Universidade Federal de Ouro Preto - UFOP

Instituto de Ciências Exatas e Biológicas - ICEB

Departamento de Computação - DECOM

UMA ABORADAGEM INCREMENTAL PARA REMOÇÃO DE AMBIGUIDADE DE NOMES EM CITAÇÕES BIBLIOGRÁFICAS

Aluno: Herculano Gripp Neto
Matricula: 07.1.419

Orientador: Anderson Almeida Ferreira

Ouro Preto
1 de julho de 2011

Universidade Federal de Ouro Preto - UFOP

Instituto de Ciências Exatas e Biológicas - ICEB

Departamento de Computação - DECOM

UMA ABORADAGEM INCREMENTAL PARA REMOÇÃO DE AMBIGUIDADE DE NOMES EM CITAÇÕES BIBLIOGRÁFICA

Relatório de atividades desenvolvidas apresentado ao curso de Bacharelado em Ciência da Computação, Universidade Federal de Ouro Preto, como requisito parcial para a conclusão da disciplina Monografia I (BCC390).

Aluno: Herculano Gripp Neto
Matricula: 07.1.419

Orientador: Anderson Almeida Ferreira

Ouro Preto
1 de julho de 2011

Resumo

Problemas de ambiguidade de nomes de autores de artigos científicos são comumente encontrados, devido a grande diversidade de fontes de dados utilizadas pelas bibliotecas digitais para a coleta dos dados, a falta de padronização na forma de escrever os nomes de autores nos artigos e a ocorrência de vários autores com o mesmo nome. Assim, é comum encontrar, em bibliotecas digitais, artigos de um mesmo autor com diferentes nomes deste autor, o que pode levar a tratar cada nome como se fosse de um autor diferente. Outra situação, que também ocorre, é quando autores distintos possuem o mesmo nome em seus artigos. Esta situação pode levar a tratar todos os artigos como se fossem do mesmo autor. Vários métodos já foram propostos para tentar resolver este problema. No entanto, nenhum deles resolve completamente este problema e normalmente eles são aplicados sobre todo o conjunto de artigos representados pelos seus meta-dados. O objetivo deste trabalho é estudar técnicas de remoção de ambiguidade de nomes de autores de artigos científicos e propor um método capaz de remover a ambiguidade de nomes de autores de modo incremental, ou seja, remover a ambiguidade apenas dos novos artigos, cujos meta-dados estão sendo inseridos em uma biblioteca digital, bem como a sua avaliação por meio de análises comparativas com outros métodos.

Palavras-chave: Ambiguidade de Nome, Bibliotecas Digitais, Citações Bibliográficas, Split Citation, Mixed Citatio

Sumário

Lista de Figuras

Lista de Tabelas

1 Introdução

O problema de ambiguidade de nomes pode ser observado em diversos contextos. Esse problema afeta principalmente os sistemas computacionais que na maioria das vezes não conseguem identificar e corrigi-lós. Alguns exemplos de ambiguidade de nomes são encontrados em nomes de lugares como a cidade Ouro Preto e o bairro Ouro Preto localizado em Belo Horizonte, em nomes de pessoas como os ex-presidentes dos EUA George W. Bush pai e filho ou ainda em veículos de publicação onde , por exemplo, SBBD e Simpósio Brasileiro de Banco de Dados se remetem ao mesmo simpósio.

Este trabalho, trata do problema de desambiguação de nomes de pessoas, mais especificamente, do nome de pessoas em citações bibliográficas de artigos científicos, que comumente são armazenados em bibliotecas digitais.

Uma citação bibliográfica em uma biblioteca digital é representada pelos seus meta-dados que são constituídos por nomes dos autores, título do trabalho, título do veículo depublicação e o ano de publicação. Neste contexto, este problema pode afetar o desempenho de recuperação de documentos, de máquinas de pesquisa, da integração de banco de dados e pode causar a atribuição indevida de crédito a um autor.

Nas subseções a seguir será introduzido o conceito e a importância das bibliotecas digitais, os problemas relativos a ambiguidade de nomes neste contexto, bem como o escopo a ser abordado neste trabalho.

Bibliotecas Digitais

Bibliotecas digitais (DLs) são sistemas de informação complexos, que são projetados para um público específico, possuem um conjunto grande de objetos digitais e seus meta-dados, várias estruturas organizacionais e fornecem diversos serviços para manter e acessar esses objetos digitais Gonçalves et al. [2004]. As bibliotecas digitais são importantes sistemas de gestão de informações na Internet. As DLs, ou *Digital Librarys*, são fontes massivas de informação para diversos segmentos. Atualmente, são de grande relevância em diversas áreas, principalmente a acadêmica e com sua ajuda vem atingindo um alto crescimento.

As DLs além de um importante veículo de referência é uma grande rede de colaboração (co-autoria), e com sua ampliação está se tornando mais complexa. Entretanto, para qualidade destas características é necessário que as bases de dados possuam informações que condizem com a realidade, ou seja, que não haja ambiguidade nas informações da DL.

