Tutorial do Metamap em Português

André Alves de Araújo[1]

[1]Ciência da Computação/ 6º período

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ

andrealvesaraujo.1995@gmail.com

Resumo. *O objetivo desse documento é mostrar num tutorial como fazer para o Metamap mapear textos de entrada em português para termos biomédicos dos vocabulários em português do UMLS. Isso é possível ao criar um novo modelo de dados que será utilizado pelo servidor do Metamap. Ele é criado a partir do uso do Metamap Data File Builder. Para que o Data File Builder seja instalado é necessário o Metamap padrão, o UMLS e a ferramenta de léxico chamada LVG. No final desse tutorial mostro exemplos de como fazer esse mapeamento e as limitações no uso do modelo. Também mostra como utilizar esse modelo de dados numa aplicação Java ao importar as bibliotecas disponíveis pelo Metamap Java API. A lógica da criação do nosso modelo pode ser estendido para ser usado em outros idiomas.*

1. Introdução

O Metamap é um programa de processamento de linguagem natural que mapeia um texto de entrada em termos biomédicos presentes nas fontes de vocabulários do UMLS. Para isso ele utiliza no seu servidor um modelo de dados próprio para acessar as fontes de vocabulários específicas do UMLS.

O modelo de dados usado por padrão é o *USABase*. Ele disponibiliza termos biomédicos de alguns vocabulários com certo nível de restrição definido pelo UMLS. Tem outros modelos disponíveis como *NLM* e *Base* que diferem entre si apenas pela quantidade de vocabulários do UMLS devido a diferentes níveis de restrição. Entretanto o servidor ao utilizar qualquer um desses modelos de dados apenas mapeia texto de entrada da língua inglesa para apenas termos biomédicos dessa língua.

Esse trabalho mostra como criar um novo modelo de dados para ser utilizado pelo servidor do Metamap para permitir o mapeamento de texto de entrada em português para termos biomédicos da língua portuguesa. Mostraremos também exemplos do uso do nosso modelo pelo servidor, suas principais limitações e de como usar esse modelo numa aplicação em Java através do uso das bibliotecas do Metamap Java API. No final o servidor do Metamap terá opção de usar tanto o modelo de dados em português proposto quanto os modelos padrões em inglês.

O presente trabalho é separado em 9 seções além da introdução. A seção 2 explica com detalhes como é feito o modelo de dados em português. A seção 3 mostra onde baixar os programas que são pré-requisitos para criar o modelo. A seção 4 mostra como é feito a preparação do ambiente de criação do modelo depois

que os programas são extraídos. A seção 5 é o momento em que é criado o nosso modelo de dados em português. A seção 6 mostra o uso do modelo pelo servidor num exemplo. A seção 7 mostra como utilizar o modelo numa aplicação em Java. A seção 8 discute as limitações e características do nosso modelo. Por fim, a seção 9 apresenta as considerações finais.

2. Sobre o Modelo De Dados Em Português

Nessa seção descrevo sobre o modelo de dados em português. O nosso modelo de dados em português chamado de **UMLS_PORT** é criado pelo programa Metamap Datafile Builder. O Metamap Datafile Builder é um programa que cria modelos de dados para serem utilizados pelo servidor do Metamap, isso é feito a partir do UMLS ou de qualquer estrutura similar a ele.

Para instalar o Metamap Datafile Builder é necessário instalar o Metamap padrão e precisa também das ferramenta léxica LVG. Precisa também baixar o UMLS pois é a partir dele que será retirado os vocabulários em português usados pelo nosso modelo. No meio do processo de criação do **UMLS_PORT** algumas alterações serão feitas em alguns scripts baixados.

Nesse tutorial vou mostrar a instalação e uso desses programas no **Linux**. Vou mostrar como criar o modelo **UMLS_PORT**. Depois vou mostrar exemplos de seu uso no servidor e como pode ser utilizado numa aplicação em Java ao utilizar as bibliotecas Java disponíveis pelo programa Metamap Java API.

3. Pré-Requisitos

Nessa seção mostro quais versões dos programas são necessários baixar. O programa principal que cria o modelo de dados **UMLS_PORT** usado pelo servidor, na sua versão mais atual, é o Metamap Datafile Builder Suite 2016.

