TP-DISCOVER: um ambiente computacional para auxílio no pré-processamento de laudos médicos não-estruturados

Daniel de Faveri Honorato[1], Maria Carolina Monard[2], Huei Diana Lee[1], Carlos Andres Ferrero[1], Feng Chung Wu[1]

[1]Centro de Engenharias e Ciências Exatas, Universidade Estadual do Oeste do Paraná – Unioeste, Laboratório de Bioinformática – LABI, Parque Tecnológico Itaipu – PTI, Foz do Iguaçu, PR, Brasil

[2]Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo – USP, Laboratório de Inteligência Computacional – LABIC, São Carlos, SP, Brasil

Resumo — *O processo de Mineração de Textos pode auxiliar especialistas na tomada de decisão por meio de extração de padrões a partir de tabelas atributo-valor nas quais os textos são representados de uma maneira estruturada. Neste trabalho é apresentado o ambiente computacional TP-DISCOVER o qual implementa um método para mapear laudos médicos não-estruturados para uma representação atributo-valor. Esse ambiente permite aplicar o método proposto de modo automático, sem interação com especialistas do domínio ou de modo semi-automático, no qual o especialista pode participar do processo de aplicação do método.*

Palavras chave — descoberta de conhecimento, extração de padrões, análise de documentos, extração de terminologias.

1. Introdução

O avanço tecnológico tem motivado o armazenamento de informações em bases de dados, tornando cada vez mais difícil analisar e extrair, manualmente, informações e padrões a partir dos dados. Em hospitais e clínicas médicas, é gerada uma quantidade de informações ao longo do tratamento e acompanhamento de pacientes. Essas informações apresentam-se em diversos formatos como laudos e formulários médicos, sendo que muitas dessas informações encontram-se atualmente armazenadas eletronicamente. Assim, é de interesse dos especialistas que essas informações sejam analisadas de modo mais completo, com a finalidade de extrair padrões que auxiliem, por exemplo, em processos de tomada de decisão associados ao diagnóstico de enfermidades. No entanto, a análise manual das informações contidas em conjuntos de laudos médicos torna-se inviável, pois, trata-se de uma tarefa que apresenta alto custo de tempo e que está sujeita à subjetividade [5,8].

Os laudos médicos são representados em diferentes formatos, sendo que um dos formatos comumente utilizados é o formato textual não-estruturado. Porém, para que as informações contidas nesses laudos possam ser analisadas por métodos computacionais é necessário que estejam no formato apropriado, de tal modo, que seja possível construir modelos que representem o conhecimento embutido nos dados. Um dos processos que pode dar apoio na realização dessa tarefa é o processo de Mineração de Textos [8], o qual consiste, basicamente, em três fases principais: (1) pré-processamento dos textos; (2) extração de padrões; e (3) pós-processamento. Na fase de pré-processamento, o conjunto de textos é transformado para uma representação adequada para ser utilizada pelos algoritmos de extração de padrões. Geralmente, os documentos são representados em uma tabela atributo-valor, na qual palavras selecionadas do conjunto de documentos são transformadas em atributos. Essa tabela é então utilizada na fase de extração de padrões para construir

modelos. Nessa etapa podem ser utilizados, por exemplo, algoritmos de aprendizado de máquina, uma subárea da Inteligência Artificial [11]. Os modelos induzidos podem ser representados por estruturas simbólicas como árvores de decisão e regras de produção, as quais permitem maior compreensibilidade humana [14]. Na terceira etapa, os modelos construídos são analisados e validados juntamente com os especialistas do domínio.

Em [4,7] foi proposto e implementado um método geral para o pré-processamento de documentos não-estruturados, ou seções específicas desses documentos, que tem como resultado a representação em uma tabela atributo-valor das informações contidas nesses documentos. Esse método pode ser aplicado a qualquer conjunto de documentos textuais que verifiquem as seguintes duas propriedades:

1. as informações são descritas utilizando um vocabulário controlado; e
2. as informações consistem de frases assertivas simples.

Em geral, os laudos médicos verificam ambas as propriedades. Um dos principais objetivos desse método é diminuir, sempre que possível, a intervenção dos especialistas, fornecendo, nas diversas fases do método, informações que facilitam o trabalho a ser realizado pelos especialistas[1], que são os responsáveis pela inclusão do aspecto semântico das informações. O método pode ser aplicado de dois modos:

- automático: sem intervenção dos especialistas, considerando somente o aspecto morfo-sintático das informações, *i.e.*, sem levar em conta o aspecto semântico.
- não-automático: com a participação de especialistas do domínio, responsáveis por levar em conta o aspecto semântico das informações que estão sendo processadas. Desse modo, informações específicas do domínio podem ser utilizadas e melhores resultados poderão ser alcançados.

Este trabalho tem por objetivo apresentar o ambiente computacional *TP-DISCOVER* o qual implementa todas as fases do método proposto em [4] para o pré-processamento de documentos não-estruturados. O trabalho está organizado da seguinte maneira: na Seção 2, é descrito resumidamente o método desenvolvido; na Seção 3, é apresentado o ambiente computacional *TP-DISCOVER*; e, na Seção 4, são apresentadas as considerações finais.

2. Método para Pré-processamento de Documentos Não-estruturados

O método proposto [4], o qual foi implementado no ambiente computacional *TP-DISCOVER*, é composto de cinco fases, ilustradas na Figura 1 e descritas a seguir.

1. Pré-processamento;
2. Extração de Terminologia;
3. Identificação de Atributos;
4. Construção do Dicionário; e
5. Construção da Tabela Atributo-valor.