As DLs são alimentadas por diversas fontes de dados e cada uma destas fontes podem adotar padrões diferentes de representação dos dados. Um dos pontos mais críticos no contexto de padrões de representação é a ambiguidade de nomes de autores em citações bibliográficas.

Ambiguidade de Nomes de Autores em Citações Bibliográficas

A ambiguidade de nomes de autores em citações bibliográficas é um problema que está presente nas mais diversas bibliotecas digitais. Segundo Lee et al. [2005] podemos dividí-lo em dois sub-problemas: os problemas *split citation*(SC) e *mixed citation*(MC).

O primeiro ocorre quando há uma variação no modo como o nome de um autor é representado. Nesse caso, publicações de um mesmo autor podem estar dividas como

se pertencessem a pessoas distintas. Isso acontece principalmente devido a um erro ortográfico, abreviatura de um nome ou sobrenome ou ainda mudança na ordem de nome e sobrenomes.

Segue um exemplo na Figura 1 para ilustrar um caso de SC retirado de uma pesquisa na BDBComp[1] para a autora "Aleksandra Silva". Neste exemplo, foram recuperadas três diferentes citações "Aleksandra Silva", "Aleksandra do Socorro Silva" e "Aleksandra do Socorro da Silva" em que todas as citaçoes se referem a uma mesma autora.

Figura 1: Exemplo de *split citation* retirado da BDBComp.

O problema *mixed citation* ocorre quando diferentes autores compartilham o mesmo nome ou a mesma variação de nome e suas publicações aparecem como se pertencessem a um mesmo autor. Na Figura 2 é apresentado um exemplo de MC retirado de uma pesquisa na DBLP para o autor "Mohammed Zaki". Analisando somente o nome exato utilizado para pesquisa "Mohammed Zaki", notamos que essa classe referência dois autores diferentes.

[1] http://www.lbd.dcc.ufmg.br/bdbcomp/

Um deles indica um "Mohammed Zaki" que pertence ao corpo docente da Universidade de Al-Zhar, na cidade de Nasr, Cairo, Egito. Enquanto o outro aponta para o "Mohammed Zaki" que pertence ao Departamento de Ciência da Computação do Instituto Politécnico Rensselaer dos Estados Unidos.

