EXTRAÇÃO DE ACRÔNIMOS E SEUS SIGNIFICADOS COM MODELOS OCULTOS DE MARKOV

Bruno Adam Osiek

DISSERTAÇÃO SUBMETIDA AO CORPO DOCENTE DA COORDENAÇÃO DOS PROGRAMAS DE PÓS-GRADUAÇÃO DE ENGENHARIA DA UNIVERSIDADE FEDERAL DO RIO DE JANEIRO COMO PARTE DOS REQUISITOS NECESSÁRIOS PARA A OBTENÇÃO DO GRAU DE MESTRE EM CIÊNCIAS EM ENGENHARIA DE SISTEMAS E COMPUTAÇÃO.

Aprovada por:

Prof. Geraldo Bonorino Xexéo, D. Sc.

Prof. Luis Alfredo Vidal Carvalho, D. Sc.

Prof. Geraldo Zimbrão da Silva, D. Sc.

Dra. Adriana Santarosa Vivacqua, D. Sc.

RIO DE JANEIRO, RJ – BRASIL

MARÇO DE 2008

OSIEK, BRUNO ADAM

Extração de acrônimos e seus significados com modelos ocultos de Markov [Rio de Janeiro] 2008

XI, 110 p. 29,7 cm (COPPE/UFRJ, M.Sc., Engenharia de Sistemas e Computação, 2008)

Dissertação - Universidade Federal do Rio de Janeiro, COPPE

1. Extração da informação
2. Processamento de linguagem natural
3. Inteligência artificial

I. COPPE/UFRJ II. Título (série)

À minha mulher, Daniela. Ao meu irmão, Victor. Sem eles eu não teria chegado aqui.

Agradecimentos

Ao Prof. Luis Alfredo Vidal Carvalho por ter me iluminado o caminho, pela amizade, pela atenção que me dispensou no momento mais difícil da minha vida e por todas as portas que me abriu.

Ao Prof. Geraldo Bonorino Xexéo por ser amigo, pela forma generosa com a qual me recebeu, pelo suporte que me deu no momento mais difícil da minha vida e pela orientação neste trabalho.

Ao amigo Dr. Eduardo Aguilar pelas aulas que me deu, pelos seminários semanais, por ter me escutado e pelas garrafas de vinho que comigo tomou.

Ao Prof. Sérgio Excel por ter me mostrado um mundo não cartesiano e por ter feito as perguntas que fez.

A toda a linha de Banco de Dados pelo carinho com o qual fui recebido.

Resumo da Dissertação apresentada à COPPE/UFRJ como parte dos requisitos necessários para a obtenção do grau de Mestre em Ciências (M. Sc.)

EXTRAÇÃO DE ACRÔNIMOS E SEUS SIGNIFICADOS COM MODELOS OCULTOS DE MARKOV

Bruno Adam Osiek

Março/2008

Orientador: Geraldo Bonorino Xexéo

Programa: Engenharia de Sistemas e Computação

Esta dissertação apresenta uma solução para a extração automática de acrônimos e seus significados de texto não estruturado. Baseado em um conjunto de regras definidas através de expressões regulares, a solução identifica no texto uma lista de termos que podem ser acrônimos e outra lista com expressões que podem ser seus respectivos significados. Uma função probabilística, baseada em modelos ocultos de Markov, calcula a probabilidade de cada uma dessas expressões emergirem de um termo. O significado de um termo consiste na expressão que obtiver a maior probabilidade. Os resultados alcançados sugerem que tal função probabilística entre o acrônimo e seu significado existe e que HMM consiste num método adequado para sua implementação, sendo esta a principal contribuição deste trabalho.

Abstract of Dissertation presented to COPPE/UFRJ as a partial fulfillment of the requirements for the degree of Master of Science (M.Sc.)

AUTOMATIC EXTRACTION OF ACRONYMS AND THEIR MEANING USING HIDDEN MARKOV MODELS

Bruno Adam Osiek

March/2008

Advisor: Geraldo Bonorino Xexéo

Department: Systems and Computer Engineering

This dissertation proposes a solution for automatically extracting acronym/meaning pairs from unstructured text. Based on a set of rules developed with regular expressions, the solution identifies within the text a list of acronym candidates and another list of expressions that could be their respective meaning. A probabilistic function implemented with Hidden Markov Models associates the correct tuple <acronym, meaning>, based on the premise that the correct meaning is the expression within the list with the highest probability to emerge from the acronym. The results achieved suggest that this probabilistic function exists and that Hidden Markov Models is an adequate method to implement it. This function consists in this dissertation major contribution.

Índice

Índice de figuras

Índice de tabelas

Capítulo 1 - Introdução

1.1 A motivação

É alta a freqüência com que novos acrônimos surgem. Normalmente são termos que referenciam um conceito ou uma entidade específica de uma área de conhecimento. O problema emerge quando a pessoa que comunica (o emissor) assume que a pessoa comunicada (o receptor) conhece o termo, não especificando a sua forma expandida na sua primeira aparição no discurso. Quando o receptor não reconhece um termo, faz uso de dicionário ou glossário para resolver a dúvida. O problema se concretiza, então, quando o termo não consta nos instrumentos de resolução. A sua solução consiste em manter atualizados dicionários ou glossários.

PUSTEJOVSKY et al. (2001) reportaram que aproximadamente 40.000 novos resumos são inseridos mensalmente na base da Medline. Ao percorrerem 40.956 resumos incluídos em apenas um mês (não especificado), encontraram 9.272 acrônimos distintos que não constavam na base de dados UMLS (Unified Medical Language System).

ZAHARIEV (2004) descreve que o maior dicionário de acrônimos na língua inglesa teve 4 edições num espaço de 3 anos e contém mais de 450.000 entradas. Também apresenta os números do site Acronym Finder (http://acronymfinder.com) que possuía em 2004 mais de 330.000 acrônimos, com 196 inclusões diárias. No momento da elaboração desta dissertação este mesmo site continha mais de 560.000 acrônimos inseridos manualmente e mais de 4.000.000 inseridos de forma automática, isto é, extraídos automaticamente.

Não se encontrou durante a fase de pesquisa dados referentes à língua portuguesa. O site http://www.siglas.com.br (similar brasileiro ao Acronym Finder) não quantificava o tamanho da sua base.

Pode-se inferir, pelos números apresentados, que manter manualmente um dicionário atualizado de acrônimos consiste numa grande tarefa e, conseqüentemente, dispendiosa, sendo esta a motivação por trás deste trabalho.

1.2 A origem e a evolução dos acrônimos

A comunicação humana é mais complexa do que qualquer outra por envolver o uso de linguagem. O que a faz tão poderosa é a combinação de três habilidades: a

primeira consiste em recursos para formar um conjunto muito grande de símbolos distintos; a segunda reside na capacidade sintática de combinar estes símbolos, resultando em um número infinito de mensagens; e a terceira é a competência semântica para comunicar estados das coisas que transcendem o real, ou seja, estados imaginários ou hipotéticos (JOHNSON-LAIRD, 1988).

O acrônimo é um símbolo da linguagem. Resulta do recurso lingüístico de concatenar as letras, normalmente as iniciais, de um conjunto de palavras (que também são símbolos da linguagem).

Nesta dissertação este conjunto de palavras, que consiste no significado do acrônimo, é denominado de expansão. Este significado pode ter cunho institucional (SRF é acrônimo para Secretaria da Receita Federal), eufemístico (DND que reduz de nada) ou de outra natureza (HENRIQUES, 2007).

Não se encontrou na literatura uma referência à origem dos acrônimos, mas sabe-se que são utilizados há mais de dois mil anos. Em hebraico o nome da bíblia é תנ"ך [1] (pronuncia-se[2] [tanach][3]) que consiste num acrônimo formado pelas primeiras letras da expansão תורה נב"אים כתובים (pronuncia-se [torah][nevi´im][ketuvim]). Esta expansão é constituída do nome de cada uma das suas três divisões que são, em tradução livre, "Os Cinco Livros" ou "Pentateuco", "Livro dos Profetas" e "Livro das Escrituras".

Os antigos romanos cunharam o símbolo SPQR para designar "Senatus Populusque Romanus" (Senado do Povo de Roma) e nas primeiras pinturas cristãs do Crucifixo está o acrônimo INRI que resume a expressão em latim "Iesus Nazarenus Rex Iudæorum" (Jesus de Nazaré Rei dos Judeus) (ZAHARIEV, 2004).

A proliferação do seu uso aconteceu no século XX e continua ainda hoje. Vários fenômenos podem explicar tal fato:

- A rápida evolução das tecnologias de informação e comunicação aumentou o volume de mensagens trocadas à distância e, visando eficiência, sempre se buscou comprimi-las. Acrônimo é um recurso lingüístico alinhado com este objetivo.

---

[1] Em hebraico lê-se da direita para a esquerda.

[2] Utilizou-se para a construção fonética dos nomes o International Phonetic Alphabet (IPA). Porém a escolha dos fonemas coube ao autor desta dissertação cujo conhecimento de hebraico é limitado.

[3] O "ch" é pronunciado como o som da letra **r** em **R**oma.

- A conjunção de palavras é utilizada para referenciar novas entidades, dentre elas tecnologias, cuja expansão recente resultou num grande acréscimo no número de acrônimos.
- As pesquisas na área biomédica os têm utilizado para referenciar suas novas descobertas.
- A comunicação síncrona escrita ("chat-room") demanda rápida digitação. Por outro lado os "torpedos", nome popular dado a Short Message System (SMS), são escritos através de uma interface desenvolvida para digitar números e adaptadas para a entrada de caracteres do alfabeto. Acrônimos são sinérgicos com as demandas dos usuários destes canais de comunicação.

Segundo HENRIQUES (2007) um fenômeno interessante a se destacar consiste na evolução da utilização de um acrônimo. Escreve ainda que em tese todos eles podem dar origem a itens lexicais ortograficamente convencionais, como tevê (televisão < TV < tevê). Alguns, além de se tornarem léxicos, têm sua conjunção original esquecida, como no caso do Laser (Light Amplification by the Stimulated Emission of Radiation).

1.3 O contexto

O número de documentos disponíveis hoje eletronicamente é muito grande. Podem estar na Internet, numa rede local ou mesmo em um computador pessoal. Além disto, podem estar gravados em formatos distintos como HTML, XML, PDF, DOC ou e-mails e seu conteúdo estar em diversas línguas.

A solução descrita nesta dissertação extrai acrônimos de um dado conjunto de documentos em formato HTML, XML, PDF, e-mail e texto simples sem formatação (TXT) escritos em qualquer língua que utilize caracteres Unicode Latin-1, conforme descritos em http://www.unicode.org/charts/PDF/U0080.pdf.

Esta solução apenas extrai os termos e as definições sem se preocupar com a busca e a recuperação dos documentos onde estão inseridos.

É importante observar que embora exemplos nas línguas portuguesa, espanhola, francesa e alemã tenham sido utilizados com sucesso, o sistema foi desenvolvido e testado somente para o inglês, pois apenas para esta língua encontraram-se os recursos lingüísticos computacionais prontos para serem utilizados no desenvolvimento da solução.

1.4 O objetivo

O objetivo do sistema de extração de acrônimos é extrair pares acrônimo-expansão de texto não estruturado, ou seja, texto escrito sem nenhuma estrutura formal pré-definida.

O processamento começa com a conversão dos documentos dos seus formatos originais em documentos de texto simples (TXT). Em seguida os termos e suas definições são localizados no documento, esta tarefa é denominada de identificação, e uma tarefa, denominada de resolução, conecta cada termo à sua respectiva definição.

Um conjunto de documentos onde os pares acrônimo-expansão foram manualmente identificados por um especialista servirá de base para a avaliação da solução. Esta avaliação se dará comparando as tuplas (ou pares) extraídas pelo sistema com aqueles identificados pelo especialista. Somente as iguais serão consideradas corretas.

1.5 A solução

Resumidamente esta consistiu na elaboração de um conjunto de cinco regras para identificar candidatos a acrônimos. Duas heurísticas diminuíram o espaço de busca das suas expansões no texto adjacente aos termos identificados. Em seguida o acrônimo e uma lista com as expansões identificadas são submetidos ao resolvedor de acrônimos, baseado em Hidden Markov Model (HMM)[4], que retorna o significado que consiste na expansão com a maior probabilidade de emergir do acrônimo (a solução é formalmente descrita no capítulo 4). Existindo uma expansão, a tupla <acrônimo, expansão> é formada, originando numa extração positiva.

1.6 Os resultados

As métricas de avaliação e comparação foram a precisão, a cobertura e o fator F. A solução alcançou uma precisão de 94%, uma cobertura de 88% e um fator F de 89,82%. Comparando-a com as demais soluções, que publicaram os resultados processando o mesmo corpus de acrônimos, esta obteve o maior fator F.

A comparação do desempenho não é conclusiva, pois não foi possível seguir o procedimento padrão de "benchmark" entre soluções distintas, uma vez que tanto as

[4] Em tradução livre para o português seria Modelos de Cadeias Ocultas de Markov. Nesta dissertação optou-se por utilizar a expressão em inglês.

soluções quanto a base nas quais estas soluções foram testadas não estavam disponíveis. Entretanto pode-se concluir que esta é no mínimo igual a das demais com uma contribuição que reside no uso de apenas 7 funções para resolver acrônimo, cinco regras para identificá-los no texto e duas heurísticas para diminuir o espaço de busca pela expansão no texto, dado o acrônimo. Esta contribuição é significativa, principalmente quando comparada com outra soluções que implementaram mais de uma centena de regras para executar a mesma tarefa.

1.7 A evolução do tema da tese

Originalmente este trabalho tinha como objetivo extrair automaticamente um conjunto de tuplas <termo, definição> de documentos escritos em português para formar um glossário da legislação brasileira que regulamenta a indústria de petróleo.

Numa análise preliminar o problema foi dividido em três classes. A primeira consistia no conjunto de tuplas cujos elementos encontravam-se adjacentes no texto, sendo que um deles entre parênteses. Um exemplo está no texto "A presidente do Supremo Tribunal Federal (STF) viajou para o Rio de Janeiro". Neste caso STF seria o termo e "Supremo Tribunal Federal" a sua definição. Chamou-se esta classe de "classe dos acrônimos".[5]

A segunda classe consistia nas tuplas cujos elementos encontravam-se em um aposto e a separação de ambos dava-se pelo caractere dois-pontos. Esta classe foi apelidade de "classe dos apostos".

Exemplo:

"Neste documento os termos a seguir têm as seguintes definições:

- Botijão de gás: cilindro metálico para acondicionar gás liquefeito de petróleo; e
- Derivados: produtos que se originam da destilação do petróleo."

As tuplas que comporiam o glossário seriam: <botijão de gás, cilindro metálico para acondicionar gás liquefeito de petróleo> e < derivados, produtos que se originam da destilação do petróleo>.

A terceira classe, denominada de "classe de texto livre", a definição do termo encontra-se em formato irrestrito. Um exemplo está no seguinte texto: "A atividade de distribuição de solvente consiste na compra, armazenagem, controle de qualidade e

[5] Nem todo termo neste formato é acrônimo, mas estes constituem a maioria, originando o nome da classe.

venda do produto". A tupla seria < A atividade de distribuição de solvente, consiste na compra armazenagem controle de qualidade e venda do produto>[6].

A pesquisa para resolver o problema de se identificar e extrair os elementos da classe acrônimos mostrou que somente a extração destes consumiria o tempo disponível para realizar este trabalho. Por esta razão, os demais termos desta classe, que não são acrônimos, juntamente com as classes restantes, não são contemplados nesta dissertação.

Neste trabalho encontra-se descrito um sistema que extrai apenas acrônimos e suas definições de documentos, desde que no texto estes se encontrem no formato descrito para a classe dos acrônimos, ou seja, os elementos da tupla estão adjacentes e necessariamente somente um deles entre parênteses.

1.8 A organização dos capítulos

No capítulo 2 descreve-se do que consiste o problema de extrair e resolver acrônimos. Definem-se alguns termos comuns na área de lingüística necessários ao entendimento do problema e da solução. Apresenta-se uma teoria geral de formação de acrônimos. Segue-se com uma breve descrição do que é e das diferentes abordagens para processamento de linguagem natural. Também se resume o que é extração da informação (EI), descreve-se um framework e uma arquitetura para desenvolvimento de sistemas de EI onde algumas técnicas de processamento de linguagem natural são utilizadas. O capítulo é encerrado com uma revisão bibliográfica com as soluções encontradas para extração e resolução de acrônimos.

No capítulo 3 descreve-se o que seja e como funciona um modelo oculto de Markov que é o algoritmo de base para a atividade de resolução de acrônimos. O capítulo prossegue com uma revisão bibliográfica da utilização bem sucedida de HMMs na tarefa de extrair informação de texto. Um resumo do que seja engenharia de texto, incluindo a descrição de um sistema para processar e extrair informação conclui este capítulo.

No capítulo 4 descrevem-se as razões para a utilização de HMM na atividade de resolução de acrônimos. Formaliza-se o problema e a solução proposta nesta dissertação.

---

[6] As vírgulas foram propositalmente retiradas da definição para não confundir o leitor na identificação dos elementos da tupla.

