Universidade Federal de Pernambuco

Centro de Informática

Graduação em Ciência da Computação

Criação de uma base de dados biológica a partir de extração da informação de bases textuais

Trabalho de Graduação

Allyson Manoel Nascimento Venceslau

Julho de 2019

Allyson Manoel Nascimento Venceslau

Criação de uma base de dados biológica a partir de extração da informação de bases textuais

Monografia apresentada ao Programa de Graduação em Ciência da Computação, como requisito parcial para a obtenção do Título de Bacharel em Ciência da Computação, Centro de Informática.

Orientador: Prof. Luciano Barbosa

Recife
Julho de 2019

Agradecimentos

Inicialmente gostaria de agradecer a meu Deus Jeová que permitiu que eu conseguisse chegar até aqui e me ajudou em todos os momentos durante a graduação permitindo que eu conseguisse continuar mesmo nos momento mais difíceis. Gostaria de agradecer a minha mãe, Auriziane Nascimento por ter me ajudado durante todo esse processo me apoiando, incentivando e me dando forças para não desistir e a minha irmã Lillyane Nascimento por mostrar que não importa as dificuldades quando colocamos a Jeová em primeiro lugar conseguimos tudo segundo a vontade dele. Também gostaria de agradecer a toda minha família por nunca duvidar na minha capacidade e torcer junto comigo pela realização do meu objetivo de me formar. Também gostaria de agradecer a todos colegas de curso que me ajudaram em algum momento durante todo este processo, em especial Luiz Reis e Arthur Embrahim que durante diversas situações estiveram comigo participando de projetos e estudando para provas.

Gostaria também de agradecer aos professores deste do Centro de Informática através de seu conhecimento permitiram que me tornasse um melhor profissional. Um agradecimento especial para o Professor Luciano Barbosa e ao seu aluno de doutorado Jhony Moreira que pacientemente me ajudaram durante todo o processo de construção deste projeto.

Gostaria de deixar eu agradecimento especial a Professora Marcela Torres que desde meu ensino médio esteve participando das minhas conquistas profissionais e a Professora Charlene Teresa que desde o começo da minha vida me ajudou e continua me ajudando.

Não haveria espeço para listar todas as pessoas que participaram e me influenciaram positivamente neste processo, então deixou este parágrafo final para todos aqueles que participaram desde últimos 5 anos de forma direta ou indireta na minha formação.

Computadores fazem arte
Chico Science

RESUMO

Este trabalho tem como objetivo mostrar o processo de criação de uma base de dados genéticos estruturada a partir da extração da informações textuais. Para isso, são propostas técnicas de obtenção de dados através de engenhos de buscas, ontologias e sentenças extraídas da Gene Expression Omnibus (GEO), base de dados que contém informações genéticas. É proposto o algoritmo CRF para realização de Extração de Entidade Nomeada (NER) e Distant Supervision é utilizado para rotulação automática de dados.

Palavras-chave: Extração da Informação, Extração de Entidade Nomeada, CRF, Distant Supervision, NLP.

ABSTRACT

In this paper we show a structured genetic database construction's process from textual information extraction. For this purpuse, data obtetion tecniques are proposed through search engines, ontologies and sentences extracted from Gene Expression Omnibus (GEO), a genetetic information database. To do Named Entity Recognition (NER) the CRF algorithm is proposed and Distant Supervision tecniques is used to automatic data labeling.

Keywords: Information Extration, Named Entity Recognition (NER), CRF, Distant Supervision, NLP.

Sumário

Lista de figuras

Capítulo 1

Introdução

1.1 Contexto

Com a popularização da web e a criação de redes sociais, a quantidade de dados trafegados cresceu exponencialmente. Essas informações podem ser encontradas de forma estruturadas, semi-estruturadas e não estruturadas. Dados estruturados têm uma boa organização, algo que permite que as informações relacionadas a eles sejam recuperadas de forma rápida. Podemos citar, como exemplo, os dados de bancos de dados relacionais, como os bancos SQL e algumas linguagens de marcação, como o XML.

Dados não-estruturados se encontram na forma de linguagem natural. São textos que não apresentam nenhuma marcação que identifique a que os dados se referem. Podemos citar como exemplo, artigos da web, artigos acadêmicos e tweets.

Dados semi-estruturados são aqueles que embora se encontrem em forma de linguagem natural apresentam algum tipo de identificação de seu conteúdo. Por exemplo, tabelas com especificação de um produto em sites de venda.

O modo mais prático de fazer análises de dados é utilizando base de dados estruturadas, porém nem sempre existe uma base dados estruturada disponível. É possível transformar base dados não estruturadas ou semi-estruturadas em bases estruturadas através de processos que envolvem utilização de recuperação da informação e técnicas de processamento de linguagem natural.

Visto que muitos dados de domínios específicos ou muito restritos estão

disponíveis na internet, porém não de forma estruturada, muitos profissionais de diversas áreas das ciências biológica tem tirado grande proveito de criação de bases de dados utilizando essas técnicas [1].

