# Indexação automática de documentos por meio de processamento de linguagem natural

**Fabio Montefuscolo, Rafael G. Câmara, Fabrício J. Barth, Orlando Rodrigues Jr.**

[1]Faculdade de Ciências Exatas e Tecnologia
Centro Universitário Senac – São Paulo, SP – Brazil

{fabio.montefuscolo,rafael.gcamara}@gmail

{fabricio.barth,orlando.rodrigues}@sp.senac.br

***Abstract.*** *This article describes the development of an application capable of analyzing a scientific text and synthesize it in keywords that represent the concept forwarded by text. To this, were constructed and used mechanisms based on natural language processing. These mechanisms are capable of filtering, inflect words and identify them as potential keywords. The intention is that implementation can be used to help professionals engaged in this activity in large centers of information and documentation.*

***Resumo.*** *Este artigo descreve o desenvolvimento de uma aplicação capaz de analisar um texto científico e sintetizá-lo em palavras-chave que representam o conceito transmitido pelo texto. Para isto, foram construídos e utilizados mecanismos com base em processamento de linguagem natural. Estes mecanismos são capazes de filtrar, inflexionar palavras e identificá-las como possíveis palavras-chave. A intenção é que a aplicação possa ser utilizada no auxílio de profissionais que exercem esta atividade em grandes centros de informação e documentação.*

## 1. Introdução

Em bibliotecas físicas e digitais, a organização da informação é baseada em descritores, que são palavras-chave cujos conceitos podem representar de forma resumida o documento onde estão contidas. Os descritores de um texto podem ser escolhidos livremente, de acordo com que o autor ou profissional de indexação imagina ser mais adequada ao documento. Outra forma é recorrer a um conjunto de descritores pré-definidos presentes em um vocabulário controlado, seguindo um conjunto de regras [LANCASTER 2004].

O vocabulário controlado contém um conjunto de descritores que são agrupados de forma hierárquica. Todo descritor é armazenado de forma que consigo estejam presentes a descrição de seu conceito, termos sinônimos e comentários deixados para ajudar o indexador em sua tarefa. Com uso do vocabulário controlado é possível organizar de forma eficiente uma coleção de documentos, pois além do conceito do descritor que sintetiza o assunto do documento, a hierarquia do vocabulário classifica este documento em tópicos mais genéricos [LANCASTER 2004].

O processo de indexação é feito por um profissional da informação, comumente chamado de indexador, que seguindo um conjunto de regras e critérios constrói

representações de documentos para que sejam incluídos em uma base de dados. Este processo é uma atividade complexa, lenta e custosa, geralmente realizada por poucas pessoas [LANCASTER 2004].

O objetivo deste trabalho reside na construção de uma aplicação capaz de auxiliar e melhorar a qualidade do trabalho desse profissional no reconhecimento de descritores para artigos científicos na área da saúde. A aplicação construída deverá apresentar palavras candidatas a descritores para um artigo, o que pode aumentar a quantidade de artigos indexados em um determinado intervalo de tempo, fazendo com que o indexador complete seu trabalho de forma mais eficiente e menos custosa.

Este artigo está estruturado da seguinte maneira: na seção 2 é descrito o método adotado; na seção 3 é descrita a implementação do extrator de descritores; na seção 4 são apresentados os resultados obtidos e, por fim, a seção 5 contém as considerações finais deste trabalho.

## 2. Método

Um vocabulário controlado é uma ferramenta que agrupa conceitos e termos de forma hierárquica que representam áreas temáticas. Os termos existentes no vocabulário controlado são usados, combinados ou não, para descrever de forma sucinta um documento. O vocabulário controlado utilizado neste trabalho foi o MeSH (Medical Subjects Heading) [NLM 2008], distribuído pela NLM (National Library of Medicine).

O MeSH pode ser encontrado em arquivos de vários padrões, mas para este trabalho foi usado o arquivo XML (Extensible Markup Language). Este trabalho analisa no MeSH três tipos de elementos para realizar o processo de indexação: descritores, conceito e termos. Os descritores são representados por termos, cuja função é expor de forma objetiva os conceitos existentes no vocabulário controlado. Descritores são representados pelo elemento descriptor no MeSH, que aninha uma lista de conceitos[NLM 2006].

Os conceitos, representados no arquivo pelo elemento *concept*, guardam um termo e uma definição conceitual para este termo. Um conceito é preferido quando seu uso é indicado para o descritor ao qual está associado. O conceito preferido guarda um termo idêntico ao termo do descritor a que pertence. Já os conceitos não preferidos têm termos diferentes do termo do descritor, mas são considerados sinônimos do conceito preferido [NLM 2006].