No capítulo 5 descreve-se formalmente a arquitetura de um sistema de extração de acrônimo seguida da implementação desenvolvida nesta pesquisa. Prossegue-se no capítulo com a descrição formal da metodologia utilizada no experimento que serviu de avaliação do sistema, incluindo uma análise quantitativa, qualitativa e uma comparação com outras soluções.

No capítulo 6 encontram-se as principais contribuições deste trabalho, as conclusões, os melhoramentos e os trabalhos futuros.

Capítulo 2 - Extração de acrônimos como uma atividade de extração da informação

Neste capítulo descreve-se o problema de extrair acrônimo. É fornecida uma lista de definições de termos utilizados nesta dissertação. Apresenta-se uma teoria universal para a formação de acrônimos. Resumem-se as áreas de conhecimento conhecidas como processamento de linguagem natural e extração da informação, sendo que para esta última apresentam-se um "framework" de atividades e uma arquitetura de sistema. Ao final é fornecida uma revisão bibliográfica com outras soluções para o tratamento do problema de se extrair acrônimos automaticamente de texto não estruturado.

2.1 Do que consiste o problema de extrair acrônimos

Como já discutido na seção 1.1 é grande o número de novos acrônimos cunhados nas mais diversas áreas do conhecimento. Somente na área biomédica surgem, segundo PUSTEJOVSKY et al. (2001), aproximadamente 40.000 novos termos mensalmente. Adicionalmente ZAHARIEV (2004) quantifica este problema com as 4 edições de um dicionário lançadas num período de 3 anos.

Uma conseqüência deste grande número de termos é a ambigüidade. Esta se caracteriza quando um mesmo acrônimo tem dois ou mais significados. Para o acrônimo ANP encontraram-se 3 expansões distintas em http://www.siglas.com.br e 176 em http://acronymfinder.com, sendo que a interseção entre os dois conjuntos é vazia. Portanto, apenas nestes dois bancos de acrônimos, existem 179 expansões distintas para ANP.

A desambigüação de acrônimos é um desafio na área de busca e recuperação da informação, pois se procurando por documentos que façam referência à Agência Nacional de Petróleo (ANP) não se deseja que no conjunto encontrado existam elementos que façam referência à ANP de Academia Nacional de Polícia (Polícia Federal).

Assim o problema de extrair acrônimos consiste em 4 tarefas principais:

- Identificação de um acrônimo, ou seja, identificar se um termo consiste ou não em um acrônimo;
- Identificação da expansão, que consiste em encontrar possíveis expansões em um contexto;
- Identificação da tupla <acrônimo, expansão>, que consiste na associação de um acrônimo com uma possível expansão; e
- Desambigüação, que trata de identificar a expansão correta de um acrônimo num dado contexto.

As três primeiras tarefas são alvo desta dissertação.

2.2 Terminologia

É necessária a definição de alguns termos no contexto desta pesquisa.

- **Acrônimo**: Não se encontrou um consenso entre os autores quanto a definição do que seja ou não um acrônimo. Freqüentemente é definido como sendo uma palavra formada pelas iniciais de outras (BORBA, 2004). Exemplo: STF (Supremo Tribunal Federal) é um exemplo perfeito desta definição. Porém em SPQR a letra Q representa a oitava letra de Populusque, ferindo a definição dada.
- **Sigla**: Letras ou sílabas iniciais dos vocábulos componentes de uma denominação ou título; e letra inicial, simples ou repetida, usada como abreviatura em medalhas, monumentos ou manuscritos antigos (BORBA, 2004). Exemplo: ONU (Organização das Nações Unidas).
- **Abreviação**: Redução da forma de uma palavra (BORBA, 2004). Não se encontrou um conjunto de regras que regem as formas possíveis de se reduzir uma palavra. Normalmente a abreviação é o resultado do truncamento da palavra como em "ex." para "exemplo". Uma abreviação é normalmente reconhecida como uma seqüência de letras seguida de um ponto no meio de uma frase.
- **Sintagma**: Unidade sintática constituída de uma estrutura binária cujos constituintes internos se relacionam para cumprir a função de comunicação pela linguagem; construção sintática (BORBA, 2004). É um termo estabelecido por Saussure que designa a combinação de elementos menores numa unidade

lingüística maior (HENRIQUES, 2007). Estes elementos menores são também chamados de constituintes. Na seqüência "O presidente da república sancionou a lei que regulamenta a Timemania." temos um sintagma constituído de dois elementos: sujeito (O presidente da república); e o predicado (sancionou a lei que regulamenta a Timemania). Neste caso este sintagma seria chamado de sintagma oracional.

- **Sintagma nominal (SN)**: Unidade sintática de uma sentença onde se agrupam as informações sobre um nome ou pronome que constitui o seu núcleo. Este constituinte é quem caracteriza a função sintática. Para efeito desta dissertação consideraremos SN somente aqueles cujo núcleo é um substantivo. Este SN é denominado de sintagma nominal reduzido em FREITAS et al. (2005). Normalmente fazem o papel de sujeito ou objeto direto e indireto. Na oração "O presidente da república sancionou a lei que regulamenta a Timemania." temos o SN "O **presidente**[7] da república" (núcleo em negrito), fazendo o papel de sujeito e o SN "a **lei** que regulamenta a Timemania", fazendo o papel de objeto direto.
- **Corpus**: Do latim e significa corpo.
- **Corpora**: O plural de corpus.

Nesta dissertação não faremos distinção entre acrônimo, sigla e abreviação. Por simplicidade trataremos como acrônimo qualquer palavra que se encaixe nas definições dadas para estes termos.

2.3 Regras para a formação de acrônimos

As definições apresentadas na seção 2.2 para acrônimo, sigla e abreviação não são suficientes para tratar da formação de todos os acrônimos encontrados em textos nas diversas áreas de conhecimento. O já mencionado acrônimo SPQR constitui um exemplo.

Em sua dissertação de doutorado ZAHARIEV (2004) tentou formalizar uma teoria universal, resumida na Tabela 2.1, para a formação de acrônimos. Sua teoria é composta por um conjunto de 15 regras (estabelecidas com o conhecimento do autor sobre acrônimos em 16 línguas) e de 7 hipóteses. Embora o português não faça parte do

[7] Em negrito esta o núcleo do sintagma nominal.

conjunto de línguas conhecidas, o autor testou estas regras num banco de acrônimos em português, verificando a sua completa cobertura.

Tabela 2.1 - Teoria universal de formação de acrônimo ZAHARIEV (2004)

Número	Regra	Descrição	Exemplo
(1)	Associação das iniciais	Cada palavra na expansão é representada no acrônimo por sua primeira letra.	ON para reduzir Observatório Nacional
(2)	Associação por morfema	A associação do caractere inicial de um morfema dentro de uma palavra da expansão com uma letra do acrônimo. Em português esta regra é bastante utilizada para termos de uso na medicina ou para gerar acrônimos com palavras compostas.	ALT para Alamina Aminitransferase
(3)	Associação por sílaba	Análoga a regra no. 2 somente trocando a primeira letra do morfema pela primeira letra da sílaba.	
(4)	Associação por grupo	Associação de um grupo de letras consecutivas em uma palavra da expansão com o mesmo grupo de letras no acrônimo.	Petrobras para Petróleo Brasileiro
(5)	Associação de caractere interno	A associação de um caractere interno de um termo da expansão com um caractere idêntico no acrônimo. Trata-se de uma generalização da regra no. 1	XML para E^{x}tensible Markup Language
(6)	Palavra funcional não associada	As palavras funcionais, como preposição, conjunção e artigo, não são consideradas na formação do acrônimo.	ABP para Academia Brasileira **de**[8] Letras

[8] Palavra em negrito não tem representação no acrônimo.

Número	Regra	Descrição	Exemplo
(7)	Não associar uma palavra precedida de pontuação	Não considerar na associação palavras na expansão precedida de certas pontuações. Na nossa pesquisa esta regra será desconsiderada, pois encontramos evidência significativa contraditórias, principalmente na área biomédica.	[Ca+]I para Intracellular Calcium Concentrations
(8)	Não associar uma palavra	Trata-se de uma generalização das regras de não associação (regras (2.6) e (2.7). São utilizada na formação de acrônimos derivados de expansões longas ou para facilitar sua pronúncia.	BANRISUL para Banco **do Estado do** Rio **Grande do** Sul
(9)	Duplicação para o plural	Duplicação de uma letra no acrônimo para caracterizar o plural de um termo (normalmente um sintagma nominal) na expansão.	MMBTU para Milhões de BTU
(10)	Associação simbólica	Associar um símbolo, caractere, morfema, grupo de caracteres, palavra ou expressão na expansão com caracteres ou grupo de caracteres no acrônimo, utilizando regras reconhecidas por membros de um meio profissional, social ou histórico. A associação simbólica também pode representar a construção de acrônimos gerados por mecanismos de soletração criativa.	A+ para Até mais
(11)	Migração	A associação de caracteres acentuados nos termos da expansão é representada sem o acento no acrônimo.	DTAE para Departamento de Tratamento de Água e Esgoto

Número	Regra	Descrição	Exemplo
(12)	Flexão	Em línguas com morfologia aglutinativa, grupos que representam morfemas inteiros podem ser associados no acrônimo de forma flexionada. Na nossa pesquisa desconsideraremos esta regra, apesar de termos exemplos de identificação e resolução positivas de acrônimos formados por palavras resultantes de morfologia aglutinativa em alemão.	
(13)	Associação em seqüência	Caracteres no acrônimo são associados em seqüência com os termos (símbolos, caracteres, palavras e expressões) na expansão. Palavras saltadas na expansão são também saltadas na mesma direção no acrônimo.	DET para Double Embryo Transfer
(14)	Inversão	A inversão acontece quando a seqüência dos caracteres no acrônimo não é igual a seqüência dos termos na expansão.	BTU para Unidade Térmica Britânica
(15)	Importação	Acrônimos podem ser importados diretamente de outra língua, ao invés de serem criados com a tradução da expansão. Isto é muito comum em áreas técnicas, como na tecnologia de informação.	HTTP e HTML

A proposta da solução descrita nesta dissertação substitui estas regras por uma função estocástica. Esta função é implementada com modelos ocultos de Markov.

As regras (1) e (6) da Tabela 2.1 dão conta da formação da maioria dos acrônimos na língua portuguesa. São listadas 3 hipóteses que levam os autores a ferirem a regra (1).

A primeira (chamada de associação para expansões curtas) seria a de que expansões com apenas dois termos gerariam muitos acrônimos ambíguos com apenas 2 letras. Como exemplo se dá QAV para Querosene de Aviação. A segunda hipótese

(chamada de pronunciabilidade) é a de que acrônimos facilmente pronunciáveis como palavras da língua são preferidos. BANESPA (para Banco do Estado de São Paulo) tem fonemas mais próximos aos das palavras em português (termina com uma vogal) do que BESP.

A terceira hipótese (chamada de conflito) é quando o acrônimo gerado já é utilizado no domínio. Mato Grosso (MT) geraria conflito com Minas Gerais (MG).

A regra (13) serve de base para o algoritmo de programação dinâmica que ZAHARIEV (2004) utilizou para desenvolver a sua solução na resolução de acrônimos. O autor menciona que este algoritmo não pode ser aplicado em todos os domínios e línguas, pois existe a regra (14) de inversão e, num estudo quantitativo, menciona que este problema é mais freqüente em línguas como alemão, finlandês, russo e muito menos freqüente no inglês.

Ainda na teoria universal de formação de acrônimos está descrita uma hipótese para justificar a inversão, baseado na análise sintática dos elementos de um acrônimo. O autor utiliza o acrônimo MFBM para "Thousand Board Feet Measure". Neste caso os sintagmas nominais "Thousand Feet" e "Board Measure" estão intercalados na expansão, mas arrumados corretamente no acrônimo, sendo esta arrumação sua justificativa para a inversão.

No exemplo da regra (14) esta hipótese não se configuraria, pois se trata de uma tradução para o português – regra (15) - de British Thermal Unit (neste caso houve uma coincidência das palavras nas duas línguas começarem com as mesmas letras). Encontrou-se nos resumos da Medline[9] o acrônimo SET para "Transfer of Single Embryo". Pode-se inferir que SET seria o acrônimo para "Single Embryo Transfer", mas a hipótese levantada para justificar a regra também não se aplicaria, pois se trata de apenas um sintagma nominal e não uma reordenação de dois ou mais sintagmas.

Um problema para a extração de acrônimos da regra (15) – Importação - é de que, normalmente, não são acompanhados da suas respectivas expansões.

2.4 Processamento e a compreensão da linguagem natural

Na literatura o processamento de linguagem natural é dividido em duas grandes tarefas: processamento de texto e processamento de voz. A segunda consiste em

[9] Banco de dados com informações bibliográficas de ciência da vida e biomédica (http://medline.cos.com/).

identificar as palavras através dos seus fonemas. Uma vez identificadas, são colocadas em forma de texto e segue o processamento de texto que descreveremos nesta seção.

Somos expostos diariamente a um grande volume de informação. Parte deste volume nos é pertinente. A separação do conjunto inicial em dois (um que contém as informações relevantes e outro que não as contém) facilitaria muito nossa iteração com o ambiente social no qual estamos inseridos. A linguagem natural é o sistema preferencial utilizado para a troca de mensagens. Compreendê-la consiste numa complexa tarefa de qualquer programa de computador cujo propósito seja auxiliar o agente humano em reagir adequadamente aos estímulos contidos nestas mensagens.

A compreensão da linguagem natural é o resultado de um processamento feito pelo aparato cognitivo humano, que hoje não é completamente compreendido. Entretanto sabemos algumas coisas sobre este processamento. Primeiro: palavras são utilizadas para simbolizar ações, conceitos, sejam estes concretos ou abstratos, e as relações entre ações diferentes, conceitos diferentes e entre ação e conceito. Segundo: representamos as ações, os conceitos e as relações de alguma forma em nossa mente. Aqui se denomina esta representação de modelo. Terceiro: utilizamos um conjunto de regras para formar enunciados com estas palavras. Normalmente o agente precede a ação e o objeto a sucede. Quarto: a compreensão se dá ao associarmos os conceitos, as ações a as relações contidas na mensagem com algum modelo conhecido. Não existindo, um novo modelo é criado. Muitas vezes este novo modelo é derivado de algum existente.

Assim define-se o processamento e a compreensão da linguagem natural como o processo de extrair (ou decodificar) o modelo da mensagem escrita ou falada e associá-lo com um modelo existente.

Existem duas abordagens para o processamento de linguagem natural, que são a determinística e a estatística.

A determinística é aquela baseada em regras. A gramática é o conjunto de regras que regem o funcionamento de uma língua e a morfologia consiste num subconjunto destas regras que teoriza sobre a estrutura, os processos de formação e classificação das palavras. Na sintaxe estão as regras de formação (ou codificação) dos enunciados. A semântica consiste no "ramo da Lingüística que estuda a significação das palavras e sua variação no tempo e no espaço, bem como a representação do sentido dos enunciados." (BORBA, 2004).

A abordagem estatística é aquela que se baseia na distribuição da freqüência na qual as palavras são usadas para inferir as regras.

2.4.1 O processamento estatístico de linguagem natural

Tanto a discussão filosófica que sustenta as hipóteses de se processar estatisticamente a linguagem natural quanto à discussão da validade destas hipóteses estão fora do escopo desta dissertação.

O trabalho aqui descrito parte de duas afirmações de MANNING (1999). A primeira é a de que o processamento estatístico de linguagem natural é melhor do que o determinístico para aprendizagem automática e para a desambigüação. A segunda é a de que a abordagem estatística objetiva atribuir probabilidades a eventos lingüísticos, permitindo distinguir o que é usual do que não é usual.

Já foi descrito que nesta dissertação entende-se que a formação de acrônimos consiste num recurso lingüístico (seção 1.2) e, abaixo, discute-se brevemente a ambigüidade.

Existe ambigüidade na linguagem. Considere-se a palavra "segundo" nas frases da Tabela 2.2.

Tabela 2.2 - Exemplo de ambigüidade morfológica

(a)	"**Segundo** pesquisa, são toleráveis os efeitos colaterais da estatina."
(b)	"Paulo não suportou a pressão de Eduardo e acabou a corrida em **segundo** lugar."
(c)	"Se reduzíssemos o tempo em que existe vida na terra para um minuto, o homem não teria completado o seu primeiro **segundo** de existência."

No exemplo (a) a palavra é classificada como preposição acidental. No (b) como numeral ordinal e no (c) como substantivo. Dada a palavra "segundo" não se pode determinar a sua classe morfológica sem o contexto na qual está inserida.

As palavras também podem ser ambíguas quanto ao seu significado. Na Tabela 2.3 descreve-se a ambigüidade do verbo "engolir".

Tabela 2.3 - Exemplo de ambigüidade quanto ao significado

(d)	"O técnico **engoliu** a desculpa dada pelo jogador para justificar seu atraso."
(e)	"João **engoliu** o remédio com dificuldade."

No exemplo (d) "engoliu" significa aceitar a desculpa com desagrado. No (e) "engoliu" tem o significado de tomar o remédio.