Para instala-lo precisa da versão do mesmo ano do LVG e do Metamap padrão que são, respectivamente, lvg 2016 e Metamap 2016v2. Essas versões do Metamap só funcionam no sistema operacional **Linux**. E por ultimo precisa do UMLS. A versão utilizada nesse tutorial é a UMLS2018AB mas pode ser usada versões antigas ou novas. Também é pre-requisito também ter instalado a versão atual do Java e defini-lo como variável de ambiente no Linux.

Para baixar esses programas é necessário ter uma conta cadastrada no UMLS que é feita em https://uts.nlm.nih.gov/license.html

Baixe os programas necessários nos links abaixo:

- MetaMap 2016v2 Linux Version: https://metamap.nlm.nih.gov/MainDownload.shtml
- Datafile Builder Suite 2016: https://metamap.nlm.nih.gov/DataFileBuilder.shtml
- LVG2016 Full Version: https://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/2016/web/download.html
- UMLS2018AB Full Release: https://www.nlm.nih.gov/research/umls/licensedcontent/umlsarchives04.html#2018AB

4. Preparação do ambiente de criação do modelo de dados

Extraia os arquivos baixados na mesma pasta que será chamada nesse trabalho de **Ambiente**. Agora será explicado o que é preciso fazer em cada pasta extraída.

4.1. Pasta do LVG

Depois de extrair o lvg2016.tgz terá a pasta lvg2016 em **Ambiente**, em seguida siga os passos a seguir:

1. Antes de instalar o LVG é necessário alterar uma linha no arquivo de sua instalação que é chamado de install_linux.sh. Então abra esse arquivo no caminho **Ambiente**/lvg2016/install/bin/install_linux.sh no editor de texto
2. Mude a linha 24 do install_linux.sh para JAVA=java
3. Na pasta lvg2016 execute pelo terminal ./install_linux.sh. Espere aparecer a mensagem de sucesso de instalação.
4. Agora é necessário fazer o lvg2016 como variável de ambiente do PATH para ser acessado pelo Metamap. Isso é feito ao executar o comando: export PATH=**CaminhoCompleto**/**Ambiente**/lvg2016/bin:$PATH. Isso mantêm apenas como variavel de ambiente durante a sessão do terminal. Para deixa-lo sempre como variável de ambiente é só copiar o comando acima e colar no arquivo .bashrc.
5. Agora edite a linha 9 do documento lvg que está em lvg2018/bin/lvg para JAVA=java
6. Use o comando lvg -f:i -m e pressione Enter algumas vezes ate surgir algumas palavras só para testar o lvg. Se aparecer é só sair ao usar o comando Control+D.

Agora o LVG esta instalado com sucesso e sua variável de ambiente pode ser acessado para a instalação do Metamap Data File Builder. Quaisquer dúvidas veja o link do site de instalação do LVG é https://lexsrv3.nlm.nih.gov/LexSysGroup/Projects/lvg/2016/docs/userDoc/install/install.html

4.2. Pastas do Metamap

Depois da extração do public_mm_linux_main_2016v2.tar.bz2 e public_mm_linux_dfb_2016.tar.bz2 terá 2 pastas **public_mm** em **Ambiente**. Uma delas é o conteúdo do Metamap padrão e o outro do conteúdo do Data File Builder. É necessário botar ambos os conteúdos na mesma pasta. Isso é feito ao copiar o conteúdo da pasta **public_mm(2)** em **public_mm**. Lembre-se de permitir que mescle pastas e que substitua os arquivos. Em seguida apague **public_mm(2)**.

Agora em **public_mm** só precisa instalar o Data File Builder. Só seguir os passos abaixo:

1. Verifique se tem as variáveis de ambiente do Lvg e o Java. Caso so queria durante a sessão do terminal use o comando: export PATH=/**Caminho_Do_Java_OU_LVG**:$PATH
2. Pelo terminal acesse o caminho **Ambiente**/public_mm/ e execute o comando ./bin/install.sh.