[1] Previamente, foi proposto em [5] outro método para realizar essa tarefa, mas que requer uma intensa interação com os especialistas do domínio, limitando a sua aplicação e motivando o desenvolvimento deste novo método.

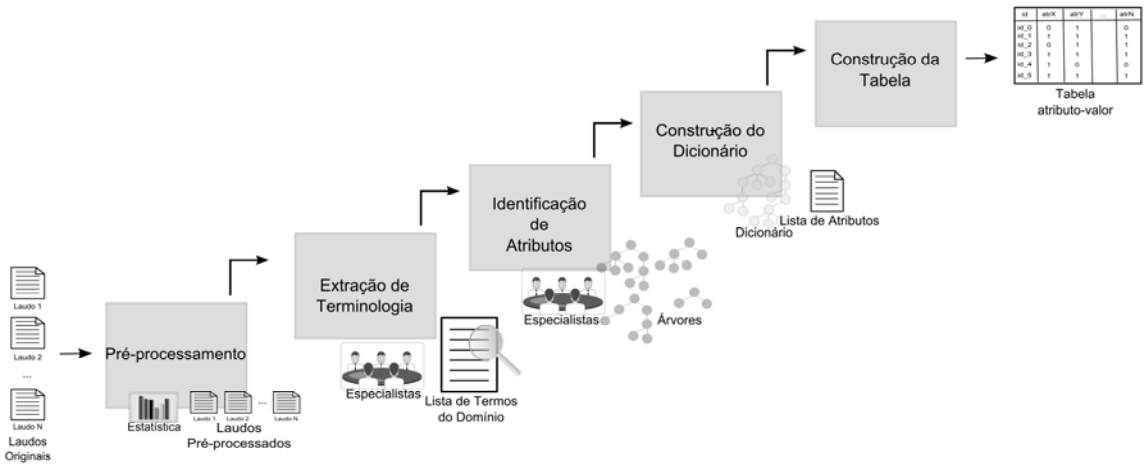

Figura 1. Método desenvolvido para pré-processamento de documentos não-estruturados [4]

***1 - Pré-processamento*:** essa fase consiste na preparação do *corpus*[2] por meio da aplicação de diferentes técnicas que permitam uniformizar as informações. Nessa fase é possível realizar tarefas como: divisão do documento de acordo com as seções que ele contém; construção do conjunto de frases únicas, que consiste de todas as frases diferentes do conjunto de documentos; remoção de *stopwords*, que são palavras consideradas não relevantes para a análise do texto; transformação para minúsculo; correção ortográfica; aplicação de substituições, tais como sinônimos ou frases que mapeiam mais de um evento; e aplicação do lematizador, o qual transforma verbos para a forma infinitiva e substantivos e adjetivos para masculino singular.

***2 - Extração de Terminologia*:** nessa fase são geradas, a partir da coleção de documentos, listas de termos do domínio que possuem determinadas propriedades sintáticas. O objetivo consiste em determinar as palavras mais apropriadas para serem consideradas como unidades terminológicas, as quais são utilizadas na próxima fase. Para isso, é adotada uma abordagem híbrida, a qual utiliza tanto conhecimento estatístico quanto linguístico e aplica algumas heurísticas sobre o conjunto de termos identificados. Nessa abordagem, é realizada uma análise morfo-sintática dos termos do domínio e, posteriormente, são geradas duas listas: uma de unigramas contendo palavras que casam com a classe gramatical N (substantivo) e outra de bigramas contendo palavras que casam com palavras consecutivas da classe gramatical N N. Nesse trabalho denominamos as classes gramaticais como "máscara". Primeiramente, o *corpus* é etiquetado, e após, são extraídas as palavras que combinam com as características apresentadas (N e N N).

Em geral, nas listas de unigramas e bigramas resultantes da aplicação do método híbrido, há duas características importantes a serem levadas em consideração: a ocorrência de termos com baixa frequência; e a presença de termos em uma determinada lista de unigramas que constituem parte de algum bigrama da lista de bigramas. Assim, para encontrar unidades terminológicas mais apropriadas para o domínio, são propostas algumas heurísticas para reduzir o número de termos identificados, usando um parâmetro Alpha, o

[2] Em lingüística, um *corpus* consiste em um conjunto de textos, os quais são utilizados em análises estatísticas, verificação de ocorrências e validação de regras lingüísticas em um universo específico.

qual, a partir da lista de unigramas e da lista de bigramas, permite favorecer a escolha de um unigrama ou de um bigrama para fazer parte da lista. Posteriormente, os termos (unigramas ou bigramas) que possuem frequência menor ou igual a um limiar Theta, (definido pelo usuário, em relação ao número de documentos) são removidos. Como mencionado, essa lista é utilizada na próxima fase do método.

***3 - Identificação de Atributos, Construção do Dicionário e da Tabela*:** a identificação dos atributos que constituem as colunas das tabelas atributo-valor é realizada por meio de três etapas:

1. definição dos termos que serão utilizados como raiz das árvores[3];
2. geração das árvores; e
3. identificação dos atributos a partir das árvores geradas.

Na primeira etapa, os termos raiz podem ser identificados de maneira automática ou não-automática. Na execução do *TP-DISCOVER*, no modo automático, todos os termos da lista de termos candidatos identificados na fase de Extração de Terminologia são considerados para serem utilizados como termos raiz das árvores na próxima etapa. No modo não-automático, uma análise, junto com os especialistas, pode ser realizada com o intuito de identificar os termos (contidos na lista de termos candidatos final) utilizados no mapeamento de informações, pois podem existir termos na lista que não sejam de interesse dos especialistas.