Neste trabalho faremos a extração de experimentos registrados na Gene Expression Omnibus (GEO)[2], que hospeda um grande núumero de experimentos de arrays de DNAm. Após isso, técnicas de limpeza de dados serão utilizadas para que apenas sentenças relevantes permaneçam. A partir desse conjunto de sentenças, utilizaremos algumas técnicas para extração da informação com objetivo de obter um melhor resultados das extrações realizadas. Aplicaremos técnicas já consolidadas na literatura como utilização de regiões de relevância para extrair informações relevantes[3] e aprendizado de padrões a partir do uso de hiperônimos [4].

Este documento está estruturado em capítulos da seguinte forma: No capítulo 2 seremos introduzidos aos conceitos fundamentais para o entendimento desse projeto. No capítulo 3 é descrito como foi implementada a nossa solução contendo informações sobre abordagens que adotamos mais não tiveram bons resultados e explicamos o porquê elas não funcionaram. No capitulo 4 explicaremos como foi construído a base de teste, bem como os resultados obtidos utilizando ela. No capitulo final, mostramos as conclusões desse projeto bem como possíveis trabalhos futuros.

Capítulo 2

Fundamentos

2.1 Ontologias

Existem muitas formas de se definir ontologias [5] [6], dentre elas podemos observar do ponto de vista filosófico ou computacional cada uma tendo uma série de definições variando de acordo com o autor. Do ponto de vista filosófico, ontologia pode ser vista como um ramo da filosofia que estuda o ser. Ela é utilizada, principalmente, para descrever domínios naturais do mundo.

Do ponto de vista computacional, porém, ontologias definem um conjunto de termos que representam conceitos dentro de um domínio e os relacionamentos que existem entre esses termos. Ontologias geralmente descrevem indivíduos, classes, atributos e relacionamentos.

Em processamento de linguagem natural, ontologia podem ser utilizadas como base de dados de léxicos para um domínio específico. Isso permite que atividades como tradução de texto e extração de informação sejam realizadas em domínios que exigem a utilização de termos específicos.

Algumas ontologias podem ser facilmente encontradas na internet e estão disponíveis para ser baixadas, como por exemplo, a Cell Ontology[1]. Ela contém informações sobre tipos de células de diferentes tipos de organismos, desde células procariontes até mamíferos, porém sem informações de células vegetal.

[1] http://www.obofoundry.org/ontology/cl.html

2.2 Distant Supervision

Para treinar modelos aprendizagem de máquina para realizar a extração informação é comum rotular um conjunto de treino a mão, extraindo relações entre sentenças. Está é uma tarefa que demanda muito tempo e caso a base de dados seja muito grande pode ser inviável rotular todos os dados fazendo com que o algoritmo potencialmente não fique bem treinado.

Distant Supervision (DS) surge como uma alternativa a este problema. Dado uma base de dados D com entidades e relações entre elas, tendo um par de entidades que sabemos que existe uma relação R entre elas, DS automaticamente rotula todas as ocorrências desse par como pertencente a relação R fazendo isso para todos os pares pertencentes a D. Essa abordagem embora muito eficiente pode potencialmente trazer erros ao conjunto de treino impactando a construção do modelo [7].

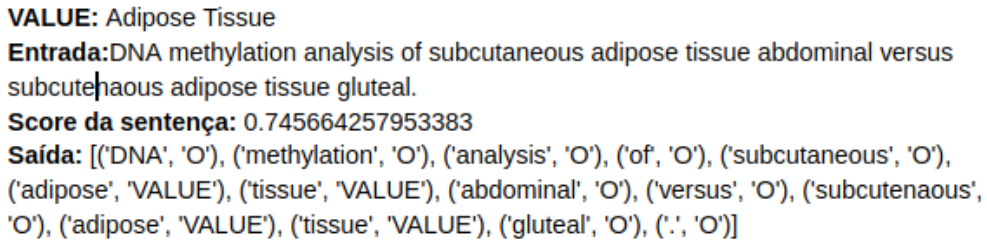

Figura 2.1: Exemplo de entrada e saída do Distant Supervision

Como podemos observar na figura 2.1 dado uma sentença de entrada e um conjunto de palavras alvo é aplicado o Soft TF-IDF com o objetivo de encontrar o padrão que mais se aproxime das palavras alvo e temos as palavras e sua classificação: Value para as palavras alvo e O para as outras. Isso permite criar um conjunto de sentenças rotuladas automaticamente

2.3 Extração de informação

Extração da Informação (IE) é uma atividade que consiste em extrair de base de dados não estruturadas ou semi-estruturadas para criação de uma base de dados estruturada. Em geral essa extração ocorre da seguinte forma: Dado um conjunto de classes de interesse e relações que existem entre elas um corpus (conjunto de documentos). Desses documentos são extraídos instâncias

das classes e relações entre elas. [8]

No processo de extrair informação de texto uma série de técnicas podem ser empregadas. Entre elas, para o processo de extrair instâncias pertencentes às classes é chamado de Named Entity Recognition. Esse processo é detalhado a seguir.

2.3.1 Named Entity Recognition

Named Entity Recognition (NER) é uma subtarefa da Extração de Informação que a partir de palavras em um texto deseja obter rótulos que representam aquelas palavras em classes [8]. Visto que uma única palavra pode ter várias aplicações e significados esta não é uma tarefa fácil. A palavra manga, por exemplo, pode ser classificado como um verbo, um substantivo, um adjetivo a depender do contexto em que ela está inserida.