Para e extração de descritores de documentos eletrônicos é necessário que sejam encontradas palavras cujo conceito, além de descrever o conteúdo de texto também estejam presentes no vocabulário controlado utilizado para a indexação.

É necessário que haja um tratamento dos textos de entrada e do vocabulário controlado para que as buscas por descritores nos textos aconteçam da forma mais eficaz possível. Este tratamento implica em reduzir as palavras aos seus radicais, através de um algoritmo de *stemming* [Porter 1997], e retirar do texto palavras que não tem função conceitual, através de uma lista de *stop words* [NLTK 2008].

O vocabulário MeSH foi tratado da mesma forma que os textos dos documentos. Foram extraídos do vocabulário controlado todos os termos dos descritores e sinônimos. Os termos retirados dos descritores, que são os termos preferidos para descrever o

conceito, são armazenados em um dicionário de descritores. Os outros termos, que são sinônimos ao termo preferido são armazenados em um dicionário de sinônimos. Todo termo presente no dicionário de sinônimos tem uma referência para um termo do dicionário de descritores. Um descritor pode ter vários sinônimos, mas um sinônimo está relacionado somente a um descritor.

O descritores e sinônimos presentes no vocabulário controlado foram associados a identificados e armazenados em estruturas do tipo *hashmap*, de forma que fosse possível mapear a relação hierarquica entre os termos do vocabulário controlado.

## 3. Desenvolvimento

No desenvolvimento deste trabalho, o vocabulário controlado usado foi o MeSH do ano 2008, que conta com 24.767 termos em inglês e mais de 97.000 formas sinônimas para os termos autorizados. A lista de *stop-words* utilizada é a lista proposta pelo projeto NLTK [NLTK 2008], composta por 571 termos. Os artigos científicos utilizados são da área da saúde e disponibilizados na biblioteca eletrônica SciELO [SciELO 2008].

O desenvolvimento foi baseado no modelo de prototipagem, onde novas funcionalidades foram desenvolvidas a cada ciclo de desenvolvimento, baseando-se nos seguintes marcos: construção de um módulo de leitura para o XML do MeSH ; desenvolvimento do processo de normalização dos termos; construção das estruturas para alocar o vocabulário controlado; construção do módulo de leitura por janelamento; construção do módulo de leitura por bigramas [Cavnar and Trenkle 1994]; desenvolvimento de um módulo de seleção de descritores.

### 3.1. Alocação do MeSH na Memória

A leitura do XML do MeSH é feita com a API SAX (Simple API for XML) [Python 2008b]. Apesar de existirem outras API mais fáceis de usar, o SAX é o mais adequado para este trabalho, pois não preciso colocar todo o arquivo em memória para manipulá-lo.

A medida que um termo era lido pela API SAX, métodos para o armazenamento eram executados. Caso o termo encontrado seja um descritor preferido, sua forma normalizada se torna a chave no *dictionary* e o termo completo é o valor associado a tal chave. Caso o termo é um sinônimo, outra estrutura é usada, onde a chave é a forma normalizada do sinônimo, e o valor é a forma normalizada do descritor correspondente.

Tendo as estruturas já formadas, elas são serializadas e gravadas em disco para que não seja necessário repetir o procedimento de leitura do XMLL. A serialização e a carga dos *dictionarys* são realizadas com auxílio da API Pickle do Python [Python 2008a].

### 3.2. Normalização de termos

Para implementar a normalização de termos, foram criadas duas classes e uma lista de *stop-words*. A primeira é uma implementação em Python do algoritmo de *Stemming* de Martin Porter [Gupta 2008], reponsável pela inflexão de palavras na aplicação. A outra classe identifica as *stop-words*, cruzando cada palavra de entrada com a lista obtida do projeto NLTK.

### 3.3. Navegação no MeSH

Para otenção de novos descritores, foi implementado um algoritmo capaz de navegar pela estrutura do MeSH utilizando a identificação hierárquica de cada descritor. Com isso, é possível contabilizar todos os ascendentes de cada descritor. Após esta etapa, é avaliada a relevância de cada ascendente para o texto. Aquele que possuir mais de 20% de seus filhos no texto será acrescentado a lista do programa.