3. Vai aparecer mensagem perguntando qual o diretório de instalação é só apertar Enter para manter o caminho.
4. Vai aparecer mensagem perguntando qual o caminho da variável de ambiente do Java depois de verificar é só apertar Enter.
5. Vai aparecer mensagem perguntando se quer o usar para criar um modelo de dados personalizado, que no nosso caso é um modelo em português, é só digitar 'y'.
6. Vai aparecer mensagem perguntando se vai usar LVG é só digitar 'y'.
7. Vai aparecer mensagem perguntando qual o caminho do LVG é so apertar Enter e esperar de terminar a execução.

Com isso o Data File Builder foi instalado com sucesso. Quaisquer dúvidas o link do site de instalalção do data file builder é https://metamap.nlm.nih.gov/Docs/README_dfb.html

4.3. Pasta do UMLS

Depois de extrair o umls-2018AB-full.zip dentro de **Ambiente** é necessário configurar essa pasta para ter apenas os vocabulários em português. Esses vocabulários são: ICPCPOR,LNC-PT-BR,MDRPOR,MSHPOR,WHOPOR

Essa pasta extraida será referenciada pelo tutorial de **DUMLS**.Isso é feito da seguinte forma:

1. Extraia **Ambiente**/**DUMLS**/mmsys.zip e move o documento release.dat dentro de mmsys.zip para o pai **DUMLS** .
2. Vai no caminho **Ambiente**/**DUMLS**/mmsys e rode o comando ./run_linux.sh. Vai aparecer uma aba do Metamorphosis
3. Selecione Install UMLS e bote como fonte e destino o caminho ate **DUMLS** e deixe marcado apenas o Specialist Lexicon e Metathesaurus e escolha OK.
4. Em seguida clique em *new configuration* e escolha *Accept*.
5. Escolha level 0 e OK
6. Na aba de Output Options verifique se está no formato para Rich Release Format(RLF) senão mude para esse formato.
7. Vai na aba de Source List e clica em Include e selecione os seguintes vocabulários em português segurando Control:
   - ICPCPOR1993
   - LNC-PT-BR_264
   - MDRPOR21_0
   - MSHPOR2018
   - WHOPOR_1997

   Cuidado pois assim que clicar num deles o vai aparecer uma aba que sugere outros vocabulários que estão ligados a esses. É só clicar em Cancelar que não serão incluídas. No final apenas estes 5 vocabulários devem estar com linhas azuis.
8. No topo da direita clique em Done e clique em Begin Subset
9. Salve e complete a configuração com qualquer nome
10. Quando terminar de carregar aperte OK. E saia do Metamorphosis.

File
MRCONSO.RRF
MRSAT.RRF
MRSTY.RRF
MRSAB.RRF
MRRANK.RRF
SM.DB

Figura 1. Arquivos Necessários do UMLS

Se tudo estiver em ordem agora temos os termos em português presentes nesses vocabulários.A informação dos termos estão salvas nos arquivos da figura 1. Os .RFF estão presentes na pasta em **Ambiente**/**DUMLS**/2018AB/META e o SM.DB está em **Ambiente**/**DUMLS**/2018AB/LEX/LEX_DB

5. Criação do modelo de dados em português

5.1. Criando pasta do modelo

Como já temos os vocabulários em português e o Data File Builder agora vamos preparar para usar o Data File Builder e ter o modelo de dados em português que será usado pelo servidor Metamap. Siga os passos a seguir:

1. Em **Ambiente**/public_mm/sourceData crie a pasta com o nome do modelo de dados em português. Como dito anteriormente o nosso modelo será chamado de **UMLS_PORT**. E dentro de **UMLS_PORT** crie a pasta chamada obrigatoriamente de **umls**.
2. Copie os arquivos da figura 1 e cole na pasta dentro de **Ambiente**/public_mm/sourceData/**UMLS_PORT**/**umls**.