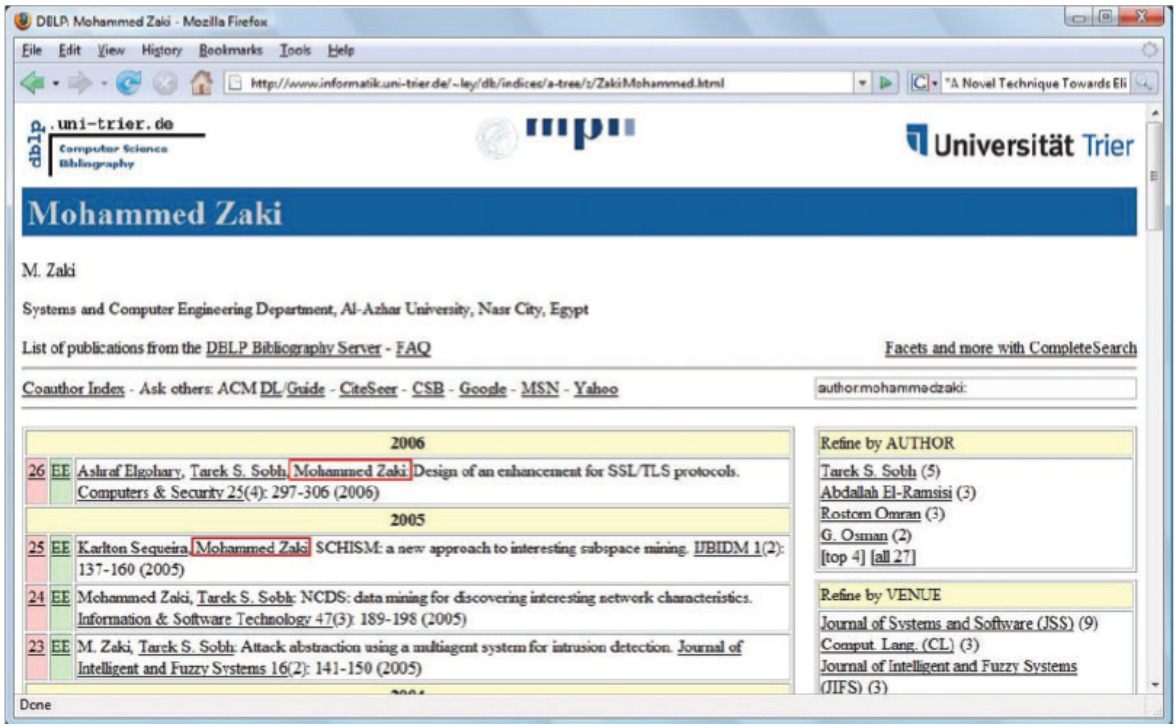

Figura 2: Exemplo de *mixed citation* retirado do DBLP

Escopo

Agora que já foi apresentado o conceito de bibliotecas digitais e os sub-problemas relacionados a ambiguidade de nomes em suas citações bibliográficas, será delimitado o escopo do trabalho. Na literatura existem diversos trabalhos que tratam do problema de remoção de ambiguidade de uma coleção de citações bibliográficas contidas em uma DL, alguns deles serão expostos e explicados na seção Trabalhos Relacionados. Entretanto, pouco se fala em uma abordagem incremental deste problema. Essa abordagem incremental consistiria em desambiguar apenas os registros das novas citações que estão sendo inseridas em uma biblioteca digital.

A propsota desta abordagem é simples de se compreender. Por exemplo, suponha que a partir de uma coleção desambiguada de registros de citações de uma DL seja necessário incluir uma publicação de um determinado autor. O método a ser proposto deverá ser capaz de identificar a publicação a ser inserida na DL já pertence a um autor que já possui artigos na biblioteca digital ou não. Caso pertença a biblioteca, a publicação a ser inserida deverá ser atribuída a esse autor. Caso contrário, deve-se identificar o trabalho como sendo de um novo autor.

2 Justificativa

O problema de ambiguidade de nomes afeta diretamente diversas áreas como os sistemas de recuperação de informação, o estabelecimento de rede de colaboração (co-autoria) e agências de fomento.

Em um sistema de recuperação de informação, se torna mais simples notar o problema de ambiguidade de nomes. Nesses sistemas, a importância de se manter o nomes desambiguados é imensa. Ao se fazer uma pesquisa sobre as publicações de um determinado autor *a1*, caso haja alguma ambiguidade neste nome os resultados da pesquisa poderiam conter publicações referentes a outros autores ou mesmo não incluir todos os trabalhos deste autor.

No caso dos outros dois exemplos a percepção deste problema não é tão simples. No estabelecimento de redes de colaboração havendo uma atribução errada de um autor a uma publicação irá gerar uma ligação que não deveria existir na rede de co-autoria. Já com relação as agências de fomento é levado em consideração o mesmo problema citado para as redes de co-autoria. Se uma publicação for atribuida erroneamente a outro autor haverá uma redução no número de publicações do autor que realmente a publicou. Como as agências de fomento são responsáveis pelo incentivo e patrocínio de publicações (exemplo CNPQ) elas levam em consideração a quantidade de trabalhos relacionados aquele autor para aprovar ou não o investimento no projeto.

3 Objetivos

3.1 Objetivo geral

- Manter uma coleção de citações bibliográficas livre de ambiguidade.

3.2 Objetivos específicos

- Fazer uma revisão bibliográfica sobre métodos de remoção de ambiguidade.
- Análise de métodos existentes, visando descobrir seus pontos fracos e fortes.
- Propor um método incremental de remoção de ambiguidade.
- Avaliar o método proposto comparando-o a métodos representativos existentes na literatura.

4 Trabalhos Relacionados

Na literatura podem ser encontrados diversos trabalhos que tratam do problema de desambiguação de nomes. Os métodos propostos por estes trabalhos possuem formas diferentes para lidar com esse problema. Existem métodos baseados em técnicas de aprendizado supervisionado, semi-supervisionado ou não supervisionados.

Nesta seção, serão apresentados trabalhos com abordagens diferentes para o problema de desambiguação.

Han et al. [2004] propoêm duas abordagens baseadas em técnicas de aprendizagem supervisionada que usam nome de co-autores, título e veículo de publicação como atributos a serem usados na remoção de ambiguidade. O primeiro baseia-se no modelo *naïve Bayes*, que é um modelo estatístico gerativo frequentemente utilizado para tarefa de classificação e visa capturar todos os padrões de nomes de autores nas citações. A segunda abordagem se baseia em *Support Vector Machines* (Máquinas de Vetor de Suporte) que também são bastantes utilizados em classificação. As duas abordagens tem uma pequena diferença, a baseada em naïve Bayes necessita somente de exemplos positivos em quanto o SVM (*Support Vector Machines*) necessita de ambos, ou seja, exemplos positivos e negativos para possibilitar a classificaçao das citações.

Han et al. [2005] propôs um método de aprendizagem não supervisionada que utiliza a técnica que agrupamento K-way Spectral Clustering, que é baseada em grafo e tem sido aplicada com sucesso na mineração de dados e na análise de clusteres. O método K-way Spectral Clustering consiste em calcular autovalores e autovetores de uma matriz Laplaciana (ou valores singulares e vetores singulares de certos dados da matriz) relacionada com o dado grafo, e construir clusteres com base em informações espectrais. A abordagem utilizada se baseia em 3 atributos para a desambiguação de nomes que são nomes dos co-autores, título do artigo e veículo de publicação.