Depois de definidos os termos raiz das árvores, na segunda etapa, é executado o algoritmo de geração de árvores, lembrando que para cada termo considerado como unidade terminológica é gerada pelo *TP-DISCOVER* uma árvore cuja raiz é definida por esse termo. As árvores geradas possuem uma estrutura semelhante à árvore ilustrada na Figura 2.

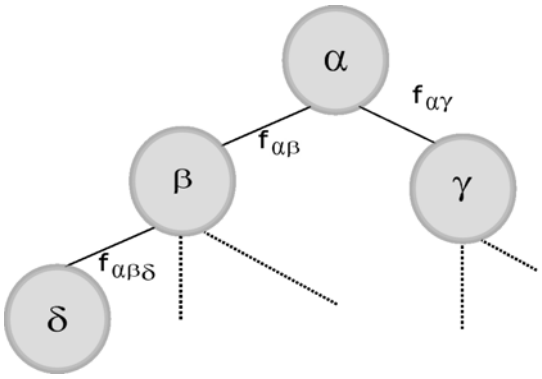

Figura 2. Árvore gerada a partir de termos raiz definidos na fase de Extração de Terminologia

Nessa árvore, o termo α, onde o tamanho de α é dado por $1 \leq |\alpha| \leq 2$, corresponde ao nó raiz identificado pelo método de extração de terminologia, os termos β e γ são filhos de α e δ é filho de β. Na árvore, todos os filhos possuem um número de palavras maior ou igual a 1 e mapeiam as palavras que aparecem no contexto de um termo raiz. Por exemplo, se a

[3] Em uma árvore são mapeados os termos que aparecem no contexto de uma unidade terminológica X, por exemplo, X A B, X C D, juntamente com a frequência desses termos. A partir das árvores são identificados os atributos que farão parte da tabela atributo-valor.

palavra *coloração* e *esbranquiçada* sempre aparecem juntas, elas serão colocadas em um mesmo nó filho, caso contrário serão colocadas em nós diferentes. Na árvore gerada também são armazenadas as frequências em que dois termos ocorrem juntos. Considerando $f_{\alpha\beta}$ a frequência com que α e β aparecem juntos e $f_{\alpha\beta\delta}$, a frequência com que $\alpha\beta\delta$ aparecem juntos, então $f_{\alpha\beta\delta} < f_{\alpha\beta}$. Essa relação se verifica em todos os ramos da árvore.

Na terceira etapa, são identificados os atributos para compor a tabela atributo-valor. Essa identificação também pode ser realizada de modo não-automático, *i.e.*, com a intervenção do especialista do domínio; ou de modo automático. Neste último caso, os ramos que possuírem frequência maior ou igual a um limiar são considerados atributos. Por exemplo, considere o limiar l e a árvore da Figura 2. Na identificação automática de atributos, primeiramente é verificado se $f_{\alpha\beta} \geq l$ e, se for, é gerado o atributo $\alpha\beta$. Após, é verificado se $f_{\alpha\beta\delta} \geq l$ e, se for, é definido $\alpha\beta\delta$ como atributo. Por último é verificado se $f_{\alpha\gamma} \geq l$ e, se for, é definido $\alpha\gamma$ como atributo.

Uma vez identificados os atributos, eles são inseridos em um dicionário denominado de dicionário de conhecimento. Desse modo, as informações dos documentos e os atributos presentes no dicionário de conhecimento são utilizados no processo de preenchimento da tabela atributo-valor. Nessa tabela, cada linha corresponde a um documento da coleção de documentos, tal que, se a sequência de termos definida por um determinado atributo do dicionário for identificada no documento, o valor desse atributo é preenchido com 1 (presente); se a sequência não for identificada no documento, o atributo é preenchido com 0 (ausente). Esse processo é repetido para todos os documentos do conjunto.

6. Ambiente Computacional

O ambiente computacional *TP-DISCOVER* possui dois módulos, sendo que o primeiro foi desenvolvido utilizando a linguagem Java[4] e o segundo foi desenvolvido utilizando a linguagem Perl[5] [13], conforme ilustrado na Figura 3.

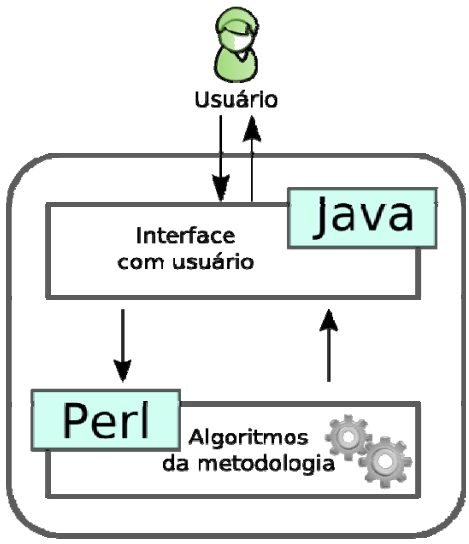

Figura 3. Módulos do ambiente TP-DISCOVER

[4] http://java.sun.com/.
[5] http://perldoc.perl.org/.

O primeiro módulo é responsável por realizar a interação com o usuário, por meio de uma interface gráfica construída utilizando os recursos para desenvolvimento de interfaces gráficas da ferramenta *NETBEANS*[6]. O segundo módulo é responsável por aplicar os algoritmos do método desenvolvido. A interação entre os dois módulos é realizada por meio de arquivos no formato XML.

No primeiro módulo, os algoritmos foram implementados utilizando o conceito de camadas, as quais são: interface, negócio e persistência. A camada de interface, geralmente chamada de *Graphical User Interface* – GUI – é responsável pela interação do usuário com o sistema. A camada de negócio é responsável pela implementação da lógica de negócio da aplicação, ou seja, nessa camada encontram-se todos os algoritmos e classes inerentes ao domínio da aplicação, neste caso, aos algoritmos de interação com o usuário. A camada de persistência é responsável pela manipulação e inserção das informações fornecidas pelo usuário em um meio de armazenamento que possa ser recuperado posteriormente. Neste trabalho foram utilizados arquivos no formato XML.