5.2. Executando Scripts Finais

Agora faça o abaixo:

1. No terminal vai para **Ambiente**/public_mm e faça ./bin/skrmedpostctl start pois é acessado pelos scripts
2. Se o lvg está como variável de ambiente por sessão faça export PATH=**CaminhoCompleto**/**Ambiente**/lvg2016/bin:$PATH

Agora tudo está pronto para rodar os scripts principais e termos o modelo de dados pronto para ser usado pelo servidor Metamap. Siga os passos abaixo:

1. No primeiro script em **Ambiente**/public_mm faça ./bin/BuildDataFiles. Bote o nome da pasta do modelo de dados **UMLS_PORT**. E digite 'yes' se o caminho ate ele estiver correto. Aperte enter 2 vezes e espera terminar. Agora vamos executar o resto dos scripts que acabou de gerar.
2. Vamos alterar o documento 01CreateWorkFiles em **Ambiente**/public_mm/sourceData/**UMLS_PORT**/01metawordindex. Altere a sua linha 41 onde estava 'ENG' para 'POR' e apague a linha 53 ate a 62.
3. Nesse mesmo caminho execute ./01CreateWorkFiles.
4. Agora execute o próximo script ./02Suppress

5. Agora execute o próximo script 03FilterPrep
6. Agora execute o próximo script ./04FilterStrict. Esse é bem demorado. Diga yes e escolha a quantidade de processadores que será usado, como por exemplo 4 processadores e aguarde.
7. Agora execute o próximo script ./05GenerateMWIFiles
8. Vai para **Ambiente**/public_mm/sourceData/**UMLS_PORT**/02treecodes e execute ./01GenerateTreecodes
9. Vai para **Ambiente**/public_mm/sourceData/**UMLS_PORT**/03variants e execute ./01GenerateVariants
10. Vai para **Ambiente**/public_mm/sourceData/**UMLS_PORT**/04synonyms e execute ./01GenerateSynonyms
11. Vai para **Ambiente**/public_mm/sourceData/**UMLS_PORT**/05abbrAcronyms e execute ./01GenerateAbbrAcronyms
12. Vai para **Ambiente**/public_mm/ e execute ./bin/LoadDataFiles e digite enter 4 vezes e 'yes'. Isso tudo gera DB.UMLS_PORT.2016AA.strict. em /**Ambiente**/public_mm/DB/.

O DB.UMLS_PORT.2016AA.strict é o modelo de dados que criamos para fazer o mapeamento de palavras de entrada em português para termos médicos em português, que está num formato que permite ser acessado pelo Metamap. Agora só precisa utiliza-lo pelo servidor do Metamap.

6. Uso do modelo no servidor

Nessa seção mostra o comando principal para executar o mapeamento de termos em português para termos biomédicos em português ao utilizar nosso modelo de dados DB.UMLS_PORT.2016AA.strict. Também mostro alguns exemplos do seu uso.

Para ser usado vai em **Ambiente**/public_mm/ e execute os dois comandos abaixo:

1. ./bin/skrmedpostctl start
2. ./bin/metamap -V **UMLS_PORT** -G -I.

A opção -V define qual modelo vai ser usado (pode usar *USABase* por exemplo), o -G permite ver as fontes de vocabulário e -I mostra o ID do termo mapeado no UMLS. Todas as opções de comando do Metamap estão em https://metamap.nlm.nih.gov/Docs/MM_2016_Usage.pdf.

Depois de rodar esse ultimo comando digite como entrada as palavras 'dor','enfermeira','sinusite'. A saída de cada uma dela esta abaixo :

- Para dor -> C0030193:DOR (Dor {MDRPOR,MSHPOR,WHOPOR}) [Sign or Symptom]
- Para enfermeira-> C0028661:Enfermeira MSHPOR [Professional or Occupational Group]
- Para sinusite-> C0037199:SINUSITE (Sinusite {MDRPOR,MSHPOR,WHOPOR}) [Disease or Syndrome]

Cada saída é o termo ou conjunto de termos biomédicos que mapeia da melhor maneira o texto de entrada. As informações da saída do mapeamento são:

- O ID do conceito no UMLS : C0030193
- O nome do conceito e a string Metathesaurus dela: DOR (Dor). Se não tem parênteses significa que a string e o nome do conceito tem o mesmo nome.
- Os vocabulários do UMLS a qual o termo encontrado pertence: {MDR-POR,MSHPOR,WHOPOR}
- E o grupo semântico que ele pertence: [Disease or Syndrome]

Se der esses resultados o modelo está sendo usado pelo servidor de forma correta. Para sair do servidor é só fazer Control+D. Se algo acontecer de errado ou se não está funcionando de uma olhada no link original da instalação do Data File Builder: https://metamap.nlm.nih.gov/Docs/datafilebuilder.pdf

7. Metamap Java API

Nessa seção mostro como usar o modelo de dados em português no sistema operacional **Linux** numa aplicação em Java. Isso é feito ao utilizar as bibliotecas disponíveis do Metamap Java API. Esse software contêm .jar que são importados na aplicação a fim de utilizar as classes e métodos que permite processar o mapeamento do Metamap. O tutorial original da sua instalação, seu uso e suas principais características estão no link a seguir: https://metamap.nlm.nih.gov/Docs/README_javaapi.shtml.