Segundo MESQUITA (1996) "Em português, é comum a construção de a oração obedecer a uma ordem sintática, ou seja, a disposição dos termos acontece de acordo com a sua respectiva função sintática". Ainda (MESQUITA, 1996) esta ordem pode ser direta, quando a disposição é a "natural e característica da Língua Portuguesa" ou inversa ou indireta, "que constitui um recurso expressivo para enfatizar algum termo da oração". Exemplos na tabela 2.4.

Tabela 2.4 - Exemplo de oração em ordem sintática natural e inversa

(f)	"Mariazinha comeu o bolo com as mãos."
(g)	"Com as mãos Mariazinha comeu o bolo."

No exemplo (f) os termos estão dispostos na sua ordem natural (sujeito -> verbo -> complemento). Aqui existe uma ambigüidade, pois "com as mãos" pode ser adjunto adverbial, modificando o verbo comeu, ou adjunto adnominal, modificando o substantivo "bolo" (podemos imaginar que as mãos constituem ornamentos do bolo, caracterizando-o). Já no exemplo (g) temos uma ordem inversa (adjunto adverbial -> sujeito -> verbo -> objeto). Neste caso, ao mudar a ordem, eliminamos a ambigüidade.

Assim, um sistema de processamento de linguagem natural precisa ser eficiente em desambigüar a classe gramatical de uma palavra ou o seu sentido. Precisa escolher entre possíveis análises sintáticas de uma oração para decidir sobre seu escopo semântico.

Em sistemas determinísticos (ou simbolistas) a maximização da cobertura é feita ampliando-se o conjunto das regras que por sua vez aumentam a ambigüidade. Segundo (MANNING, 1999 apud LAKOFF, 1987) "experiências com abordagens em inteligência artificial para a análise sintática e desambigüação, que procuram modelos para uma profunda compreensão de enunciados, mostraram que demora a conclusão da priorização de regras e restrições sintáticas elaboradas manualmente. Além da demora as regras estabelecidas manualmente não generalizam bem e são muito frágeis em face ao uso extensivo de metáforas."

A abordagem estatística busca resolver estes problemas através do aprendizado automático feito em corpora, buscando suas estruturas sintáticas e léxicas preferidas. Entendendo que existe um padrão no uso real da linguagem e de que as pessoas tendem a repetir expressões utilizadas por outras, entende-se que os métodos estatísticos

constituem numa boa alternativa para o problema da desambigüação, por serem robustos, generalizarem bem e terem comportamento adequado na presença de ruído ou de nova informação. Como seus parâmetros são aprendidos automaticamente de corpora, reduzem significativamente o esforço humano.

Em (LUGER, 2004 apud MAGERMAN, 1994) existe uma comparação entre o desempenho de analisadores sintáticos, programas que identificam as classes sintáticas dos termos de uma sentença, baseados em gramática (determinístico) e analisadores sintáticos estatísticos. Utilizando uma medida popular para avaliação deste tipo de aplicação, chamada de "crossing-bracket" (parênteses cruzados), o determinístico obteve o resultado de 69% e o estatístico de 78%. A gramática utilizada pelo primeiro foi o resultado de um conjunto de regras elaboradas por um lingüista ao longo de 10 anos de trabalho.

Nossa pesquisa apontou para a complementaridade entre as duas abordagens. Conforme (TURMO et al., 2006) os métodos estatísticos não são suficientes para o endereçamento de muitas tarefas envolvidas na extração da informação e precisam ser combinados com abordagens baseadas em conhecimento. Nesta pesquisa consistiram em regras para elencar termos candidatos a acrônimos e heurísticas para reduzir o espaço de procura das suas respectivas expansões.

2.5 A extração da informação

"Extração da informação é o processo de popular um banco de dados através de uma varredura superficial de um texto, procurando por ocorrências de uma classe particular de objetos ou eventos e a relação entre estes objetos e eventos" (RUSSELL, 2003). Quando particularizado, ou seja, o objeto é nomeado, suas propriedades são discriminadas e as relações são formalmente descritas temos um cenário para a extração.

Define-se o cenário para a extração de acrônimo aquele em que procura-se extrair tuplas em um texto onde cada uma contém dois elementos sendo um o acrônimo e o outro a sua expansão, onde elementos do segundo são representados de alguma forma no primeiro, tornando o acrônimo numa referência à expansão.

11/09/2007.

Brasil pode deixar de ser 'molenga das Bric', diz 'FT'

Plantão | Publicada em **11/09/2007** às 07h16m

O Brasil pode ter uma oportunidade de se livrar de sua "imagem de molenga" dentro das economias em desenvolvimento das chamadas Bric (Brasil, Rússia, Índia e China) esta semana, quando "o dado de crescimento no segundo trimestre deve chegar a 5,5% - mais do dobro da média dos últimos 15 anos", diz artigo na edição desta terça-feira do jornal britânico Financial Times.

Figura 2.1 - Exemplo de extração de acrônimo de texto[10]

Do cenário da Figura 2.1 extraem-se dois pares: <Bric, Brasil, Rússia, Índia e China> e <FT, Financial Times>. O banco de dados, como exemplo, seria um glossário da área de economia.

2.5.2 Um framework para a extração de informação

O desenvolvimento das tecnologias de extração de informação (EI) está ligado à conferência de compreensão de mensagem ou "Message Understanding Conference" (MUC) em inglês. Este conjunto de eventos que aconteceram entre 1987 e 1998 tinha como objetivo padronizar os conceitos, as tarefas de extração e suas respectivas métricas de avaliação, dando aos participantes uma base de estruturação, comparação, consolidação e troca de conhecimento. No MUC-7, segundo TURMO et al. (2006), existiam 5 tarefas cujos resultados eram avaliados:

- Reconhecimento de entidades (NE do inglês "Named Entity") consiste na tarefa de reconhecer conceito independente de domínio tais como nome de pessoas, organizações, locais, datas entre outros;
- Resolução de Correferência (CO do inglês "Coreference") consistia em reconhecer referências distintas de um mesmo elemento;
- Template de Elementos (TE do inglês "Template Element") foi proposta, visando à portabilidade das tecnologias desenvolvidas. Para isto foi criado um template com conceitos básicos (pessoas, organizações, etc) que participavam de vários tipos de eventos;

[10] Artigo extraído de http://www.oglobo.com.br em 11/09/2007.

- Template de Relação (TR do inglês "Template Relation") visava encontrar tipos de relação (local de, empregado de, produto de, etc) entre os TE; e
- Template de Cenário (ST do inglês "Scenario Template") consistia na descrição de uma classe particular de evento com os seus respectivos TE e TR.

Os sistemas que competiram no MUC-7 tiveram desempenho considerado aceitável para NE, TE e TR, mas inaceitáveis para CO e ST. Estas conclusões serviram de base para as estratégias nas quais os novos frameworks de avaliação se estruturaram.

Considerado como a continuação do MUC o ACE[11] (NIST ACE, 2007) redefiniu as tarefas para as seguintes:

- Reconhecimento e Detecção de Entidades (EDR do inglês "Entity Detection and Recognition") que consiste na extração das entidades estipuladas nos textos fornecidos. No MUC existiam dois tipos de entidades: entidade e local, sendo a entidade classificada ainda em organização, pessoa e artefato. No ACE as entidades são classificadas em tipo, subtipo e classe. A tarefa CO do MUC está embutida aqui, pois as classes são referências a uma entidade na forma de nome próprio, substantivo ou pronome;
- Detecção e Reconhecimento de Valores (VAL do inglês "Value Detection and Recognition") que consiste na extração de todos os valores dos elementos considerados no ACE.
- Detecção e Reconhecimento de Tempo (TERN do inglês "Time Expression Recognition and Normalization") que consiste na extração de valores temporais, normalmente de eventos;
- Detecção e Reconhecimento de Relação (RDR do inglês "Relation Detection and Recognition") que consiste na extração dos relacionamentos entre as entidades;
- Detecção e Reconhecimento de Eventos (VDR do inglês "Event Detection and Recognition") que consiste na extração de todos eventos com seus argumentos suas entidades e relacionamentos;

[11] Acrônimo para "Automatic Content Extraction".

- Detecção de Menção de Entidade (EMD do inglês "Entity Mention Detection") que consiste na extração de todas as menções as entidades especificadas para EDR;
- Detecção de Menção de Relação (RMD do inglês "Relation Mention Detection") que consiste na extração de todas as menções ao relacionamento definidos em RDR; e
- Detecção de Menção de Evento (VMD do inglês "Event Mention Detection") que consiste na extração de todos os eventos definidos para VDR.

De certa forma estas tarefas são hierárquicas, pois ter um bom desempenho nas 3 últimas tarefas consiste num pré-requisito para se alcançar desempenho adequado nas 5 primeiras.

2.5.3 A arquitetura de um sistema de extração da informação

A arquitetura aqui descrita consiste numa adaptação daquela encontrada em (TURMO et al., 2006). Segundo (HOBBS, 1993) "a arquitetura de um sistema de extração de informação consiste numa cascata de transformadores ou módulos que a cada passo adicionam estrutura e freqüentemente perdem informação, esperançosamente irrelevante, aplicando regras que são adquiridas manualmente e/ou automaticamente". Na Figura 2.2 encontra-se um desenho desta arquitetura.

No módulo de "Estruturação do Texto" o documento é preparado para o processo de extração. Esta preparação consiste na conversão do tipo do documento (o formato de destino é normalmente o tipo texto), sua segmentação em sentenças e estas desmembradas em seus tokens (uma seqüência de caracteres entre espaços) constituintes.

O objetivo do transformador "Análise morfológica" é classificar ou rotular cada token de acordo com a sua função gramatical na sentença. Esta classificação é feita através da consulta a dicionários e da aplicação de regras, que levam em consideração o seu contexto, para a desambigüação. Não constando no dicionário o analisador morfológico "adivinha" a classe através da extração do radical (na literatura a função de extração do radical é conhecida como stemmer) da palavra e da aplicação de regras gramaticais.

Nomes de empresas, de organizações, de unidades geográficas, de cargos e de pessoas devem ser classificados como nomes próprios, mas não constam de dicionários. O módulo de "Reconhecimento de entidades nomeadas" tem como objetivo reconhecer e, muitas vezes, classificá-las quanto ao seu tipo.

No transformador "Análise sintática parcial" é feita a rotulação dos sintagmas existentes em uma sentença.

No módulo de "Extração" é preenchido o valor dos elementos que compõe o template de extração na medida em que são encontrados no texto.

No transformador de "Resolução de co-referência" são resolvidas, normalmente, as anáforas encontradas.

No módulo de "Análise do discurso" os elementos encontrados são agrupados de acordo com o cenário de extração.

No transformador "Geração de resultado" elementos (como data, hora e número de telefone) tem seu formato normalizado.

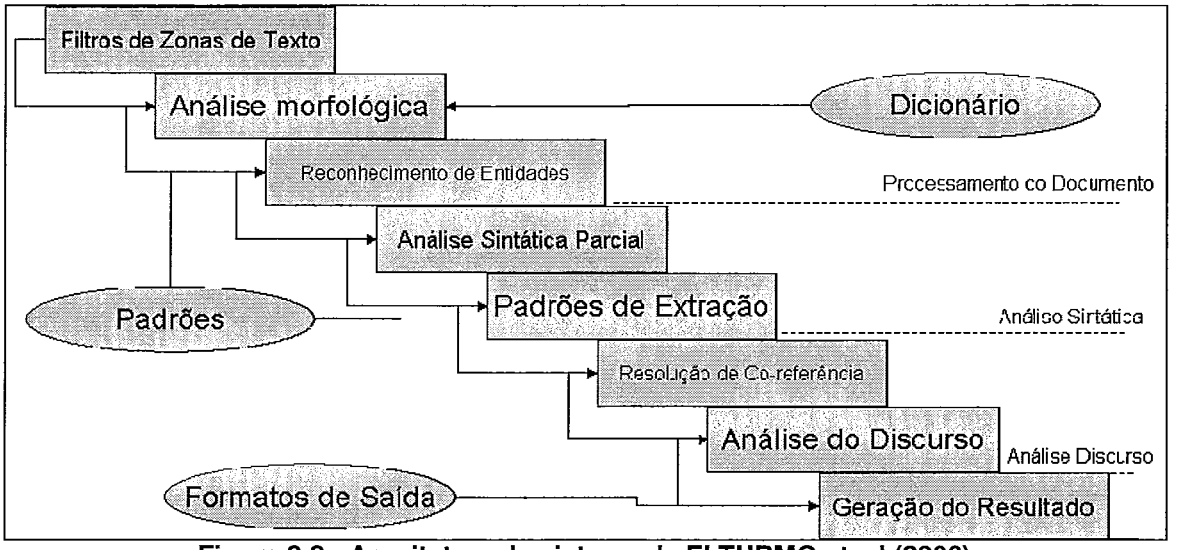


Figura 2.2 - Arquitetura de sistema de El TURMO et. al (2006)

2.6 Revisão bibliográfica para a extração de acrônimos

A resolução de acrônimos, como já vista, consiste na identificação em um dado texto de candidatos a acrônimos e expressões candidatas a expansões, na associação da tupla <acrônimo, expansão> correta e na desambigüação.

Em (PUSTEJOVSKY et al.,2001) o objetivo consistiu no desenvolvimento de um sistema automático de identificação e resolução de acrônimos (chamado de Acromed) extraídos de textos da área biomédica. Este sistema faz parte de um conjunto

de outros cujo objetivo final é a extração de informação nesta mesma área. O algoritmo segmenta o texto em frases. Cada uma é submetida a um analisador sintático parcial. Em seguida os sintagmas nominais extraídos são submetidos a um conjunto de 4 expressões regulares que identificam possíveis pares. Uma função de mapeamento entre as iniciais das palavras que constituem o sintagma nominal candidato, a expansão e cada letra do sintagma nominal candidato a acrônimo é processada, gerando um resultado. A identificação foi considerada positiva caso o resultado desta função de associação fosse igual ou superior a um determinado limite. Seguindo o procedimento padrão de avaliação de desempenho em IE foi construída uma base, denominada de padrão-ouro de desenvolvimento, com 86 resumos selecionados aleatoriamente da base de dados da Medline para o desenvolvimento das regras, teste do algoritmo e verificação do limite de identificação positiva mais adequada. Depois este algoritmo foi submetido a uma nova base de resumos (100 selecionados aleatoriamente entre os resultados da procura pela palavra "gene" numa base de jornais de prestígio na área biomédica), denominada de padrão-ouro de teste. Em ambas as bases um especialista identificou e rotulou manualmente os pares <acrônimo, expansão>. A avaliação de precisão e cobertura foi comparando os pares extraídos pelo Acromed com os pares rotulados pelo especialista. A cobertura foi definida como a divisão do número de pares corretamente identificados/número total de pares na base. A precisão foi definida com a divisão do número de pares corretamente identificados/número total de pares extraídos. Sendo NPA o número de palavras cuja primeira letra foi encontrada no acrônimo (desconsiderando as "stopwords") e N o número de letras do acrônimo a fórmula da função de mapeamento era NPA/N. Os autores reportam uma cobertura na base de teste de 72% com uma precisão de 98%. As duas bases estão disponibilizadas na Web e por isto tornaram este projeto um padrão de comparação para novas abordagens.

CHANG et al. (2002) decompõe o problema de localizar o acrônimo e sua expansão em 5 passos:

1. Percorrem as frases para encontrar candidatos utilizando apenas uma expressão regular. Esta basicamente consiste numa seqüência de palavras seguidas por um (ou 2 no máximo) termo entre parênteses. Encontrada a expressão na frase o termo entre parênteses é o candidato ao acrônimo e as $3*N$ (onde N é o

número de letras do acrônimo candidato) palavras a sua esquerda constituem na sua expansão candidata;

2. Calculam o melhor alinhamento entre as letras do acrônimo candidato e de uma palavra da expansão candidata utilizando uma adaptação do problema "Longest Common Substring".
3. Com as informações obtidas no passo anterior é construído um vetor com 8 atributos, tais como: se a letra associada está no início; meio ou fim da palavra da expansão; se letra é minúscula; e se delimita sílaba.
4. Utilizam um classificador de regressão lógica binária treinado com 1000 acrônimos (para os 93 considerados verdadeiros o vetor de alinhamento foi manualmente definido) para otimizar os pesos relativos de cada atributo.
5. O escore da abreviação é a máxima probabilidade dos escores dos alinhamentos possíveis da expansão candidata.

Rodando este algoritmo para o padrão-ouro de teste os autores alegaram os seguintes valores para cobertura e precisão: cobertura de 83%; e precisão de 80%.

Um algoritmo simples e equivalente aos demais em cobertura e precisão é apresentado por (SCHWARTZ, 2003). Semelhantemente a (Pustejovsky et al., 2001) tem como objetivo identificar pares <acrônimo, expansão> (através das mesmas expressões regulares) onde existe um mapeamento de qualquer tipo entre os caracteres do elemento acrônimo com os do elemento expansão. O processamento é dividido em duas etapas. Na primeira são identificadas todas as seqüências de palavras candidatas a expansão numa janela cujo número de palavras é definido pela fórmula a seguir.

$$\min(|A|+5,|A|*2)$$

Esta janela está adjacente (à esquerda ou à direita) a um acrônimo encontrado na sentença. Na segunda etapa um ponteiro percorre cada letra do acrônimo no sentido direita -> esquerda e outro percorre (no mesmo sentido) as letras que compõe a expansão. O ponteiro que percorre a expansão é decrementado até que o caractere para o qual aponta seja igual ao caractere apontado pelo ponteiro do acrônimo. Caso o ponteiro da expansão alcance o seu início antes de o ponteiro do acrônimo fazê-lo o algoritmo não reconhece o par. A restrição adicional é que a primeira letra do acrônimo seja igual à primeira letra da expansão. Para aumentar a precisão são descartadas

expansões cujo comprimento seja menor do que o do acrônimo e ou cujo acrônimo seja parte da expansão. Os autores alegam uma cobertura de 82% e uma precisão de 96%.