Huange et al. [2006] apresentou um framework para o problema de desambiguação de nomes em que primeiro utiliza-se um método de blocagem que cria blocos de citações de de autores com nomes semelhantes e então emprega em cada bloco um método de clusterização baseado em densidade para realizar a desambiguação.

Ferreira et al. [2010] propõem um método híbrido de desambiguação de nomes dividido em duas fases. Na primeira, são obtidos de forma automática os exemplos para compor o conjunto de treino que será utilizado na segunda fase. Na segunda, uma função de desambiguação é inferida usando-se os exemplos. A fase inicial elimina a necessidade de qualquer rotulagem manual para formar o conjunto de treino,pois os registros de citações são organizadas utilizando um método de clusterização que separa os registros de um mesmo autor em um mesmo cluster. Após a formação dos clusteres é montado a base de treinamento, que é constituída de exemplos retirados de cada clusters. Essa é chamada de fase não supervisionada. Na segunda fase são utilizados os exemplos de treino para criar uma função de desambiguação de autores empregando características frequentes presentes nas citações (nome de co-autores, título e local da publicação). A função criada é usada para prever o autor correto das demais publicações e assim remover a ambiguidade. Essa é a fase supervisionada. Por este fato é um método hibrido pois utiliza técnicas supervisionadas e não supervisinadas.

Xiaoming et al. [2011] surgerem um método de desambiguação de autores baseado em grafos. A abordagem apresentada por esses autores é construída por meio de grafos direcionais. Somente um único atributo é utilizado para a remoção da ambiguidade que

é o de co-autoria. O Método consiste em dividir as publicações a serem desambiguadas de forma que cada cluster contenha somente as publicações de um mesmo autor. Inicialmente os clusters criados contém registros de autores com nomes ambíguos. Para remove-lá é utilizado um framework de desambiguação de nomes chamado GHOST (abreviatura para *GrapHical framewOrk for name diSambiguaTion*). Este framework cria um grafo de co-autoria a partir dos nomes ambiguos. Baseado nisso, é utilizado um algoritmo de caminhamento para definir se dois nomes se referem ao mesmo autor. Se dois nomes se referem ao mesmo autor ele ficam no mesmo grupo, caso contrário, são colocados em grupos distintos.

5 Metodo Proposto

Na seção Introdução, foi apresentado o problema e o escopo do trabalho. Nesta será apresentado o método proposto detalhadamente.

Como o objetivo é manter uma coleção de registros de citações bibliográficas livre de ambiguidade o desafio é , dado um conjunto de citações bibliográficas a serem inseridos nesta coleção, identificar de forma única o autor a que se refere cada nome de cada citação bibliográfica. Para isso, um dos grandes desafios é identificar se os autores a que se referem os nomes nas citações bibliográficas já possuem trabalhos cadastrados na biblioteca digital. Pois, se já possuem, essas citações devem ser atribuídas a esses autores já existentes e se não possuirem essas citações devem ser atribuídas a um novo autor.

Indicar se uma citação a ser inserida em uma DL se refere a autores com trabalhos já cadastrados nessa DL, requer, inicialmente, uma comparação dos nomes desta citação com os nomes dos autores de trabalhos da DL e o modo como será feita essa comparação influência a eficiência e a eficácia do método de desambiguação.

Imagine por exemplo, que para reconhecer um determinado autor *a1*, tivessemos que comparar *a1* com cada autor presente na DL para garatir que é um novo autor, o custo computacional seria muito alto. Com o intuito de minimizar comparações excessivas iremos separar o método em duas etapas: a primeira será a seleção de citações da DL de possivéis autores e a segunda comparação entre os registros dessas citações com os das citações a serem inseridas. Abaixo a Figura 3 ilustra a idéia de funcionamento do método.