O segundo módulo é constituído por *scripts*, os quais implementam os algoritmos do método desenvolvido, e são invocados a partir do primeiro módulo. Também, o *TP-DISCOVER* utiliza diversos programas auxiliares na implementação das fases do método, tais como N-GRAM STATISTICS PACKAGE – NSP[7] [1], GRAPHVIZ[8] [3], SENTER [9], MXPOST [10] e TREETAGGER[9] [11].

Para executar o *TP-DISCOVER* é necessário que estejam disponíveis os documentos a partir dos quais serão aplicados os algoritmos e gerada a tabela atributo-valor. Esses documentos devem estar na forma textual em arquivos digitais no formato TXT. A seguir é descrito como o *TP-DISCOVER* implementa as diversas fases da metodologia proposta.

1 - Pré-processamento: implementado pelas seguintes cinco tarefas[10]: padronização das informações; aplicação de substituições; correção ortográfica; aplicação de lematização; e estatísticas sobre o conjunto de frases únicas – CFU – dos laudos.

Na Figura 4 é mostrada a interface de padronização de informações. O sistema gera automaticamente um arquivo contendo o CFU identificado no conjunto de documentos que está sendo processado e um arquivo de estatísticas desse CFU. Nessa interface o usuário pode indicar algumas modificações que deverão ser aplicadas sobre o conjunto de documentos tais como: transformar o texto para minúsculo, informar a lista de *stopwords* (o usuário pode carregar um arquivo de *stopwords* a partir de um arquivo TXT ou inserir individualmente cada *stopword*) que deverão ser retiradas do texto e também informar, caso existam, as substituições de termos utilizando expressões regulares.

É importante destacar que as alterações que são configuradas nessa interface são aplicadas sobre todo o conjunto de documentos. Após serem aplicadas as alterações, o sistema gera um novo CFU e as estatísticas respectivas.

[6] http://www.netbeans.org/.
[7] http://www.d.umn.edu/~tpederse/nsp.html.
[8] http://www.GraphViz.org/.
[9] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.
[10] Por questões de espaço, ilustramos somente as interfaces correspondentes a primeira e terceira tarefas.

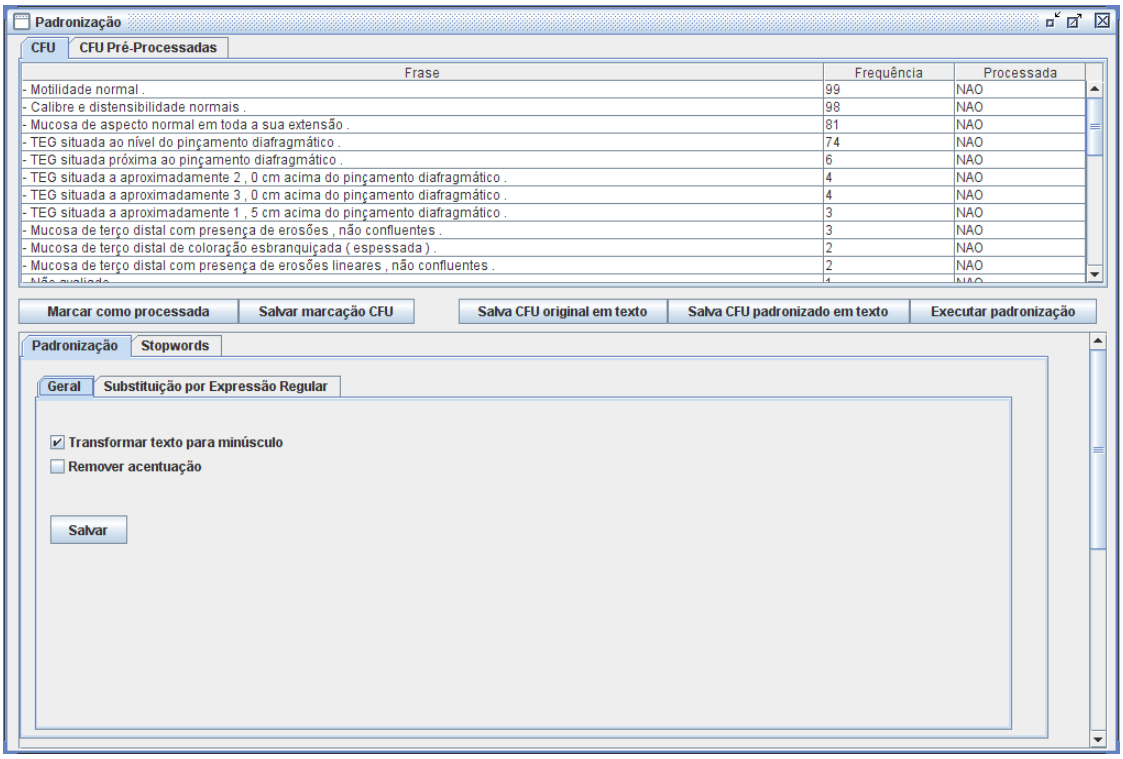

Figura 4. Interface de padronização do ambiente TP-DISCOVER

O *TP-DISCOVER* permite que seja realizada a correção ortográfica nas informações contidas nos documentos. Duas opções são permitidas: manual ou automática. Na Figura 5 é ilustrada a interface utilizada para realizar a correção ortográfica, a qual pode ser utilizada tanto para realizar a correção manual como a automática.

