Descrição do projeto da metodologia de mapeamento de informações semi-estruturadas em uma representação atributo-valor *

Daniel de Faveri Honorato
Everton Alvares Cherman
Huei Diana Lee
Feng Chung Wu
Maria Carolina Monard

Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação e Estatística
Laboratório de Inteligência Computacional
Caixa Postal 668, 13560-970 - São Carlos, SP, Brasil
e-mail: {dfaverih@gmail.com, evertoncherman@gmail.com, huei@unioeste.br}

Resumo

O processo de Mineração de Dados auxilia na análise, compreensão e extração de conhecimento de conjunto de dados. Freqüentemente, esse processo requer que os dados encontrem-se armazenados em uma tabela atributo-valor. Neste trabalho é apresentado o projeto de uma metodologia desenvolvida para auxiliar na transformação de informações semi-estruturadas encontradas em laudos médicos em informações estruturadas representadas em tabela atributo-valor.

Palavras-Chave: Processamento de Texto, Extração de Informação, Aprendizado de Máquina.

Outubro 2007

*Trabalho desenvolvido com apoio da Fundação Parque Tecnológico Itaipu - FPTI - e do CNPq.

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas

BLOB *Binary Large Object*

CC Conjunto de Características

CFU Conjunto de Frases Únicas

CLD Conjunto de Locais Definidos

CLI Conjunto de Locais Indefinidos

CSC Conjunto de Subcaracterísticas

EDA Endoscopia Digestiva Alta

FC Ferramenta Computacional

LABI Laboratório de Bioinformática

LABIC Laboratório de Inteligência Computacional

LM Laudos Médicos

MD Mineração de Dados

SGBD Sistema Gerenciador de Banco de Dados

UML *Unified Modeling Language*

XML *Extensible Markup Language*

CAPÍTULO

1

Introdução

Com o avanço tecnológico, a quantidade de informações armazenadas digitalmente está cada vez maior. Para que possam ser realizadas análises mais completas, é necessário que essas informações sejam representadas de maneira apropriada, processadas e que um modelo que represente o conhecimento embutido nesses dados seja construído, uma vez que a análise manual é inviável. Um dos processos utilizados para realizar a análise de dados é a Mineração de Dados — MD —, cujo objetivo é identificar padrões válidos, novos, potencialmente úteis e compreensíveis embutidos em base de dados (Fayyad et al., 1996).

O processo de MD é interativo e iterativo. Ele é composto, basicamente, por três etapas: pré-processamento, extração de padrões e pós-processamento (Hand and Kamber, 2006; Weiss and Indurkhya, 1998). O pré-processamento é, freqüentemente, a etapa mais custosa, consumindo em torno de 80% do tempo usado para realizar o processo (Pyle, 1999). Essa etapa tem como objetivo realizar tarefas como preparação, redução e transformação dos dados. Ainda em pré-processamento, é necessário que os dados estejam representados no formato apropriado para a próxima etapa sendo um dos formatos mais comumente utilizados o atributo-valor, conforme ilustrado na Tabela 1.1.

Nessa tabela estão representados N exemplos compostos por M atributos. Cada exemplo e_i é um vetor $e_i = (v_{i1}, v_{i2}, ...v_{iM})$, no qual o valor v_{ij} refere-se ao valor associado ao j-ésimo atributo do exemplo i. Na tabela também pode estar associada a classe C a qual cada exemplo pertence.

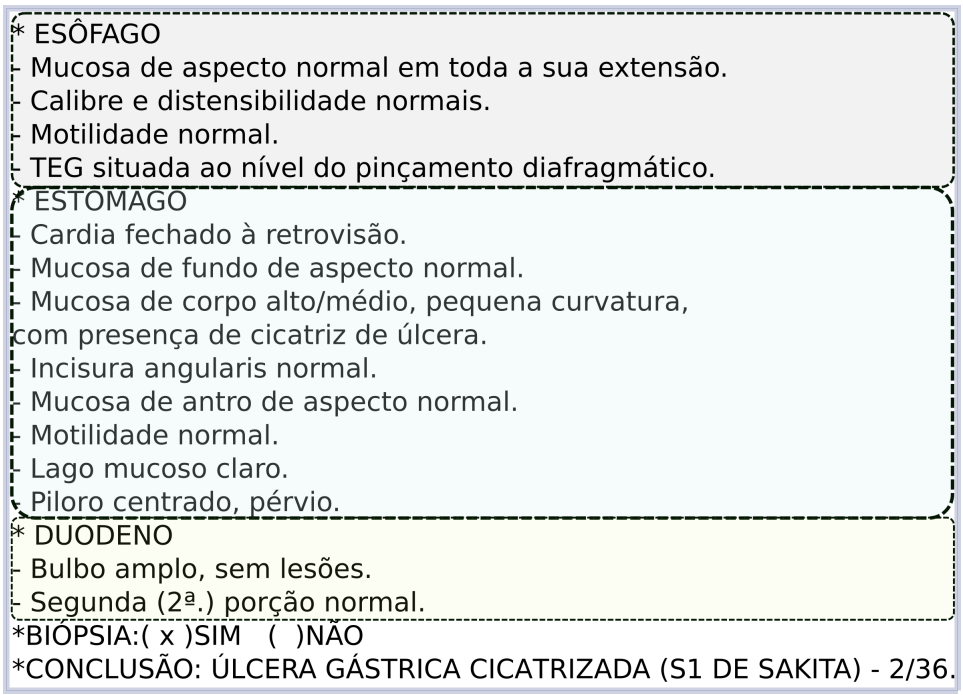
A etapa de extração de padrões tem como característica a configuração,

	atr_1	atr_2	...	atr_M	C
e_1	v_{11}	v_{12}	...	v_{1M}	c_1
e_2	v_{21}	v_{22}	...	v_{2M}	c_2
...	...	...	...	...	...
e_N	v_{N1}	v_{N2}	...	v_{NM}	c_N

Tabela 1.1: Tabela atributo-valor

a escolha e a execução de um ou mais algoritmos de extração de padrões sobre os dados selecionados na etapa de pré-processamento. Essa etapa é realizada de maneira iterativa, sendo necessário realizar diversos ajustes nos parâmetros dos algoritmos de extração de padrões utilizados, com o objetivo de construir modelos do conhecimento extraído dos dados pré-processados (Alpaydin, 2004; Mitchell, 1997). Após a extração de padrões, inicia-se a etapa de pós-processamento, na qual os modelos construídos são avaliados e validados. Depois de concluído o processo, o conhecimento extraído é disponibilizado ao usuário, o qual pode ser utilizado para auxiliar no processo de tomada de decisões.

No cenário atual de desenvolvimento tecnológico, hospitais e clínicas médicas registram cada vez mais informações sobre pacientes e resultados laboratoriais. Essas informações são freqüentemente armazenadas em Laudos Médicos — LM — semi-estruturados com informações descritas em língua natural. Nas Figuras 1.1 e 1.2 estão ilustrados LM de Endoscopia Digestiva Alta— EDA — e Colonoscopia, respectivamente.


* ESÔFAGO
- Mucosa de aspecto normal em toda a sua extensão.
- Calibre e distensibilidade normais.
- Motilidade normal.
- TEG situada ao nível do pinçamento diafragmático.

* ESTÔMAGO
- Cardia fechado à retrovisão.
- Mucosa de fundo de aspecto normal.
- Mucosa de corpo alto/médio, pequena curvatura,
com presença de cicatriz de úlcera.
- Incisura angularis normal.
- Mucosa de antro de aspecto normal.
- Motilidade normal.
- Lago mucoso claro.
- Piloro centrado, pérvio.

* DUODENO
- Bulbo amplo, sem lesões.
- Segunda (2ª.) porção normal.

*BIÓPSIA:(x)SIM ( )NÃO
*CONCLUSÃO: ÚLCERA GÁSTRICA CICATRIZADA (S1 DE SAKITA) - 2/36.

Figura 1.1: Laudo semi-estruturado de Endoscopia Digestiva Alta

No laudo da Figura 1.1, há uma divisão clara nas informações relacionadas ao esôfago, ao estômago, ao duodeno, a biopsia e a conclusão do exame. Nos três primeiros segmentos as informações estão descritas em língua natural. No último segmento, a informação referente à biópsia está estruturada, enquanto que a informação referente à conclusão do exame está descrita em língua natural. No laudo da Figura 1.2, de modo semelhante, uma parte do

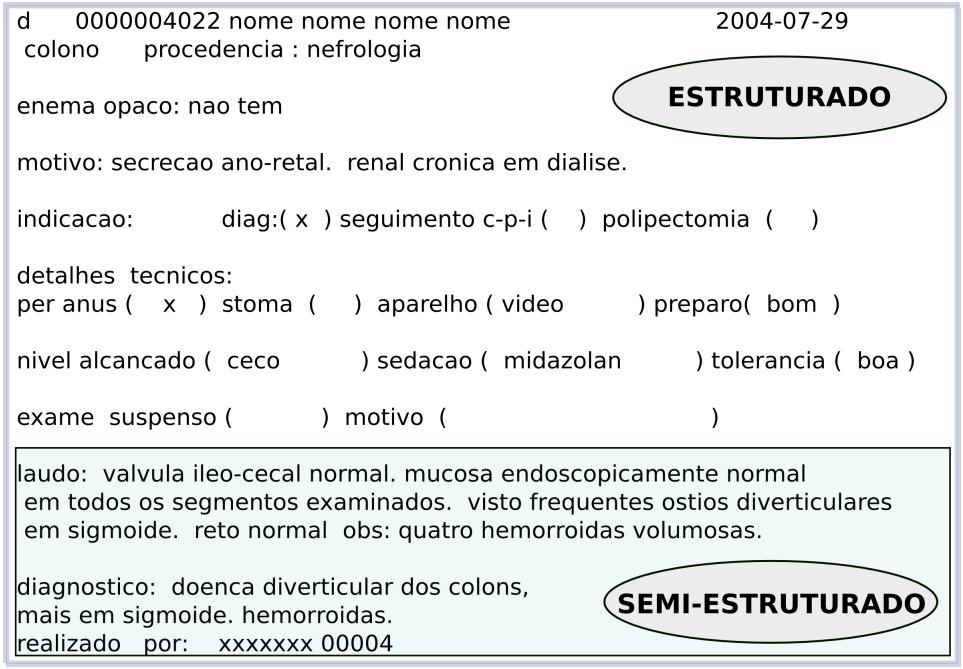


Figura 1.2: Laudo estruturado e semi-estruturado de Colonoscopia

laudo está estruturada e outra parte está semi-estruturada.

Conforme mencionado, para que possa ser realizada a MD, geralmente é necessário que as informações, sobre as quais serão aplicados os algoritmos de extração de padrões, estejam no formato atributo-valor. Portanto, é necessário que as informações armazenadas em LM sejam interpretadas[1] e transformadas para o formato utilizado por esses algoritmos. Esse processo, além de ter um custo de tempo elevado, está sujeito à interpretação subjetiva de quem o está realizando (Monard and Lee, 2003; Ferro et al., 2002). Desse modo, processos para auxiliar na semi-automatização dessa tarefa poderiam prover ganho em tempo, além da padronização no tratamento das informações contidas nos laudos médicos (Lee, 2005).

O objetivo deste trabalho é apresentar uma descrição detalhada da metodologia de transformação de laudos semi-estruturados em tabela atributo-valor (Honorato et al., 2005) e descrever a implementação, a qual foi desenvolvida

[1]Nesse contexto, a interpretação refere-se à identificação de padrões de relacionamento nas informações e não em uma análise profunda usando processamento de língua natural.

pelo LABI — Laboratório de Bioinformática da Universidade Estadual do Oeste do Paraná em parceria com o LABIC — Laboratório de Inteligência Computacional, do Instituto de Ciências Matemáticas e de Computação da Universidade de São Paulo.

Este trabalho está organizado da seguinte maneira: no Capítulo 2 são descritas as tarefas necessárias para a execução da metodologia. No Capítulo 3 é apresentada a ferramenta computacional proposta para auxiliar na aplicação da metodologia e no Capítulo 4 é apresentada a biblioteca de classes desenvolvida que implementa os algoritmos utilizados na implementação da metodologia. No Capítulo 5 são apresentadas as considerações finais do trabalho. Nos Apêndices A e B são apresentadas as interfaces da ferramenta computacional desenvolvida e exemplos de arquivos gerados pela ferramenta, respectivamente.