7.1. Instalação e uso

Primeiramente precisa instalar o Metamap Java API. Como pré-requisito da sua instalação é necessário ter o Metamap padrão antes. Como já temos esse padrão só precisamos baixar a versão no **Linux** do Metamap Java API do mesmo ano chamada de The Java API Release (2016v2). O link para o baixar é esse: https://metamap.nlm.nih.gov/JavaApi.shtml.

Depois de o extrair na pasta **Ambiente** siga as etapas abaixo para instalar o Metamap Java API:

1. Cole os arquivos do Metamap Java API na pasta **Ambiente**/public_mm que contém o Metamap padrão junto do Data File Builder.
2. No terminal vai para o caminho **Ambiente**/public_mm e execute o comando ./bin/install.sh
3. Vai aparecer mensagem perguntando qual o diretório de instalação é só apertar Enter para manter o caminho
4. Vai aparecer mensagem perguntando qual o caminho da variável de ambiente do Java depois de verificar é só apertar Enter. Depois disso a instalação termina.

7.2. Aplicação Java

Agora que instalou o Metamap Java API vamos fazer o código em Java de exemplo. Os .jar importantes que devem ser importados no projeto são MetaMapApi.jar e prologbeans.jar que estão presentes em /public_mm/src/javaapi/dist/. Não vou explicar como que usa .jar e nem como rodar um programa em java. O código está abaixo:

```
MetaMapApi api = new MetaMapApiImpl();
api.setOptions("-V UMLS_PORT");
String entrada="Estou com sinusite e muitas dores.";
List<Result> resultList = api.processCitationsFromString(entrada);
for(Result r : resultList){
    for(Utterance ut : r.getUtteranceList()){
        for(PCM pcm: ut.getPCMList()){
                System.out.println("Phrase:" + pcm.getPhrase().getPhraseText());
                int cont=1;
                if(pcm.getMappingList().size() == 0){
                    System.out.println("Não foi mapeado");
                }else{
                    for(Mapping m :pcm.getMappingList()){
                        System.out.println("MAPEAMENTO " + cont);

                        for(Ev v : m.getEvList()){
                        System.out.println("[Score]:" +  v.getScore());
                        System.out.println("[Conceito]:"+ v.getPreferredName());
                        System.out.println("[G.Semantico]"+v.getSemanticTypes();
                        System.out.println("[Fontes]" +  v.getSources());
                        System.out.println();
                        }
                        cont++;
                        }
                    }
                System.out.println("-----------------------------------");
        }
    }
}
```

O código acima faz o processo de mapeamento do Metamap na frase de entrada "Estou com sinusite e muitas dores"utilizando nosso modelo de dados **UMLS_PORT** no processo. Caso o mapeamento seja possível as informações de cada termo biomédico do UMLS encontrado nesse mapeamento são apresentados.

As informações sobre cada termo são: seu valor(Score) no mapeamento, nome do seu conceito, grupo semântico que pertence, fontes de vocabulário em que está presente o conceito. Caso mais de um mapeamento seja possível ele apresenta o de maior valor ou caso tenha valores iguais como maior apresenta ambos. No nosso exemplo apenas a palavra "sinusite' será mapeada para um termo biomédico.

A interface "MetaMapApi"é a classe que representa o servidor do Metamap, ou seja, nela envolve todo o processo de mapeamento que é ativado pelo método

"processCitationsFromString()"no texto de entrada. Nessa interface é possível escolher certas opções para dar variação no processamento("setOptions()") como, por exemplo, qual modelo de dados usar.