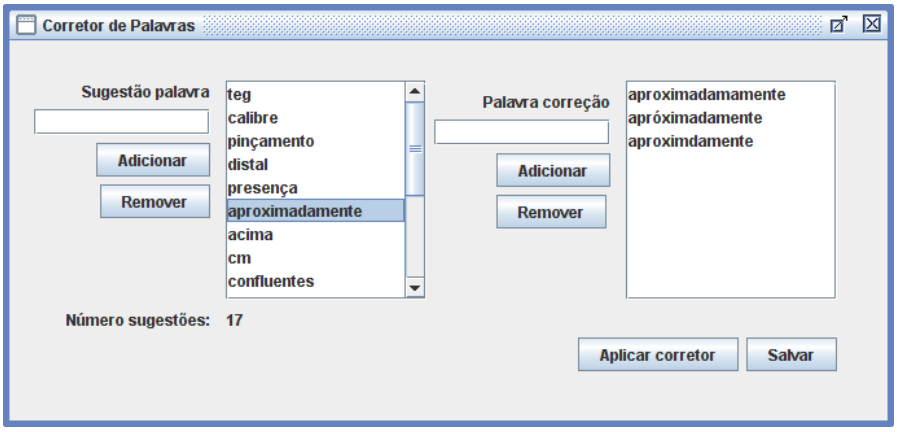

Figura 5. Interface do corretor de palavras do ambiente TP-DISCOVER

No primeiro modo, manual, o usuário deve analisar manualmente o CFU padronizado, o qual pode ser acessado na interface ilustrada na Figura 4, e identificar as palavras

incorretas. À medida que essas palavras são identificadas, elas podem ser inseridas nessa interface. Depois de inseridas as palavras a serem corrigidas, o usuário pode aplicar a correção de palavras, a qual irá substituir a palavra incorreta pela palavra correta, por ele informada, no conjunto de documentos padronizados.

No modo de correção automática, o ambiente computacional gera automaticamente, utilizando um algoritmo simples de correção ortográfica por nós implementado [4,7], uma lista de sugestões de correções baseado no cálculo de medida de distância entre cada termo do texto. O usuário pode então analisar essa lista de sugestões de correções, e remover sugestões incorretas, e/ou inserir manualmente outras correções.

***2 - Extração de Terminologia*:** como mencionado, a extração de terminologia utilizando a abordagem híbrida proposta neste trabalho, necessita que sejam definidas máscaras, a partir das quais serão extraídas as unidades terminológicas do conjunto de documentos etiquetados. Na interface ilustrada na Figura 6 é possível definir as máscaras que serão aplicadas na extração de terminologia.

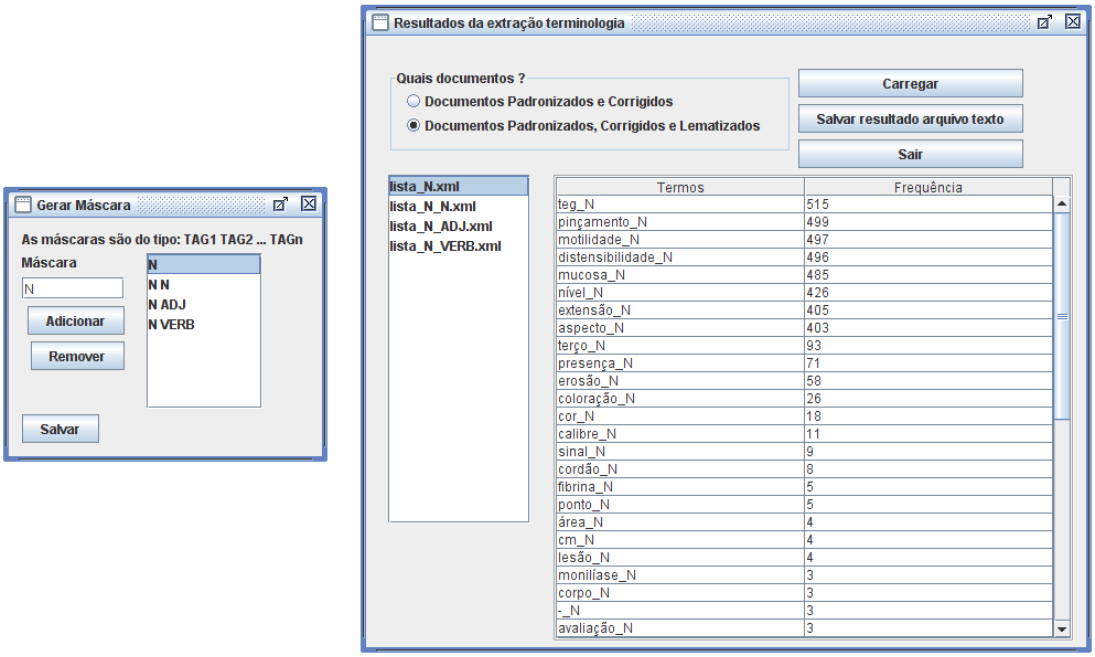

Figura 6. Interfaces para definição de máscaras e resultados do ambiente TP-DISCOVER

As máscaras utilizadas podem ser: N (substantivo), ADJ (adjetivo), VERB (verbo), LOCU (locução), ADV (advérbio), PREP+ART (preposição + artigo), ou qualquer combinação das mesmas. Após definidas as máscaras, deve ser aplicado o algoritmo de extração de terminologia [6], o qual irá identificar, em todo o conjunto de documentos selecionado, todos os termos que casam com as máscaras que foram definidas. Depois de realizada a extração de terminologia, o usuário pode visualizar, por meio da interface ilustrada na Figura 6, as unidades terminológicas que foram extraídas a partir de cada máscara definida.