Um método de identificação de acrônimo através de aprendizagem supervisionada é apresentado em (NADEAU, 2005). Os autores entendem que a utilização de expressões regulares, principalmente com o uso de parênteses, como em (PUSTEJOVSKY, 2001) restringem muito a identificação de um acrônimo. É proposto um vetor com 17 atributos: número de letras do acrônimo casadas com as primeiras letras das palavras da expansão; o atributo anterior dividido pelo comprimento do acrônimo; número de letras maiúsculas do acrônimo; o atributo anterior dividido pelo comprimento do acrônimo; o número de palavras da expansão candidata; a distância em palavras entre o acrônimo e a expansão candidata; número de palavras da definição que não participam do acrônimo; o atributo anterior dividido pelo número de palavras da mesma expansão; o comprimento médio das palavras que não participam do acrônimo; se a primeira palavra é uma preposição, conjunção ou artigo; se a última palavra é uma preposição, conjunção ou artigo; número de preposições, conjunções e artigos na expansão candidata; maior número de letras do acrônimo que participam de uma única palavra da definição; número de letras do acrônimo que não participam; número de dígitos e pontuação do acrônimo que não participam; se a definição ou o acrônimo estão entre parênteses; e o número de verbos na definição. Adicionalmente são propostas heurísticas para restringir o espaço de busca de pares <acrônimo, expansão>, sendo 4 regras para localizar o acrônimo e 5 para a expansão candidata. Vários métodos de aprendizagem de máquina foram utilizados, seguindo a metodologia e corpora utilizados em PUSTEJOVSKY et al. (2001). O método de aprendizagem de máquina com o melhor desempenho foi o "Support Vector Machine" com cobertura de 84,4% e precisão de 92,5%. Apesar de alegarem que este algoritmo é menos restritivo do que os encontrados na literatura por não utilizarem expressões regulares para a identificação do acrônimo, concordam que o resultado reportado só foi possível graças ao atributo no vetor que caracteriza ou não o uso dos parênteses.

Os artigos até aqui descritos são aqueles que encontramos em nossa pesquisa que utilizam a base de resumos da área biomédica (PUSTEJOVSKY et al., 2001) como base para comparação de resultados. Porém precisamos destacar outros que serviram de inspiração para o nosso desenvolvimento.

Em sua tese de doutorado ZAHARIEV (2004) contribui com: uma teoria universal (a universalidade decorre do estudo de 14 línguas, sendo 4 européias ocidentais, 1 nórdica, 4 européias orientais, 3 do oriente médio e duas asiáticas) para a formação de acrônimo; com um algoritmo de identificação do par <acrônimo, expansão> quando adjacentes; com um algoritmo de identificação do par quando não adjacentes; e com um algoritmo para desambigüação (que é parte do problema da extração de acrônimos não abordado nesta pesquisa). Quando adjacentes, os pares candidatos são extraídos com expressões regulares. A associação entre acrônimo e expansões candidatas numa frase é feita com uma adaptação do problema "Longest Common Substring" como em (CHANG et al., 2002), mas, diferentemente deste, o escore (apelidado pelo autor de escore de alinhamento) não é uma função probabilística de um vetor de atributos, mas , como também denominando pelo autor, uma função de "plausibilidade lingüística" que utiliza uma escala decrescente de pesos (definidos empiricamente) para cada tipo de alinhamento. Os alinhamentos podem ser de 5 tipos: primeira letra da palavra, primeira letra de uma preposição, conjunção ou artigo; primeira letra de um morfema dentro de uma palavra; primeira letra de uma sílaba; e uma letra alinhada com nenhuma dos tipos de alinhamento anteriores. Em adição ao escore de alinhamento é adicionado um escore de continuidade que consiste na média entre o escore da primeira letra da palavra com o escore de alinhamento da letra atual. Como a adjacência entre os elementos pode ter dois formatos: acrônimo "("expansão")" e expansão "("acrônimo")" o autor desenvolveu a otimização do alinhamento utilizando os procedimentos "forward" e "backward" de programação dinâmica. A associação é considerada positiva quando os alinhamentos ótimos dos dois procedimentos são iguais. Zahariev dividiu a tarefa de resolução de acrônimos à distância em três fases. A primeira consiste em definir uma lista de candidatos para acrônimos e expansões, utilizando como entrada no algoritmo de resolução de acrônimos adjacentes todos os candidatos a acrônimos (que são identificados no texto de acordo com um padrão da capitalização das letras) e candidatos a expansão (que são todas as palavras, sintagmas e sentenças identificadas). Para cada par possível encontrado, um vetor com 46 atributos é definido. A classificação entre pares positivos e negativos acontece em três etapas. Na primeira o objetivo é eliminar os pares improváveis, diminuindo o espaço de procura. Uma "Support Vector Machine" (SVM) é treinada com os vetores de atributos calculados com os exemplos positivos e negativos anotados manualmente de um corpus

treinamento. Na segunda é calculada a perplexidade dos exemplos positivos resultados da primeira. Esta perplexidade indica a correlação dos elementos dos pares aparecerem na literatura concomitantemente, numa tentativa do autor de quantificar o que chamou de "cultura geral", ou seja, o conhecimento necessário para resolver acrônimos. Na terceira etapa os pares resultantes da primeira são novamente submetidos à SVM, que para esta fase foi treinada com os valores dos vetores de atributos mais a perplexidade dos exemplos positivos do corpus de treinamento, saindo a lista final dos pares considerados positivos. A perplexidade de cada par é encontrada, tendo como base um conjunto de documentos (extraídos da Web com uma máquina de busca) onde os elementos dos pares aparecem junto ou separados. ZAHARIEV (2004) reporta uma cobertura de 91,91% e uma precisão de 92,85%.

SCHUMANN (2005) utiliza HMM para aprender um conjunto de regras de formação dos acrônimos. Trata-se de uma evolução do trabalho proposto por PARK (2001). As expansões são formadas por morfemas classificados em 5 categorias:

- "s" para "stop words";
- "p" para prefixo;
- "h" para o resto de uma palavra excluído o prefixo;
- "n" para uma seqüência de números; e
  "w" para uma palavra.

O acrônimo é formado por letras, classificadas como "c" e por números, classificados como "n". Um conjunto de 7 operadores:

- "F" para associação da primeira letra do morfema;
- "I" para a associação de uma letra interna;
- "L" para a associação da última letra;
- "E" para uma associação exata entre uma seqüência de letras do acrônimo com um morfema inteiro;
- "R" para a substituição de um morfema por um caractere;
- "C" para associação contínua; e
- "Ins" para uma inserção de uma unidade do tipo "n" ou "c" no acrônimo que não tem uma correspondência na expansão.

Estes operadores associam um morfema na expansão a um caractere do acrônimo. A expansão e o acrônimo são desmembrados em suas respectivas classes. Cada regra tem dois lados: no lado esquerdo estão as classes dos morfemas e do lado direito as unidades do acrônimo, sendo que cada unidade é precedida pelo respectivo operador. O exemplo da figura 2.3 contém uma regra para descrever a formação do acrônimo PAI-1 que reduz "plasminogen activator inhibitor type-1". A regra fica no formato a seguir.

$$wwwwn \rightarrow [F(1):c][F(2):c][F(3):c][E(5):n]$$

Neste formato o número entre "()" indica o morfema. O problema apontado é que as regras não são estáveis, ou seja, bancos de acrônimos distintos têm conjunto de regras diferentes, dificultando o seu reaproveitamento. Nesta proposta a cadeia oculta mapeia os operadores (que são seus estados) de onde emergem os símbolos, que são os caracteres do acrônimo. O HMM é treinado com uma base de acrônimos onde as regras são manualmente anotadas. Após o treino o HMM é submetido a uma nova base, apelidada como base de teste, e a regra com a maior probabilidade é escolhida para resolver o acrônimo. Os autores reportam que a diminuição da cobertura foi pequena entre a base de treino e de teste (96% -> 94%) enquanto que sem a utilização de HMM a diminuição foi grande (96% -> 78%). Os autores aplicaram seu trabalho na base de acrônimos elaborados por PUSTEJOVSKY (2001) e reportam uma "taxa de detecção" de 82,5%.[12]

[12] Esta taxa é igual à cobertura, assumindo que todos os acrônimos detectados foram corretamente resolvidos. Esta cobertura está alinhada com os trabalhos dos demais autores que utilizaram esta base de acrônimos para comparação.

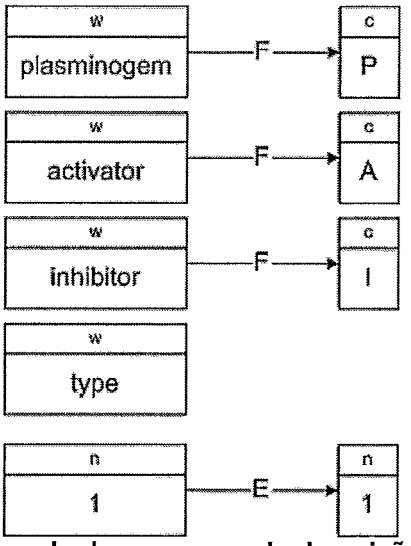


Figura 2.3 - Exemplo de uma regra de descrição de acrônimo

Capítulo 3 - Ferramentas e Métodos

O objetivo deste capítulo é apresentar as ferramentas e métodos utilizados no algoritmo de extração de acrônimo. Aqui se descreve para que servem e do que consistem os modelos ocultos de Markov, incluindo seus algoritmos. Uma revisão bibliográfica é apresentada, mostrando como estes modelos vêm sendo utilizados para a extração de informação.

É apresentada a plataforma GATE utilizada para o desenvolvimento da solução proposta nesta dissertação. Descrevem-se sua arquitetura, os tipos de documentos tratados e a sua linguagem específica para o tratamento de expressões regulares.

3.1 Hidden Markov models ou HMM

3.1.1 Introdução

Podemos caracterizar sinais, sejam eles discretos ou contínuos, como um conjunto de resultados observados dos processos que acontecem no mundo real. RABINER (1989) aponta duas razões para modelá-los: a primeira é servir de base teórica descritiva para o desenvolvimento de um sistema que possa reproduzi-los; e a segunda consiste em permitir o aprendizado sobre o processo real sem que tenhamos acesso direto a ele.

Geralmente modelamos sinais de duas formas: deterministicamente ou estatisticamente. Quando conhecemos suas propriedades descritivas, ou seja, quando podemos determinar a observação como uma função das propriedades conhecidas do processo, utilizamos o modelo determinístico. Quando não conhecemos estas propriedades ou quando podemos caracterizar estes sinais como resultantes de um processo aleatório parametrizável com formas precisas e bem definidas de estimar seus parâmetros, utilizamos a modelagem estatística.

3.1.2 Modelos de Markov

Um processo é dito markoviano caso a probabilidade de transitar para um próximo estado seja condicionalmente independente dos estados anteriores, dado o estado atual.

Seja $X = (X_1, \ldots, X_T)$ uma seqüência de variáveis aleatórias observadas e seja $S = \{s_1, \ldots, s_N\}$ um conjunto de estados finitos onde $X_t = s_n, 1 \leq t \leq T, 1 \leq n \leq N$.

Sejam, também, 2 propriedades (chamadas de propriedade de Markov) definidas na Tabela 3.1.

Tabela 3.1 - Propriedades de uma cadeia de Markov

(1)	$P(X_{t+1} = s_n \mid X_1, \ldots, X_t) = P(X_{t+1} = s_n \mid X_t)$	Horizonte limitado
(2)	$P(X_{t+1} = s_n \mid X_t, \forall t)$	Invariância temporal

Se as propriedades números (1) e (2) da Tabela 3.1 forem verificadas em X, esta é denominada de Processo ou Cadeia de Markov.

Podemos descrever esta cadeia através de uma matriz de transição estocástica A, onde $A = \{a_{ij}\}$ e $a_{ij} = P(X_{t+1} = s_j \mid X_t = s_i)$, sendo $a_{ij} \geq 0 (\forall i, j)$, $\sum_{j=1}^{N} a_{ij} = 1 (\forall i)$ e $s_i, s_j \in S$.

Para completar o modelo precisamos de um vetor com as probabilidades iniciais dos estados contidos em S. Este vetor, referenciado com a letra Π, onde $\Pi = \{\pi_i\}$, $\pi_i = P(X_1 = s_i)$, $\sum_{i=1}^{N} \pi_i = 1$ e $s_i \in S$.

Um modelo de Markov é, portanto, um conjunto de parâmetros $\lambda = \{S, \Pi, A\}$ e a probabilidade $P(X_1, \ldots, X_T \mid \lambda) = P(X \mid \lambda)$ é dada pela equação a seguir:

$$P(X \mid \lambda) = \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}} \tag{3.1}$$

3.1.3 Um exemplo de modelos de Markov

Na Figura 3.1 representamos uma cadeia de Markov através de um diagrama de estado. Trata-se de um pequeno alfabeto cujas letras representam os estados. A probabilidade de transição entre os estados está associada ao arco que os conecta. A cadeia pode ser iniciada pelos estados A, C, O. Formalmente temos:

- Conjunto de estados $S = \{A, C, O, R\}$
- Vetor de inicialização $\Pi = (0{,}3 \quad 0{,}4 \quad 0{,}3 \quad 0{,}0)$

- Matriz de Transição de Estados $A = \begin{bmatrix} & A & C & O & R \\ A & 0,0 & 0,0 & 0,0 & 1,0 \\ C & 0,8 & 0,0 & 0,2 & 0,0 \\ O & 0,0 & 0,0 & 0,0 & 1,0 \\ R & 0,0 & 0,4 & 0,4 & 0,2 \end{bmatrix}$

Dado o modelo, qual a probabilidade de observarmos as seqüências CARRO e ORAR? Com a equação (3.1) calcula-se:

- P(CARRO) = $\pi_C * a_{CA} * a_{AR} * a_{RR} * a_{RO} = 0,4 * 0,8 * 1,0 * 0,2 * 0,4 = 0,256$; e
- P(ORAR) = $\pi_O * a_{OR} * a_{RA} * a_{AR} = 0,3 * 1,0 * 0,0 * 1,0 = 0,0$

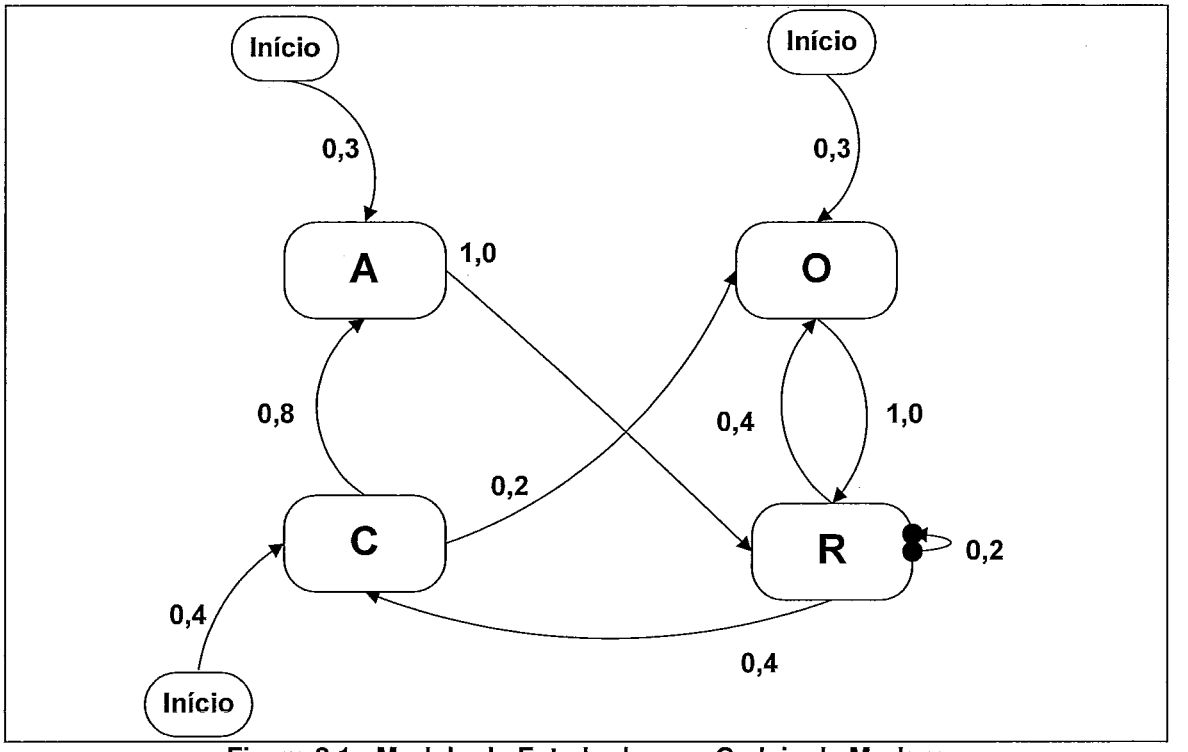


Figura 3.1 - Modelo de Estado de uma Cadeia de Markov

Concluímos que a observação da seqüência CARRO, dado o modelo, é possível e de que a seqüência ORAR não é.

