ExpTrunc.*

Média expRound	fold1	fold2	fold3	fold4	fold5
Seed1	3.9433075277	4.0146996283	3.9313454913	4.1847011548	3.9378609206
Seed2	3.7580667747	3.6065122127	3.8265418403	4.030083626	3.8798893644
Seed3	3.7912740803	4.2647967014	4.0875588506	3.6921044057	3.7519482704
Seed4	3.2710396287	3.6829132999	3.9381279651	4.0142634761	3.9693322619
Seed5	3.743663606	3.7489969122	3.8339376229	3.8854577707	4.0776878656
Média	3.7014703235	3.8635837509	3.923502354	3.9613220867	3.9233437366

***Tabela 9**: Média para cada semente e a média dos folds do método ExpRound.*

Um comportamento parecido com o do Exponencial e do Linear é esperado para os experimentos utilizando a função Logarítmica. Os resultados para UTIs truncados e arredondados podem ser verificado nas tabelas a seguir:

Média logTrunc	fold1	fold2	fold3	fold4	fold5
Seed1	4.0806577447	3.9227483682	3.9679455959	4.0477862651	4.0484778262
Seed2	4.0001200264	4.0296026478	4.0078828219	4.0331788727	4.0083537349
Seed3	3.6854107904	3.7193757743	4.0114905208	3.6726821851	4.0701564223
Seed4	4.0805377183	3.9894791339	4.0109854429	4.0482749882	3.8463436057
Seed5	4.0614935286	3.9905885398	4.0031206595	4.0353509756	3.9352258499
Média	3.9816439617	3.9303588928	4.0002850082	3.9674546574	3.9817114878

***Tabela 10**: Média para cada semente e a média dos folds do método LogTrunc.*

Média logRound	fold1	fold2	fold3	fold4	fold5
Seed1	4.0367080758	4.0364809644	4.0098309793	4.048437896	4.0906202695
Seed2	4.0644941887	3.9946748516	4.0112379819	4.048437896	4.0906202695
Seed3	4.0803376743	3.9760368323	3.9890506341	4.048437896	4.0700442916
Seed4	3.8467462842	4.0364809644	4.0595811462	4.048437896	4.0906202695
Seed5	4.0644941887	4.032265222	4.0114905208	4.0358577997	4.0677829898
Média	4.0185560823	4.015187767	4.0162382524	4.0459218767	4.081937618

***Tabela 11**: Média para cada semente e a média dos folds do método LogRound.*

A tabela 12 apresenta os resultados médios de cada fold e a média geral dos métodos. Como visto abaixo, é possível observar que até mesmo a média dos folds para o método não normalizado possui uma variação superior em relação aos métodos normalizados. Já era aguardado que os resultados de bit médio utilizados para armazenamento nos métodos normalizados seria inferior aos resultados do LePrEF Inteiro, o que pode ser observado na tabela:

	LePrEF Inteiro	**ExpRound**	**ExpTrunc**	**LinearRound**	**LinearTrunc**	**LogRound**	**LogTrunc**
Fold1	6.39916	3.70147	3.68611	3.80581	3.77133	4.01856	3.98164
Fold2	7.74783	3.86358	3.65716	3.70506	3.65925	4.01519	3.93036
Fold3	8.15764	3.92350	3.73569	3.85368	3.67513	4.01624	4.00029
Fold4	8.07516	3.96132	3.50928	3.86673	3.83024	4.04592	3.96745
Fold5	6.99067	3.92334	3.77517	3.88880	3.77738	4.08194	3.98171
Média	7.47409	3.87464	3.67268	3.82402	3.74267	4.03557	3.97229

***Tabela 12**: Média de cada fold e a média geral dos folds para cada método.*

Quanto menor a média de bits utilizados pra cada UTI, menor o espaço necessário para armazenar os impactos, desta forma, os métodos normalizados apresentaram uma vantagem bastante significativa nessa característica sobre o LePrEF Inteiro. Essa vantagem pode ser vista na figura a seguir:

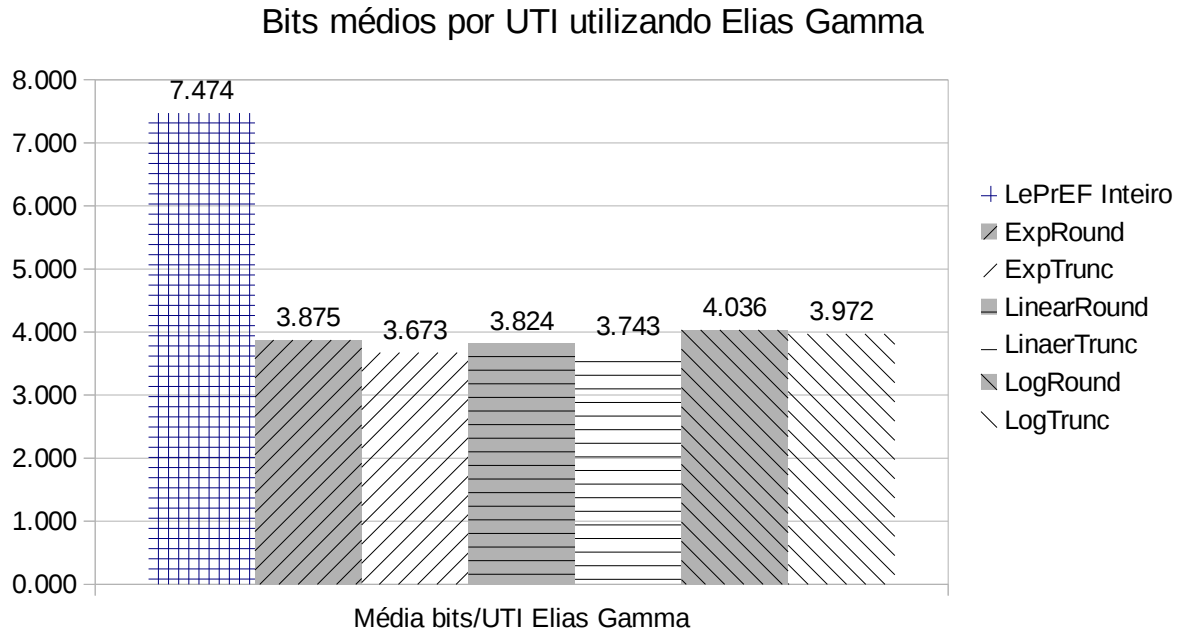

***Figura 5**: Comparação do valor médio de bits utilizados pelo método base e os métodos normalizados.*

A Figura 5 apresenta de forma visual os resultados resumidos da tabela 12. Por ela podemos realizar várias observações importantes. Por exemplo, assim como no teste de

qualidade das consultas, os métodos Linear e Exponencial obtiveram resultados melhores que o Logarítmico também na quantidade de bits por UTI. Ambos os resultados podem ser reflexos da distribuição numérica característica do da função Logarítmica.

Outro ponto a se observar é que para cada método de normalização, o método de discretização que obteve menor valor necessário para armazenamento, foi o truncamento, resultado explicado pelo fato de o arredondamento escolher o valor do inteiro mais próximo do valor real, tanto antecessor quanto sucessor, enquanto que no truncamento, o inteiro escolhido é sempre o antecessor ao número real.

Neste experimento o Exponencial Truncado obteve média de bits por UTI inferior a metade da média calculada para o LePrEF Inteiro neste experimento, ou seja, os impactos para o método Exponencial Truncado ocupam menos que a metade do espaço necessário para armazenar os UTIs do método base. Outro método que atingiu um resultado bastante parecido foi o Linear Trucado, que obteve média de bits por UTI ligeiramente superior a metade do LePrEF Inteiro. Assim, Exponencial Truncado e Linear Truncado obtiveram os melhores resultados no teste de bits por UTI utilizando Elias gamma para compressão.

Em ambos os testes os métodos Linear e Exponencial obtiveram resultados superior em relação ao Logarítmico, e no caso do teste de qualidade de consulta, tanto Linear quanto Exponencial atingiram resultados bem próximos ao método base, com isso, é possível afirmar que estes dois métodos competitivos com o LePrEF Inteiro, e a vantagem em armazenamento justifica estudos futuros.

AGRADECIMENTOS

Este trabalho foi realizado com o auxílio da FAPEAN (PIB-E/0196/2014).

CONCLUSÕES

De acordo com os resultados obtidos pelos experimentos com a implementação em DEAP, foi possível confirmar a validade do método LePrEF para geração de UTIs, e também que é possível gerar impactos normalizados sem grande perda de qualidade de consultas.

Para o teste de qualidade de consultas as estratégias Linear e Exponencial se mostram bastante competitivas com o LePrEF Inteiro, já a estratégia Logarítmica não obteve resultados satisfatórios.