Nessa interface o usuário deve selecionar o arquivo correspondente à máscara definida e, depois de selecionada a máscara, o sistema retorna as unidades terminológicas que casaram com ela e a frequência do termo no conjunto de documentos.

***3 - Identificação de Atributos e Construção da Tabela Atributo-valor*:** antes da geração das árvores, é necessário definir quais serão os termos (unidades terminológicas) que serão utilizados como raiz das árvores. Essa tarefa é realizada por meio da geração da lista de termos candidatos e posterior aplicação das heurísticas utilizando os parâmetros Alpha e Theta. Após, a lista de termos final que será utilizada para a geração da árvore pode ser definida a partir de uma análise manual. As duas listas de termos e os parâmetros que serão utilizados para identificar as unidades terminológicas podem ser configurados em uma interface específica ilustrada na Figura 7.

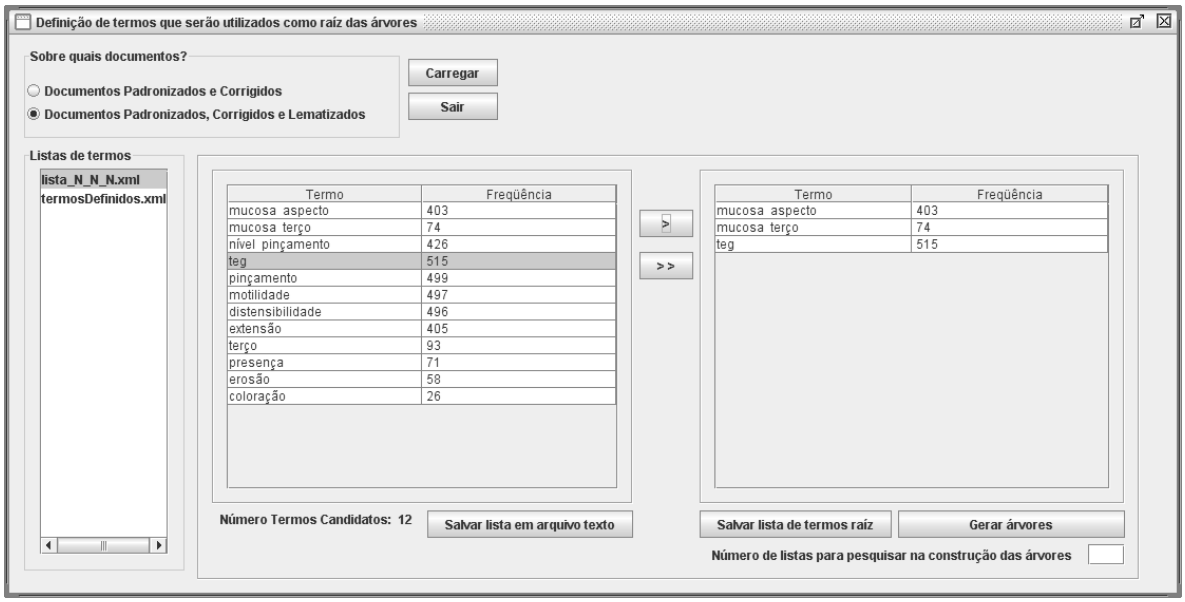

Figura 7. Interface de definição de termos raiz do ambiente TP-DISCOVER

Nessa interface, primeiramente o usuário seleciona o conjunto de termos candidatos extraídos, o qual está armazenado, no exemplo da Figura 7, no arquivo *listaN_NN.xml*, por meio da aplicação de heurísticas. No caso aqui ilustrado, nesse arquivo são armazenados: uma lista de termos gerada por meio da aplicação das heurísticas sobre uma lista de unigramas N; e outra de bigramas N N. O usuário pode selecionar os termos manualmente com a ajuda de um especialista do domínio ou pode selecionar todos os termos identificados pelo algoritmo. Para cada termo definido, o ambiente computacional irá gerar uma árvore[11] que será utilizada posteriormente para a identificação de atributos. No caso de interações com o especialista para a identificação manual dos atributos, as árvores geradas podem ser visibilizadas por meio da interface ilustrada na Figura 8. Nessa mesma figura, é ilustrada a árvore gerada a partir do termo selecionado, **mucosa terço**.

[11] A altura máxima da árvore corresponde ao número médio de palavras das frases do CFU ou pode ser definido pelo usuário.