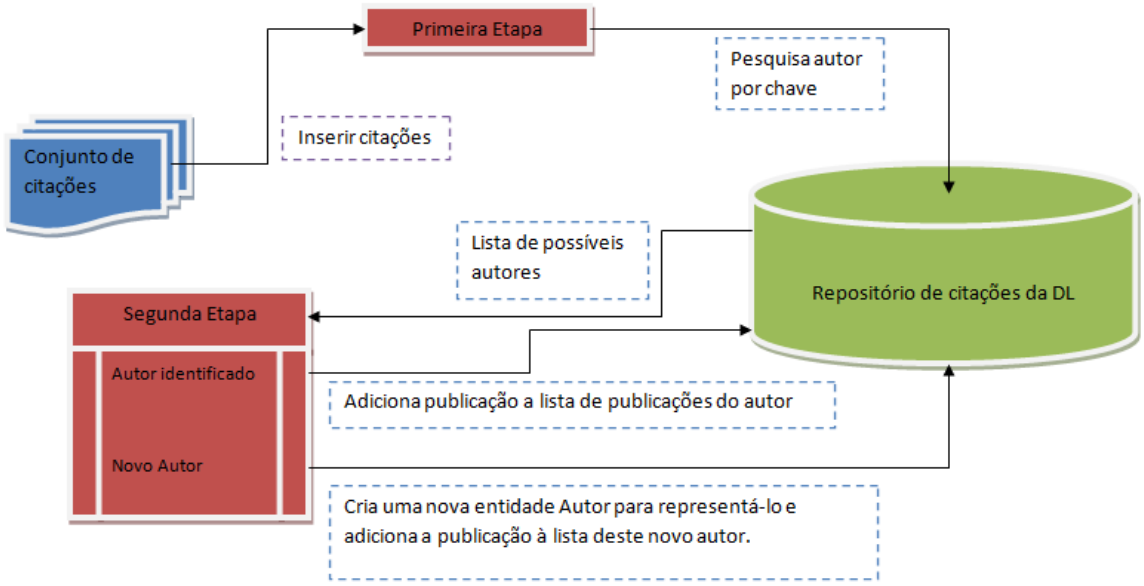

Figura 3: Fluxograma Funcionamento do método

Na etapa inicial serão selecionados registros de citações cujos nomes de autores são similares aos nomes de autores dos registros de citações a serem inseridos. Na próxima etapa serão avaliadas técnicas supervisionadas e não supervisionadas para identificar se os autores da citação a ser inserida já possuem publicações cadastradas na DL.

5.1 Primeira Etapa

Na etapa inicial dado uma citação c a ser inserida com n autores sendo eles $a_1, a_2, \cdots, a_n$, para cada nome de c deve-se recuperar uma lista de possiveis autores contidos na DL. Como este primeiro refinamento deve ser computacionalmente barato será usado uma tabela hash para armazenar todos os autores contidos na DL que usará como chave a inicial do nome do autor e o último sobrenome. Cada registro nesta tabela será uma lista da entidade *Autor*. Essa entidade é uma tripla contendo o nome representativo do autor , uma lista de publicações deste autor e o valor da similaridade que será definido somente na segunda etapa. Cada elemento da lista de publicações tem um par publicação e a posição do autor na lista de autores. Um exemplo de registro da tabela é a chave *A. Ferreira* que irá armazenar uma lista de entidades *Autor* que possuam o nome representativo iniciado com a letra *A* e tenham *Ferreira* como seu último sobrenome, ou seja, a lista de autores por exemplo pode conter os seguintes nomes *Anderson Almeida Ferreira, André Luiz Ferreira ou Aida Silva Ferreira* desde que estes autores possuam alguma publicação na DL.

5.2 Segunda Etapa

Na segunda etapa cada nome $(a_1, \cdots a_n)$ da citação c terá uma lista $(l_1, \cdots l_n)$ de possivéis autores associados a ele. Como nesta lista podem haver nomes de pessoas totalmente diferentes ao nome do autor da citação c será necessário utilizar um método de similaridade entre strings para fazer um refinamento. Na seção 6 serão apresentados diferentes métricas de similaridade strings. As funções de similaridade de strings retornam um valor entre 0 e 1 de acordo com o grau de similaridade entre duas strings, neste caso o nome de dois autores. Baseado neste valor será realizada uma filtragem na lista de possiveis autores de cada nome pertencente a c, onde um valor α será tomado como base. Isso quer dizer que só irá permanecer na lista os nomes de autores o qual a similaridade entre strings retornar um valor maior ou igual ao valor base α, sendo ela aplicada entre o nome da lista e um autor de c. Este valor da similiaridade será armazenado no atributo valor da similaridade (SM) , como dito na primeira etapa.

Com a lista de possiveis autores refinada, serão utilizados outros atributos para ajudar a identificar se alguns dos autores pertencentes a essa lista é realmente um autor da citação c. Os atributos que serão utilizados são o nome do autor, nome dos co-autores, titulo das publicações e veículo de publicação. Agora é necessário selecionar uma citação de cada possível autor em sua lista de publicações. Para isso será usado um critério simples, primeiro a posição do autor nesta publicação deve ser 1, ou seja , ele deve ser o autor da publicação e segundo a citação a ser escolhida deve o maior número de co-autores em comum com a citação c entre as citações da lista.

Como foi escolhida uma citacão de cada possível autor, podemos utilizar os outros atributos para definir se algum destes autores é o autor da citação c. Agora serão usados os co-autores para definir uma similaridade entre os autores. Serão comparados a lista de de co-autores de c com a lista de co-autores da citação escolhida. Essa similaridade de co-autores (SCA) será definida assim:

$$SCA = \frac{N_c}{N_t}$$

onde, N_c é número de co-autores de c que são iguais aos co-autores da citação

escolhida e N_t é o número total de co-autores de c.