CAPÍTULO

2

Metodologia

A metodologia é composta por duas fases, ilustradas na Figura 2.1. A primeira fase caracteriza-se pela construção de um dicionário do domínio do conhecimento considerado, o qual é empregado para o processamento de laudos desse mesmo domínio durante a próxima fase. Na primeira fase, o auxílio de especialistas é de fundamental importância para o sucesso da construção do dicionário. Na segunda fase, o dicionário é utilizado para a transformação de laudos médicos desse domínio, por meio de casamento de padrões, para a construção da tabela atributo-valor. Para exemplificar a aplicação da metodologia, são utilizadas informações contidas em laudos de Endoscopia Digestiva Alta, especificamente informações relacionadas ao esôfago e ao estômago, os quais foram utilizadas como base para o desenvolvimento da metodologia descrita neste trabalho. Os arquivos do tipo *XML*[1] necessários para a aplicação da metodologia, descritos neste capítulo, podem ser construídos por meio da ferramenta computacional apresentada no Capítulo 3.

2.1 Primeira fase

A construção do dicionário é realizada por meio das seguintes quatro etapas iterativas e interativas:

1. identificação de frases únicas;
2. construção de arquivo de padronização;

[1] eXtensible Markup Language, http://www.w3.org/XML

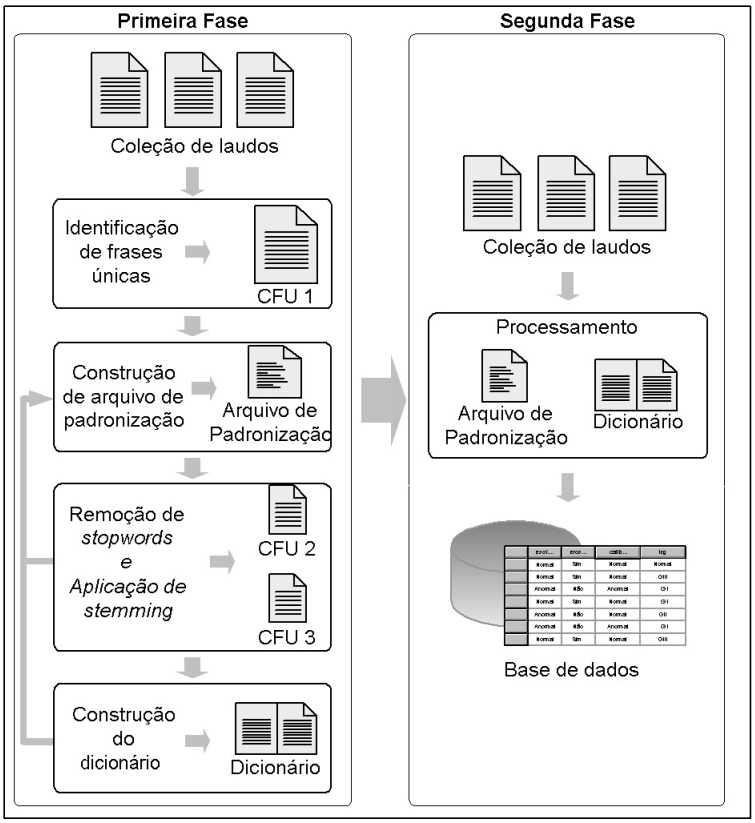


Figura 2.1: Metodologia inicial proposta para o tratamento de dados semi-estruturados (Honorato et al., 2005)

3. remoção de *stopwords* e aplicação de *stemming*; e

4. construção do dicionário.

O objetivo das três primeiras etapas, descritas a seguir, é auxiliar no processo de identificação dos padrões de escrita contidos nos laudos para que esses laudos possam ser mapeados para o dicionário.

2.1.1 Identificação de frases únicas

Consiste na identificação de frases únicas existentes na coleção de laudos utilizada para a construção da base de dados. Os laudos podem estar armazenados, dependendo do domínio que está sendo trabalhado, em diferentes repositórios, por exemplo, em um Sistema Gerenciador de Banco de Dados — SGBD — (Date, 2000), em arquivos binários ou mesmo arquivos texto. Nesse caso, cada arquivo correspondente a um laudo. Desse modo, é necessário criar procedimentos para encontrar as frases únicas específicas para cada tipo de repositório. Para exemplificar, é apresentado o procedimento realizado para a construção de frases únicas para o domínio de EDA.

Os laudos de EDA estavam armazenados no SGBD Interbase[2], especificamente em um atributo do tipo BLOB[3]. Desse modo, para a construção de frases únicas, foi desenvolvido um *script*, denominado **ordena.pl**, o qual tem por objetivo realizar automaticamente a extração de laudos do SGBD e reunir a lista de frases da coleção de laudos. Nos laudos de EDA, conforme ilustrado na Figura 1.1, as informações estão dispostas nos segmentos esôfago, estômago, duodeno e conclusões do exame. Em cada segmento, por sua vez, as informações estão mapeadas por meio de frases, nos quais cada frase refere-se a um diagnóstico, um prognóstico ou uma observação do médico sobre o exame realizado. Desse modo, neste trabalho decidimos criar listas de frases únicas relacionadas a cada segmento do laudo, uma vez que foi decidido processar cada segmento do laudo separadamente. O *script* **ordena.pl**, portanto, efetua a conexão com o SGBD e, iterativamente, para todos os laudos, extrai a informação de cada segmento, por exemplo, esôfago e estômago, e insere em uma lista de frases correspondentes, conforme ilustrado na Figura 2.2.

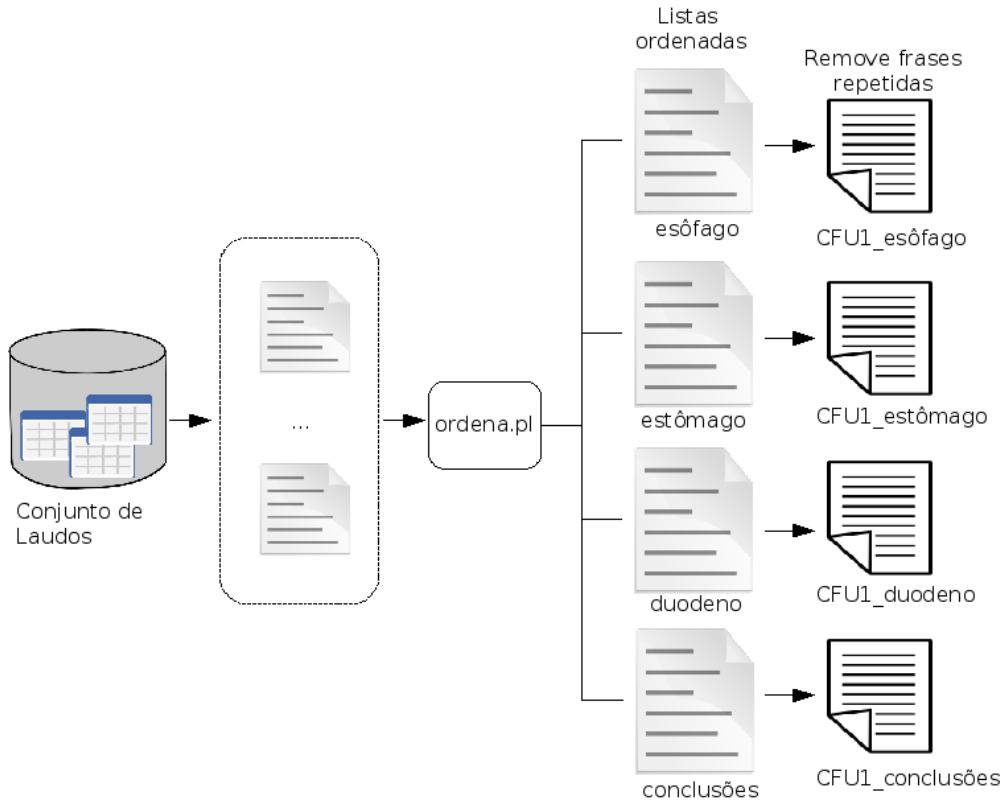


Figura 2.2: Identificação de frases únicas

Depois de construir a lista de frases correspondentes a cada segmento,

[2] http://www.borland.com/br/products/interbase/index.html

[3] O BLOB (Binary Large Object - grande objeto binário) é um campo criado para o armazenamento de qualquer tipo de informações em formato binário, dentro de uma tabela de um banco de dados.

o *script* **ordena.pl** permite realizar a ordenação de cada uma das listas, de modo que as frases iguais fiquem juntas. Depois, as listas ordenadas são armazenadas em arquivos, por exemplo, a lista de frases do esôfago é armazenada no arquivo **esôfago.txt** e, manualmente, com o apoio de um editor de textos, as frases que se repetem no arquivo são removidas, deixando apenas um exemplar de cada frase. No final do processo, tem-se o conjunto de frases únicas denominado CFU1, de cada segmento. Na Figura 2.3 é apresentado um fragmento do CFU1 do segmento do estômago.

- Operado, </- 50%, com grande quantidade de resíduos alimentares, dificultando a avaliação
- Operado, coto gástrico < 50 %, com enantema
- Operado, coto gástrico </= 50 %
- Operado, coto gástrico < 50%, com enantema
- Lago mucoso tinto de bile
- Alça percorrida pérvia,sem lesões
- Antro aparentemente normal
- Antro com distensibilidade pouco diminuída, e aspecto infiltrado, com presença de lesões ulceradas
- Antro com entantema
- Antro com presença de erosões planas e piloro sem alterações
- Antro com áreas de enantema e piloro sem alterações
- Boca anastomótica ampla, pérvias, sem lesões
Boca anastomótica ampla, sem lesões
- Boca anastomática ampla, com áreas de metaplasia intestinal
- Boca anastomótica ampla, com cicatriz de úlcera
- Boca anastomótica ampla, com enantema e áreas de metaplasia intestinal
- Boca anastomótica ampla, com área hiperemiada (cicatriz)
- Boca anastomótica ampla, com áreas de metaplasia
- Boca anastomótica ampla, com áreas de metaplasia intestinal
- Boca anastomótica ampla, sem lesões
Boca anastomótica ampla, sem lesões, com presença de fio de sutura
- Boca anastromótica ampla, sem lesões
- Boca anastómotica ampla, sem lesões com presença de fios de sutura
- Boca anostomótica ampla, com presença de lesão ulcerada, de aproximadamente 10 mm, rasa, com fibrina
- Cardia aberto com presença de resíduos alimentares dificultando a avaliação
- Cardia aberto a retrovisão
- Cardia entreaberto à retrovisão
Cardia fechada à retrovisão
- Com presença de resíduos alimentares, dificultando a avalição
- Corpo aparentemente normal
- Coto gástrico com enantema

Figura 2.3: Fragmento do CFU do segmento do estômago

2.1.2 Construção dos arquivos de padronizações

Para que seja possível trabalhar com um vocabulário controlado, além de permitir que as informações estejam em um formato adequado para serem utilizadas na construção do dicionário, devem ser construídos os arquivos de padronizações. Por exemplo, o termo *coloração esbranquiçada*, geralmente utilizado para descrever a característica de um tecido biológico, é sinônimo de que o tecido está *anormal*. Um outro fato freqüente é o médico inserir mais de um evento em uma única sentença. Por exemplo, a frase *calibre e distensibilidades normais*, representa a descrição de dois eventos, ou seja, essa frase indica que *calibre* está *normal* e *distensibilidade* está *normal*. Portanto, no

arquivo de padronização deverá ser mapeada a transformação dessa frase de modo a aparecer, depois da padronização, apenas um evento em cada frase, por exemplo, *calibre normal* e *distensibilidade normal*.

Nesta etapa, portanto, é iniciada com a análise de CFU1, a construção dos arquivos de padronizações, os quais permitem trabalhar com um vocabulário controlado na construção do dicionário e também auxiliam na padronização de frases com mais de um evento. Os arquivos de padronização são construídos a partir da análise do CFU1, juntamente com os especialistas, e devem ser do tipo *XML* com uma estrutura pré-definida. Dois arquivos são utilizados para mapear os padrões identificados, sendo um arquivo para mapear os sinônimos e outro arquivo para mapear os casos em que aparecem mais de um evento por frase. Nas Figuras 2.4 e 2.5 são apresentados exemplos desses arquivos, ilustrando os dois tipos de mapeamento de padrões permitidos.

```
<pattern number="1" >
 <synonym>
  <new>terco_distal</new>
  <old>40\s*cm\s*[da]*\s*ads</old>
 </synonym>
</pattern>
```

Figura 2.4: Exemplo de padronização de sinônimos

```
<patterntwo number="1" >
  <synonymtwo n="2" >
    <old>calibre distensibilidade normais</old>
    <new>calibre normal</new>
    <new>distensibilidade normal</new>
  </synonymtwo>
</patterntwo>
```

Figura 2.5: Exemplo de padronização de mais de um evento

Na estrutura apresentada na Figura 2.4, o atributo *number* indica o número de sinônimos existentes no arquivo. Os sinônimos são mapeados utilizando os marcadores <*old*> e <*new*>, os quais sempre aparecem juntos dentro de um nó <*synonym*>. O primeiro corresponde à palavra que será padronizada e o segundo corresponde a nova palavra. Na estrutura da Figura 2.5 são mapeados os casos onde aparecem mais de um evento em uma frase. Também são utilizados os marcadores <*old*> e <*new*> dentro do nó <*synonymtwo*>. Porém, neste caso podem existir mais de um marcador <*new*> associado a um <*old*>. De modo semelhante o marcador <*old*> corresponde à frase com mais de um evento e os marcadores <*new*> correspondem à divisão da frase com mais de um evento em várias frases com apenas um evento.