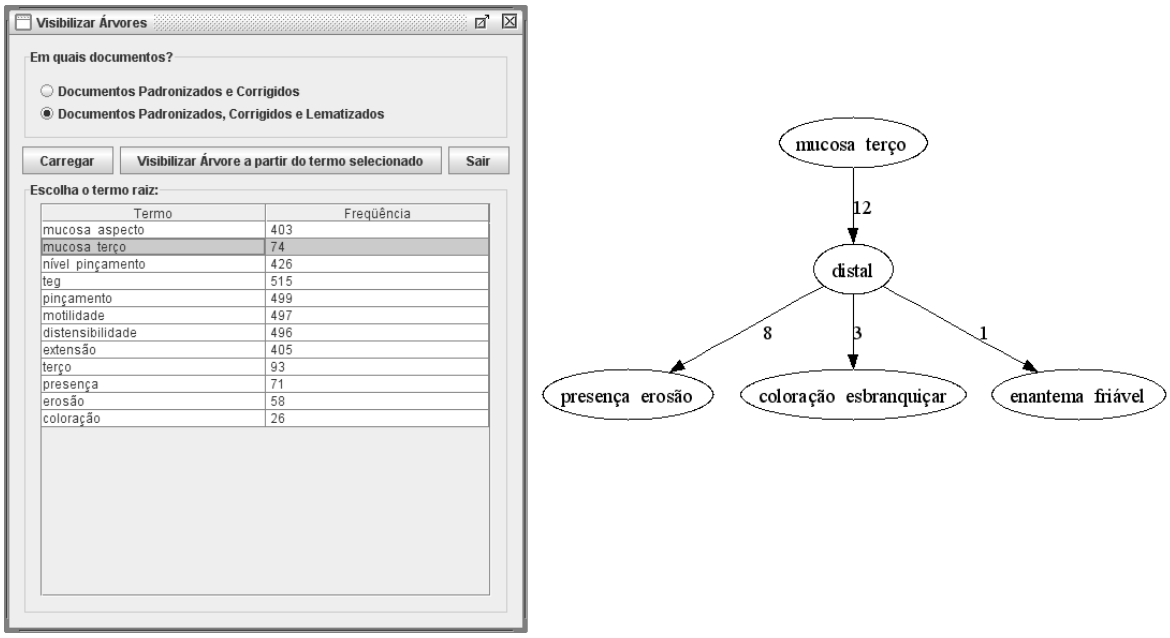

Figura 8. Interface de visibilização das árvores do ambiente TP-DISCOVER

Após a geração das árvores, o usuário pode executar as funcionalidades para a geração de atributos e preenchimento da tabela atributo-valor. Para esse fim, como mencionado, existem dois modos de construir o dicionário de atributos: modo manual e modo automático. No modo manual o usuário, com a ajuda dos especialistas, deve analisar as árvores geradas e identificar os atributos, os quais são inseridos em uma interface específica do sistema. No modo automático, o usuário deve acessar a interface ilustrada na Figura 9 e definir o limiar de criação dos atributos.

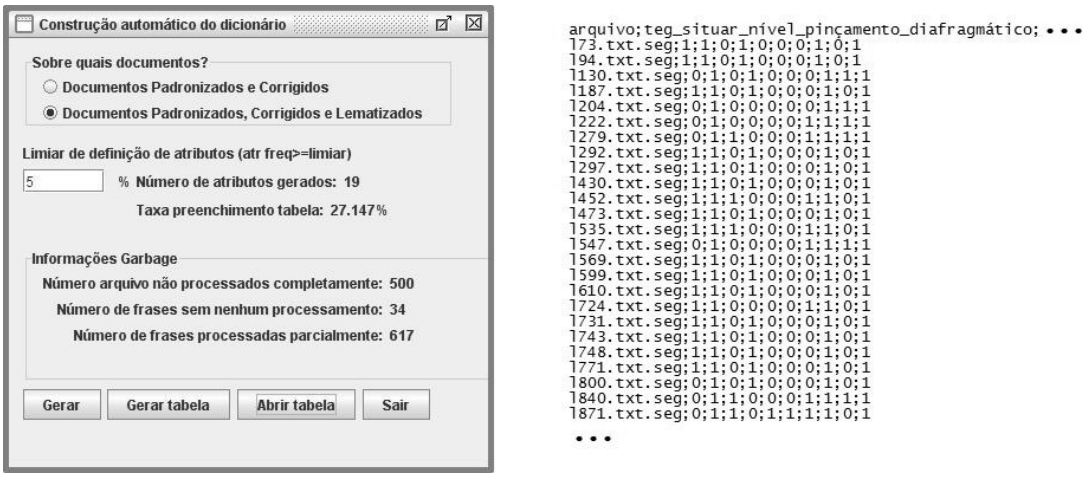

Figura 9. Interface de construção automática do dicionário e a tabela gerada do ambiente TP-DISCOVER

Esse limiar refere-se à porcentagem relacionada à frequência com que um determinado ramo da árvore aparece em relação ao número de laudos. Por exemplo, a partir de um conjunto de 100 laudos e definindo-se o limiar 5, o ambiente computacional irá selecionar

todos os ramos que possuem frequência maior ou igual a 5 para serem atributos da tabela atributo-valor. Depois de definidos os atributos, pode ser gerada automaticamente a tabela atributo-valor (fragmento ilustrado na Figura 9), a partir da qual o ambiente computacional fornece a taxa de preenchimento, ou seja, a porcentagem de células que foram preenchidas como presentes. Nessa tabela, a primeira linha contém os atributos que constituem a tabela atributo-valor. As linhas restantes contêm o nome do arquivo do documento processado, seguido dos valores dos atributos identificados.

Na geração da tabela atributo-valor é também gerado um arquivo, denominado de *garbage*, no qual são armazenadas as frases dos documentos que não sofreram nenhum processamento e as frases que foram processadas parcialmente. Ao final, o ambiente computacional permite calcular o número dessas frases e exibe o resultado para o usuário. É interessante ressaltar que essa é uma informação importante, uma vez que indica a porcentagem de frases que estão sendo processadas completamente, parcialmente ou não estão sofrendo nenhum processamento, em relação ao número total de frases do conjunto de documentos.

7. Considerações Finais

Este trabalho teve por objetivo apresentar o ambiente computacional *TP-DISCOVER*, o qual implementa todas as fases do método proposto em [4]. Esse ambiente permite ao usuário, por meio de interfaces amigáveis, a interação com o sistema e possibilita que o especialista também participe de modo interativo. A possibilidade da participação do especialista é muito importante neste tipo de sistema, devido ao fato de que permite fornecer informações semânticas específicas do domínio que poderão auxiliar para alcançar a obtenção de melhores resultados.