Para ajudar a identificar o autor poderão ser usados os atributos título e veículo de publicação. Sobre esses atributos serão aplicadas métricas de similaridade de strings assim como às aplicadas nos nomes dos autores. No entanto, serão aplicadas métricas mais específicas visto que esses atributos possuem maior número de strings. Esses valores serão chamados de similaridade de título (ST) e similaridade de veículo de publicação (SVP).

Para identificar se o autor da citação c é um autor presente na DL utilizaremos uma média aritmética ponderada utilizando a similaridade de todos os atributos, ou seja, a similaridade de nomes de autores (SM), co-autores (SCA), título (ST) e de veículo de publicação (SVP). Cada um destes atributos terão um peso P decrescente começando por P_{SM}, P_{SCA}, P_{ST} até P_{SVP}. Sendo assim os atributos mais relevantes terão um peso maior. A função de identificação de autores é mostrada abaixo :

$$F = \frac{SM * P_{SM} + SCA * P_{SCA} + ST * P_{ST} + SVP * P_{SVP}}{P_{SM} + P_{SCA} + P_{ST} + P_{SVP}}$$

Para que a identificação do autor seja confirmada será necessário que o valor de F seja maior que um valor δ pré-definido. O possível autor que tiver o maior resultado de F será identificado como um autor da citação c, desde que $F \geq \delta$. Uma vez que nome da citação c tenha sido identifcado como um autor da DL, a publicação é inserida na lista deste publicações do autor.

Caso os possíveis autores da citação c obtenham um valor de $F \leq \delta$ será considerado como um novo autor. Considerado um novo autor será criada uma entidade $Autor$ para representá-lo. O nome representativo da entidade será o mesmo nome contido em c e será criada uma lista de publicações contendo apenas uma publicação, ou seja, a publicação a qual a citação c se refere.

Os valores base citados acima α e δ serão definidos na implementação visto que eles influênciam diretamente no resultado do método, assim também como os pesos P_{SM}, P_{SCA}, P_{ST} e P_{SVP}. Testes serão executados afim de identificar valores aos quais o método produz um melhor resultado.

6 Métricas de Similaridade de Strings

Nesta seção serão apresentadas as principais métricas de similaridade entre strings utilizadas. No entanto, somente algumas serão utilizadas na construção do método de remoção ambiguidade.

6.1 Similaridade de Distância de Levenshtein

Distância de Levenshtein [1966] foi nomeada em homenagem ao cientista russo Vladimir Levenshtein, que desenvolveu o algoritmo em 1965. Também sendo conhecida pelo nome de *edit distance*. O cálculo de similaridade utilizando distância de edição baseia-se no número de mínimo de transformações (inserção,exclusão e substituição) necessário para transformar uma string S em uma string T. Quanto menor a distância de edição mais similares são as strings. Existem diversas variações deste método: alguns atribuem pesos diferentes para cada operação e outros utilizam métricas diferentes da distância de Levenshtein como a distância de Hamming.

6.2 Similaridade por Comparação de Fragmentos

Segundo French et al. [2000] a similaridade por comparação de fragmentos é uma função de casamento de padrão que, através do algoritmo de distância de edição, avalia um a um cada fragmento de duas cadeias de caracteres que representam nomes.
Os parâmetros necessários são duas cadeias de caracteres normalizadas, S e T, e o limite L utilizado para a distância de edição. O resultado retornado é verdadeiro se S e T são compatíveis e eventualmente podem representar a mesma entidade do mundo real, e falso caso contrário. O limite L pode ser um número inteiro e, neste caso é utilizado diretamente pelo algoritmo, ou um número entre 0 e 1, em cujo caso o número máximo de erros permitidos será igual ao tamanho do menor fragmento comparado multiplicado por L.

6.3 Coeficiente de Jaccard

Segundo Cohen et al. [2003] a métrica de similaridade de Jaccard é baseada em *tokens*. Jaccard considera as strings a serem comparadas como palavras separadas por espaços para definir a similaridade. Sua equação é dada a seguir:

$$jaccard = \frac{S \cap T}{S \cup T}$$

onde, S e T são conjuntos de *tokens* derivados da quebra em palavras das duas strings de entrada s e t. Esta métrica retorna o quociente do número de *tokens* que representam a intersecção dos conjuntos S e T pelo número de tokens que representam a união desses conjuntos. Observe o exemplo a seguir:

Jaccard ("‘Universidade Federal Ouro Preto"’, "‘Universidade Ouro Preto"’) = 0.750

Neste exemplo há três *tokens* em comun (Universidade, Ouro e Preto), e há 4 *tokens* no total desta forma basta realizar a divisão $\frac{3}{4}$ para obter o resultado final como no exemplo.