Um recurso importante que pode ser utilizado nos arquivos de padronizações é o uso de expressões regulares, pois durante a análise do CFU podem

ser identificados algumas características de escrita nas frases, como espaços excedentes entre palavras, ou mesmo a existência de termos que ocorrem raramente e que devem ser levados em consideração no momento da padronização. Por exemplo, no padrão `40\s*cm\s*[da]*\s*ads` da Figura 2.4, a expressão regular `\s*` indica que no local onde está mapeada podem ocorrer zero ou mais espaços. A expressão regular `[da]*` indica que pode existir mais de uma vez o termo `da`. Portanto, durante a construção dos arquivos de padronizações, deve ser analisada a necessidade de utilização de expressões regulares nos padrões mapeados.

2.1.3 Remoção de stopwords e aplicação de stemming

Nessa etapa são removidas do CFU1 palavras consideradas irrelevantes (*stopwords*) para análise do texto, as quais encontram-se em uma *stoplist*. Essa lista é formada por um conjunto de *stopwords* do tipo artigos, preposições, conjunções, além de algumas palavras consideradas irrelevantes pelo especialista. As *stopwords* são armazenadas cada uma em uma linha do arquivo **stopwords.txt**. Para a remoção de *stopwords* dos conjuntos de frases únicas é utilizado o *script* **sstopwords.pl**. Nesse *script*, deve ser configurado o nome do arquivo de *stopwords* e o local onde se encontra a pasta com o arquivo de frases únicas sobre o qual será aplicado o algoritmo de remoção de *stopwords*. Na Figura 2.6 é ilustrado o processo de remoção de *stopwords*.

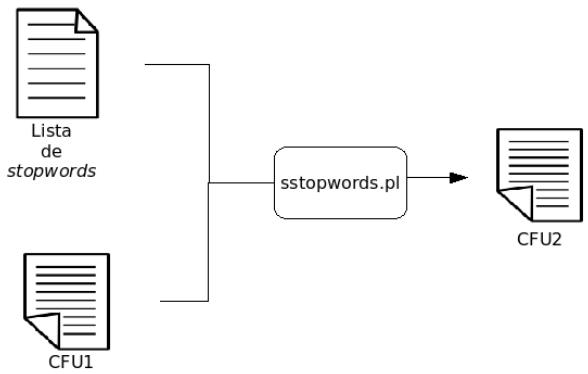


Figura 2.6: Processo de remoção de *stopwords* do CFU1

Sobre o CFU2 construído é aplicado o processo de *stemming*, com o intuíto de remover frases redundantes, uma vez que muitas frases diferem apenas pela variação de uma palavra, por exemplo, *terço distal com erosão* e *terço distal com erosões*. O algoritmo de *stemming* utilizado é baseado no método de **Porter** (Porter, 1980). Na Figura 2.7 é ilustrado o processo de aplicação de *stemming*.

Figura 2.7: Processo de aplicação de *stemming* sobre CFU2

2.1.4 Construção da estrutura da tabela atributo-valor e do dicionário

A estrutura da tabela atributo-valor e do dicionário são definidas com base na forma com que as informações estão dispostas nos laudos. Para identificar qual o formato de distribuição das informações, devem ser analisados os conjuntos de frases únicas gerados. Para exemplificar como é realizada a análise da distribuição das frases, é apresentado como foi realizada a análise das informações presentes nos laudos do domínio de EDA, na qual foram analisadas as informações dos segmentos do esôfago e estômago.

As informações contidas nos laudos de EDA estão representadas por meio de frases e estão geralmente organizadas na forma de estrutura anatômica (local), características associadas a essa estrutura e subcaracterísticas associadas às características. Por exemplo, na frase *antro com úlcera de aproximadamente 5mm*, *antro* é o local, *úlcera* é a característica e *5mm* é a subcaracterística associada a úlcera. Quanto à distribuição das palavras nas frases, é indispensável a presença de um local e de uma característica, caso contrário, as informações contidas na frase não representam nenhum valor de atributo que poderá ser preenchido na tabela. Como a subcaracterística é a especificação de uma característica, sua presença não é imprescindível, porém pode tornar a descrição do que está sendo apresentado na frase mais detalhada e, consequentemente, serão criados mais atributos na tabela atributo-valor. Como resultado da análise das frases, foram identificados os seguintes padrões na disposição das informações:

- Para toda característica é necessário um ou mais locais descritos anteriormente, de modo que todos os locais consecutivos encontrados imediatamente antes compartilham dessa característica;
- Uma característica somente pode ser a especificação de um ou mais locais consecutivos encontrados posteriormente, se não houver nenhuma característica após ou antes desses locais;

- Toda subcaracterística descrita é precedida de uma característica, as quais estão relacionadas.

Na Figura 2.8 é apresentada uma frase que exemplifica os padrões descritos anteriormente.

Frase original	"Mucosa de antro com presença de erosões planas e úlcera pré-pilórica, de aproximadamente 06 mm, média profundidade"
Padronizada	"antro erosao plana ulcera pre-pilorica 06mm media_profundidade"
Classificada	"L C SC C L SC SC" (L = Local; C = Característica; SC = SubCaracterística)
Atributo e valor	antro_erosao = sim antro_erosao_plana = sim antro_ulcera = sim antro_ulcera_extensao = 06mm antro_ulcera_profundidade = media pre-pilorica_ulcera = sim pre-pilorica_ulcera_extensao = 06mm pre-pilorica_ulcera_profundidade = media

Figura 2.8: Padronização e classificação de uma frase original

Dessa maneira, os locais "antro" e "pré-pilórica" compartilham a característica "úlcera", sendo a característica "erosões" pertencente somente ao "antro". A subcaracterística "planas" refere-se às "erosões", e as subcaracterísticas "06mm" e "média profundidade" à "úlcera".

Depois de identificada a distribuição das informações, são analisadas, em conjunto com os especialistas, cada frase de CFU3 e do arquivo de padronização, identificando a relação local-característica-subcaracterística a partir da qual são criados os atributos e definida a estrutura do dicionário. Na implementação dessa metodologia a definição da estrutura da tabela é realizada em duas etapas. A primeira é a construção da tabela em um SGBD. Neste trabalho foi adotado o SGBD MySql[4]. A segunda etapa consiste na construção de um arquivo *XML*, o qual é consultado pelos algoritmos que implementam a metodologia para identificar os atributos da tabela atributo-valor. Na Figura 2.9 é ilustrado um arquivo exemplo constituído de dois atributos. No arquivo, cada atributo da tabela é inserido em um marcador <*attribute*>.

```
<attributes number="2" >
  <attribute>calibre</attribute>
  <attribute>distensibilidade</attribute>
</attributes>
```

Figura 2.9: Arquivo de atributos

O dicionário pode ser construído paralelamente e é composto por uma estrutura hierárquica de três níveis: locais, características e subcaracterísticas, conforme ilustrado na Figura 2.11.

[4] http://dev.mysql.com/doc/refman/5.0/en/

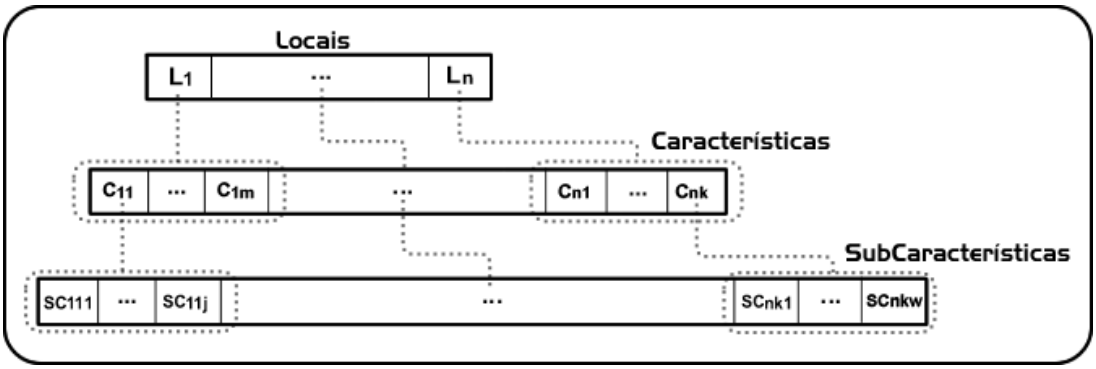


Figura 2.10: Estrutura de dados para representar o dicionário

Nessa estrutura, há uma lista de locais e cada local (L_i) possui uma lista de uma ou mais características associadas (C_{ia}), onde $i = 1, 2, ..., n; a = 1, 2, ..., k$; n é o número total de locais e k é o número total de características. A lista de características, por sua vez, é uma lista de uma ou mais subcaracterísticas associadas (SC_{iab}), onde $b = 1, 2, ...w$ e w é o número total de subcaracterísticas. Na lista de características e subcaracterísticas, são também armazenadas informações relacionadas ao atributo da tabela que será preenchido e o valor que será preenchido nesse atributo. As informações do dicionário são também mapeadas em arquivos texto do tipo *XML*. Na Figura 2.11 é ilustrado um arquivo *XML* no qual está mapeada a frase *antro com ulcera de 5mm*, onde local=*antro*, característica=*úlcera* e subcaracterística=*5mm*.

```
<dictionary number="1" >
  <condition>
    <local>antro</local>
    <AllCharacteristic numberc="1" >
      <characteristics>
        <attribute>
          <attributeName>antro_ulcera</attributeName>
          <position>0</position>
          <valueToSave>sim</valueToSave>
        </attribute>
        <characteristicName>com ulcera</characteristicName>
      </characteristics>
      <allSubCharacteristic numbersc="1" >
        <subcharacteristics>
          <attribute>
            <attributeName>antro_ulcera_profundidade</attributeName>
            <position>0</position>
            <valueToSave>5mm</valueToSave>
          </attribute>
          <subcharacteristicName>5mm</subcharacteristicName>
        </subcharacteristics>
      </allSubCharacteristic>
    </AllCharacteristic>
  </condition>
</dictionary>
```

Figura 2.11: Estrutura do dicionário

Os marcadores

<*attributeName*>antro_ulcera</*attributeName*>

<*attributeName*>antro_ulcera_profundidade</*attributeName*>

indicam o atributo da tabela que deverão ser preenchidos e os marcadores

<*valueToSave*>sim</*valueToSave*>
<*valueToSave*>5mm</*valueToSave*>

indicam os valores que deverão ser preenchidos nos respectivos atributos citados.

2.2 Arquivo de parâmetros

Antes que seja iniciada a segunda fase, o usuário deve configurar um arquivo de parâmetros o qual tem como função armazenar informações necessárias para os algoritmos que implementam a metodologia tais como tarefas a serem executadas e diretórios onde estão os arquivos necessários para aplicação da metodologia, como o do dicionário e dos atributos. O arquivo de parâmetros é do tipo *XML* e seus marcadores são descritos em detalhes a seguir.

- **applyNormalize**: Indica se o conteúdo dos laudos deve ser transformado para minúsculo e se devem ser removidos os acentos;
  Valores possíveis: `sim` ou `não`;
  Valor *default*: `sim`;
  Consultado por: **TControl.pm**

- **applyStopwords**: Indica se devem ser removidas *stopwords* dos laudos;
  Valores possíveis: `sim` ou `não`;
  Valor *default*: `sim`;
  Consultado por: **TControl.pm**

- **applyStemming**: Indica se deve ser aplicado *stemming* sobre o conteúdo dos laudos;
  Valores possíveis[5]: `sim` ou `não`;
  Valor *default*: `não`;
  Consultado por: **TControl.pm**

- **applyPattern**: Indica se devem ser aplicas as padronizações nos laudos com base nos arquivos de padronizações criados;
  Valores possíveis: `sim` ou `não`;

[5]Se for definido para aplicar *stemming*, o dicionário e o arquivo de padronização deverão ser construídos considerando que os laudos estarão apenas com o *stem* das palavras.