Por ser uma atividade complexa, uma série de técnicas podem ser utilizadas com o objetivo de solucionar este problema, desde técnicas manuais até técnicas que utilização aprendizagem de máquina (ML) para se fazer essas extrações. Atualmente diversas abordagens utilizando ML foram propostas, entre elas Conditional Random Fields (CRF) [9] afinidade semântica e Modelos de Markov Escondidos (HMM) [10]

2.3.2 Conditional Random Fields (CRF)

Este modelo foi inicialmente definido por Lafferty, McCallum e Pereira [11] e tem se tornado bastante popular em aplicações de processamento de linguagem natural devido ao fato de analisar o contexto ao classificar uma amostra. Isso é muito útil ao fazer NER visto que a classificação de uma palavra depende do contexto que ela foi empregada, ou seja, a depender do sentido da frase a palavra pode ter um significado e uma classificação morfológica diferentes.

Para utilizar este modelo é necessário um conjunto de sentenças rotulado para treino e um conjunto de parâmetros associados a cada amostra, que permite ao classificador identificar qual a importância de cada amostra no contexto.

Por exemplo, tipos de tecido são substantivos, em português muitas vezes o tipo do tecido vem precedido pela palavra tecido. Essas e outras informações podem ser utilizadas pelo CRF para identificar tipos de tecidos em um sentença e assim classifica-las. Essas informações são extraídas das sentenças enviadas ao classificador no momento de seu treino. Isso permite que quando o CRF recebe uma nova entrada seja capaz de reconhecer pelas características da sentença contém um tipo de tecido e qual o tipo de tecido identificado.

Capítulo 3

Solução

Neste trabalho pretendemos criar um vocabulário que irá nós auxiliar no processo de extração de informações relevantes para a nossa base de dados. A base conterá informações sobre tipos de tecidos e tipos de células. Para a criação desse Gazetteer testamos duas abordagens: Utilização de engenhos de buscas para pesquisar por sentenças relevantes extraindo novos termos a partir delas e extração de termos a partir de ontologias. O link para os repositórios do projeto estão na referência[12] [13].

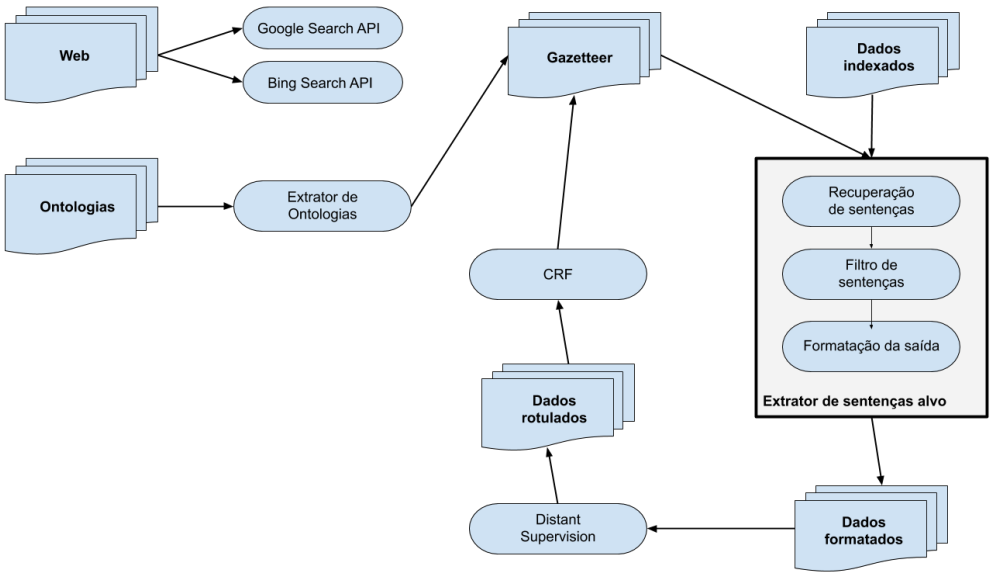

Figura 3.1: Fluxograma da solução proposta

a ideia do sistema é focado em técnicas para aumentar o Gazetteer. Inici-

almente nós foi enviada informação sobre três tipos de tecido que serviram de sementes para o novo vocabulário. Para aumentar isso utilizamos extratores da Web e Ontologias. Com os novos termos partimos para extrair novos termos nos dados indexados. Para realizar está atividade foi necessário extrair as sentenças alvo formatá-los, rotula-los automaticamente utilizando Distant Supervision e utilizar esses dados rotulados no CRF. O CRF treinado extrai esses novos termos que são adicionados ao Gazetteer que pode repetir este processo.

A seguir detalharemos cada uma dessas etapas explicando como foi feito a implementação. Fazemos também uma comparação entre a utilização de extratores Web e Extração de ontologias, quais as vantagens e desvantagens, bem como por que foi escolhido seguir com apenas uma das abordagens.