3.2 Modelo oculto de Markov – HMM

Quando os estados não são diretamente observáveis, mas sim uma seqüência de sinais resultantes de um conjunto de processos estocásticos que produzem a seqüência observada se tem um modelo oculto de Markov - em inglês chama-se "Hidden Markov Model", cujo acrônimo é HMM e através do qual passaremos a referenciar esta tecnologia.

Um HMM consiste, portanto, de dois processos estocásticos, sendo um subjacente e não observável (oculto), mas que pode ser percebido através do segundo que produz a seqüência de observações visíveis (RABINER, 1989).

HMM são úteis quando desejamos modelar processo que percebemos indiretamente através de uma seqüência de observações, ou símbolos. Um exemplo seria o reconhecimento de voz (RABINER, 1989) onde percebemos os sons emitidos pelas cordas vocais, mas não observamos diretamente os seus estados. Outro seria a rotulação (ou classificação) das palavras segundo sua função gramatical (MANNING, 1999) onde imaginamos uma cadeia de Markov oculta cujos estados são as classes das palavras (em inglês "Parts of Speech" ou POS) dos quais as palavras lidas emergem. LUGER (2002) afirma que "as cadeias de Markov oferecem uma ferramenta poderosa para capturar os padrões da linguagem e o relacionamento entre estes e o mundo que descrevem".

Para que tenhamos um HMM precisamos adicionar dois parâmetros ao modelo (λ) da seção anterior. O primeiro deles é o conjunto de símbolos K, cujos elementos correspondem às observações do sistema que queremos modelar. Formalmente: $K = \{k_1, \ldots, k_M\}$, onde M consiste no número de elementos distintos. O segundo é a matriz de emissão de símbolos (B) onde modelamos a função estocástica de um símbolo emergir de um estado. Formalmente seja $O = \{o_1, \ldots, o_T\}$ a seqüência observada com $o_t \in K$ e $1 \leq t \leq T$, temos: $B = \{b_{s_i s_j k_m}\}$ e $b_{s_i s_j k_m} = P(O_t = k_m \mid X_t = s_i, X_{t+1} = s_j)$, onde $s_i, s_j \in S, k_m \in K$. O modelo passa a ter 5 parâmetros: $\lambda = \{S, K, \Pi, A, B\}$ e a probabilidade $P(O \mid \lambda)$ é dada pela equação a seguir.

$$P(X \mid \lambda) = \pi_{X_1} \prod_{t=1}^{T-1} a_{X_t X_{t+1}} \tag{3.2}$$

Uma aplicação com base em HMM precisa responder a uma das três perguntas abaixo para que seja útil na solução de problemas no mundo real:

- Problema 1: Dada uma seqüência de obsercações O e um modelo λ, como podemos computar eficientemente $P(O \mid \lambda)$? O aspecto prático deste problema é decidir dentre 2 ou mais modelos qual aquele que melhor descreve a seqüência observada;

- Problema 2: Dada uma seqüência de obsercações O e um modelo λ, como nós escolhemos a seqüência de estados X que melhor explica O? Seu aspecto prático é classificar cada elemento observado; e
- Problema 3: Como ajustar os parâmetros Π, A *e* B para maximizar $P(O \mid \lambda)$? Seu aspecto prático é definir os parâmetros do modelo quando estes são desconhecidos.

3.2.1 Um exemplo de HMM

No diagrama de estados da figura 3.2 representamos uma cadeia onde os estados constituem num pequeno alfabeto de apenas 3 letras e os símbolos num conjunto com apenas 4 palavras. Define-se então: $S = \{A \quad N \quad R\}$ e $K = \{Associação, Nacional, Racionalista, Behaviorista\}$. O objetivo consiste em calcular a probabilidade das seqüências de observações "Associação Nacional Racionalista" e "Associação Nacional Behaviorista" emergirem da cadeia $X = (A \quad N \quad R)$, sabendo que um estado $s_n \in S$ emite um símbolo $k_n \in K$, sendo l_1 a primeira letra de o_t, através da seguinte função:

- $f_{emissão} \begin{cases} 0 & s_n \neq l_1(o_t) \\ 1 & s_n = l_1(o_t) \end{cases}$;
- O vetor de inicialização é $\Pi = \begin{pmatrix} A & N & R \\ 1{,}0 & 0{,}0 & 0{,}0 \end{pmatrix}$;
- A matriz de transição de estados é $A = \begin{bmatrix} & A & N & R \\ A & 0{,}0 & 1{,}0 & 0{,}0 \\ N & 0{,}0 & 0{,}0 & 1{,}0 \\ R & 0{,}0 & 0{,}0 & 0{,}0 \end{bmatrix}$;
- A matriz de emissão de símbolos é $B = \begin{bmatrix} & Associação & Nacional & Racionalista & Behaviorista \\ A & 1{,}0 & 0{,}0 & 0{,}0 & 0{,}0 \\ N & 0{,}0 & 1{,}0 & 0{,}0 & 0{,}0 \\ R & 0{,}0 & 0{,}0 & 1{,}0 & 0{,}0 \end{bmatrix}$

Utilizando a fórmula da equação 3.2 calculam-se nos quadros a seguir as probabilidades das observações O_1=(Associação Nacional Racionalista) e O_2=(Associação Nacional Behaviorista).

O_1=(Associação Nacional Racionalista)

$$P = \pi_A * a_{an} * b_{a,n,Associação} * a_{nr} b_{nrNacional} * a_{rf} b_{rfRacionalista} = 1{,}0 * 1{,}0 * 1{,}0 * 1{,}0 * 1{,}0 = 1{,}0$$

O_2=(Associação Nacional Behaviorista)

$$P = \pi_A * a_{an} * b_{a,n,Associação} * a_{nr} b_{nrNacional} * a_{rf} b_{rfBehaviorista} = 1{,}0 * 1{,}0 * 1{,}0 * 1{,}0 * 0{,}0 = 0{,}0$$

Neste exemplo o cálculo da probabilidade foi simplificado por existir apenas um caminho possível de se percorrer a cadeia. A complexidade do algoritmo para o cálculo da probabilidade descrita na equação 3.2 é $O((2T+1).|S|^T)$ onde T é o número de palavras observadas e $|S|$ é o número de estados. Os algoritmos (descritos nas seções 3.2.2, 3.2.3 e3.2.4), baseados em programação dinâmica, para resolver os problemas apresentados na seção anterior reduzem esta complexidade para $2T.N^2$.

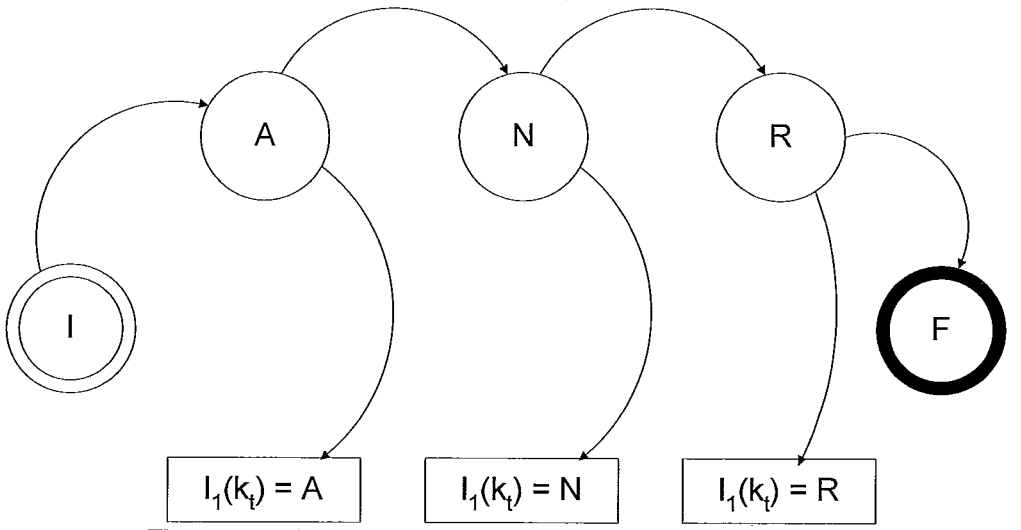


Figura 3.2 - Diagrama de estados e símbolos de um HMM

3.2.2 Probabilidade de uma observação dado um modelo

A probabilidade é computada através do algoritmo progressivo (do inglês forward procedure), tradução encontrada em KEPLER (2005), descrito a seguir:

Seja a matriz de propagação progressiva, cujas linhas referenciam cada elemento de S e as colunas cada elemento de O, $A = \alpha_{s_i}(t)$, onde $\alpha_{s_i}(t) = P(o_1, \ldots, o_{t-1}, X_t = s_i \mid \lambda)$ e é calculado recursivamente em 3 etapas:

Seja a probabilidade de se observar $O(o_1,\ldots,o_{t-1})$ e de se chegar ao estado s_i em tempo *t*, calculamos:

- Inicialização: $\alpha_{s_i}(1)=\pi_{s_i}$, $1\le i\le N$;
- Indução: $\alpha_{s_j}(t+1)=\sum_{i=1}^{N}\alpha_{s_i}(t)a_{s_is_j}b_{s_is_jo_t}$, $1\le t\le T, 1\le j\le N$; e
- Finalização: $P(O\mid\lambda)=\sum_{i=1}^{N}\alpha_i(T+1)$.

O algoritmo regressivo (do inglês backward procedure), tradução encontrada em KEPLER (2005), não é necessário para o cálculo desta probabilidade, mas será descrito aqui e utilizado nas duas próximas seções.

Seja $\beta_i(t)=P(o_t,\ldots,o_T\mid X_t=s_i,\lambda)$ cada elemento da matriz de propagação regressiva $\mathrm{B}(s_i,t)$, este algoritmo também divide-se em 3 fases:

- Inicialização: $\beta_{s_i}(T+1)=1$, $1\le i\le N$;
- Indução: $\beta_{s_j}(t)=\sum_{j=1}^{N}\beta_{s_j}(t+1)a_{s_is_j}b_{s_is_jo_t}$, $1\le t\le T, 1\le i\le N$; e
- Finalização: $P(O\mid\lambda)=\sum_{i=1}^{N}\pi_i\beta_i(1)$.

Para finalizar para qualquer tempo: $P(O\mid\lambda)=\alpha_{s_i}(t)\beta_{s_i}(t)$, $1\le t\le T+1$

3.2.3 Definição da seqüência de estados percorridos

A melhor seqüência de estados ocultos é encontrada através do algoritmo de Viterbi, descrito a seguir.

Seja $\Delta(s_j,t)$ a matriz que acumula a máxima probabilidade de estar no estado s_j em *t*, onde calculamos $\delta_{s_j}(t)=\max_{X_1,\ldots,X_{t-1}} P(X_1,\ldots,X_{t-1},o_1,\ldots,o_{t-1},X_t=s_j\mid\lambda)$, e seja também a matriz $\Psi(s_j,t)$ onde é armazenado o argumento, que neste caso é um estado, referente a máxima probabilidade de $\delta_{s_i}(t)$, o algoritmo recursivo segue em 3 fases:

- Inicialização: $\delta_{s_j}(1)=\pi_{s_j}, 1\le j\le N$;
- Indução: $\delta_{s_j}(t+1)=\max_{1\le i\le N}\delta_{s_i}(t)a_{s_is_j}b_{s_is_jo_t}$, $1\le j\le N$ e

  $\psi_{s_j}(t+1)=\arg\max_{1\le i\le N}\delta_{s_i}(t)a_{s_is_j}b_{s_is_jo_t}$, $1\le j\le N$; e

Finalização, onde a seqüência de estados é lida da direita para a esquerda: $\overline{X}_{T+1} = \arg\max_{1\le i\le N} \delta_{s_i}(T+1)$ e $\overline{X}_t = \psi_{\overline{X}_{t+1}}(t+1)$.

3.2.4 Maximização dos parâmetros do modelo

Ajustamos os parâmetros através do algoritmo de Baum-Welch, ou algoritmo progressivo-regressivo, que funciona como descrito a seguir:

Dado um modelo $\lambda = \{S, K, \Pi, A, B\}$, inicializado baseado em heurísticas ou randomicamente, procura-se encontrar um novo que melhor explique as observações encontradas. Ou seja, queremos encontrar $\{\overline{\Pi}\ \ \overline{A}\ \ \overline{B}\}$ maximizado por *O*.

Duas matrizes são necessárias: $P_t = \{p_t(s_i, s_j)\}$ que contém a probabilidade de passar $s_i \longrightarrow s_j$ em qualquer momento *t*; e a matriz $\mathrm{H}_{s_i} = \{\gamma_{s_i}(t)\}$ que contém o número de vezes em que o estado de origem foi s_i *em* o_t .

Calculam-se:

- $p_t(s_i, s_j) = \dfrac{\alpha_{s_i}(t) a_{s_i s_j} b_{s_i s_j o_t} \beta_{s_j}(t+1)}{\sum_{y=1}^{N}\sum_{z=1}^{N} \alpha_y(t) a_{s_y s_z} b_{s_y s_z o_t} \beta_z(t+1)}$; e
- $\gamma_{s_i}(t) = \sum_{j=1}^{N} p_t(s_i, s_j)$.

Os novos parâmetros são então processados como descrito a seguir:

- $\overline{\pi}_{s_i} = \gamma_{s_i}(1)$;
- $\overline{a}_{s_i s_j} = \dfrac{\sum_{t=1}^{T} p_t(s_i, s_j)}{\sum_{t=1}^{T} \gamma_i(t)}$;
- $\overline{b}_{s_i s_j k_m} = \dfrac{\sum_{t:o_t = k_m, 1\le t\le T, 1\le m\le M} p_t(s_i, s_j)}{\sum_{t=1}^{T} p_t(s_i, s_j)}$.

3.3 HMM e a Extração de Informação

3.3.1 Uma introdução

A opção por HMM para a elaboração desta pesquisa deveu-se principalmente pela forma com a qual o problema da resolução de acrônimo foi abordado. Aqui se

entende que existe uma distribuição conjunta de probabilidades, maximizada com as observações fornecidas ao modelo, que explica, ou modela, a relação entre a seqüência de caracteres que compõe o acrônimo com a sua forma expandida. Trata-se, portanto, de um modelo gerativo.

Encontraram-se vários artigos na literatura que utilizam HMM para a extração de informação. Na seção 3.3.2 resumem-se alguns deles.

3.3.2 Revisão bibliográfica

FREITAG (1999) propõe uma metodologia onde cada HMM é manualmente construído para a extração de um dado específico. Este dado consiste num estado-alvo, as palavras adjacentes à esquerda são rotuladas como prefixo, as palavras adjacentes à direita são rotuladas como sufixo e as demais são rotuladas como não-alvo. A janela de adjacência em ambas as direções foi fixada em 4 palavras. A aprendizagem é feita em documentos manualmente rotulados. Visando diminuir o erro oriundo da escassez de dados, pois o vocabulário é muito maior do que aquele fornecido para treinar o modelo, os autores propõem a técnica estatística de interpolação linear (do inglês shrinkage) para melhorar a probabilidade de emissão de símbolos. Estes são hierarquizados numa estrutura de árvore onde as palavras são as folhas, que por sua vez são agrupadas em prefixos e sufixos. Estes são agrupados num nível chamado de contexto, que agrupado com o estado-alvo chega-se no nível global. Este agrupado com o estado não-alvo leva ao nível mais alto da hierarquia onde a distribuição é uniforme. A cada nível é atribuído um peso, maximizado via EM (Expectation Maximization), sendo a probabilidade final de uma palavra ser emitida por um estado a média ponderada entre os pesos de cada nível da hierarquia multiplicada pela probabilidade do respectivo símbolo na hierarquia ser emitido pelo estado.

Em FREITAG (2000) o esforço adicional a FREITAG (1999) consiste em encontrar a melhor estrutura do HMM para cada dado a ser extraído. Sete operações são definidas para os estados (aumentar a janela do prefixo, aumentar a janela do sufixo, dividir o prefixo, dividir o sufixo, aumentar a janela do estado-alvo, dividir um estado-alvo e adicionar um estado não-alvo). Um algoritmo de "Hill-Climbing" é iniciado com uma estrutura simples. Todas as operações possíveis em cada etapa são realizadas e para cada estrutura resultante são calculados os parâmetros da rede conforme FREITAG (1999). O escore F1 (média harmônica entre precisão e cobertura) de cada

uma é calculado sobre a base de desenvolvimento rotulada. Aquela que tiver o maior escore substitui a rede anterior na nova iteração. Quando um número pré-definido de iterações ou estados é alcançado o algoritmo pára. Os autores reportam resultados melhores do que os alcançados no trabalho anterior (FREITAG, 1999).