Valor *default*: `sim`;
Consultado por: **TControl.pm**

- **dirStopWord**: Indica o caminho onde está o arquivo de *stopwords*;
  Consultado por: **TControl.pm**
- **dirDocuments**: Indica o caminho onde estão os laudos a serem processados;
  Consultado por: **TControl.pm**
- **dirDictionary**: Indica o caminho do arquivo do dicionário contruído;
  Consultado por: **TControl.pm**
- **dirAttributes**: Indica o caminho do arquivo que contém a lista de atributos da tabela;
  Consultado por: **TControl.pm**
- **dirPatterns**: Indica o caminho dos arquivos de padronizações;
  Consultado por: **TControl.pm**
- **dirOutput**: Indica o caminho onde serão colocados os laudos processados de acordo com as tarefas especificadas no arquivo de parâmetros;
  Consultado por: **TControl.pm**

2.3 Segunda fase

O objetivo dessa fase é, com base no dicionário construído, no arquivo de padrões, na lista de atributos e no arquivo de parâmetros, processar a coleção de laudos e preencher a tabela atributo-valor. Nessa fase, uma vez que os arquivos necessários estão criados, não há interação do usuário, ou seja, o processo é realizado automaticamente, utilizando os arquivos que foram criados na primeira fase. Durante o processamento dos laudos, as palavras que não foram mapeadas na estrutura do dicionário e, portanto, não são reconhecidas, são inseridas em um arquivo de palavras não processadas para serem analisadas posteriormente.

Cada laudo corresponde a um registro na tabela atributo-valor. A Figura 2.12 ilustra o processo realizado para a transformação das informações. Esse processo é aplicado da seguinte maneira: primeiramente é consultado o arquivo de parâmetros para verificar quais tarefas de pré-processamento do laudo devem ser realizadas (normalização, remoção de *stopwords*, aplicação de *stemming* e padronização) e as tarefas configuradas para serem realizadas são executadas. Em seguida, é aplicado o algoritmo de busca e preenchi-

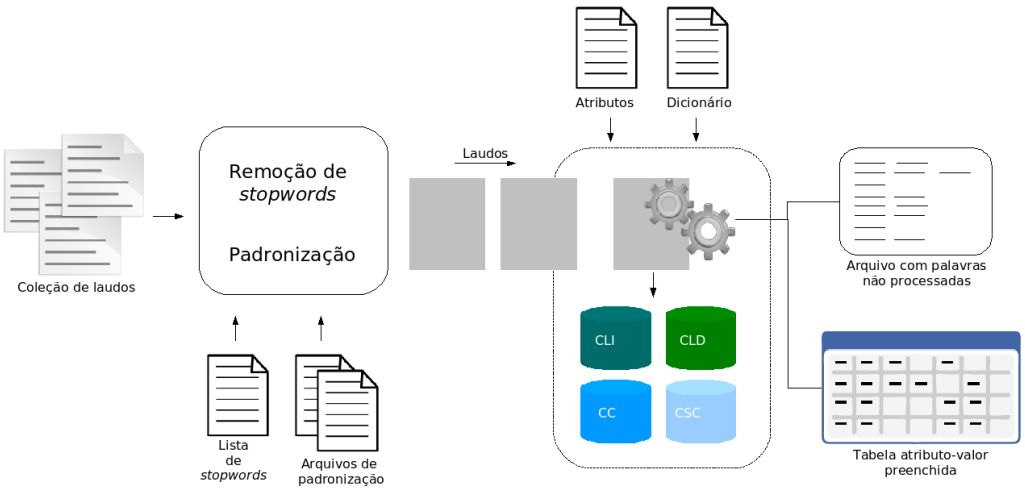


Figura 2.12: Processo realizado na segunda fase

mento, o qual realiza o processamento de uma frase do LM de cada vez. A Figura 2.13 ilustra o fluxograma que representa o algoritmo executado para o processamento da frase.

Essa frase é avaliada da esquerda para a direita e, utilizando as informações mapeadas na estrutura do dicionário, são identificadas as categorias das palavras, ou seja, se a mesma é um local, uma característica ou uma subcaracterística. Se a palavra não pertencer a nenhuma dessas categorias, o algoritmo a insere em um arquivo de palavras não processadas para ser analisada posteriormente. Para auxiliar na identificação de relações entre local, característica e subcaracterística, definiu-se quatro estruturas auxiliares, as quais são utilizadas para memorizar os locais, as características e as subcaracterísticas identificadas durante o processamento inicial da frase. As quatro estruturas são: Conjunto de Locais Indefinidos — CLI, Conjunto de Locais Definidos — CLD, Conjunto de Características — CC — e Conjunto de Subcaracterísticas — CSC. Toda característica e subcaracterística detectada na frase é armazenada no CC e CSC respectivamente. O CLI tem como objetivo armazenar os locais identificados na frase até o momento em que é identificada uma característica ou subcaracterística. Se for identificada uma característica, os locais serão enviados para o CLD e se relacionarão com a característica encontrada. Se for identificada uma subcaracterística, os locais também serão enviados para o CLD, mas se relacionarão com a característica que o precede. Com essas informações identificadas é preenchida a tabela atributo-valor. O preenchimento é realizado por meio de consultas no dicionário construído, nas quais são relacionados os locais de CLD e as características de CC, e relacionadas às características de CC com as subca-

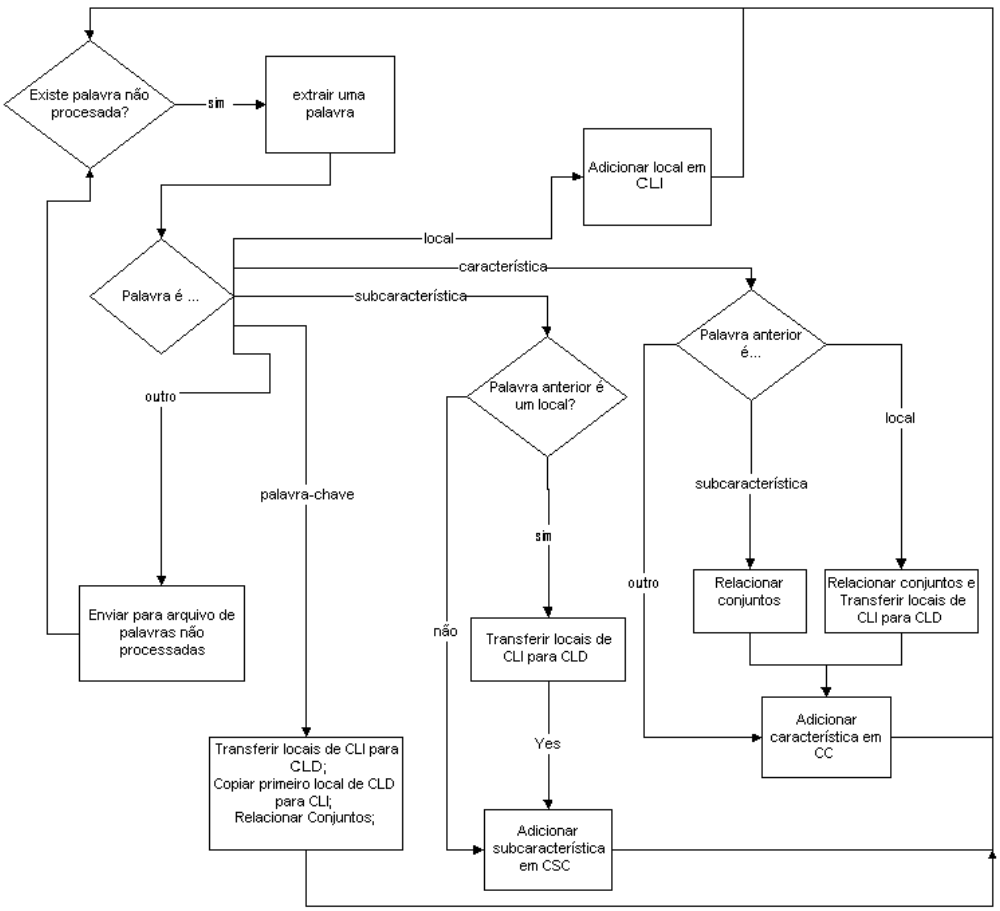


Figura 2.13: Algoritmo de processamento de frase

racterísticas de CSC. As consultas permitem identificar os atributos da tabela que deverão ser preenchidos e quais valores deverão ser atribuidos a esses atributos. Também foi definida uma lista de palavras que ao serem descritas auxiliam o algoritmo de busca e preenchimento a executar ações que possibilitam o mapeamento correto da frase. Essas palavras são denominadas de palavras-chaves e estão descritas no arquivo **keywords.txt**. O processo descrito é realizado para todas as frases de cada laudo e para todo o conjunto de laudos. Ao final do processo tem-se a tabela preenchida com os padrões identificados nos laudos.

CAPÍTULO

3

Ferramenta Computacional

Para auxiliar na aplicação da metodologia foi proposta uma Ferramenta Computacional — FC —, a qual tem por objetivo facilitar a construção dos arquivos utilizados para a aplicação da metodologia descrita no Capítulo 2, por meio de um ambiente amigável para inserção das informações, além de possibilitar uma visão conjunta das informações que são utilizadas na aplicação da metodologia. Essa visão conjunta das informações é importante uma vez que, na construção do dicionário, por exemplo, é necessário saber quais padrões foram mapeados e qual é a lista de atributos da tabela.

Na Figura 3.1 é ilustrado o processo de aplicação da metodologia utilizando a FC. Conforme pode ser observado na figura, a partir dos conjuntos de frases únicas CFU1, CFU2 e CFU3, o usuário realiza a construção dos arquivos na FC. Depois de gerados os arquivos necessários, os mesmos são utilizados pelos algoritmos da metodologia proposta para o preenchimento dos padrões identificados nos laudos na tabela atributo-valor.

3.1 Funcionamento da ferramenta

Como mencionado, para aplicar a metodologia devem ser construídos os arquivos do dicionário, o arquivo de padronização, a lista de atributos e a lista de *stopwords*. Na FC, a qual possui a interface principal ilustrada na Figura 3.2, primeiramente deve ser construído o arquivo de projeto, no qual são armazenadas informações tais como diretório do projeto, descrição do domínio que está sendo trabalhado e nome dos arquivos que armazenam as

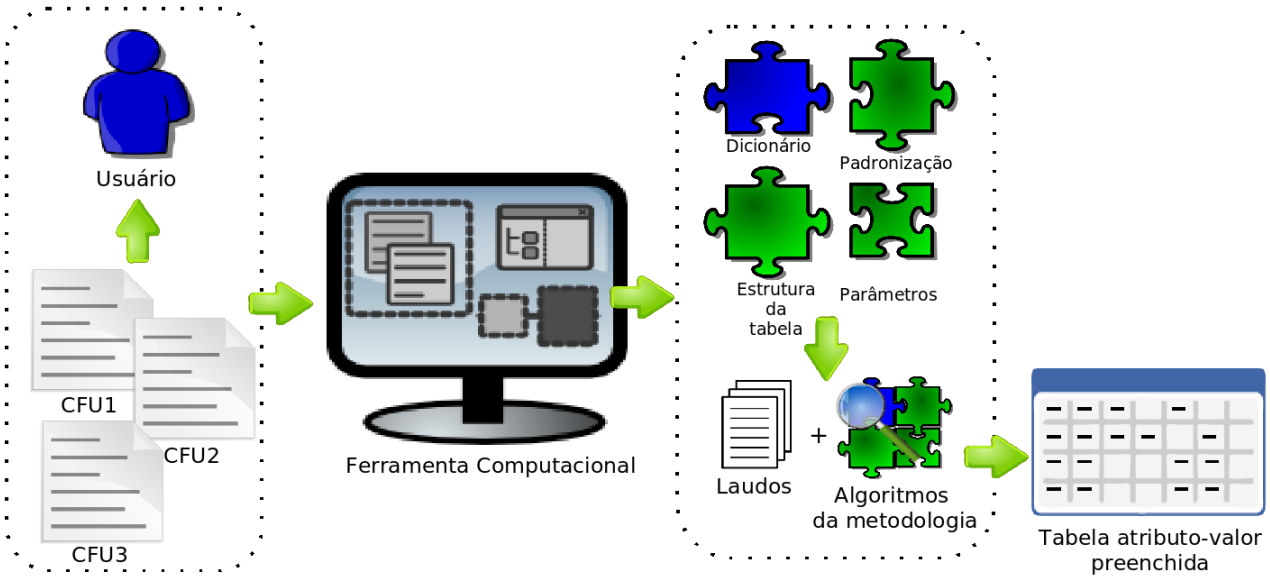


Figura 3.1: Processo de aplicação da metodologia utilizando a ferramenta computacional

informações utilizadas pelos algoritmos da metodologia.