3.1 Engenhos de buscas

Utilizamos engenho de buscas para construir um Gaziteer de forma automática utilizando padrões iniciais como sementes e utilizamos bootstrapping para aumentar o tamanho do vocabulário. Com esse objetivo utilizamos APIs do Bing e do Google para capturar possíveis sentenças relevantes e com elas extrair novos termos para o nosso vocabulário.

Ao realizar uma consulta na web um conjunto de informações é retornado podendo variar de um engenho de busca para outro, porém algumas informações básicas podem ser encontradas, entre elas: a url que redireciona para a página, um título relacionado a página e uma pequena descrição do conteúdo daquela página. Ambas as APIs utilizadas retornam json com caracteristicas próprias, mas contendo essas três informações descritas.

Queríamos utilizar sentenças que contivessem uma relação de hiponímia e extrair novos termos aumentando o nosso vocabulário [14]. Para uma melhor eficácia da captura dos temos buscamos fazer pesquisas que contivessem a expressão "tissue source such as"e buscar por relações na descrição de cada página retornada.

Uma limitação encontrada em ambas as apis foi a cobertura muito pequena. Por se tratar de um domínio muito específico poucas sentenças foram retornadas impedindo que fosse possível obter bons resultados. A api do Google foi a que apresentou os melhores resultados e conseguimos obter apenas

27 sentenças diferentes nos testes realizados. Outra limitação encontrada foi ao tentar realizar buscas exatas com a api do Bing. Diferentemente da api do Google, ela não apresenta um parâmetro permita fazer buscas exatas, isso fez com que os resultados obtidos tivessem uma boa precisão tornando a api de pouca utilidade.

3.2 Ontologias

A segunda técnica utilizada foi a extração de termos de ontologias. A ideia consiste em extrair os termos de ontologias relacionadas a células e tipos de tecidos. Visto que muitos desses termos são empregados apenas naquele contexto, ontologias são ótimas fontes de dados para criação desses vocabulários.

```
In [30]: cells_type

Out[30]: ['cell',
          'primary cultured cell',
          'obsolete immortal cell line cell',
          'native cell',
          'obsolete cell by organism',
          'fibroblast neural crest derived',
          'neuronal receptor cell',
          'early embryonic cell',
          'migratory cranial neural crest cell',
          'obsolete fusiform initial',
          'cultured cell',
          'migratory trunk neural crest cell',
          'obsolete cell by class',
          'obsolete dentine secreting cell',
          'germ line stem cell',
          'male germ cell',
          'male germ line stem cell',
          'spermatocyte',
          'spermatid',
```

Figura 3.2: Exemplo de sentenças encontradas na Cell Ontology

Para isto, utilizamos duas ontologias. A Cell Ontology que trás informações sobre dados relacionadas a células e a Ktao-merged [1] que contém

[1] https://raw.githubusercontent.com/KPMP/KTAO/master/src/ontology/ktao-

informações sobre tipos de tecidos. Essas informações se encontram em formato OWL contendo termos aplicados aqueles domínio. A figura 3.2 mostra algumas termos que pertencem a Cell Ontology e foram extraídos dessa ontologia. Para extraí-las utilizamos um módulo Python chamado Alton [15] que faz o parser desse formato e permite que seja extraído os termos. Utilizando essa técnica foi possível obter 2319 tipos de células e 5251 tipos de tecido. Após a extração desses termos enviamos para análise do Prof. Dr. Ivan G. Costa que, junto com sua equipe, analisou os resultados e comprovou serem confiáveis.

3.3 Extrator de sentenças alvo

Com o objetivo de aumentar o nosso vocabulário, utilizamos os dados indexados extraídos do GEO [2] e algumas palavras já contidas no nosso vocabulário. No capitulo 4 detalhamos o pipeline necessário para que os dados ficassem no formato desejado para esta etapa do pipeline. Nosso objetivo nesta etapa é extrair sentenças dessa base de dados indexados que tenham potencial de conter informações relevantes para o CRF conseguir realizar boas extrações.

Para conseguir sentenças com maior qualidade dividimos essa etapas em processos menores. Começamos pela recuperação de sentenças, a seguir filtramos as sentenças encontradas e, por fim, formatamos as sentenças para que fiquem no formato adequado para as fazes seguintes. A seguir, descrevemos as subetapas desse processo.

3.3.1 Recuperação de sentenças

Utilizando utilizamos os termos do vocabulário como entrada fizemos buscas nos textos indexados utilizando Lucene [16]. Esse framework permite recuperar informações que foram indexadas por ele e fazer uma série de pesquisas de modo eficiente. Como resultado desta etapa, temos uma série de sentenças. Porém, analisando cuidadosamente percebemos que nem todas estavam devidamente relacionadas com os termos pesquisados. Para resolver este problema foi criada a etapa de filtro de sentenças que é descrito em sequência.