RAY (2001) tinha como objetivo extrair tuplas <PROTEINA, LOCALIZAÇÃO> e <GEN, DOENÇA>. A principal contribuição foi considerar para o efeito da extração sintagmas específicos, ou seja, aqueles cujo núcleo consiste em uma palavra do domínio da extração (utilizando dicionário específico), e não palavras avulsas. Cada frase foi processada por um analisador sintático parcial (do inglês "shallow parser"). Os sintagmas extraídos recebiam um rótulo adicional indicativo se pertencentes ao domínio da extração. A rede de Markov totalmente conectada foi estabelecida e otimizada para representar a estrutura das frases que contém a relação desejada. Utilizando o algoritmo de Viterbi a seqüência de estados internos era verificada e a extração era consumada na ocorrência simultânea dos sintagmas rotulados.

SKOUNAKIS (2003), num trabalho que estende RAY (2001), acrescenta os modelos de Markov hierárquicos, cujo objetivo era ampliar a representação gramatical dos sintagmas relativos ao domínio da extração com a função gramatical de cada uma das suas palavras, adicionando, posteriormente, informações obre o contexto (através de vetores de atributos), levando a uma melhora em precisão e cobertura.

MCCALLUM (2000) propõe modelos de Markov baseado em máxima entropia, onde a distribuição conjunta de probabilidades das seqüências dos pares estado/observação é substituída por uma função exponencial que combina estes parâmetros em termos de atributos como, capitalização, parte do discurso, posição no documento, endentação entre outros) que são ajustados para um modelo exponencial por Máxima Entropia. A estrutura do modelo de Markov é definida a priori, apesar de um corpus rotulado de treinamento não ser estritamente necessário para estimar seus parâmetros.

A utilização de HMM hoje é diversificada. Além das já mencionadas classificação de palavras e reconhecimento de voz, encontramos aplicações destes modelos no reconhecimento de estrutura genética (WINTERS-HILT, 2006), na mineração de bancos de dados com atributos gen/proteina para associá-los ao genoma

de protozoários (OLIVEIRA et al., 2007) e na análise e classificação de imagem (SÖDEBERG, 2007).

3.4 General Architecture for Text Engineering – GATE

3.4.1 Uma introdução

Como descrito em CUNNINGHAM (1999) Engenharia de Linguagem (do inglês Language Engineering – LE) consiste na "...disciplina ou ato de construir sistemas de software que realizam tarefas envolvendo o processamento da linguagem humana, onde tanto os processos de construção quanto seus resultados são possíveis de serem previstos e medidos."

O GATE consiste numa arquitetura, numa plataforma e num ambiente de desenvolvimento de sistemas de engenharia de linguagem. É uma arquitetura, pois define a organização e as responsabilidades dos componentes de um sistema de LE. É uma plataforma, pois fornece implementações reutilizáveis de componentes e aplicações que engenheiros de linguagem podem utilizar, ampliar ou adaptar para necessidades específicas. É um ambiente de desenvolvimento, pois simplifica o desenvolvimento de novas aplicações ou módulos e fornece mecanismos para depurá-los (BONTCHEVA et al., 2004).

A sua adoção no desenvolvimento desta dissertação deveu-se as seguintes razões:

- Seus componentes são facilmente integráveis a outras aplicações;
- Código aberto, documentado e com adequado nível de suporte;
- Utilização dos padrões abertos como Unicode e XML;
- Por ser muito utilizado existem várias componentes de processamento de linguagem natural prontos;
- Trabalha com vários tipos de formatos de documentos;
- Gratuito e totalmente desenvolvido em Java; e
- Robustez e maturidade, conseqüência dos mais de 10 anos de desenvolvimento.

É desenvolvido com o suporte financeiro da Engineering and Physical Sciences Research Council (EPSRC), Biotechnology and Biological Science Research Council (BBSRC), Arts and Humanities Research Board (AHRB), European Union (EU) e parceiros comerciais.

3.4.2 A arquitetura do GATE

Segundo CUNNINGHAM et al. (2002) a arquitetura do GATE é composta de três objetos principais cujos nomes são:

- Language Resource (LR), em português recurso lingüístico, ao qual estão associados conceitos como dicionário, corpora e ontologia;
- Processing Resource(PR), em português recurso de processamento, representam entidades cuja principal natureza seja algorítmica como rotuladores de classe de palavras, analisadores sintáticos, resolução de co-referência e reconhecimento de entidades nomeadas; e
- Visual Resource (VR), em português recurso visual, que são componentes de visualização ou edição dos LR e PR.

Estes objetos recebem o nome genérico de CREOLE (Collection of Reusable Objects for Language Engineering), tendo seus atributos descritivos registrados em arquivos XML.

Podem residir em servidores ou em máquinas dos usuários.

Uma aplicação no GATE consiste num conjunto de PR, processados em série, chamado de "pipeline", sobre um conjunto de LR, tendo ou não VR.

A principal vantagem desta arquitetura é que, uma vez separada a parte algorítmica de uma aplicação da sua parte lingüística, seu desenvolvimento pode ser feito de forma independente pelas pessoas com competência nas respectivas partes.

3.4.3 Os documentos e suas anotações

O GATE suporta documentos em vários formatos, incluindo texto puro, HTML, XML, RTF e SGML. A adoção destes padrões simplifica a comunicação com outras plataformas de processamento de linguagem.

Uma vez aberto, o documento é analisado e convertido para um modelo interno de anotações onde é registrado o processamento de todos os PR, incluindo sua formatação original. As anotações são organizadas em um ou mais níveis de grafos ancorados no conteúdo do documento por offsete, no caso de texto, ou por milisegundo, no caso de áudio-visual. Cada anotação tem um identificador, nós inicial e final e conjunto de atributos.

3.4.4 Expressões regulares sobre anotações

JAPE (Java Annotations Pattern Engine) consiste numa máquina de estados finitos baseada em expressões regulares processada sobre as anotações. Sua gramática consiste num conjunto de regras, agrupadas em fases, que contém o padrão a ser comparado e a ação a ser aplicada sobre a anotação. As fases são processadas seqüencialmente como uma cascata de transformadores de estados finitos sobre as anotações. Não é permitida a recursão, mas regras de uma fase podem ser aplicadas sobre anotações geradas na fase anterior.

Uma regra tem dois lados (direito e esquerdo). No lado esquerdo está o padrão a ser encontrado e no lado direito está um conjunto de declarações (Java ou JAPE) que constituem a ação. Um rótulo é associado a cada anotação casada com o padrão e funciona como um sinalizador para o lado direito sobre quais anotações a ação deve ser aplicada. Na Figura 3.3 está descrita uma regra em JAPE que identifica endereço IP. O símbolo "-->" separa o lado esquerdo do direito.

O padrão a ser encontrado constitui numa seqüência de 7 anotações do tipo "Token", cujos atributos tratados são o "kind" e o "string". O atributo "kind" pode assumir somente o valor "number". Já o atributo "string" somente pode assumir o caractere ".". Este conjunto, quando encontrado, é rotulado como "ipAddress".

A ação consiste em criar uma nova anotação do tipo "Address", inserindo nela o atributo "kind" com o valor "ipAddress" uma vez encontrado o padrão. Esta nova anotação tem como nó inicial o nó inicial do primeiro token e como nó final o último token.

```
Rule: IPAddress
(
 {Token.kind == number}
 {Token.string == "."}
 {Token.kind == number}
 {Token.string == "."}
 {Token.kind == number}
 {Token.string == "."}
 {Token.kind == number}
)
:ipAddress -->
 :ipAddress.Address = {kind = "ipAddress"}
```

Figura 3.3 - Uma anotação no formato JAPE para identificar endereço IP

3.4.5 Reutilização de objetos GATE

Um valioso atributo do GATE está no reaproveitamento dos recursos desenvolvidos. Um exemplo consiste no conjunto de PR, os recursos de processamento, normalmente utilizados em aplicações de EI foram empacotados e disponibilizados com a plataforma. Este pacote é chamado de ANNIE e consiste dos seguintes elementos: tokenizador, separador de sentenças, rotulador de palavras, gazeteers, rotulador de entidades nomeadas, identificador de relações entre entidades nomeadas e identificador de sintagmas nominais.

3.4.6 Funcionalidades adicionais

Foi dada ênfase nos recursos e funcionalidades do GATE utilizados nesta dissertação. A plataforma, porém, oferece outros recursos que merecem uma menção.

Um é a integração com aprendizado de máquina. Os atributos de um conjunto de anotações são estruturados na forma de vetores que, por sua vez, podem alimentar qualquer base de treinamento.

Existe um modelo de dados para ontologias com suporte para hierarquia de classes, hierarquia de relações e indivíduos. Este modelo, implementado na forma de API, isola o PR das várias implementações de ontologias existentes.

Um sistema de avaliação de desempenho facilita a avaliação dos recursos desenvolvidos. Funciona comparando as anotações feitas manualmente com as automáticas, fornecendo as métricas comuns de cobertura e precisão.

Capítulo 4 - Extração de acrônimos com HMM

Neste capítulo aborda-se o porquê de adotar-se HMM para a solução proposta nesta tese. As atividades de identificação de acrônimos e suas expansões são descritas. São formalizadas as funções de transição de estados e de emissão de símbolos e o processo de construção do modelo de Markov a partir delas.

4.1 Introdução

Esta tese baseia-se na sugestão de que a formação de acrônimos consiste num processo de linguagem natural. Como os demais, este também evolui, tornando difícil manter atualizado um conjunto de regras que definam a formação de todos os acrônimos. A teoria universal proposta por ZAHARIEV (2004) já não consegue explicar alguns dos acrônimos encontrados nesta pesquisa. Um exemplo interessante, que não se encaixa nesta teoria, foi encontrado na área biomédica. ME2SO reduzindo "dimethyl sulfoxide". O autor deste acrônimo parece ter adaptado a fórmula química do composto, que é $(CH_3)_2SO$ para cunhá-lo, substituindo o CH_3 por ME, que são as iniciais de "methyl"[13].

O processo de escolher o caractere que representará a palavra no acrônimo parece aleatório. Normalmente os autores escolhem a primeira letra da palavra, de uma sílaba ou de um morfema. A aleatoriedade, portanto, estaria no processo de escolher um caractere dentre o conjunto dos caracteres possíveis.

O alinhamento entre um acrônimo e sua expansão pode ser ambíguo. A expansão de TUNEL é "terminal deoxynucleotidyl transferase-mediated deoxyuride triphosphate nick-end labeling". As letras TU podem ser representadas na expansão pelo binômio "Terminal deoxynUcleotidyl" ou pelo binômio "Transferase-mediated deoxyUride". Ou por ambos. Outro possível alinhamento é que cada binômio seria representado por uma letra. Então T seria representado por "Terminal deoxynucleotidyl" e U por "transferase-mediated deoxyUride".

Assim pode-se pensar que o processo de formação de um acrônimo é aleatório e, vendo-o desta forma, podem-se adotar técnicas probabilísticas, como HMM, pois estas "generalizam o ponto de vista determinístico" e "podem modelar precisamente tanto

[13] Encontramos em http://en.wikipedia.org/wiki/Dimethyl_sulfoxide o acrônimo DMSO para o mesmo composto químico.

aquelas partes da linguagem que são bem definidas como também aquelas partes que realmente têm algum grau de aleatoriedade" (LUGER, 2004).

4.2 Adoção de HMM para a resolução de acrônimos

CHANG et al. (2002) e ZAHARIEV (2004) adaptaram o algoritmo de encontrar a subseqüência de caracteres comum mais longa (LCS) para resolver acrônimos, incluindo a ambigüidade. CORMEN et al. (2001) descrevem a solução para a LCS através de um algoritmo baseado em programação dinâmica onde cada elemento de uma matriz bidimensional (com linhas contendo os caracteres do padrão a ser encontrado e com colunas contendo os caracteres da seqüência com a qual o padrão é alinhado) armazena o comprimento da subseqüência ótima calculado com uma fórmula recursiva, que considera o valor de elementos adjacentes anteriores. O comprimento considerado é aquele que se encontra armazenado no último elemento da matriz, considerando que esta é varrida da esquerda para a direita e de cima para baixo. Aplicando este algoritmo para encontrar a maior subseqüência comum entre ANP e as palavras que constituem sua expansão (Agência, Nacional e Petróleo) obtém-se o alinhamento demonstrado na Figura 4.1.

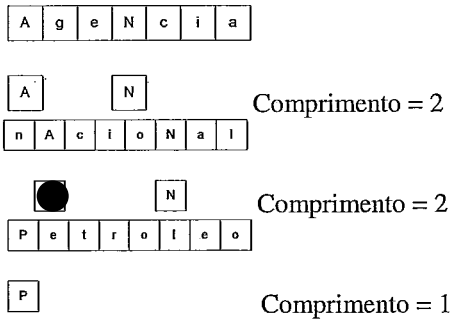


Figura 4.1 - Subseqüência comum mais longa para ANP

Verifica-se que este algoritmo não alinha o N de ANP com o primeiro N da palavra Nacional, que é seu alinhamento mais provável.

Com HMM nós substituímos a função recursiva de cálculo do maior comprimento baseada na comparação entre caracteres por funções probabilísticas para transição de estados e emissão de símbolos, onde um caractere, ou conjunto de caracteres, transita para o próximo depois de emitir uma palavra. Por exemplo: considerando o mesmo acrônimo da Figura 4.1 **Erro! Fonte de referência não encontrada.**a cadeia transitaria do caractere "A" para o caractere "N" depois de emitir a

palavra "agência"; transitaria para o caractere "P" depois de "N" emitir a palavra "nacional"; e chegaria ao seu fim depois de "P" emitir "Petróleo".

Outra restrição do LCS é que a varredura é feita da esquerda para a direita, utilizando o algoritmo progressivo, ou no sentido inverso, utilizando o algoritmo regressivo. Isto restringe a identificação de acrônimos como "SET", que reduz "Transfer of Single Embryo", onde a primeira palavra ("Transfer") corresponde à última letra do acrônimo e a terceira palavra ("Single") alinha-se com a primeira letra do acrônimo.

Com HMM é associada uma probabilidade de se transitar de um estado de origem para um estado de destino. Com isto consegue-se mapear transições nas cadeias nos dois sentidos (da esquerda para a direita e da direita para a esquerda).

"Partial Pressure of Oxygen in Arterial Blood" tem como acrônimo "PaO2" (PUSTEJOVSKY et. al. (2001) desconsideraram este termo como sendo um acrônimo). Nesta resolução pode-se associar as letras "Pa" com a palavra "Partial". Os caracteres "O2" (fórmula química do gás oxigênio) emitiriam a palavra "Oxygen". O conjunto $R = \{\text{Pr}\,essure, of, in, Arterial, Blood\}$ de símbolos não teria uma representação no acrônimo.

Ruído, para efeito desta dissertação, consiste em um símbolo (ou palavra) que consta na expansão, mas que não foi emitido por nenhum caractere do acrônimo. As palavras do conjunto R acima são ruídos para efeito da extração e a expansão é formada, portanto, por símbolos e ruídos[14].

CORMEN et al. (2001) propõe encontrar a semelhança entre duas cadeias de caracteres, sendo uma o padrão (uma subcadeia) a ser encontrada na outra, utilizando autômatos finitos. HMM consiste num autômato finito com funções estocásticas de transição de estado e emissão de símbolos.

Considerando a frase "O diretor do Observatório Nacional (ON) viajou para Houston." queremos resolver ON, tendo como expansão candidata "diretor do Observatório Nacional". Imaginando que neste caso o conjunto de ruídos é $R = \{diretor, do\}$ emitidos pelo caractere invisível ®, se quer comparar as cadeias

[14] A letra "a" do acrônimo "PaO2" pode emitir a palavra "arterial" e não trata-se de fato de um ruído. Aqui esta palavra permaneceu no conjunto dos ruídos apenas para fixar o conceito.

"®®ON" com o padrão "ON", almejando como resultado a seqüência em negrito "®®**ON**".

Colocado desta forma podemos ver a analogia que existe entre o problema de encontrar um padrão em uma cadeia de caracteres, como proposto por CORMEN et al. (2001) com a resolução de acrônimos e, portanto, a pertinência de se utilizar HMM para esta tarefa.

Portanto a utilização de HMM foi inspirada no algoritmo de autômatos finitos para encontrar a semelhança entre duas cadeias de caracteres e para tratar dos acrônimos cujo alinhamento não segue um único sentido.

4.3 A extração dos acrônimos e das suas expansões

A figura 4.2 descreve o processo de extração de acrônimo. No Anexo C está descrito um exemplo com o resultado do processamento de cada uma das atividades da Figura 4.2 sobre o texto não estruturado.

Uma vez iniciado o processo, a primeira atividade consiste em identificar no texto todas as strings candidatas a serem acrônimos. A próxima atividade é a identificação de todas as expressões candidatas a serem expansões do acrônimo identificado na etapa anterior. O processo prossegue com a resolução que consiste na definição da tupla <acrônimo, expansão> correta. Após a impressão de todas a tuplas resolvidas o processo de extração se encerra.