### 3.4. Janelamento

No processo de identificação de descritores para o artigo, existem duas ocasiões em que a técnica de janelamento é utilizada. A primeira delas é diretamente na comparação entre palavras do artigo e vocabulário controlado. Existem descritores no vocabulário que possuem mais de uma palavra, então na leitura do artigo utilizamos uma janela de leitura para verificar a existência de várias palavras como sendo um descritor válido. Esse processo é parametrizável, possibilitando assim definir quantas palavras serão utilizadas. A segunda ocasião é para a análise dos bigramas, onde a janela representa a quantidade de palavras a frente de uma que está em leitura.

### 3.5. Bigrama

A aplicação da técnica de bigramas é dependente do conceito utilizado na leitura em janelamento no artigo, já que este conceito é reutilizado nesta técnica. Quando uma palavra esta no processo de leitura do texto, são feitas combinações com as palavras seguintes dentro de uma janela pré-definida e buscadas no vocabulário controlado. Porém a primeira combinação é descartada, já que a leitura do texto em janelamento cobre o primeiro caso de bigrama.

### 3.6. Corte de descritores

Como o programa é capaz de cruzar todas as palavras do artigo com vocabulário controlado, acontece de descritores com poucas ocorrências e irrelevantes para o assunto aparecerem como resultados do programa. Sendo assim, a eliminação destes é fundamental para a qualidade dos resultados e para isto, duas técnicas de corte foram implementadas.

O corte por ocorrência, como diz o nome, leva em consideração o número de aparições do descritor no texto. Todos os descritores são ordenados por ordem de aparições e então o corte é feito baseado no descritor de maior ocorrência no texto. Aqueles descritores que tiveram um número de aparições menor que 20% do número de aparições do descritor de maior ocorrência são eliminados do resultado.

Já o corte por quantidade simplesmente ordena os descritores, e mantém 10% dos descritores que tiveram maior numero de ocorrências.

### 3.7. Corte de descritores pré-definidos não existentes no texto

Para a validação dos descritores obtidos pelo programa foram utilizados artigos indexados manualmente por um profissional da área. Com isso, foram armazenados artigos e descritores. Para tornar a validação dos resultados coerente, é necessário verificar se os descritores pré-definidos para o artigo realmente estão no texto. Este procedimento é necessário, pois caso não realizado, os descritores obtidos pelo programa serão

comparados com descritores que foram escolhidos subjetivamente por um profissional. Como a técnica utilizada é apenas a definição de termos encontrados no texto, é correto comparar os resultados apenas com os descritores pré-definidos que realmente aparecem no artigo. São verificados se os descritores e seus sinônimos possuem alguma ocorrência no texto, e quando o caso é negativo, estes são eliminados dos descritores que serão usados para avaliar a qualidade dos resultados obtidos.

## 4. Resultados

Para avaliar as abordagens propostas e desenvolvidas neste trabalho, foi utilizada uma coleção de artigos para obter a média dos valores de precisão, cobertura e F-measure [Manning et al. 2008]. Os artigos utilizados foram extraídos da base de dados da SciELO. Esta fonte foi escolhida tendo em vista artigos completos, disponíveis para download e indexados.

Foi obtido através do site da BIREME, um volume de 419 artigos para serem avaliados. Estes com seus respectivos descritores foram armazenados em arquivos de texto puro para a validação de resultados. Cada artigo desta massa possui uma média de 11,28 descritores definidos manualmente, com um desvio padrão de 4,57.

O programa teve diferentes técnicas aplicadas, cada uma podendo ser executada de forma independente. Para a execução do programa, foram feitas combinações das técnicas sobre os 419 artigos, visando obter a melhor combinação possível usando as medidas apresentadas anteriormente. Também foram medidos nestes testes a eficiência dos diversos tamanhos de janela de leitura e os tipos de cortes usados. Primeiro foi executado o processo utilizando um tamanho de 3 palavras para a janela de leitura, com técnicas que não dependam de nenhum tipo de corte.

Na tabela 1 é possível visualizar os resultados de precisão, cobertura e F-measure encontrados para as três técnicas (existência dos descritores no artigo, navegação na árvore do MeSH e navegação na árvore do MeSH com filtragem dos descritores pré-definidos) sem cortes.