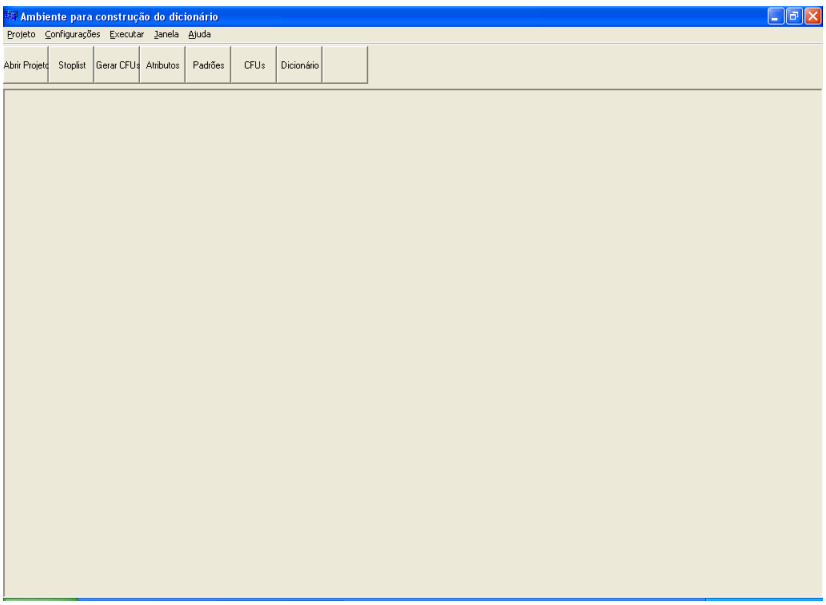


Figura 3.2: Interface principal da ferramenta computacional

Na Figura 3.3 é ilustrada a interface de construção de um novo projeto.

As informações preenchidas nessa interface são armazenadas em um arquivo do tipo *XML*, conforme ilustrado na Figura 3.4. Os nomes de arquivos com extensão *xml*, ilustrados no arquivo da Figura 3.4 são utilizados como *default*. O nome dos arquivos podem ser modificados, dependendo da escolha do usuário.

Quando o usuário necessitar abrir um projeto, deve referenciar o arquivo

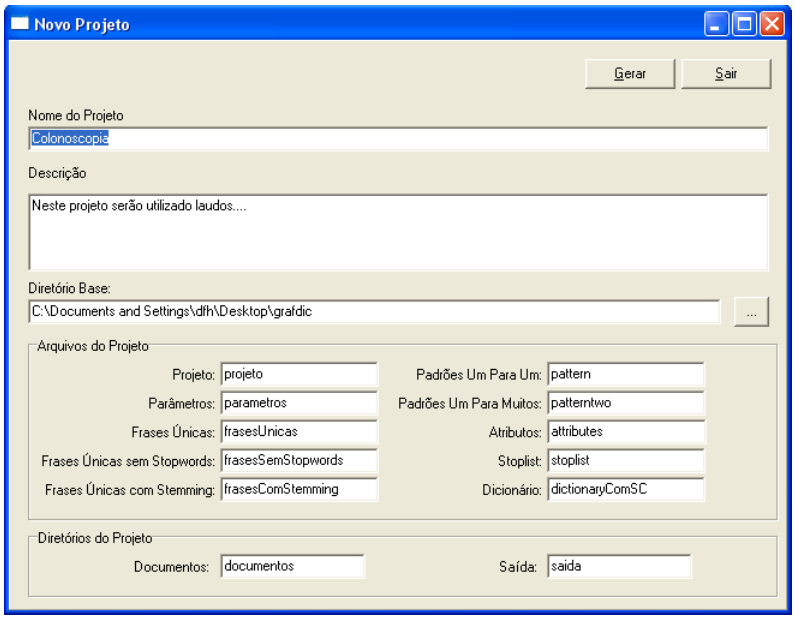


Figura 3.3: Interface de construção de novo projeto

```
<parametros>
  <nome>Colonoscopia</nome>
  <descricao>Neste projeto serão utilizados
             laudos de colonoscopia...</descricao>
  <dirBase>[DIRBASE]</dirBase>
  <arquivoProjeto>projeto</arquivoProjeto>
  <arqParametros>[DIRBASE]\parametros.xml</arqParametros>
  <arqStoplist>[DIRBASE]\stoplist.xml</arqStoplist>
  <dirDocumentos>[DIRBASE]\documentos.xml</dirDocumentos>
  <arqDicionario>[DIRBASE]\dictionaryComSC.xml</arqDicionario>
  <arqAtributos>[DIRBASE]\attributes.xml</arqAtributos>
  <arqPadroes1>[DIRBASE]\pattern.xml</arqPadroes1>
  <arqPadroes2>[DIRBASE]\patterntwo.xml</arqPadroes2>
  <dirSaida>[DIRBASE]\saida.xml</dirSaida>
</parametros>
```

Figura 3.4: Arquivo *xml* de projeto

do projeto criado e o sistema realiza a leitura dos arquivos que foram mapeados no arquivo de projeto carregado[1]. Depois de construído o arquivo de projeto, já pode ser iniciada a construção dos arquivos necessários para a aplicação da metodologia. Na Figura 3.5 são ilustrados os arquivos que podem ser gerados na FC.

Para a construção dos arquivos, o usuário interage com a interface da FC e as informações que estão sendo tratadas são armazenadas em arquivos *XML*. A seguir é apresentada uma descrição sucinta de cada arquivo que é mapeado utilizando a FC. No Apêndice B são ilustrados exemplos de cada um desses arquivos.

Parâmetros **parameters.xml**: Esse é um arquivo de configuração no qual são mapeadas informações relacionadas às tarefas que devem ser realizadas

[1]O sistema não gera os arquivos *XML* utilizados para armazenar as informações. A estrutura desses arquivos já deve ter sido previamente criada seguindo as definições apresentadas no Capítulo 2 e usando um editor de texto. A FC irá apenas gerenciar essa estrutura, isto é, possibilitar a inserção e remoção de informações.

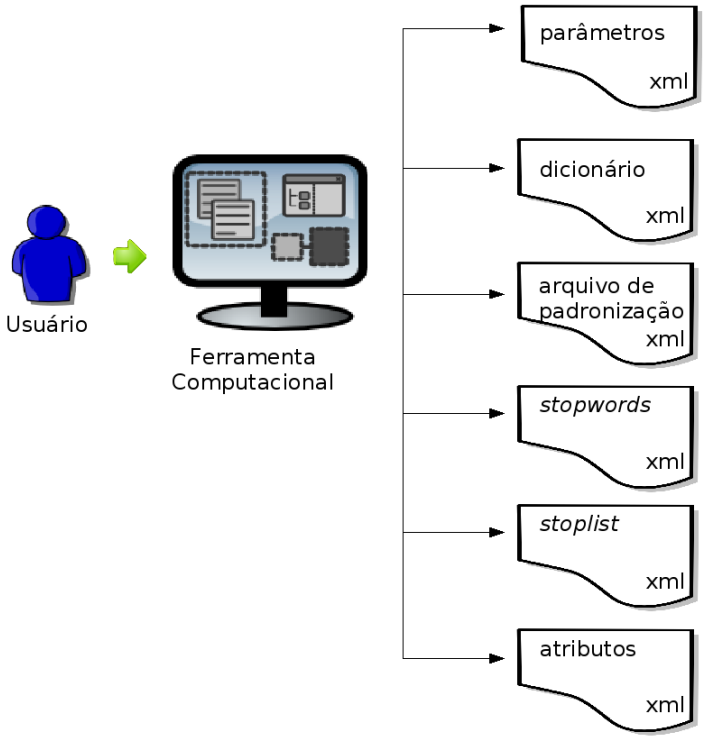


Figura 3.5: Arquivos gerados pela ferramenta computacional

durante o pré-processamento dos laudos. Algumas das tarefas que podem ser configuradas para serem realizadas são: remoção de *stopwords*, aplicação de *stemming* e aplicação de padronização.

Dicionário **dictionary.xml**: Nesse arquivo são mapeados os padrões de relacionamento de informações identificados nos CFUs, por meio da estrutura hierárquica de locais, características e subcaracterísticas.

Padrões **pattern.xml**: Nesse arquivo são mapeadas as padronizações de sinônimos.

Padrões **patterntwo.xml**: Nesse arquivo são mapeadas os casos nos quais existe mais de um evento em cada frase.

Stoplist **stoplist.xml**: Nesse arquivo são mapeados os endereços de onde se encontram os arquivos de *stopwords*.

Stopwords **stopword.xml**: O usuário pode mapear diversos arquivos de *stopwords* os quais devem ser do tipo *XML*.

Atributos **atributos.xml**: Nesse arquivo são mapeados os atributos da tabela atributo-valor, na qual serão inseridas as informações durante a aplicação da metodologia.

3.2 Projeto do sistema

Nesta seção é apresentado o projeto da FC. O objetivo desse seção é fornecer uma visão geral de como a ferramenta foi implementada. Na construção da ferramenta foi utilizado o paradigma de orientação a objetos na linguagem C++ (Stroustrup, 1997). Na Figura 3.6 é apresentado o diagrama de classes da ferramenta desenvolvida e a descrição de cada uma das classes.

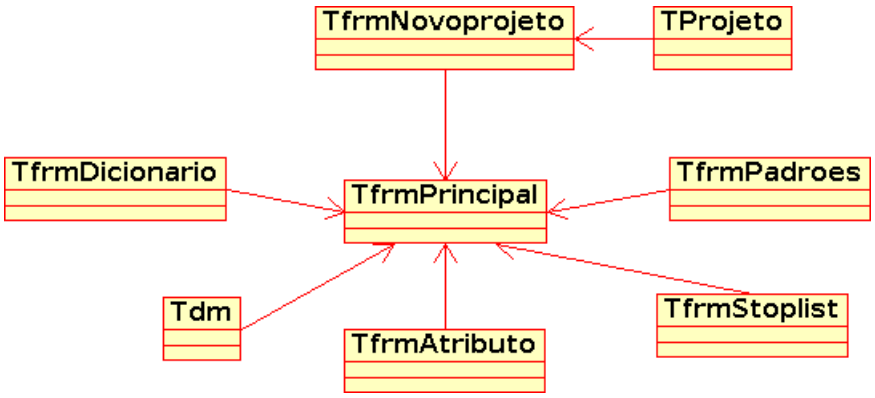


Figura 3.6: Diagrama de classes da ferramenta computacional

TfrmPrincipal: Essa classe representa a interface principal da ferramenta na qual as outras classes estão associadas;

TfrmNovoProjeto: Essa classe representa o formulário de criação de novos projetos.

TProjeto: Nessa classe estão implementadas funções para realizar a abertura dos arquivos *XML* de um projeto.

TfrmDicionario: Essa classe representa o formulário de construção do dicionário e é também responsável por gerenciar o arquivo *XML* do dicionário de três níveis. Ela implementa métodos que possibilitam a realização de inserções e remoções de locais, características e subcaracterísticas, além de métodos que implementam filtros por locais e características.

TfrmAtributo: Essa classe representa o formulário de atributos e implementa métodos de inserção e remoção de atributos que irão fazer parte da tabela atributo-valor.

TfrmStoplist: Nessa classe são gerenciadas as *stoplists*, ou seja, inserção e remoção de endereços de arquivos de *stopwords*. Também é responsável pela inserção e remoção de *stopwords* em *stoplists*. Por exemplo,

o usuário poderá escolher inserir uma *stopword* em uma determinada *stoplist*. Isso possibilita que sejam criadas *stoplists* diversas, por exemplo, uma geral da língua com artigos e preposições, e outra específica do domínio.

TfrmPadroes: Essa classe é responsável pela construção dos arquivos de padronização.

Tdm: Essa classe serve como repositório dos componentes *XML* que fazem a ligação entre a ferramenta computacional e os arquivos *XML*.

No Apêndice A são apresentadas as interfaces relacionadas a cada uma dessas classes, exceto as classes **Tprojeto** e **Tdm**, as quais são utilizadas como classes auxiliares.

CAPÍTULO

4

Biblioteca de Classes

Os algoritmos que implementam a metodologia descrita no Capítulo 2 constituem um conjunto de *scripts* os quais foram implementados na linguagem *Perl* (Schwartz et al., 1997), utilizando o paradigma de orientação a objetos. Na Figura 4.1 é ilustrado o diagrama de classes implementado.

4.1 Classe **TControl**

A classe **TControl** é uma das principais classes da metodologia. Essa classe é responsável por gerenciar e utilizar as demais classes, de modo que cada documento[1] é extraído do conjunto original e, depende dos parâmetros definidos no arquivo de parâmetros (**parameters.xml**), são invocados os métodos correspondentes a normalização, remoção de *stopwords*, aplicação de *stemming* e padronização. Depois de realizado o pré-processamento é aplicado o algoritmo de busca e preenchimento da tabela atributo-valor.

4.2 Classes auxiliares

Constants: Essa classe contém a definição de algumas constantes utilizadas pelos algoritmos da metodologia.

TDocument: Essa classe representa uma abstração de um documento. Disponibiliza funcionalidades para leitura de um documento (arquivo texto), in-

[1] Neste capítulo utilizamos indistintamente os termos *documentos* e *laudos*.