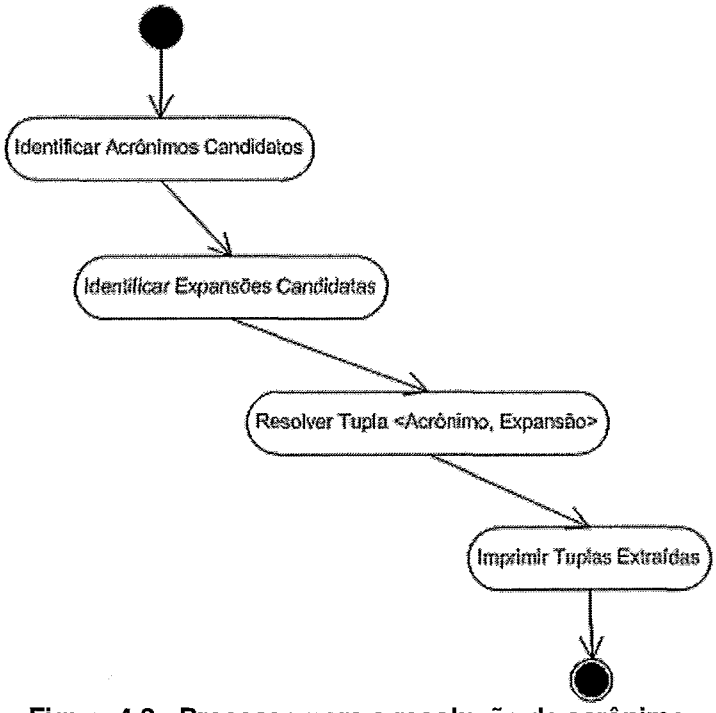


Figura 4.2 - Processo para a resolução de acrônimo

4.3.2 A identificação dos acrônimos e expansões candidatas

Dado um conjunto de documentos as duas primeiras tarefas, conforme a Figura 4.2, consistem na identificação no texto de todas as expressões candidatas a acrônimo (forma-curta) e de todas as expressões candidatas à sua expansão (forma-longa).

A maioria das abordagens estudadas utiliza a adjacência aos parênteses para esta identificação. Nesta dissertação somente serão extraídos os acrônimos cuja expansão encontra-se na sua adjacência. Segundo ZAHARIEV (2004) 97% dos acrônimos são definidos desta forma. Aqueles definidos a distância, ou seja, cuja forma expandida não se encontra adjacente a parênteses, não foram abordados nesta pesquisa.

Com estas restrições as tuplas <acrônimo, expansão> podem se encontrar nos 4 padrões descritos na Tabela 4.1.

Na prática a maioria dos pares está em conformidade com o padrão (4.1), onde a dificuldade é determinar o tamanho da janela de busca da forma-longa.

Tabela 4.1 - Padrões para a extração de candidatos a acrônimo e sua expansão

(4.1)	forma-longa "(" forma-curta ")"
(4.2)	forma-curta "(" forma-longa ")"
(4.3)	"(" forma-curta ")"forma-longa
(4.4)	"(" forma-longa ")"forma-curta

4.3.2.2 A identificação dos candidatos a acrônimos

A identificação da forma-curta nos padrões é feita utilizando-se expressões regulares que encontrem texto entre parênteses. Se esta string for constituída de um ou dois termos identifica-se positivamente um candidato a acrônimo, rotulando-o com os respectivos rótulos dos padrões (4.1) e (4.3) da Tabela 4.1. Se constituída de três ou mais termos rotula-se o termo adjacente à esquerda do primeiro parêntese com o respectivo rótulo do padrão (4.2) e o termo adjacente à direita do segundo parêntese com o rótulo que caracteriza um candidato a acrônimo no padrão (4.4).

4.3.2.3 A identificação das expressões candidatas à expansão

Dois conjuntos de expressões constituem as expressões candidatas a expansão. No primeiro está o sintagma nominal, conforme proposto por PUSTEJOVSKY (2001), à esquerda do acrônimo candidato rotulado no padrão (4.1) e o sintagma nominal à direita do acrônimo candidato rotulado no padrão (4.3). No segundo está o conteúdo da janela, conforme proposto por PARK (2001), à esquerda do acrônimo candidato rotulado no padrão (4.1) e o conteúdo da janela à direita do acrônimo candidato rotulado no padrão (4.3).

O tamanho T destas janelas, isto é, o número de palavras que as constitui, é dado pela equação (4.5), onde $|A|$ é o número de letras do acrônimo candidato.

$$T = \min\{|A|+5, |A|*2\} \qquad (4.5)$$

PARK (2001) chegou a esta fórmula analisando 4.500 acrônimos e suas respectivas expansões de textos da área da ciência da computação. Considerando as "stopwords" e as palavras sem representação no acrônimo verificou que quanto menor o acrônimo (até 4 caracteres) o número de palavras na janela não deveria ser maior do que o dobro do número de caracteres do acrônimo. Para acrônimos grandes (5 ou mais

caracteres) o número de palavras na janela não deveria ultrapassar o número de caracteres do acrônimo mais 5.

As expansões candidatas para os acrônimos candidatos rotulados com os padrões (4.2) e (4.4) consistiram na string entre os parênteses.

4.3.3 A resolução das tuplas <acrônimo, expansão>

4.3.3.1 Uma introdução

Os exemplos da figura 4.3, extraídos do banco de acrônimos elaborado por PUSTEJOVSKY et al.(2001), ilustra a necessidade de um processamento adicional à identificação. Chamamos esta tarefa de resolução das tuplas <acrônimo, expansão> ou, resumidamente, resolução de acrônimo.

O objetivo desta tarefa é encontrar a correta expansão de um acrônimo, caso ela exista.

"Nucleotide sequences were analyzed by Dr. Xiao Jianguo (Texas University Medical School and School of Public Health, Center for Infectious Diseases) using a suit of computer program (NIH)."

"The purpose of this study was to develop and evaluate a rapid microdose 14C-urea breath test (14C-UBT) with a simplified protocol for detecting the infection of hilicobacter pylori (HP)."

Figura 4.3 - O porquê da necessidade de se identificar os candidatos extraídos

Nos dois exemplos da figura 4.3 somente o rótulo do padrão (4.1) da Tabela 4.1 seria associado às strings "HP" e "NIH". Para este padrão, nos dois casos, tanto o sintagma nominal quanto o conteúdo da janela estão à esquerda do primeiro parêntese. O caractere "." elimina a possibilidade do padrão (4.3).

Assim as tuplas passíveis de serem extraídas seriam:

- <suit of computer program, NIH> (<sintagma nominal, acrônimo candidato>);
- <using a suit of computer program, NIH> (<janela, acrônimo candidato>);
- <hilicobacter pylori, HP> (<sintagma nominal, acrônimo candidato>); e
- < infection of hilicobacter pylori, HP> (<janela, acrônimo candidato>).

Nestes exemplos observam-se os dois objetivos da tarefa de resolução de acrônimos. O primeira é reconhecer que existe pelo menos uma tupla correta. A segunda é escolher dentre as várias possíveis qual é a melhor. "NIH", provavelmente,

consiste no apelido da suíte de programas de computador, mas não trata-se de um acrônimo e as duas tuplas devem ser rejeitadas. Já para "HP" a tupla <HP, hilicobacter pylori> deve ser escolhida.

4.3.3.2 Função estocástica de associação

A função estocástica que associa a expansão ao acrônimo é definida na função 4.6.

$$f(EC, AC) = \begin{cases} \max(P(EC / AC)) > 0 \; positivo \\ \max(P(EC / AC)) = 0 \; negativo \end{cases} \tag{4.6}$$

Esta função recebe dois argumentos como entrada, que são os elementos das tuplas identificadas. As expansões candidatas são associadas ao argumento EC e o acrônimo candidato é associado ao argumento AC. A tupla correta é aquela que submetida à função obtenha o maior valor. Existindo esta (cujo maior valor seja maior do que zero) a resolução é considerada positiva. Se todas as tuplas submetidas à função obtiverem valor igual a zero a resolução é considerada negativa.

Para o exemplo da figura 4.3 o valor esperado da função para as tuplas que têm "NIH" como acrônimo candidato é zero. Já o valor esperado para cada tupla que têm "HP" como acrônimo candidato é maior do que zero e, ainda, que o valor da tupla <HP, hilicobacter pylori> seja maior do que o da tupla <HP, infection of hilicobacter pylori>. Acontecendo isto a tupla <HP, hilicobacter pylori> é considerada correta (resolução positiva) e seus elementos passam à condição de acrônimo e respectiva expansão.

Nesta dissertação esta função estocástica é implementada com a "Forward Procedure" definida para HMM na seção 3.2.2.

4.3.4 A resolução das tuplas <acrônimo, expansão> com HMM

4.3.4.1 Definições

Como visto na seção 3.2 um HMM é definido como a cinco-tupla $\lambda = \{S, K, \Pi, A, B\}$, onde S é o conjunto de estados, K o conjunto de símbolos, Π o vetor de inicialização, A é a matriz de transição de estados (onde é descrita a função estocástica de transição de estado) e B é a matriz de emissão de símbolos (onde é descrita a função estocástica de emissão de símbolos).

Para configurar-se λ para a extração de acrônimos são necessárias algumas definições. São elas:

- Um **estado-alvo** consiste num subconjunto de caracteres do acrônimo candidato. Cada subconjunto pode ser constituído de no mínimo um elemento (um caractere do acrônimo candidato). O seu número máximo de elementos é igual ao número de caracteres do acrônimo candidato. Os elementos do subconjunto, ou seja, os caracteres que constituem o estado, devem ter a mesma ordenação dos caracteres do acrônimo candidato. A razão para isto está baseado na hipótese de que os autores escolhem aleatoriamente as letras e seu número para compor um acrônimo;
- Os termos tanto do sintagma nominal quanto da janela de busca constituem o conjunto de **símbolos** possíveis.
- O sintagma nominal ou a expressão que constitui a janela de busca são as **observações**;
- Existe um **estado-fim** que determina o término da forma-curta. Qualquer outro estado pode transitar para o estado-fim, mas este não transita para nenhum outro. Já o estado-início transita para qualquer estado-alvo, porém nenhum estado transita para ele;
- Existem estados, denominados de **estado-ruído**, que não aparecem na forma-curta, mas emitem símbolos que aparecem nas observações. Cada estado-ruído deve ter adjacente à direita um estado-alvo ou um estado-fim. Adjacente à sua esquerda somente um estado-alvo. Os estados-ruído foram introduzidos para tratar das palavras que não têm representação no acrônimo (regras (6) e (8) da Tabela 2.1); e
- Um estado-alvo pode transitar para outro à sua direita (adjacente ou não), outro à sua esquerda (adjacente ao não) ou transitar para si próprio.
- Nesta dissertação utilizaremos as palavras "termo", "símbolo" e "palavra" como sinônimos para designar na nomenclatura usual de HMM os símbolos que emergem de um estado.

A Figura 4.4 mostra a cadeia formada para o acrônimo ANP. Para simplificá-la somente representamos transições adjacentes à direita e omitimos, dentre estas, as transições de estados-alvo com mais de um caractere para um estado-ruído

(representados pelo caractere "®"). Cada transição, na verdade, é simétrica, ou seja, se é possível ir do estado A para o estado B, também é possível ir de B para A. O estado-fim é representado pelo caractere "¶".

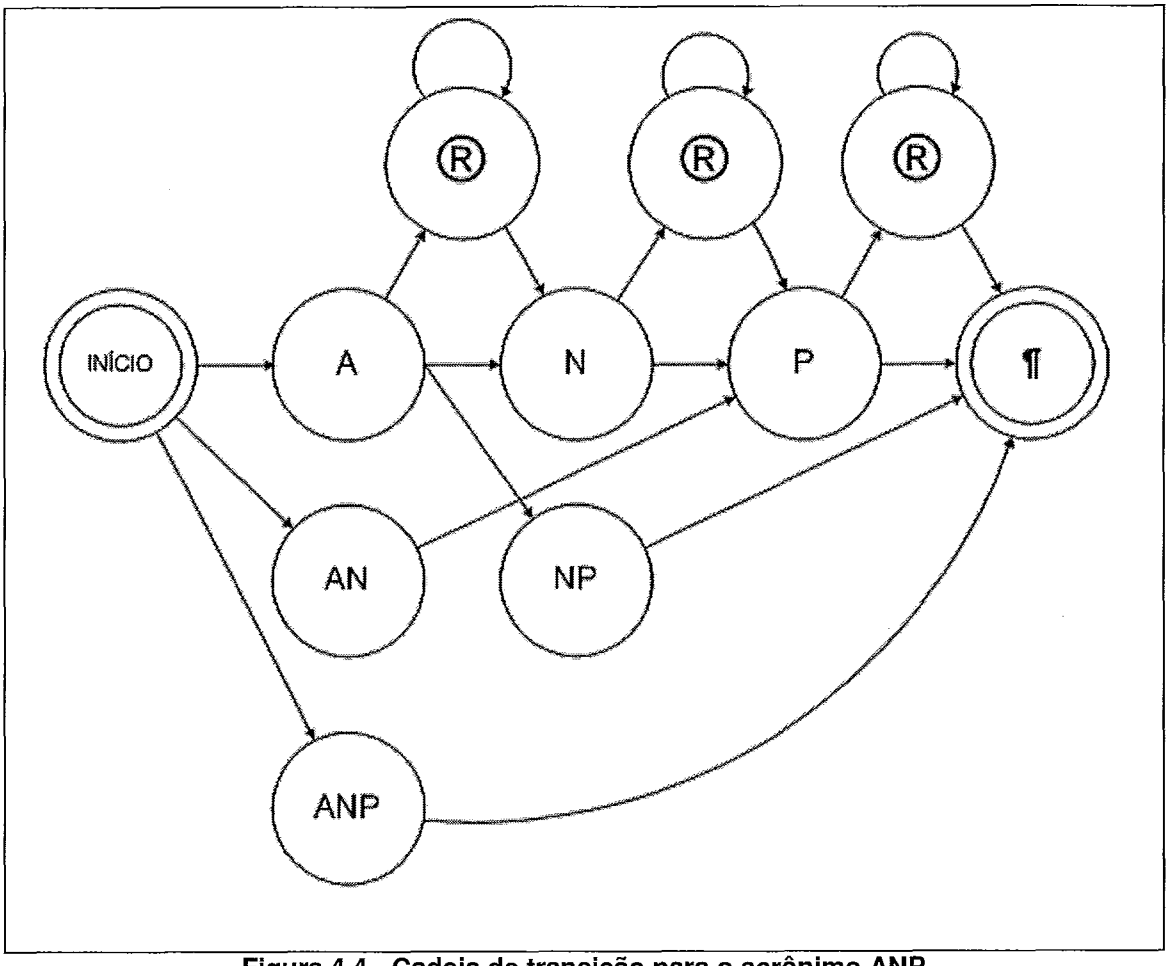


Figura 4.4 - Cadeia de transição para o acrônimo ANP

4.3.4.2 A formalização da resolução de acrônimos

Nesta dissertação adotaremos as seguintes definições:

Definição 4.1: **Acrônimo candidato** é todo termo extraído do texto de acordo com o padrão para forma-curta definido na tabela 4.1, constituído de caracteres com pelo menos um deles não numérico. Seja $C=\{c_1,\ldots,c_x,\ldots,c_X\}$ o conjunto de todos os caracteres possíveis[15] e seja $D=\{1,2,3,4,5,6,7,8,9,0\}\supset C$. $AC=(\alpha_1,\ldots,\alpha_y,\ldots\alpha_Y)$ é dito "acrônimo candidato" se $\alpha_y \in C$ e $\exists\alpha_y \notin D$.

[15] No caso desta dissertação foi utilizado o padrão Unicode "C1 Controls and Latin-1 Supplement" obtido em http://unicode.org/charts/PDF/U0080.pdf .

Definição 4.2: **Expansão candidata** é toda seqüência de termos extraída de acordo com o padrão para forma-longa definido na tabela 4.1, onde cada termo consiste numa seqüência de caracteres. Seja $EC = (ec_1,\ldots,ec_z,\ldots ec_Z)$ uma expansão candidata, onde $ec_z = (e_1,\ldots,e_w,\ldots e_W)$ e $e_w \in C$. O conjunto de expansões candidatas é dado por $E = \{EC_1,\ldots,EC_F\}$.

Definição 4.3: **Acrônimo** e sua **expansão** é qualquer conjunto de acrônimo candidato e expansão candidata que submetidos a função definida na figura 4.4 tenha o valor maior do que zero.

Definição 4.4: **Estado-alvo** é qualquer subconjunto de caracteres de um acrônimo candidato respeitada a ordem em que aparecem no acrônimo. Dado $AC = (\alpha_1,\ldots,\alpha_y,\ldots\alpha_Y)$ - o acrônimo candidato – um estado é um estado-alvo $EA = (ea_1,\ldots ea_k,\ldots ea_K)$ se $(ea_1 = \alpha_y) \wedge (ea_k = \alpha_{y+k-1})$, $K \leq Y$ e $1 \leq k \leq K, 1 \leq y \leq Y$.