---

merged.owl

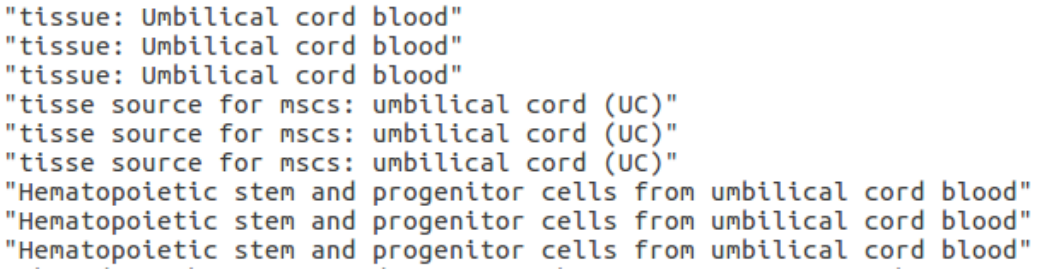

Figura 3.3: Sentenças antes de serem processadas

A figura 3.3f mostra como fica um conjunto de sentenças recuperadas da base de dados indexadas, porém sem ter passado pelas etapas de processamento.

3.3.2 Filtro de sentenças

Podemos identificar que alguns sentenças não tinham um casamento exato com o padrão pesquisado. Além disso, percebemos que tinham varias sentenças repetidas. Por isso, utilizando o grep [17] foram feito pesquisas por ocorrências exatas e também removemos duplicatas. Assim, embora a quantidade de sentenças resultante seja menor, teremos uma representação mais eficaz dos dados.

3.3.3 Formatando saída

Percebemos que cada sentença estava entre aspas e muitas vezes sem um ponto final. Isso estava atrapalhando o resultado das etapas posteriores. Por isso, como última subetapa, formatamos os dados de saída removendo as aspas e adicionando ponto final nas sentenças que não possuíam.

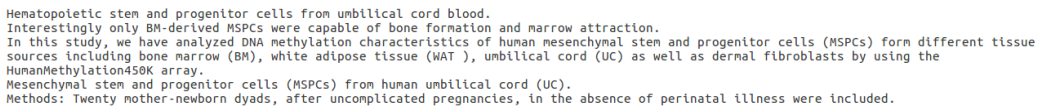

Figura 3.4: Sentenças após etapas extração das sentenças alvo

Como pode ser observado na figura 3.4, algumas sentenças são removidas ao final do processo, bem como as sentenças duplicadas. Ao final do processo de extração de sentenças, obtivemos um conjunto de dados úteis, limpos e

formatados.

3.4 Distant Supervision

Para a realização da rotulagem automática dos dados foi utilizado uma variação do Distant Supervision [18] que utiliza o Soft TF-IDF para realizar o casamento de padrões.

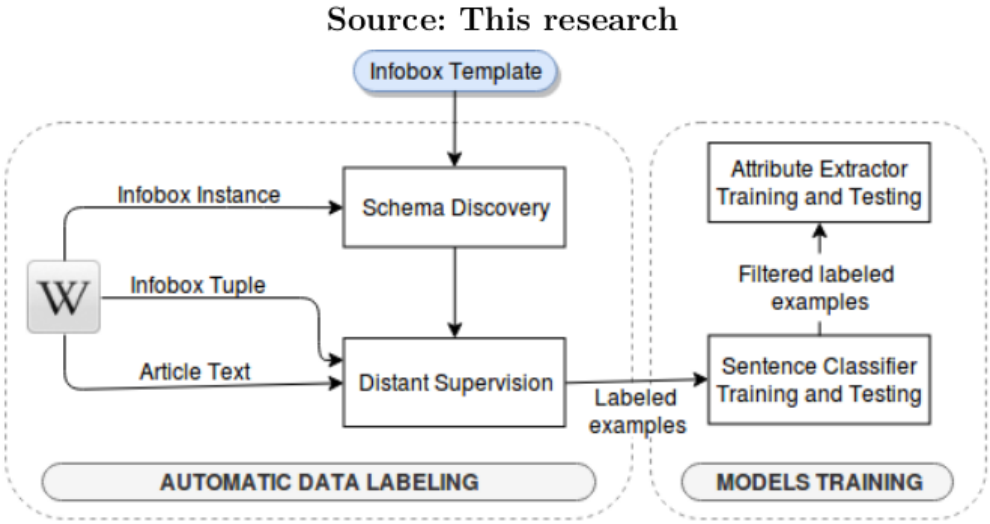

Figura 3.5: Fluxograma de extração da informação utilizando Distant Supervision

O fluxograma da figura 3.5 descreve as etapas utilizadas para a extração da informação de páginas do Wikipédia utilizando um templante de entrada. Embora esse fluxograma tenha vários módulos para este projeto utilizamos apenas o Distant Supervision. Detalhamos ele a seguir.

O Distant Supervision (DS) recebe como entrada um template, Schema Discovery, que pode receber dois componentes como entrada. Primeiro, um domínio e o segundo é uma instância daquele domínio. Como no nosso caso temos apenas um domínio analisado, tipos de célula, deixamos o primeiro campo em branco e preenchemos o segundo com o nome da célula que utilizar na hora de fazer a rotulagem do conjunto dos dados. A saída deste módulo é uma conjunto de dados rotulados automaticamente.