As informações do resultado do mapeamento são recuperadas através de laços aninhados em cima de uma série de coleções entre elas dos objetos "Result", "Utterance", "PCM", "Phrase", "Mapping". O objeto "Mapping"contêm o conjunto de termos biomédicos de um mapeamento encontrado. E o "Ev"é um objeto que representa o termo biomédico nesse evento. E a partir dele extrai suas principais informações entre elas nome, grupo semântico e outros.

Sabendo disso, para rodar a aplicação em Java precisa fazer o seguinte:

1. No terminal vai no caminho **Ambiente**/public_mm e execute o comando ./bin/skrmedpostctl start
2. Execute o comando ./bin/mmserver16 que ativa o servidor do Metamap
3. Rode o código em Java e veja o resultado

Qualquer dúvida aqui está o link original: https://metamap.nlm.nih.gov/Docs/README_javaapi.shtml. E sobre as possíveis opções que podem ser usadas no processo de mapeamento : https://metamap.nlm.nih.gov/Docs/MM_2016_Usage.pdf. E sobre as classes do Metamap Java API: https://metamap.nlm.nih.gov/javaapi/javadoc/index.html.

8. Discussão

Nessa seção mostro quais são as limitações e problemas estão ligados ao nosso modelo de dados em português. Primeiro, mesmo com o modelo em português que contêm termos biomédicos com acentos nos seus vocabulários, o servidor do Metamap não aceita como entrada texto com acentuações nem nenhum outro caractere especial. Pois o Metamap só aceita palavras cujos caracteres estão presentes na tabela ASCII. Por causa disso qualquer tentativa com esses caracteres o Metamap aborta o processo de mapeamento.

Segundo, quaisquer variações gramaticais em substantivos e verbos na entrada não serão mapeadas corretamente na saída. Só serão mapeadas aquelas que forem totalmente iguais aos conceitos que estão nos vocabulários que por padrão não tem todas as variações possíveis. Por exemplo, a entrada "dor"é mapeada corretamente ao termo biomédico "DOR", entretanto não é mapeado suas variações como "dolorido"ou "dores"pois não existe os termos biomédicos "dolorido"ou "dores"nos vocabulários. O correto seria que as entradas "dolorido"ou "dores"fossem mapeadas para "DOR"por exemplo.

Isso ocorre porque o servidor não consegue estabelecer a ligação entre "dor"e suas variações já que ele utiliza no processo de mapeamento um Léxico em inglês. Esse léxico é usado numa fase do processamento como meio para gerar os melhores candidatos para o mapeamento, que em outra fase é comparado aos termos dos vocabulários. Entretanto ele faz isso usando regras da língua inglesa ao invés das regras em português o que afeta bruscamente o mapeamento.

Mesmo assim esse tutorial pode ser usado para fazer modelos em qualquer idioma ou um modelo com mais de um idioma. Só precisa ter cuidado na seleção

dos vocabulários ao usar o mmsys da seção 3.3. e na alterações das linha na seção 4.2 para a linguagem especifica.

Além disso, o servidor do Metamap ainda pode acessar os modelos de dados padrões ao fazer o comando:

- ./bin/metamap -V USABase.

9. Conclusão

Esse trabalho mostra a criação de um modelo de dados que é acessado pelo servidor do Metamap para mapear as palavras em português de entrada para termos biomédicos em português dos vocabulários do UMLS. Isso é feito ao instalar e usar o Metamap Data File Builder no ambiente Linux. Mostro o principal comando no servidor para utilizar o modelo num exemplos do seu uso e explico o que significa a saída do comando

Também foi mostrado um aplicação simples em Java que usa o modelo de dados junto do servidor para mapear uma frase de entrada em português. Expliquei a sua lógica e as principais classes e métodos utilizados. Isso foi possível através da importação das bibliotecas em Java do Metamap Java API.

O nosso modelo de dados em português não é utilizado por completo pelo servidor do Metamap. Pois o servidor não aceita como entrada texto com acentuações nem nenhum outro caractere especial fora da tabela ASCII. Além disso o servidor não consegue mapear corretamente na maioria das vezes as variações gramaticais no texto de entrada em português.

E por fim, a lógica desse tutorial pode ser estendido a modelos de outros idiomas ou em um modelo com mais de um idioma só precisando escolher os vocabulários que deseja ter no modelo.

Referências