6.4 Medida do Cosseno

Segundo Salton et al. [1975] uma string é nada mais que um conjunto de palavras. Existe um conjunto finito de palavras que determinam o vocabulário. Logo em cada elemento da string podem aparecer quaisquer palavras do vocabulário. Sendo assim é possível criar uma representação vetorial em um espaço Euclidiano multidimensional para cada elemento do conjunto. Cada eixo deste espaço corresponde a uma palavra. A coordenada de um elemento e na direção correspondente a uma palavra p é determinada por duas medidas:

- TF (*term frenquecy*), que corresponde ao número de vezes que uma palavra p aparece em um elemento e.
- IDF (*inverse document frenquecy*), que corresponde ao peso da palavra de acordo com o inverso da frequência no elemento

A medida IDF deve ser utilizada, pois os eixos no espaço vetorial não são igualmente importantes. Assim, palavras de maior freqüência no conjunto de elementos devem ser penalizadas. Sendo E o conjunto de todos os elementos de mesmo rótulo pertencentes a um repositório R e Ep o conjunto dos elementos de E que contêm determinada palavra p, uma forma comum para o cálculo do IDF de p é:

$$w_p = \log\left(1 + \frac{|E|}{|E_p|}\right)$$

Elementos longos tendem a ser favorecidos por conterem um maior número de palavras diferentes. Com isso, é necessário realizar uma normalização em função do tamanho do elemento, que é determinada pela fórmula:

$$w_e = \sqrt{\sum w_{e,p}^2}$$

onde $w_{e,p}$ corresponde ao peso das palavras em relação ao elemento e é calculado através da regra $TF * IDF$

$$w_{e,p} = (1 + \log(f_{e,p})) * w_p$$

sendo $f_{e,p}$ o número de ocorrências da palavra p no elemento e.

A similaridade entre dois elementos é calculada através da medida do cosseno entre suas representações vetoriais. Quanto maior o cosseno, maior a similaridade. Para isto, utilizamos a fórmula:

$$\cos(e_1, e_2) = \frac{1}{w_{e1} * w_{e2}} * \sum_{p \in (e_1 \cap e_2)} (w_{e_1,p} * w_{e_2,p})$$

7 Avaliação

Nesta seção apresentaremos a forma como será feita a avaliação do método proposto na disciplina Monografia II. Nas subseções abaixo serão mostrados e explicados as métricas de avaliação, os métodos existentes na literatura que serviram de parâmetro para a comparação e as bases de dados a serem utilizados nos testes para a avaliação.

7.1 Métricas de Avaliação

Para avaliar a eficácia do método proposto para a remoção de ambiguidade de nomes será utilizada a métrica k. A seguir, é descrita essa métrica.

Métrica k

A métrica K Lapidot [2002] determina o equilíbrio entre duas métricas específicas de agrupamento: pureza média do cluster (PMC) e pureza média do autor (PMA). PMC avalia a pureza da clusteres gerados com relação a clusteres de referência desambiguados manualmente, ou seja, verifica se os clusters gerados incluem apenas os registros pertencentes aos clusteres de referência (são puros). Assim, se os clusteres gerados são puros, o resultado desta métrica será 1. A fórmula para calcular PMC é:

$$PMC = \frac{1}{N} \sum_{i=1}^{q} \sum_{j=1}^{R} \frac{n_{ij}^2}{n_i}$$

onde R é o número de clusteres gerados manualmente (clusteres de referência), N é o número total de registros na coleção da DL, q é o número de clusters automaticamente gerados pelo método, n_{ij} é o número total de elementos do cluster i gerado automaticamente pertencente ao cluster j gerado manualmente, e n_i é o número total de itens do cluster i gerado automaticamente.

A PMA avalia a fragmentação dos clusteres gerados automaticamente em relação aos clusteres de referência. Se houver uma baixa proporção de clusters fragmentados, o resultado está mais próximo 1. Seus valores variam entre 0 e 1. A fórmula para calcular PMA é

$$PMA = \frac{1}{N} \sum_{j=1}^{R} \sum_{i=1}^{q} \frac{n_{ij}^2}{n_j}$$

onde R é o número de clusters gerados manualmente (clusteres referência), N é o número total de registros no grupo ambíguo, q é o número de clusters automaticamente gerados pelo método, n_{ij} é o número total de elementos do cluster i gerado automaticamente pertencente ao cluster j gerados manualmente, e n_j é o número total de itens de o cluster gerado manualmente j. A métrica K consiste na média geométrica entre a duas métricas. Ela combina a avaliação de ambos a pureza e a fragmentação dos clusteres gerados por cada método.

A fórmula para calcular K é:

$$k = \sqrt{PMC * PMA}$$

7.2 Baseline

Como o problema de remoção de ambiguidade de nomes já vem sido estudado há alguns anos, existem diversos trabalho relacionados na literatura. Será utilizado um trabalho como base de comparação para o novo método a ser desenvolvido. O método a ser utilizada para a comparação é o método HHC de Cota et al. [2007] que significa Heuristica baseada em método hierárquico de clusterização.