Para este projeto foi necessário fazer algumas alterações no projeto inicial. O DS proposto retorna as sentenças que melhor preenchem os campos requeridos no esquema. Porém, nosso objetivo é ter um conjunto de sentenças rotuladas de um único domínio. Por isso, tivemos de modificar a quantidade de sentenças retornadas. Inicialmente uma função definia a melhor sentença de acordo com o Soft TD-IDF, porém para a nossa aplicação modificamos de modo que fosse retornado todas as sentenças que tivessem sido classificadas com um certo grau de confiabilidade baseado do Soft TD-IDF. O grau foi arbitrariamente definido como 0.5. Desse modo uma grande quantidade de sentenças pode ser classificada automaticamente.

3.5 CRF

A implementação do CRF foi utilizando Sklearn-crfsuite [19], uma wrapper que encapsula as funcionalidades do CRFsuite[20] para que seja possível utilizar os módulos do Scikit-learn junto com está biblioteca. Os dados de entrada consistem nas sentenças que foram rotuladas pelo DS. A partir dessas sentenças, características são extraídas delas. Entre as varias características que foram usadas para treinar o classificador, podemos citar como exemplo a verificação se a palavra começa com letras maiúsculas, se ela contém números, se ela contém underline (_) e se a palavra é uma palavra de ligação (stopword).

Além disso, para a análise do desempenho foi utilizada o módulo Metrics [19] para cálculo de acurácia, precisão e cobertura.

Capítulo 4

Experimentos e resultados

Buscando avaliar os resultados do nosso experimentos utilizamos a base de dados da GEO para o experimento e avaliação de seus resultados. Porém, está base não se encontra em um formato próprio para os experimentos desejados. Por isso, uma série de atividade tiveram de ser realizadas com o objetivos de tornar os dados próprios para uso na nossa solução.

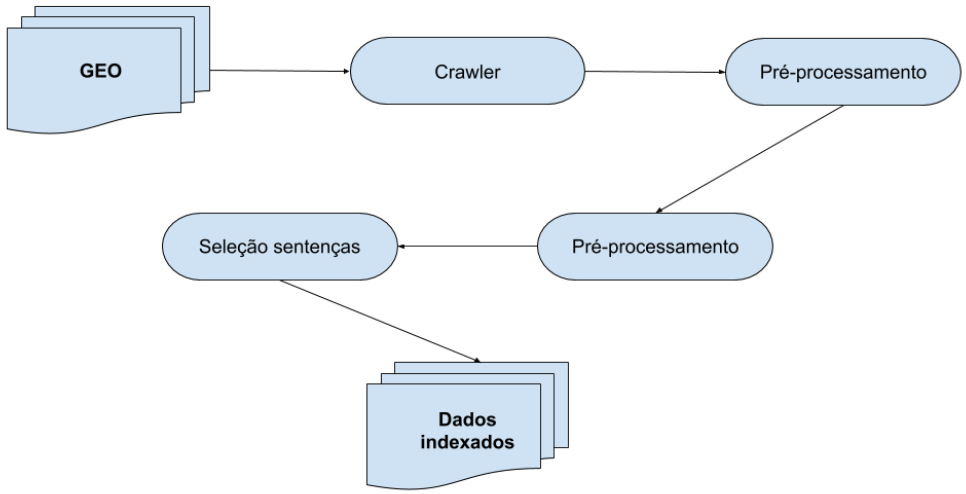

Figura 4.1: Fluxograma da criação da base de dados

Na figura 4.1 podemos observar o fluxograma que descreve as etapas seguidas para criação dos experimentos. A seguir descrevemos o passo a passo do pipeline utilizado para a criação da base de dados utilizada para o teste.

4.1 Aquisição dos dados

A base de dados para teste foi criada a partir do Gene Expression Omnibus (GEO) [2], que hospeda um grande número de experimentos de arrays de DNAm, porém eles se encontram em formato de texto livre, por sua vez desestruturados.

Uma série de etapas foi necessária para transformar esse dados de modo que fosse possível utiliza-los. Inicialmente, utilizando scripts em python baixamos todos os dados. Após isso, tivemos uma etapa de pré-processamento onde removemos markeups, extraímos sentenças que poderiam ser relevantes e indexamos as sentenças utilizando Lucene [16].

A seguir descrevemos os processos utilizados em cada etapa.

4.1.1 Baixando os dados

A GEO disponibiliza uma conjunto de dados separados em grupos de acordo com números de série. Nossa primeira etapa foi criar um crawler capaz de baixar os documentos dentro de cada grupos determinado. Nos foi dado um conjunto de dados que deveriam ser utilizados e seguindo esses ponteiros o crawler identificou e baixou cada um deles. Percebemos que a url que identificava cada documento seguia um padrão e a partir de modificação nessa url pudemos obter esses dados.

4.1.2 Pré-processamento

Após baixar e descomprimir os documentos, foi necessário fazer algumas etapas de pré-processamento. Primeiro, separamos o texto em linhas. Consideramos como uma linha, as sentenças que estavam dentro de uma mesma etiqueta. Logo após, removemos todas as etiquetas que estavam em cada linha. Percebemos que algumas sentenças eram tão pequenas que não eram capazes de conter informações relevantes. Por isso, criamos uma filtro que removia sentenças menores seguindo um dado limiar de 3 palavras, ou seja, linhas com menos de 3 palavras foram descartadas.