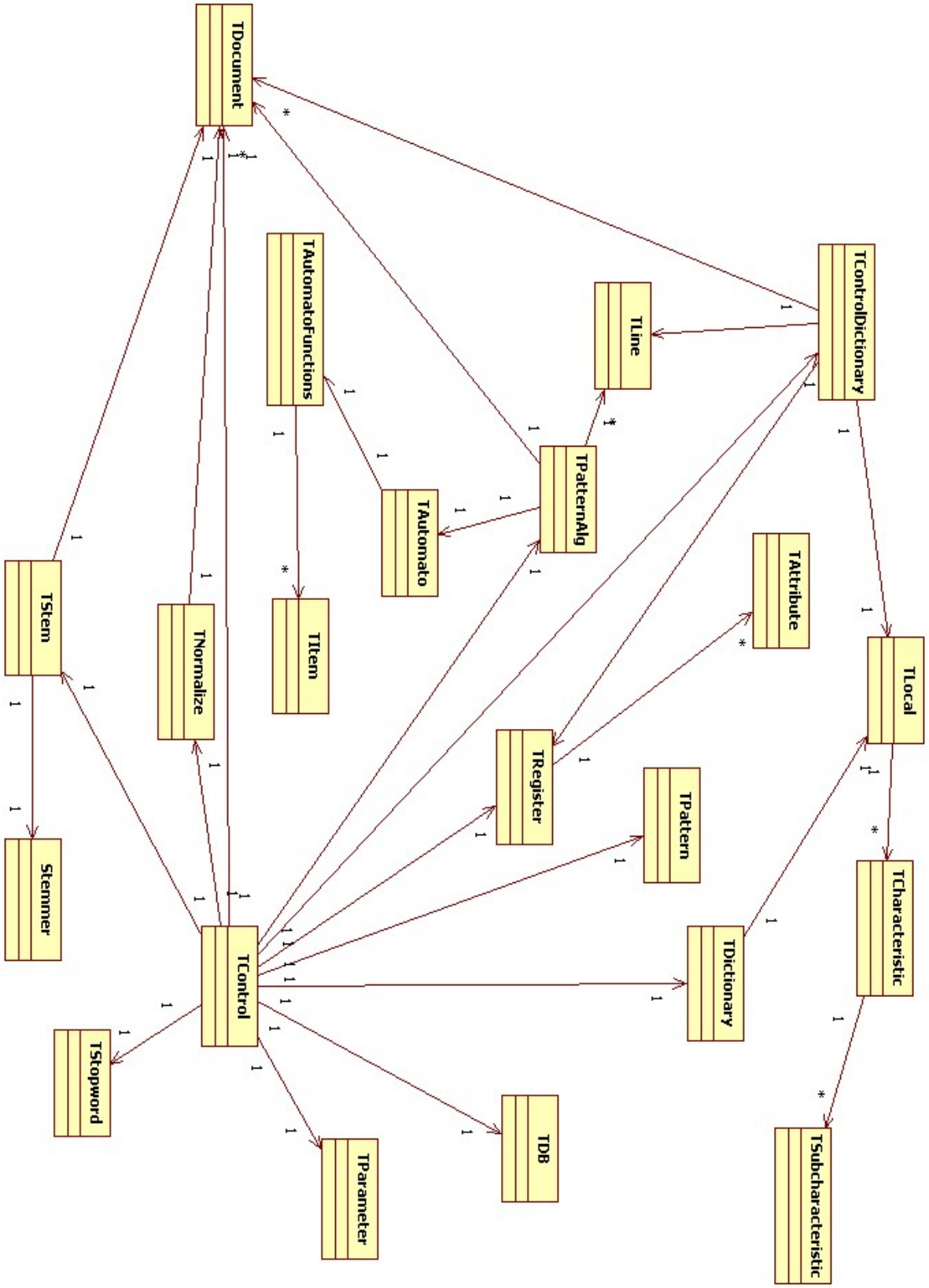


Figura 4.1: Diagrama de classes

serção e alteração de uma linha específica do arquivo, sendo que todas essas operações ocorrem na memória. Também oferece um método para salvar o arquivo, sobrescrevendo o documento antigo.

TLine: Essa classe contém métodos que se destinam a processar uma linha do laudo, para que seja desagregada em palavras e armazenada numa fila, permitindo o processamento de cada palavra na ordem de leitura.

TParameter: Essa classe é responsável por realizar a leitura para a memória dos parâmetros contidos no arquivo de parâmetros (**parameters.xml**).

TStatistic: Nessa classe estão implementados métodos para realização de uma análise estatística sobre um conjunto de documentos, como, por exemplo, a freqüência de determinada frase ou palavra.

TDB: Nessa classe estão implementadas as funcionalidades que permitirão aos algoritmos da metodologia configurar a conexão[2] com o SGBD e inserir registros na tabela atributo-valor.

4.3 Pré-processamento dos documentos

Conforme mencionado, antes que seja realizada a transformação das informações dos laudos para a tabela atributo-valor, é realizado o pré-processamento dos laudos, de acordo com as informações que estão mapeadas no arquivo de parâmetros (**parâmetros.xml**). A Figura 4.2 ilustra o diagrama UML das classes envolvidas no pré-processamento dos laudos.

TNormalize: Essa classe é responsável por remover os acentos e transformar para minúsculo as informações contidas nos documentos. Outra funcionalidade implementada nessa classe é a identificação de palavras chaves, que é executada antes da normalização.

TStopword: Nessa classe estão implementadas as técnicas necessárias para remover *stopwords* de um documento por meio de expressões regulares baseando-se nos termos cadastrados nos arquivos de *stopwords*

TStem: Essa classe é responsável por aplicar o algoritmo de *stemming* nos documentos. O algoritmo de *stemming* utilizado é o algoritmo de Porter (Porter, 1980).

[2]Neste trabalho a conexão com a tabela da base de dados foi realizada por meio do módulo de acesso *Database Interface — DBI*, implementado na linguagem *Perl*.

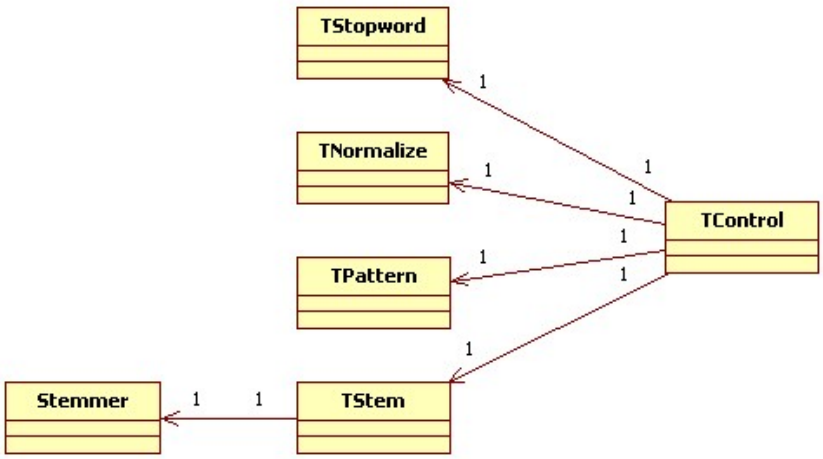


Figura 4.2: Classes responsáveis por realizar o pré-processamento

TPattern: Nessa classe estão implementados métodos para a realização da padronização em um documento, de acordo com as informações dos arquivos de padronizações.

4.4 Armazenamento da estrutura do dicionário

A estrutura do dicionário é armazenada nas classes **TLocal**, **TCharacteristic** e **TSubCharacteristic**. A classe **TDictionary** é responsável por realizar a leitura do arquivo *XML* e inserir as informações nas classes correspondentes. Na Figura 4.3 é ilustrado o diagrama UML das classes envolvidas na leitura e armazenamento da estrutura *XML* na memória.

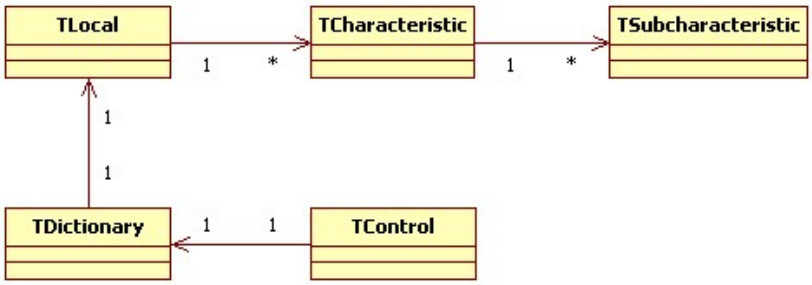


Figura 4.3: Classes responsáveis por realizar a leitura da estrutura do dicionário

TDictionary: Essa classe tem como objetivo realizar a leitura do arquivo *XML* do dicionário, armazenando-o na memória. Contém uma instância de **TLocal**, na qual é armazenada as listas de locais e características.

TLocal: Essa classe contém a lista de locais armazenadas do dicionário. Cada elemento da lista contém o nome do local e uma lista de características relacionada ao local, a qual é representada por um objeto da classe **TCharacteristic**.

TCharacteristic: Essa classe contém a lista de características armazenadas do dicionário. Cada elemento da lista contém o nome da característica, o nome e a posição correspondente ao atributo na tabela atributo-valor e também uma lista de subcaracterística que é representada por um objeto da classe **TSubCharacteristic**.

TSubCharacteristic: Essa classe contém a lista de subcaracterísticas. Cada elemento da lista é composto pelo nome da subcaracterística, nome e posição do atributo na tabela atributo-valor, assim como o valor a ser preenchido no atributo.

4.5 Mapeamento das informações na tabela atributo-valor

Depois de realizado o pré-processamento do laudo e realizada a leitura da estrutura do dicionário para a memória, o mapeamento das informações é executado em duas etapas, as quais são realizadas por meio das classes ilustradas na Figura 4.4.

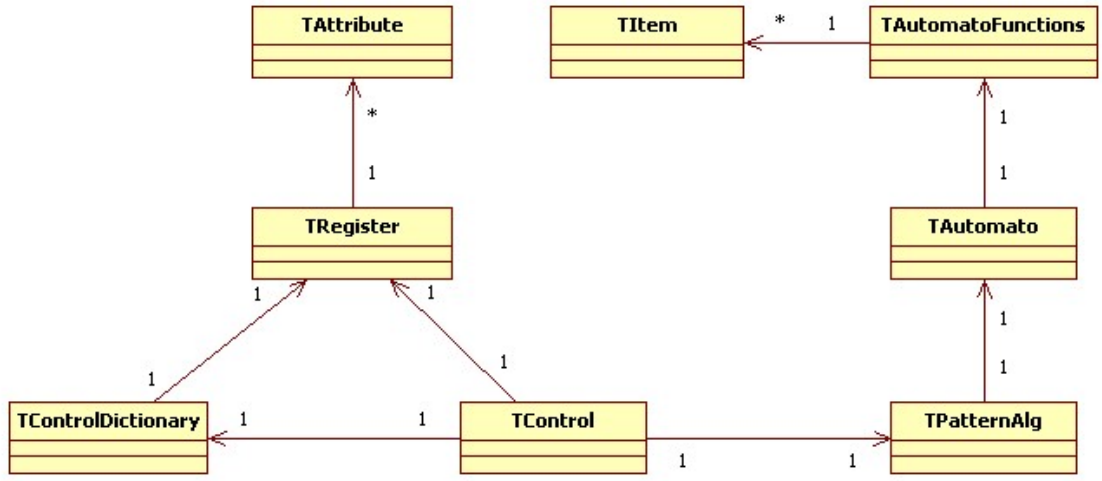


Figura 4.4: Classes responsáveis pelo mapeamento do laudo para a tabela atributo-valor

Na primeira etapa é realizado um processamento no laudo, por meio das classes **TPatternAlg**, **TAutomato**, **TAutomatoFunctions** e **TItem**. Depois de

realizado esse processamento, as informações estarão no formato adequado para serem mapeadas na tabela atributo-valor. Neste trabalho, as informações processadas poderão ter as seguintes formatações:

- Local Característica(s)
- Local Característica(s) SubCaracterística(s)

A classe **TControlDictionary** processa o documento com as informações nesse formato e preenche os atributos, representados pela classe **TAttribute**, do registro da tabela atributo-valor, o qual está representado pela classe **TRegister**.

TAttribute: Essa classe representa um atributo da tabela atributo-valor, contendo o nome do atributo e seu valor que serão utilizados pela classe **TRegister** para a definição de registros na tabela.

TAutomato: Nessa classe está implementado o algoritmo de busca e preenchimento utilizando os métodos básicos implementados na classe **TAutomatoFunctions**, de modo a invocá-los conforme a seqüência de palavras que é fornecida como entrada.