O *TP-DISCOVER* foi aplicado, no modo automático, i.e., sem intervenção de especialistas, para realizar a análise de laudos de Endoscopia Digestiva Alta e estruturar a informação contida nesses laudos na tabela atributo-valor correspondente. Ainda que os resultados obtidos pelo *TP-DISCOVER* executado no modo automático representam os resultados no pior caso, pois somente é considerada informação morfo-sintática dos textos, os resultados obtidos, publicados em [4,7], mostram a adequabilidade da nossa proposta.

Trabalhos futuros incluem estender a implementação do *TP-DISCOVER* para considerar unidades terminológicas de termos trigramas; avaliar a interação homem-máquina do ambiente *TP-DISCOVER* com diversos usuários; e analisar o grau de aceitação dos especialistas no domínio; pesquisar métodos para o tratamento do diagnóstico incluído nos laudos, os quais requerem um pré-processamento diferenciado; bem como a aplicação do *TP-DISCOVER* a outros domínios do conhecimento que verifiquem as duas propriedades requeridas para aplicar a metodologia implementada no sistema.

Agradecimentos

Trabalho realizado com o auxílio do Programa de Desenvolvimento Tecnológico Avançado – PDTA – da Fundação Parque Tecnológico Itaipu – FPTI-BR – e do Conselho Nacional de Desenvolvimento Científico e Tecnológico – CNPq - Brasil.

Referências

[1] Banerjee, S. and Pedersen, T. (2003). The design, implementation, and use of the Ngram Statistic Package. In *Proceedings of the Fourth International Conference on Intelligent Text Processing and Computational Linguistics*, pages 370–381, Cidade do México, México.

[2] Feldman, R. and Sanger, J. (2006). *The Text Mining Handbook: Advanced Approaches in Analyzing Unstructured Data*. Cambridge University Press, Nova Iorque, EUA.

[3] Gansner, E. R. and North, S. C. (2000). An open graph visualization system and its applications to software engineering. *Software – Practice and Experience*, 30(11):1203–1233.

[4] Honorato, D. D. F. (2008). Metodologia de transformação de laudos médicos não-estruturados e estruturados em uma representação atributo-valor. Dissertação de Mestrado, ICMC-USP, http://www.teses.usp.br/teses/disponiveis/55/55134/tde-10062008-154826/publico/dissertacaoDanielHonorato.pdf.

[5] Honorato, D. D. F., Cherman, E. A., Lee, H. D., Monard, M. C., and Wu, F. C. (2008a). Construction of an attribute-value representation for semi-structured medical findings knowledge extraction. *CLEI Electronic Journal*, 11(2):1–12.

[6] Honorato, D. D. F. and Monard, M. C. (2008). Descrição de uma metodologia de mapeamento de informações não estruturadas em uma representação atributo-valor. Technical Report 317, ICMC-USP. http://www.icmc.usp.br/˜biblio/BIBLIOTECA/rel_tec/RT_317.pdf.

[7] Honorato, D. D. F., Monard, M. C., Lee, H. D., and Wu, F. C. (2008b). Uma abordagem de extração de terminologia para a construção de uma representação atributo-valor a partir de documentos não-estruturados. In *Conferencia Latinoamericana de Informáica*, pages 190–199, Santa Fe, Argentina.

[8] Lee, H. D. (2005). Seleção de atributos importantes para a extração de conhecimento de bases de dados. Tese de Doutorado, ICMC-USP, http://www.teses.usp.br/teses/disponiveis/55/55134/tde-22022006-172219/.

[9] Pardo, T. A. S. (2006). Senter: Um segmentador sentencial automático para o português do brasil. Technical Report NILC-TR-06-01, ICMC-USP. http://www.icmc.usp.br/˜taspardo/NILCTR0601-Pardo.pdf.

[10] Ratnaparkhi, A. (1996). A maximum entropy model for part-of-speech tagging. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 133–142, Nova Jérsei, EUA.

[11] Rezende, S. O. (2003). *Sistemas Inteligentes: Fundamentos e Aplicações*. Editora Manole, São Paulo, Brasil.

[12] Schmid, H. (1994). Probabilistic part-of-speech tagging using decision trees. In *Proceedings of the International Conference on New Methods in Language Processing*, Manchester, Reino Unido.

[13] Schwartz, R., Christiansen, T., and Pyle, L. W. (1997). *Learning Perl*. California.

[14] Witten, I. and Frank, E. (2005). *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufman Publishers Inc., San Francisco, Califórnia, EUA.

Dados de Contato

Daniel de Faveri Honorato. Laboratório de Bioinformática, Universidade Estadual do Oeste do Paraná. Av. Presidente Tancredo Neves, 6731, 85866-900, Foz do Iguaçu, PR, Brasil. dfaverih@gmail.com.

Maria Carolina Monard. Laboratório de Inteligência Computacional, Instituto de Ciências Matemáticas e de Computação, Universidade de São Paulo. Av. Trabalhador São-carlense, 400, 13560-970, São Carlos, SP, Brasil. mcmonard@icmc.usp.br

Huei Diana Lee. Laboratório de Bioinformática, Universidade Estadual do Oeste do Paraná. Av. Presidente Tancredo Neves, 6731, 85866-900, Foz do Iguaçu, PR, Brasil. hueidianalee@gmail.com.

Carlos Andres Ferrero. Laboratório de Bioinformática, Universidade Estadual do Oeste do Paraná. Av. Presidente Tancredo Neves, 6731, 85866-900, Foz do Iguaçu, PR, Brasil. anfer86@gmail.com.

Feng Chung Wu. Laboratório de Bioinformática, Universidade Estadual do Oeste do Paraná. Av. Presidente Tancredo Neves, 6731, 85866-900, Foz do Iguaçu, PR, Brasil. wufengchung@gmail.com.