4.1.3 Seleção de possíveis sentenças candidatas

O próximo passo foi selecionar sentenças candidatas. Foi utilizado o framework Python NLTK para realização de algumas atividades nessa etapa. Primeiramente, separamos as informações das linhas agora no formato de sentenças. Em seguida, as sentenças foram filtradas de modo a conter apenas sentenças que possuíam verbos. Em ambas as etapas foi utilizado funções do NLTK.

O resultado final dessa etapa foi um conjunto de arquivos onde cada arquivo continha uma sentença verbal.

4.1.4 Indexação e busca

Para este processo utilizamos o framework Java chamado Lucene [16] que facilita a atividade de indexação de arquivos. Com ele pegamos cada um dos arquivos gerados no arquivo anterior e indexamos.

Logo após, ainda utilizando o Lucene fizemos buscas utilizando os três termos sementes relacionado a tipos de células que nos foi passado: Bone Marrow, Adipose Tissue e Umbilical Cord. Essas buscas resultaram em três arquivos, cada um contendo informações sobre um tipo de tecido.

Após uma primeira rodada de testes, percebemos que seria necessário mais tipos de célula, por isso repetimos o processo acima para os seguintes célula: epithelial,fibroblast, kidney, neural cell, precursor cell, e stem cell. Houve a tentativa de recuperar informação para mais alguns tipos de células, porém devido a baixa quantidade de sentenças elas foram descartadas da análise. Entre elas podemos citar neuronal receptor cell, early embryonic cell, neuroplacodal cell, apocrine cell e totipotent stem cell. Todas retornaram menos que 8 sentenças cada e algumas após remover as duplicações ficavam com cerca de duas ou três sentenças.

4.2 Extração da informação utilizando CRF

Para a extração da informação treinamos um classificador CRF para que ele aprendesse os padrões que identifique os padrões de uma expressão que contém informações sobre tipos de tecidos ou células.

4.2.1 Separação dos dados

A partir do conjunto de dados rotulados pelo Distant Supervision, cada tipo de célula separado em um arquivo diferente, separamos 1 conjunto para teste e os outros para treino. No total tínhamos 9 arquivos que somados davam 586 sentenças.

4.3 Resultados

Conforme explicitado na seção anterior, dos 9 tipos de células analisadas, foi feito a separação 8:1 onde 8 arquivos foram utilizados para treino e 1 para teste. Foi implementado uma variação do K-fold Cross Validation que permitiu que todos os arquivos fossem utilizados para teste apenas uma vez. A partir daí tiramos a média da acurácia, precisão e cobertura. Os resultados são mostrados na figura 4.2. A classe pos está relacionada com palavras referentes a tipos de células enquanto a neg refere-se a todas as outras. Além disso foi feito testes variando alguns parâmetros como o valor de L1, L2 e o algoritmo de treinamento.

	Acurácia	Precisão pos	Precisão neg	Cobertura pos	Cobertura neg
alg = lbfgs c1 = 0.1 c2 = 0.1	**92,26%**	28,97%	92,98%	4,42%	99,15%
alg = lbfgs c1 = 0.01 c2 = 0.01	92,07%	28,12%	93,04%	5,42%	98,85%
alg = l2sgd c2 = 0.1	92,18%	**30,08%**	93,06%	5,42%	**98,97%**
alg = l2sgd c2 = 0.01	92,04%	28,22%	**93,07%**	**5,69%**	98,80%

Figura 4.2: Desempenho médio do modelo após análise

De acordo com a figura 4.2 podemos observar que a variação nos parâmetros fez com que as métricas variassem. Utilizando o Stochastic Gradient Descent com regularização no termo L2 (l2gd) e l2 (c2) de 0,01 podemos observar que a precisão na classe de interesse (pos) foi a melhor. Porém, a melhor cobertura na classe de interesse foi com o mesmo algoritmo, porém com o valor de c2 de 0,01. Dado essas variações iremos seguir a nossa análise com a amostra

que apresentou a melhor acurácia, a utilizando o algoritmo que utiliza Gradiente descendente usando o método L-BFGS.

Embora o modelo apresente uma acurácia 92,3% ao observarmos a precisão e a cobertura por classe podemos observar que esse resultado é fortemente influenciado pela classe com as informações não referentes a tipo de células (neg) que é maior na base. Porém, como nosso objetivo é identificar as classes que possuem informações sobre tipos de célula (pos) a precisão e a cobertura nas classes pos são os principais pontos que devem ser observados.

Dito isso, podemos observar que os resultados foram baixos, com 28,97% de precisão e 4,42% de cobertura. Isso pode ter ocorrido por alguns motivos. Primeiro, devido a pequena quantidade de sentenças analisadas o modelo não conseguiu convergir e extrair com qualidade as características das sentenças. Outra possibilidade é que devido a variabilidade das sentenças o modelo não conseguiu extrair um padrão generalista o suficiente para conseguir capturar os tipos de sentenças. Para ambos os casos acredito que a solução seja similar, seria necessário uma amostra maior de dados, para que o modelo conseguisse extrair melhor as características das sentenças e conseguir convergir melhor.

Capítulo 5

Conclusão

Neste trabalho buscamos construir uma base de dados biológicas a partir de extrações utilizando diversas abordagens, como extrações de sentenças utilizando buscadores, através de ontologias, e de extração de informação de sentenças da base de dados biológicas GEO [2]. Para isso diversas técnicas foram explicitadas e implementadas.