TAutomatoFunctions: Nessa classe estão implementadas as funções básicas necessárias para o funcionamento do algoritmo de busca e preenchimento, por exemplo, inserir um termo nos conjuntos auxiliares (CLD, CLI, CC e CSC) ou relacionar os conjuntos para formar atributos.

TItem: Essa classe tem como objetivo auxiliar a classe **TAutomatoFunctions**, representando um local e suas características ou uma característica e suas subcaracterísticas, bem como a implementação de algumas funcionalidades para a manipulação dos mesmos. Os conjuntos CLI, CLD, CC, CSC contêm objetos da classe **TItem**.

TControlDictionary: Essa classe é responsável por extrair informações do documento processado pela classe **TAutomato**, correspondentes a local, característica e subcaracterística. Estes dados coletados serão utilizados na criação de uma instância da classe **TRegister**, que posteriormente é adicionada na tabela atributo-valor.

TPatternAlg: Essa classe tem como objetivo aplicar o algoritmo de busca e preenchimento em um documento, bem como o gerenciamento de possíveis palavras não processadas encontradas no documento. Essa classe é responsável por invocar os métodos da classe **TAutomato**.

TRegister: Nessa classe são implementadas funcionalidades para simular um registro da tabela. Juntamente com a classe **TAtributte**, essa classe disponibiliza métodos que permitem a inserção e a alteração dos dados nos atributos da tabela atributo-valor, os quais estão descritos no arquivo de atributos (**attributes.xml**).

CAPÍTULO

5

Considerações Finais

Hospitais e clínicas médicas registram cada vez mais informações sobre pacientes e exames laboratoriais. Essas informações geralmente são armazenadas em laudos semi-estruturados descritos em língua natural. Para que possa ser aplicado o processo de mineração de dados sobre essas informações, é necessário transformá-las para um formato adequado, como o atributo-valor. Neste trabalho apresentamos o projeto de uma metodologia desenvolvida para auxiliar na transformação de informações semi-estruturadas encontradas em laudos médicos em informações estruturadas representadas em tabela atributo-valor. Foi desenvolvida uma ferramenta computacional de modo a prover um ambiente amigável para a construção dos arquivos utilizados pela metodologia. A ferramenta computacional, bem como os algoritmos da metodologia foram desenvolvidos utilizando o paradigma de orientação a objetos e, desse modo, estão preparados para receberem novas funcionalidades que possam ser incorporadas no futuro. A metodologia e a ferramenta foram utilizadas em estudos de casos do domínio de Endoscopia Digestiva Alta e Andrologia. Os resultados obtidos foram promissores. Porém, é necessário um intenso trabalho do especialista a fim de identificar local, característica e subcaracterística nas frases únicas. Atualmente estamos investigando técnicas da área de extração de terminologia (de Barcellos Almeida et al., 2006; Pavel and Nolet, 2002) para auxiliar na identificação de unidades terminológicas do domínio.

Referências Bibliográficas

Alpaydin, E. (2004). *Introduction to Machine Learning*. MIT Press, Massachusetts, EUA.

Date, C. J. (2000). *Introdução a Sistemas de Banco de Dados*. Campus, Rio de Janeiro.

de Barcellos Almeida, G. M., de Oliveira, L. H. M., and Aluísio, S. M. (2006). A terminologia na era da informática. *Ciência e Cultura*, 58(2):42–45.

Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., and Uthurusamy, R. (1996). *Advances in Knowledge Discovery and Data Mining*. AAAI Press.

Ferro, M., Lee, H. D., and Esteves, S. C. (2002). Intelligent data analysis: A case study of the diagnostic sperm processing. In *Proceedings of the ACIS - CSITeA02*, pages 352–356, Paraná, Brasil.

Hand, J. and Kamber, M. (2006). *Data Mining: Concepts and Techniques*. Morgan Kaufmann, Califórina, EUA.

Honorato, D. D. F., Lee, H. D., Monard, M. C., Wu, F. C., Machado, R. B., Neto, A. P., and Ferrero, C. A. (2005). Uma metodologia para auxiliar no processo de construção de bases de dados. In *Anais do V Encontro Nacional de Inteligência, XXV Congresso da Sociedade Brasileira de Computação*, pages 593–601, Rio Grande do Sul, Brasil.

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/.

Mitchell, T. (1997). *Machine Learning*. McGraw Hill.

Monard, M. C. and Lee, H. D. (2003). *Processamento de Sêmen Diagnóstico*, pages 461–463. 1. Editora Manole, Barueri, SP, Brasil.

Pavel, S. and Nolet, D. (2002). *Handbook of terminology*. Minister of Public Works and Government Services Canada, Québec, Canadá. http://www.bureaudelatraduction.gc.ca/pwgsc_internet/en/publications/documents/handbook.pdf.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program*, 14(3):130–137.

Pyle, D. (1999). *Data Preparation for Data Mining*. Morgan Kaufmann, Califórnia, EUA.

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

Stroustrup, B. (1997). *C++ Progamming Language*. Addison-Wesley, Nova Jérsei, EUA.

Weiss, S. M. and Indurkhya, N. (1998). *Predictive Data Mining: A Practical Guide*. Morgan Kaufmann, Califórnia, EUA.

APÊNDICE

A

Interfaces da Ferramenta Computacional

Neste capítulo são apresentadas as interfaces da ferramenta computacional desenvolvida. Para cada interface é apresentada uma breve descrição de sua função.

A.1 Atributos

O formulário de inserção de atributo, ilustrado na Figura A.1, tem por objetivo possibilitar a construção da lista de atributos que compõem a tabela atributo-valor.

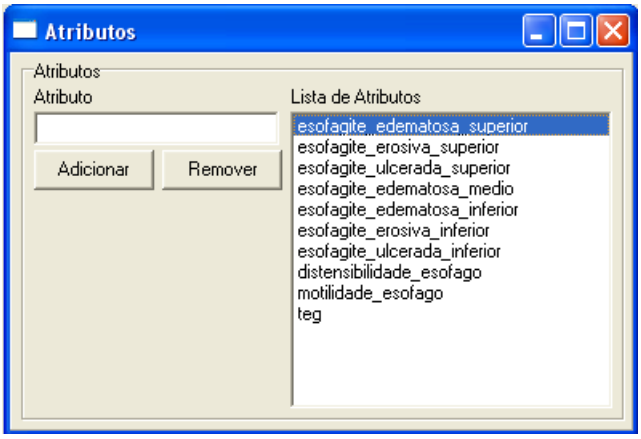


Figura A.1: Formulário de construção da lista de atributos

A.2 Padrões

O formulário de construção dos arquivos de padronização é apresentado nas Figuras A.2 e A.3.

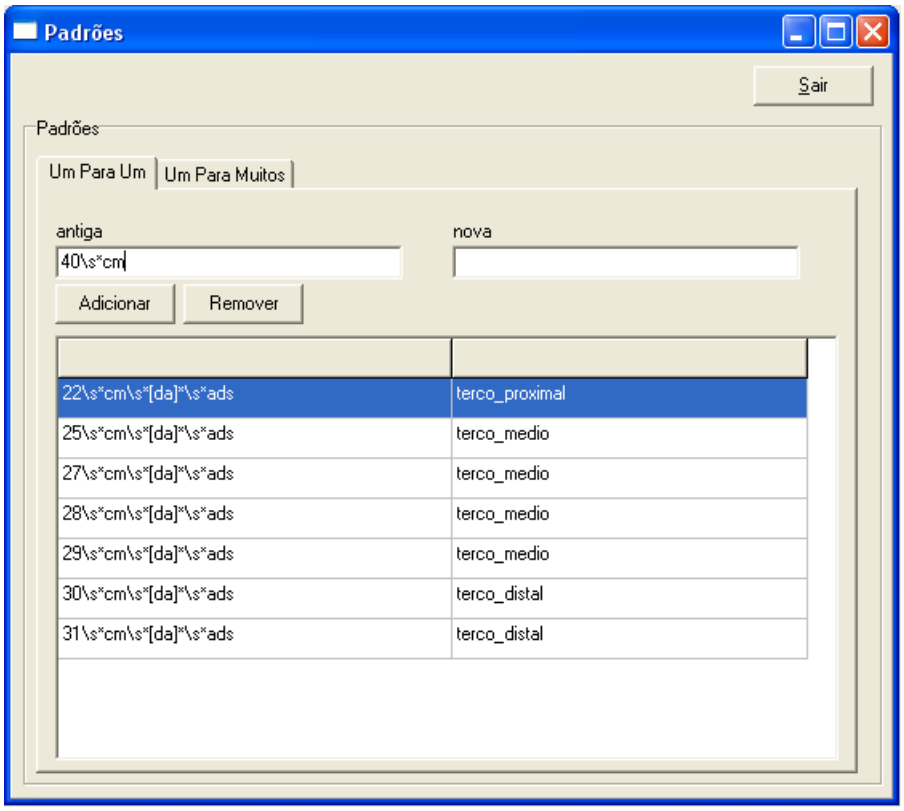


Figura A.2: Formulário de construção dos arquivos de padronização de sinônimos

Na primeria figura é possível cadastrar os sinônimos que serão transformados e, na segunda figura, são cadastradas as padronizações das frases que possuem mais de um evento.

A.3 Stopwords

A construção da lista de *stopwords* deve ser realizada por meio do formulário apresentado na Figura A.4. No lado esquerdo do formulário deve ser selecionado o arquivo no qual serão inseridas as *stopwords* e o lado direito do formulário é utilizado para a inserção de *stopwords*.

A.4 Dicionário

A construção do dicionário é realizada por meio da inserção de informações no formulário apresentado na Figura A.5. Para cada local inserido podem ser

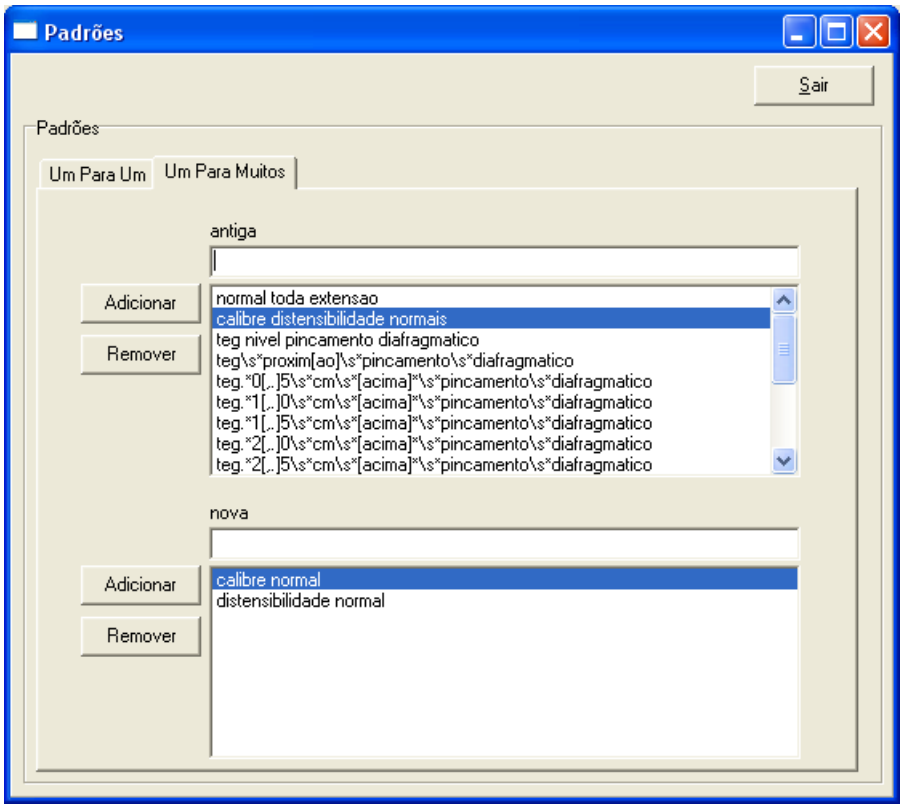


Figura A.3: Formulário de construção dos arquivos de padronização de frases

cadastradas uma ou mais características e, para cada característica, podem ser inseridas zero ou mais subcaracterísticas. No cadastro de características e subcaractísticas também devem ser inseridas informações como o nome do atributo da tabela atributo-valor que será preenchido e o valor que será atribuído.