**Tabela 1. Técnicas sem corte e seus resultados**

| Tecnicas | Precisão | Cobertura | F-Measure |
|---|---|---|---|
| Existência dos descritores no Artigo | 4,803% | 48,809% | 8,746% |
| Navegação na arvore do Mesh | 4,406% | 51,153% | 8,114% |
| Navegação na arvore do Mesh + Filtragem dos descritores pré-definidos | 4,264% | 68,946% | 8,031% |

É possível observar que o programa conseguiu uma cobertura de mais de 60% de descritores dos artigos. Entrentanto, a não realização do corte faz com que um grande número de descritores seja retornado, diminuindo então a precisão do programa. Também é observado que com a utilização da navegação na estrutura do MeSH, a cobertura também aumenta. A filtragem dos descritores em muito dos casos aumentará a cobertura também, já que descritores que não estão no texto saem da comparação.

O próximo passo foi avaliar qual o tipo de corte seria mais eficiente. Também foi utilizada a navegação na estrutura do MeSH para obter mais descritores. Esta navegação

também foi realizada antes e após o corte, para avaliar o melhor momento que ela deveria ser realizada no processo. Na massa de dados utilizada, a navegação após o corte acrescentou uma média de 3,31 descritores novos por texto, tendo um desvio padrão de 3,34. Na tabela 2 são apresentados os resultados desta avaliação.

**Tabela 2. Comparação entre os possíveis cortes em diferentes técnicas**

| TÉCNICAS | CORTE POR OCORRÊNCIA | | | CORTE POR QUANTIDADE | | |
|---|---|---|---|---|---|---|
| | Precisão | Cobertura | F-Measure | Precisão | Cobertura | F-Measure |
| Apenas utilizando o corte | 19,184% | 20,226% | 19,691% | 17,993% | 19,038% | 18,501% |
| Corte antes da navegação no Mesh | 15,647% | 20,532% | 17,760% | 15,042% | 19,240% | 16,884% |
| Navegação no Mesh antes do corte | 16,989% | 20,514% | 18,586% | 15,839% | 19,238% | 17,374% |

Com os últimos resultados, já é possível afirmar que o corte por ocorrência é mais eficiente que o corte por quantidade, porém mais alguns testes serão feitos com ambos os cortes. Como já foi visto que o filtro nos descritores melhora a cobertura. O principal motivo é o fato de que, no conjunto de artigos utilizado, temos uma média de 3,19 descritores com desvio padrão de 2,17 que são definidos subjetivamente e não estão presentes no texto. O próximo resultado apresenta a combinação desta técnica mais a do bigrama na tabela anterior (tabela 3).

**Tabela 3. Resultados com as técnicas de filtragem e bigramas**

| TÉCNICAS | CORTE POR OCORRÊNCIA | | | CORTE POR QUANTIDADE | | |
|---|---|---|---|---|---|---|
| | Precisão | Cobertura | F-Measure | Precisão | Cobertura | F-Measure |
| Corte + Filtragem + Bigramas | 18,476% | 28,522% | 22,425% | 17,196% | 26,752% | 20,935% |
| Corte antes da navegação + Filtragem | 14,989% | 28,787% | 19,714% | 14,314% | 27,034% | 18,717% |

A tabela 4 revela que tanto o bigrama como o filtro aumentam entre 7 e 8% o cobertura do programa, mas a precisão cai aproximadamente 0,8%. Entretano, a medida harmônica F-measure aponta que o melhor resultado fica por conta da não utilização da navegação do MeSH.

Com todos esses resultados entre técnicas e cortes, a melhor abordagem para o programa foi a utilização do corte por ocorrência e a técnica de bigramas para obter os melhores descritores de um artigo. Com isso, um novo teste foi realizado alterando a janela de leitura do programa para quatro, e o resultado obtido pode ser visto na tabela 4.

Os novos valores para a média harmônica representam uma sensível melhora nos resultados obtidos. Isso porque o numero de combinações aumenta para a análise de bigramas, e a janela de leitura pode encontrar um descritor de até quatro palavras no texto. Números menores não são aconselhados por eliminar possíveis descritores existentes no vocabulário controlado, e uma janela maior pode distorcer os resultados do bigrama.

**Tabela 4. Comparações entre janelas**

| TÉCNICAS | Precisão | Cobertura | F-Measure |
|---|---|---|---|
| Corte por Ocorrência + Filtragem + Bigramas (Janela 3) | 18,476% | 28,522% | 22,425% |
| Corte por Ocorrência + Filtragem + Bigramas (Janela 4) | 18,631% | 28,792% | 22,623% |

Com isso, é possível afirmar que a melhor configuração para o programa é a utilização de uma janela de leitura igual a quatro palavras, com as técnicas de corte por ocorrência e bigramas. O programa com esta configuração final, retornou, para os artigos utilizados na validação, uma média de 35,19 descritores por artigo, com um desvio padrão de 28,8.