Foi testada variações de parâmetros do CRF e o melhor classificador, segundo a acurácia, teve 92,26%, porém quando levamos em conta o a média da precisão, 29% e da cobertura, 4% podemos perceber que nas classes alvo ainda podem ser feitas melhorias para alcançar melhores resultados.

Buscando melhor o desempenho dos resultados, possíveis trabalhos futuros podem abordar modo de melhorar o classificador. Talvez por aumentar o tamanho da base por realizar extrações de outros tipos de células ou por aumentar a quantidade de parâmetros que obtém características das palavras. Outro possível trabalho futuro consiste em aumentar essa expansão para tipos de tecidos. Termos foram extraídos de Ontologias mas não foi possível realizar experimentos com eles.

Referências

[1] M. Craven, J. Kumlien et al., "Constructing biological knowledge bases by extracting information from text sources.", em *ISMB*, vol. 1999, 1999, pp. 77–86.

[2] T. Barrett, S. E. Wilhite, P. Ledoux, C. Evangelista, I. F. Kim, M. Tomashevsky, K. A. Marshall, K. H. Phillippy, P. M. Sherman, M. Holko et al., "NCBI GEO: archive for functional genomics data sets—update", *Nucleic acids research*, vol. 41, nº D1, pp. D991–D995, 2012.

[3] S. Patwardhan e E. Riloff, "Effective information extraction with semantic affinity patterns and relevant regions", em *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007.

[4] R. Snow, D. Jurafsky e A. Y. Ng, "Learning syntactic patterns for automatic hypernym discovery", em *Advances in neural information processing systems*, 2005, pp. 1297–1304.

[5] Wikipédia, *Ontologia (ciência da computação) — Wikipédia, a enciclopédia livre*, [Online; accessed 11-maio-2019], 2019. endereço: https://pt.wikipedia.org/w/index.php?title=Ontologia_(ci%C3%AAncia_da_computa%C3%A7%C3%A3o)&oldid=55102955.

[6] E. A. M. Morais e A. P. L. Ambrósio, "Ontologias: conceitos, usos, tipos, metodologias, ferramentas e linguagens", *Relatório Técnico–RT-INF-001/07, dez*, 2007.

[7] M. Mintz, S. Bills, R. Snow e D. Jurafsky, "Distant Supervision for Relation Extraction Without Labeled Data", em *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2 - Volume 2*, sér. ACL '09, Suntec, Singapore: Association for Computational Linguistics, 2009, pp. 1003–1011, ISBN: 978-1-932432-46-6. endereço: http://dl.acm.org/citation.cfm?id=1690219.1690287.

[8] J. Piskorski e R. Yangarber, "Chapter 2 Information Extraction : Past , Present and Future", 2018.

[9] J. R. Finkel, T. Grenager e C. Manning, "Incorporating non-local information into information extraction systems by gibbs sampling", em *Proceedings of the 43rd annual meeting on association for computational linguistics*, Association for Computational Linguistics, 2005, pp. 363–370.

[10] G. Zhou e J. Su, "Named entity recognition using an HMM-based chunk tagger", em *proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Association for Computational Linguistics, 2002, pp. 473–480.

[11] J. Lafferty, A. McCallum e F. C. Pereira, "Conditional random fields: Probabilistic models for segmenting and labeling sequence data", 2001.

[12] A. Nascimento, *Criação de uma base de dados biológica a partir de extração da informação de bases textuais*, 2019. endereço: %5Curl%7Bhttps://github.com/amnv/projetoIC%7D.

[13] J. Moreira e A. Nascimento, *Rotulação automática de bases de dados utilizando Distant Supervision*, 2019. endereço: %5Curl%7Bhttps://github.com/amnv/sentence_scoring%7D.

[14] M. A. Hearst, "Automatic acquisition of hyponyms from large text corpora", em *Proceedings of the 14th conference on Computational linguistics-Volume 2*, Association for Computational Linguistics, 1992, pp. 539–545.

[15] M. Larralde, *althonos/pronto: v0.10.2*, nov. de 2017. DOI: 10.5281/zenodo.1061770. endereço: https://doi.org/10.5281/zenodo.1061770.

[16] Wikipedia contributors, *Apache Lucene — Wikipedia, The Free Encyclopedia*, [Online; accessed 25-June-2019], 2019. endereço: https://en.wikipedia.org/w/index.php?title=Apache_Lucene&oldid=902590967.

[17] Wikipédia, *Grep — Wikipédia, a enciclopédia livre*, [Online; accessed 18-abril-2019], 2019. endereço: https://pt.wikipedia.org/w/index.php?title=Grep&oldid=54866301.

[18] J. M. da Silva, "Extracting Structured Information from Text to Augment Knowledge Bases", Master's Thesis, UFPE, Recife - PE, Brazil, 2019.

[19] *sklearn-crfsuite*, https://sklearn-crfsuite.readthedocs.io/en/latest/, Accessed: 2019-06-12.

[20] *CRFsuite*, http://www.chokkan.org/software/crfsuite/, Accessed: 2019-06-12.