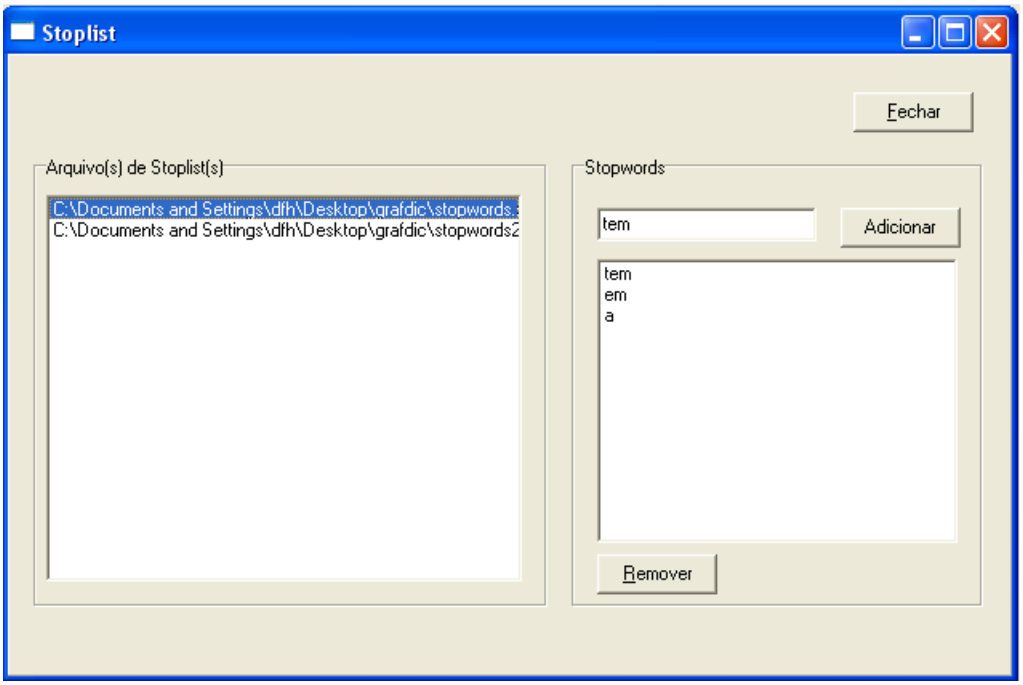


Figura A.4: Formulário de construção dos arquivos de *stopwords*

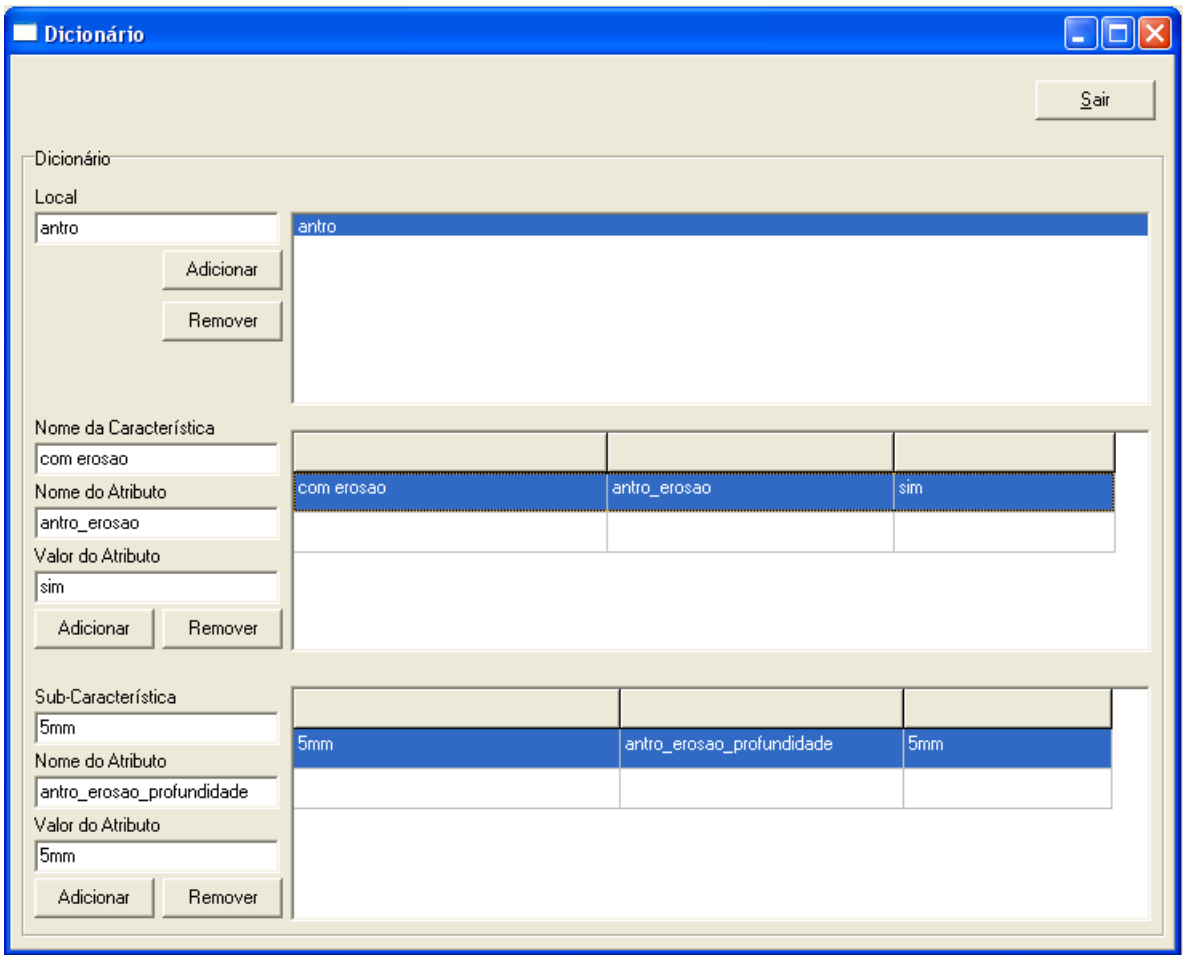


Figura A.5: Formulário de construção do dicionário

APÊNDICE

B

Arquivos Gerados pela Ferramenta Computacional

Neste capítulo são apresentados os arquivos gerados pela ferramenta computacional desenvolvida. Para cada arquivo é apresentada uma breve descrição de sua estrutura.

B.1 Attributes.xml

Na Figura B.1 é ilustrado o arquivo xml no qual são armazenados os atributos que compõem a tabela atributo-valor. Como pode ser observado, nesse

```
<attributes number="10" >
   <attribute>esofagite_edematosa_superior</attribute>
   <attribute>esofagite_erosiva_superior</attribute>
   <attribute>esofagite_ulcerada_superior</attribute>
   <attribute>esofagite_edematosa_medio</attribute>
   <attribute>esofagite_edematosa_inferior</attribute>
   <attribute>esofagite_erosiva_inferior</attribute>
   <attribute>esofagite_ulcerada_inferior</attribute>
   <attribute>distensibilidade_esofago</attribute>
   <attribute>motilidade_esofago</attribute>
   <attribute>teg</attribute>
 </attributes>
```

Figura B.1: Arquivo de atributos

arquivo é armazenado o número de atributos e um conjunto de **tags**, no qual cada uma dessas **tags** representam um atributo da tabela.

B.2 Pattern.xml

Na Figura B.2 é ilustrado o arquivo *XML* que contém as padronizações de sinônimos a serem realizadas. Nesse arquivo são mapeados os sinôni-

```
<pattern number="7" >
  <synonym>
    <new>terco_proximal</new>
    <old>22\s*cm\s*[da]*\s*ads</old>
  </synonym>
  <synonym>
    <new>terco_medio</new>
    <old>25\s*cm\s*[da]*\s*ads</old>
  </synonym>
  <synonym>
    <new>terco_medio</new>
    <old>27\s*cm\s*[da]*\s*ads</old>
  </synonym>
  <synonym>
    <new>terco_medio</new>
    <old>28\s*cm\s*[da]*\s*ads</old>
  </synonym>
  <synonym>
    <new>terco_medio</new>
    <old>29\s*cm\s*[da]*\s*ads</old>
  </synonym>
  <synonym>
    <new>terco_distal</new>
    <old>30\s*cm\s*[da]*\s*ads</old>
  </synonym>
  <synonym>
    <new>terco_distal</new>
    <old>31\s*cm\s*[da]*\s*ads</old>
  </synonym>
</pattern>
```

Figura B.2: Arquivo de padrões

mos identificados na coleção de documentos, bem como são armazenados o número de padrões e uma coleção de nós que indicam os mapeamentos de sinônimos antigos (<*old*>) e novos (<*new*>).

B.3 Patterntwo.xml

Na Figura B.3 é ilustrado o arquivo *XML* que contém as padronizações de mais de um eventos a serem realizadas. Nesse arquivo são mapeados os casos nos quais existem dois eventos em cada frase do documento. No arquivo são armazenados o número de padrões e uma coleção de nós que indicam os mapeamentos de frases com mais de um evento (<*old*>) e novos conjuntos de frases (<*new*>).

```
<patterntwo number="2" >
  <synonymtwo n="12" >
    <old>normal toda extensao</old>
    <new>esofago_superior normal</new>
    <new>esofagite_edematosa_superior nao</new>
    <new>esofagite_erosiva_superior nao</new>
    <new>esofagite_ulcerada_superior nao</new>
    <new>esofago_medio normal</new>
    <new>esofagite_edematosa_medio nao</new>
    <new>esofagite_erosiva_medio nao</new>
    <new>esofagite_ulcerada_medio nao</new>
    <new>esofago_inferior normal</new>
    <new>esofagite_edematosa_inferior nao</new>
    <new>esofagite_erosiva_inferior nao</new>
    <new>esofagite_ulcerada_inferior nao</new>
  </synonymtwo>
  <synonymtwo n="2" >
    <old>calibre distensibilidade normais</old>
    <new>calibre normal</new>
    <new>distensibilidade normal</new>
  </synonymtwo>
</patterntwo>
```

Figura B.3: Arquivo de padrões para mapeamento de mais de um evento por frase

B.4 Stoplist.xml

Na Figura B.4 é ilustrado o arquivo *xml* no qual são armazenados os arquivos de *stopwords* que serão utilizados como referência para extração de termos irrelevantes. Nesse arquivo são armazenados os endereços de arquivos

```
<stoplist number="2" >
  <stopwordFile>[DIR]\stopwords.xml</stopwordFile>
  <stopwordFile>[DIR]\stopwords2.xml</stopwordFile>
</stoplist>
```

Figura B.4: Arquivo de *stoplist*

cujos conteúdo são *stopwords* que serão removidas do texto.

B.5 Stopwords.xml

Na Figura B.5 é ilustrado o arquivo *xml* no qual são armazenados os termos que serão utilizados como referência para extração de termos irrelevantes. Nesse arquivo são armazenados as *stopwords* que serão removidas do texto.

```
<stopwords number="1" >
  <stopword>a</stopword>
</stopwords>
```

Figura B.5: Arquivo de *stopwords*

B.6 Dictionary.xml

Na Figura B.6 é ilustrado o arquivo *XML* no qual são armazenados os termos que serão utilizados como referência para extração de termos relevantes. Nesse arquivo é armazenada a estrutura do dicionário composto por locais,

```
<dictionary number="1" >
  <condition>
    <local>fundo</local>
    <AllCharacteristic numberc="3" >
      <characteristics>
        <attribute>
          <attributeName>fundo_cicatriz_ulcera</attributeName>
          <position>0</position>
          <valueToSave>sim</valueToSave>
        </attribute>
        <characteristicName>cicatriz_ulcera</characteristicName>
      </characteristics>
      <characteristics>
        <attribute>
          <attributeName>fundo_erosao</attributeName>
          <position>0</position>
          <valueToSave>sim</valueToSave>
        </attribute>
        <characteristicName>erosao</characteristicName>
        <allSubCharacteristic numbersc="2" >
          <subcharacteristics>
            <attribute>
              <attributeName>fundo_erosao_fibrina</attributeName>
              <position>0</position>
              <valueToSave>sim</valueToSave>
            </attribute>
            <subcharacteristicName>fibrina</subcharacteristicName>
          </subcharacteristics>
          <subcharacteristics>
            <attribute>
              <attributeName>fundo_erosao_distribuicao</attributeName>
              <position>0</position>
              <valueToSave>esparso</valueToSave>
            </attribute>
            <subcharacteristicName>esparso</subcharacteristicName>
          </subcharacteristics>
        </allSubCharacteristic>
      </characteristics>
      <characteristics>
        <attribute>
          <attributeName>fundo_edematosa</attributeName>
          <position>0</position>
          <valueToSave>sim</valueToSave>
        </attribute>
        <characteristicName>edematosa</characteristicName>
        <allSubCharacteristic numbersc="1" >
          <subcharacteristics>
            <attribute>
              <attributeName>fundo_edematosa_intensidade</attributeName>
              <position>0</position>
              <valueToSave>leve</valueToSave>
            </attribute>
            <subcharacteristicName>leve</subcharacteristicName>
          </subcharacteristics>
        </allSubCharacteristic>
      </characteristics>
    </AllCharacteristic>
  </condition>
</dictionary>
```

Figura B.6: Arquivo do *dicionário*

características e subcaracterísticas.