7.2.1 Método HHC

Este método trata ao mesmo tempo os problemas *split citation* (SC) e o *mixed citation* (MC). Utilizando uma combinação de funções de similaridade com algumas heurísticas que usam evidências presentes nos registros de citação dos autores que se desejam desambiguar. Este método agrupa registros de citações de forma hirarquica levando em consideração as similaridades entre os atributos contidos nos registros de citação.

HHC inicialmente separa a lista de registros de citações em duas listas, uma com os autores que possuem nomes curtos (apenas a inicial do primeiro nome e o último nome) e a segunda com os demais (nomes longos). Ele começa processando primeiro a lista com nomes longos, que são nomes com mais informação, e depois processa a que possui nomes curtos.

Neste processamento, o HHC agrupa inicialmente os registros que possuem o nome do autor similar e pelo menos um nome de co-autor similar. Depois, ele usa os títulos das citações e dos veículos de publicação para agrupar citações de um mesmo autor, mas que não possuem co-autores em comum.

8 Trabalhos Futuros

Baseado no método proposto neste trabalho efetuar alterações visando adaptá-lo para funcionar como um método não supervisionado. Seguindo o mesmo escopo do método citado na seção 5 com a única diferença de que ao fim da primeira etapa seria obtido uma base de treinamento para inferir se nome da citação c a ser inserida é um autor da biblioteca digital. A primeira etapa permaneceria sem nenhuma modificação e a segunda seria modificada. Na segunda etapa, dada uma base de treinamento de citações treino c_t gerar uma função para inferir se o nome de c já possui publicações cadastradas na DL e caso verdadeiro definir qual seria esse autor.

9 Cronograma de atividades

Na Tabela 1, estão representadas todas as etapas de desenvolvimento da monografia contemplando as disciplinas monografia I e II .

Atividades	**Abr**	**Mai**	**Jun**	**Jul**	**Ago**	**Set**	**Out**	**Nov**	**Dez**
Revisão Bibliográfica	X								
Estudo dos métodos	X	X							
Projetar um novo método		X	X						
Implementar o método				X	X	X	X		
Testar o método							X	X	
Análise Comparativa							X	X	
Redigir a Monografia							X	X	X
Apresentação do Trabalho									X

Tabela 1: Cronograma de Atividades.

Referências

W. W. Cohen, P. Ravikumar, and S. E. Fienberg. A comparison of string distance metrics for name-matching tasks. pages 73–78, 2003.

R. G. Cota, M. A. Gonçalves, and A. H. F. Laender. A heuristic-based hierarchical clustering method for author name disambiguation in digital libraries. 2007.

A. A. Ferreira, A. Veloso, M. A. Gonçalves, and A. H. Laender. Effective self-training author name disambiguation in scholarly digital libraries. *Joint Conference on Digital Library*, pages 39–48, 2010.

J. C. French, A. L. Powell, and E. Schulman. Using clustering strategies for creating authority files. *Journal of the American Society for Information Science and Technology*, 51:774–786, 2000.

M. A. Gonçalves, E. A. Fox, L. T. Watson, and N. A. Kipp. Streams, structures, spaces, scenarios, societies (5s): A formal model for digital libraries. *ACM Trans. Inf. Syst.*, 22(2):270–312, 2004.

H. Han, C. L. Giles, H. Zha, C. Li, and K. Tsioutsiouliklis. Two supervised learning approaches for name disambiguation in author citations. *Joint Conference on Digital Library*, pages 296–305, 2004.

H. Han, H. Zha, and C. L. Giles. Name disambiguation in author citations using a k-way spectral clustering method. *Joint Conference on Digital Library*, pages 334–343, 2005.

Huange, J., Ertekin, S., Giles, and C.L. Efficient name disambiguation for large-scale databases. *In Proceedings of 10th European Conference on Principles and Praticle of Knowledge Discovery in Databases*, (4213):536–544, 2006.

I. Lapidot. Self-organizing-maps with bic for speaker clustering. *IDIAP Research Report 02-60, IDIAP Research Institute*, 2002.

D. Lee, B.-W. On, J. Kang, and S. Park. Effective and scalable solutions for mixed and split citation problems in digital libraries. *Proceedings of the 2nd international workshop on Information quality in information systems*, pages 69–76, 2005. doi: http://doi.acm.org/10.1145/1077501.1077514. URL `http://doi.acm.org/10.1145/1077501.1077514`.

I. V. Levenshtein. Binary codes capable of correcting deletions, insertions and reversals. pages 707–710, 1966.

G. Salton, A. Wong, and C. S. Yang. A vector space model for automatic indexing. *Commun. ACM*, 18:613–620, 1975. ISSN 0001-0782.

F. Xiaoming, W. Jianyong, P. Xu, Z. Lizhu, and L. Bing. On graph-based name disambiguation. *J. Data and Information Quality*, 2:10:1–10:23, 2011.