Da mesma forma, para o teste da quantidade de bits necessários para o armazenamento em cada método, Exponencial e Linear obtiveram os melhores resultados. Em destaque os métodos que utilizaram a estratégia de truncamento para discretização dos UTIs, que no caso do método Exponencial Truncado, os impactos obtidos ocuparam menos que a metade do espaço necessário para armazenar os UTIs do método base, e também o método Linear Trucado, que obteve média de bits por UTI ligeiramente superior a metade do LePrEF Inteiro.

Os resultados são indicadores suficientes de que a normalização dos UTIs gerados pelo LePrEF é uma boa estratégia para obter distribuições com melhores taxas de compressão sem grande impacto na qualidade das consultas, o que motiva a continuação dos estudos e experimentos utilizando esta estratégia com o objetivo de minimizar e quem sabe superar os resultados dos métodos já propostos em qualidade das consultas.

REFERÊNCIAS BIBLIOGRÁFICAS

[1] WorldWideWebSize.com. The size of the Internet. Acesso em: 29 de janeiro de 2015. Disponível em: <http://www.worldwidewebsize.com>.

[2] Baeza-Yates, R., e Ribeiro-Neto, B. (2011). Modern Information Retrieval. Addison-Wesley Professional, Boston, MA, USA.

[3] Dean, J.: Challenges in building large-scale information retrieval systems: invited talk. In: Proc. WSDM ’09. (2009)

[4] Elias, P.: Universal codeword sets and representations of the integers. Trans. Info. Theory 21(2) (1975).

[5] da Costa Carvalho, A. L., Rossi, C., de Moura, E. S., da Silva, A. S., e Fernandes, D. (2012). Lepref: Learn to precompute evidence fusion for efficient query evaluation. Journal of the American Society for Information Science and Technology, 63(7):1383–1397.

[6] T. Westerveld, W. Kraai, D. Hiemstra, Retrieving web pages using content, links, urls and anchors, in: Notebook of 10th Text Retrieval Conference-TREC, 2001.

[7] I. Silva, B. Ribeiro-Neto, P. Calado, E. Moura, N. Ziviani, Link-based and content-based evidential information in a belief network model, in: SIGIR ’00: Proceedings of the 23rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, ACM Press, New York, NY, USA, 2000, pp. 96–103.

[8] P. Calado, M. Cristo, E. Moura, N. Ziviani, B. Ribeiro-Neto, M.A. Gonçalves, Combining link-based and content-based methods for web document classification, in: CIKM ’03: Proceedings of the 12th International Conference on Information and Knowledge Management, ACM Press, New York, NY, USA, 2003, pp. 394–401.

[9] W. Fan, M. Gordon, P. Pathak, W. Xi, E. Fox, Ranking function optimization for effective web search by genetic programming: an empirical study, in: System Sciences, Proceedings of the 37th Annual Hawaii International Conference on System Sciences (HICSS’04)-Track 4, vol. 4, IEE CNF, 2004.

[10] William F. Punch and Douglas Zongker and Erik D. Goodman. The Royal Tree Problem, a Benchmark for Single and Multiple Population Genetic Programming. In Peter J. Angeline and K. E. Kinnear, Jr. editors, Advances in Genetic Programming 2, chapter 15, pages 299-316. MIT Press, Cambridge, MA, USA, 1996.

[11] Catena, M., Macdonald, C., & Ounis, I. (2014). On Inverted Index Compression for Search Engine Efficiency. In Advances in Information Retrieval (pp. 359-371). Springer International Publishing.

[12] Félix-Antoine Fortin, François-Michel De Rainville, Marc-André Gardner, Marc Parizeau and Christian Gagné, “DEAP: Evolutionary Algorithms Made Easy”, Journal of Machine Learning Research, pp. 2171-2175, no 13, jul 2012.

[13] Liu, T.-Y., Xu, J., Qin, T., Xiong, W., e Li, H. (2007). LETOR: Benchmark Dataset for

Research on Learning to Rank for Information Retrieval. In LR4IR 2007, in conjunction with SIGIR 2007.

CRONOGRAMA

Nº	Descrição	Ago 2014	Set	Out	Nov	Dez	Jan 2015	Fev	Mar	Abr	Mai	Jun	Jul
1	Revisão Bibliográfica.	x	x	x									
2	Estudo sobre distribuição de inteiros.			x	x	x							
3	Estudo sobre compressão de índices de máquinas de busca.					x	x	x					
4	Criação do método.						x	x	x				
5	Experimentos qualitativos.								x	x	x		
6	Experimentos quantitativos.									x	x	x	
7	Elaboração do Resumo e Relatório final.											x	x
8	Preparação da apresentação final para o progresso.												x