## 5. Considerações sobre os resultados e trabalhos futuros

Através dos resultados obtidos pela validação, algumas considerações podem ser realizadas sobre as mesmas. O fato da média final de descritores sugeridos ser alta reflete a capacidade do computador de cruzar todas as palavras do artigo com as existentes no vocabulário controlado, e assim, retornar um grande número de descritores. Esta atividade é dificilmente realizada por uma pessoa, já que a mesma precisaria saber todo o vocabulário controlado, ou então ter que procurar uma a uma as palavras ao ler o texto. Outro ponto importante e interessante é a questão de que, com a navegação na estrutura do MeSH para descobrir novos descritores possíveis para o texto ter prejudicado a qualidade dos resultados. Foram identificadas duas razões para este fato. A primeira é que a heurística utilizada para definir novos descritores está imprecisa. Isto acarreta em trazer descritores diferentes dos escolhidos por um indexador manual, e diminuir a cobertura e a precisão da validação. A outra razão é a de que, para estes artigos, os profissionais que definiram os descritores para os artigos preferem utilizar aqueles que são de assuntos mais específicos, evitando a utilização de um descritor mais genérico que consiga representar uma parte significativa destes assuntos.

Existem algumas alterações possíveis ao programa para que este consiga obter resultados melhores para sua proposta. A criação de um módulo para o aprendizado de maquina poderia personalizar o programa com as características de um único indexador manual, aprendendo com os artigos já indexados. Assim o resultado será aproximado de um profissional da área. Para melhorar a leitura no artigo, seria interessante atribuir pesos para algumas seções do texto, como por exemplo, o título, resumo e conclusões, já que estes são trechos que provêem uma síntese do conteúdo tratado no documento. Conseguindo atribuir um peso maior aos descritores encontrados nestas partes, acredita-se que melhores resultados podem ser alcançados. Para melhorar a leitura no artigo, seria interessante atribuir pesos para algumas seções do texto, como por exemplo, o título, resumo e conclusões. Todo o conteúdo do artigo esta em ênfase nesses pontos. Conseguindo atribuir um peso maior aos descritores encontrados nestas partes, acredita-se que os resultados possam retornar resultados melhores. Para indexar um artigo, o contexto onde este será inserido deve ser levado em consideração [LANCASTER 2004]. Dependendo da biblioteca onde será inserido o artigo, podem ser escolhidos descritores mais abrangentes ou que tratem de um assunto mais específico. Este seria outro aprimoramento possível de se realizar neste programa.

## Referências


Cavnar, W. B. and Trenkle, J. M. (1994). N-gram-based text categorization. In *In Proceedings of SDAIR-94, 3rd Annual Symposium on Document Analysis and Information Retrieval*, pages 161–175, Obtido em: http://www.dcs.fmph.uniba.sk/diplomovky/obhajene/getfile.php/Ng-based-tc.pdf?id=1&fid=3&type=application%2Fpdf, Acessado em: 4 de Março de 2009.

Gupta, V. (2008). *Python Implementation of Porter Stemming Algorithm*. Obtido em: http://tartarus.org/ martin/PorterStemmer/python.txt, Acessado em 5 de Março de 2009, 2 edition.

LANCASTER, F. W. (2004). *Indexação e Resumos: Teoria e Prática*. Briquet de Lemos, Brasília.

Manning, C. D., Raghavan, P., and Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press, Cambridge.

NLM (2006). *Concept Structure in XML MeSH*. Obtido em http://www.nlm.nih.gov/mesh/concept_structure.html, Acessado em 5 de Março de 2009.

NLM (2008). *Medical Subject Headings*. Obtido em: ftp://nlmpubs.nlm.nih.gov/online/mesh/.xmlmesh/desc2008.gz, Acesso em 5 de Març de 2009.

NLTK (2008). *Natural Language Toolkit*. http://www.nltk.org, Acessado em 4 de Março de 2009.

Porter, M. F. (1997). An algorithm for suffix stripping. pages 313–316.

Python (2008a). *Python object serialization*. Obtido em: http://docs.python.org/library/pickle.html?highlight=pickle, Acessado em 4 de Março de 2009.

Python (2008b). *Support for SAX2 Parsers*. Obtido em: http://docs.python.org/library/xml.sax.html, Acessado em 4 de Março de 2009.

SciELO (2008). *Scientific Eletronic Library*. http://www.scielo.org, Acessado em 4 de Março de 2009.