Definição 4.5: **Símbolo** é qualquer termo que constitui as expansões candidatas. Portanto o conjunto de símbolos é igual ao conjunto de termos que constitui as expansões candidatas. Logo: $S = E = (EC_1 = (e_1,\ldots,e_W),\ldots,EC_F = (e_1\ldots,e_q))$.

Definição 4.6: As **observações** são constituídas dos sintagmas nominais ou os termos da janela de busca adjacentes ao acrônimo candidato. O conjunto de observações O é igual ao conjunto de símbolos. Logo $S = O$.

Definição 4.7: O caractere "¶", introduzido à direita do acrônimo candidato, estabelece o seu fim. O **estado-fim** é o estado que contém apenas este caractere.

Definição 4.8: O caractere "®" é introduzido entre dois caracteres do acrônimo candidato, incluindo o caractere "¶". **Estado-ruído** é o estado que contém apenas este caractere.

4.3.4.3 Geração dos estados de um acrônimo candidato

Seguindo as definições 4.4, 4.7 e 4.8 definem-se agora todos os estados possíveis, dado um acrônimo candidato. A geração de todos estes estados decorre de duas razões: não se sabe a priori quais as palavras representadas no acrônimo; e quais caracteres de cada palavra foram escolhidos para representá-lo.

Da definição 4.4 deduz-se que o número de estados-alvo possíveis é dado pelo seguinte somatório: $\sum_{a=1}^{Y} a$, onde Y representa o número de caracteres do acrônimo

candidato. Da definição 4.7 existe apenas um estado-fim e da definição 4.8 Y estados-ruído. Portanto o número total de estados possíveis é dado pela equação:

$$Num = \frac{1}{2}\left(Y^2 + 3Y + 2\right) \quad (4.7)$$

A função de geração de estados-alvo é dada pela equação 4.3:

$$f_{ger}(AC) = (c_b, \ldots, c \quad (4.8)$$
$$1 \le b \le Y; 0 \le d \le Y - b$$

A Figura 4.4 exemplifica os estados-alvo, que são "A", "N", "P", "AN", "NP" e "ANP", gerados através da função na equação 4.8 para o acrônimo ANP, os estados-ruído e estado-fim representados respectivamente pelos caracteres "¶" e "®". Os estados-ruído e estado-fim são inseridos de acordo com as definições 4.7 e 4.8.

Uma conseqüência destas definições é que os estados-alvo serão compostos apenas por caracteres nas posições pares (ver o exemplo na Figura 4.4). Os estados-ruído serão compostos por um caractere numa posição ímpar. O estado-fim é aquele que contém o caractere da última posição da seqüência.

Exemplificando, na Figura 4.5, para o acrônimo ANP o estado "NP" é composto pelos caracteres que estão nas posições 2 e 4 respectivamente. Todos os estados-ruído estão numa posição ímpar.

O conceito de distância, que é definido na próxima seção, baseia-se nesta abstração.

```
A ® N ® P ® ¶
0 1 2 3 4 5 6
```

Figura 4.5 - Exemplo de inserção dos estados ruído e fim dado um acrônimo

4.3.4.4 Definição de transição entre estados

Para calcular a distância entre dois estados-alvo é preciso, primeiro, ordená-los segundo a seqüência de caracteres do acrônimo. Na Figura 4.6 "BAN" e "SUL" estão ordenados corretamente e, em contrapartida, os estados "UL" e "NRI" estão ordenados

incorretamente. O estado "SUL" está à direita do estado "BAN". O estado "BAN" está à esquerda do estado "SUL".

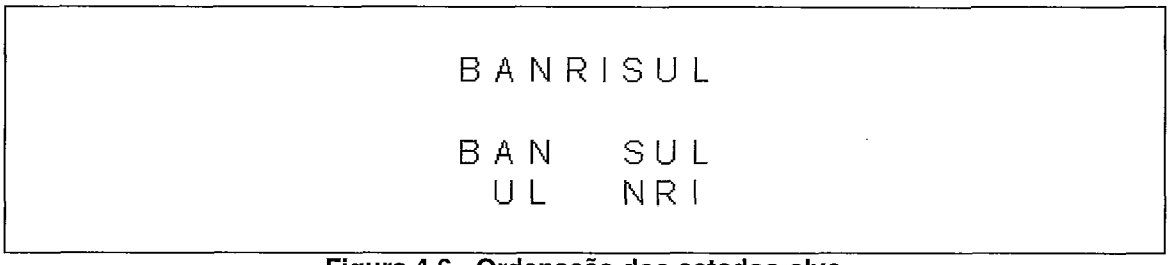


Figura 4.6 - Ordenação dos estados-alvo

O cálculo da distância entre dois estados-alvo é feito, respeitada a ordenação, subtraindo-se da posição do primeiro caractere do estado à direita a posição do último caractere à esquerda. Na **Erro! Fonte de referência não encontrada.** o estado à direita é o "SUL", cujo primeiro caractere está na posição 10. O estado à esquerda é o "BAN", cujo último caractere está na posição 4. A distância entre ambos é igual a 6.

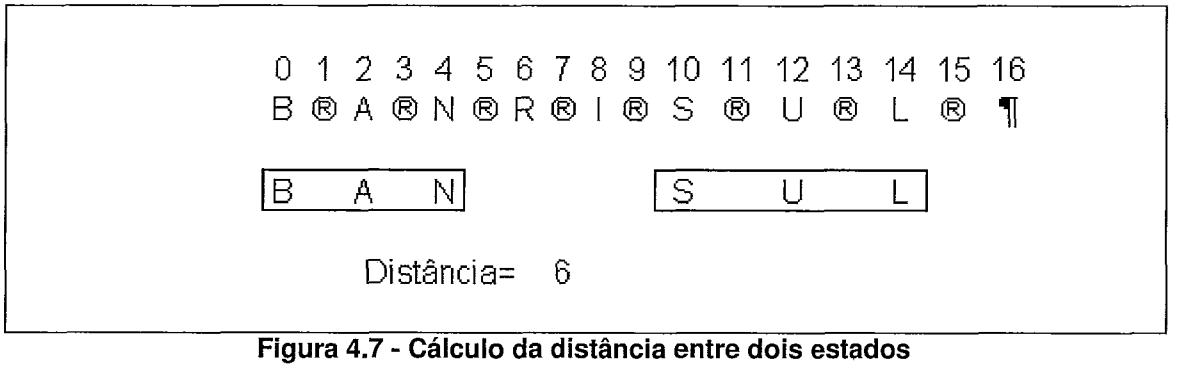


Figura 4.7 - Cálculo da distância entre dois estados

Agora podem ser definidos os conceitos de distância e transição entre estados conforme abaixo.

Definição 4.9: Distância entre dois estados é calculada subtraindo-se da posição do primeiro caractere do estado à direita a posição do último caractere do estado a esquerda. Portanto consiste num número inteiro maior do que zero.

Definição 4.10: Uma transição é dita **à direita** se a diferença entre a posição do primeiro caractere do estado de destino menos a posição do último caractere do estado de origem for positiva e maior do que dois.

Definição 4.11: Uma transição é dita **adjacente à direita** se a diferença entre a posição do primeiro caractere do estado de destino menos a posição do último caractere do estado de origem for positiva e igual a dois, se o estado de destino for do tipo alvo, ou

igual a um, se o estado de destino for do tipo ruído.

Definição 4.12: Uma transição é dita **à esquerda** se a diferença entre a posição do primeiro caractere do estado de origem e a posição do último caractere do estado de destino for positiva e maior do que dois.

Definição 4.13: Uma transição é dita **adjacente à esquerda** se a diferença entre a posição do primeiro caractere do estado de origem e a posição do último caractere do estado de destino for positiva e igual a dois, se o estado de destino for do tipo alvo, ou igual a um, se o estado de destino for do tipo ruído.

Definição 4.14: Uma transição é dita **auto-transição** quando o estado de origem for igual ao estado de destino.

A transição do estado de origem "BAN" para o estado destino "SUL" na Figura 4.7 é caracterizada como uma transição à direita, pois a distância é positiva e igual a 6.

Na Figura 4.8, abaixo, a transição do estado "BAN" para o estado "RISU" consiste num exemplo de transição adjacente à direita.

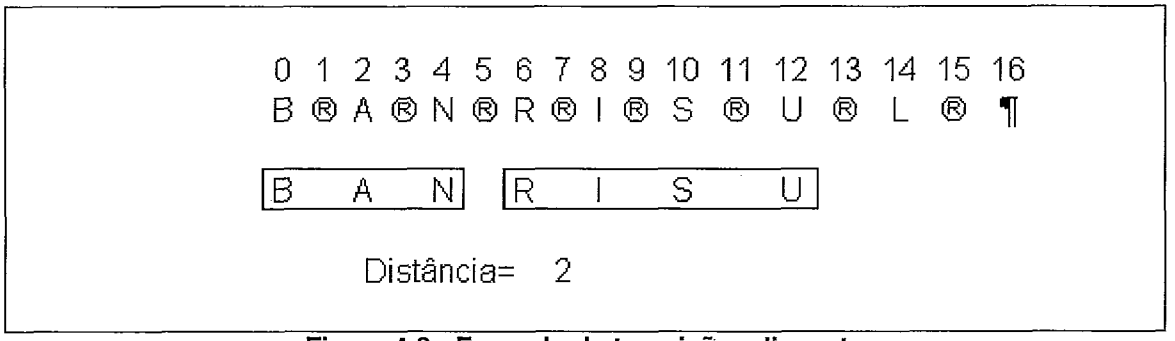


Figura 4.8 - Exemplo de transição adjacente

A transição do estado de origem "SUL" para o estado destino "BAN" na Figura 4.7 é caracterizada como uma transição à esquerda, pois a distância é positiva e igual a 6.

Na Figura 4.8 a transição do estado "RISU" para o estado "BAN" consiste num exemplo de transição adjacente à esquerda.

Uma conseqüência da definição 4.9 é a de que transições de estados com caracteres sobrepostos não é possível. Na Figura 4.9 estão exemplificadas duas transições não permitidas. A primeira seria uma transição do estado "ANR" para o estado "RIS". Neste caso a distância é igual a 0. A segunda é a transição do estado

"UL" para o estado "®" que está na posição 13. Neste caso a distância é negativa e igual a -1.

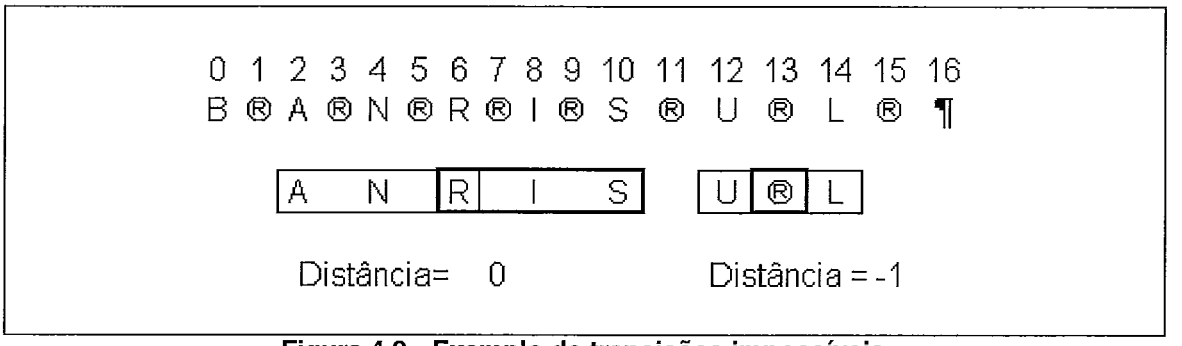


Figura 4.9 - Exemplo de transições impossíveis

4.3.4.5 As funções e a matriz de transição de estados

Com estas definições descrevem-se agora as funções de transição de estados, sendo uma para cada tipo de transição. Seu funcionamento é simples. Dados dois estados quaisquer, sendo um o de origem e o outro o de destino, a função retorna um valor igual ou maior do que zero. Se igual a zero a transição não é possível. Valores superiores a zero foram introduzidos para valorizarem-se determinados tipos de transição em detrimento de outros. Na verdade deseja-se que transições adjacentes à direita tenham maior probabilidade de acontecer do que uma transição à direita. A razão é que na maioria dos acrônimos todas as palavras da expansão estão representadas no sentido em que são lidas[16] (ver regras (1), (2), (3), (4) e (5) da Tabela 2.1). Voltando ao exemplo de ANP da Figura 4.4 se deseja que a transição do estado A para o estado N seja mais provável do que a transição do estado A para o estado P.

Estados-ruído foram introduzidos para tratarem palavras que estão na expansão, porém sem representação no acrônimo. Sem as restrições abaixo se poderia migrar na cadeia do estado de início para o estado fim transitando apenas por estados-ruído. Isto levaria o HMM a reconhecer qualquer expansão candidata como possível de emergir do acrônimo. As restrições abaixo diminuem esta possibilidade.

As restrições são:

- Do estado-início somente podemos transitar para um estado-alvo;

[16] Nesta dissertação estudaram-se acrônimos nas línguas portuguesa, inglesa, espanhola, francesa e alemã onde o sentido da leitura é da esquerda para a direita.

- Estado-ruído pode transitar para si mesmo ou para um estado-alvo adjacente; e
- Não existe transição possível do estado-fim.

Assim os tipos de transições de um estado-alvo para um estado-ruído, ou sua transição inversa, são as transições adjacentes à direita e adjacente à esquerda cuja distância é igual a 1. Na Figura 4.10 está um exemplo onde do estado "BA" se transita para o estado ruído que está na posição 3.

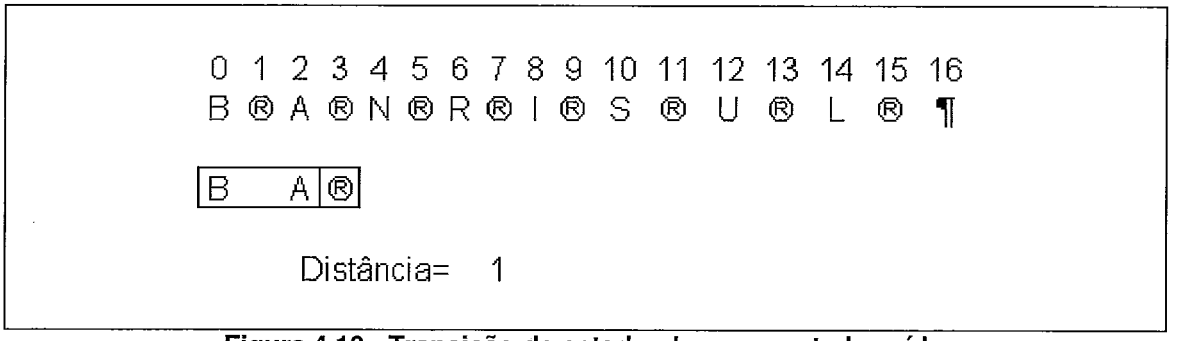


Figura 4.10 - Transição de estado-alvo para estado-ruído

Na Tabela 4.2 estão as funções de transição de estado. Nelas EO representa o estado de origem, ED o estado de destino. $C(Posição)_{Estado}$ indica o caractere na última posição do estado. O vetor W possibilita atribuir diferentes pesos aos tipos de transições, visando restringir a transição pela cadeia. Por exemplo, se $W_E = 0$ não será possível transição à esquerda.

Tabela 4.2 - Funções de transição de estado

(4.9)	$ft(EO,ED)_{DIREITA} = \begin{cases} 1 \times W_D, & C(1)_{ED} - C(P)_{EO} > 2 \\ 0, & Contrário \end{cases}$ e *ED e EO não ruído*
(4.10)	$ft(EO,ED)_{DIREITA(ADJACENTE)} = \begin{cases} 1 \times W_{DA}, & 1 \le C(1)_{ED} - C(P)_{EO} \le 2 \\ 0, & Contrário \end{cases}$
(4.11)	$ft(EO,ED)_{ESQUERDA} = \begin{cases} 1 \times W_E, & C(1)_{EO} - C(P)_{ED} > 2 \\ 0, & Contrário \end{cases}$ e *ED e EO não ruído*
(4.12)	$ft(EO,ED)_{ESQUERDA(ADJACENTE)} = \begin{cases} 1 \times W_{EA} & 1 \le C(1)_{EO} - C(P)_{ED} \le 2 \\ 0, & Contrário \end{cases}$ $ED \neq Inicio, EO \neq Fim$
(4.13)	$ft(EO,ED)_{AUTO} = \begin{cases} 1 \times W_{AU}, & EO = ED \\ 0, & Contrário \end{cases}$

A matriz de transição de estado é quadrada e é preenchida por linha, submetendo os estados às funções. Uma vez preenchida, normaliza-se esta matriz por linha, resultando na matriz de transição de estado.

4.3.4.6 A matriz de emissão de símbolo

A matriz de emissão de símbolo indica a probabilidade relativa de um estado emitir um símbolo em comparação com todos os símbolos possíveis de serem emitidos pelo respectivo estado. Assim consiste numa matriz onde nas linhas estão todos os estados (um por linha) e nas colunas todos os símbolos (um por coluna). Uma função de emissão de símbolo $f_{EMISSÃO}$ indica se um estado pode emitir um símbolo se seu valor for maior do que zero. Se igual a zero o estado não pode emitir o símbolo. A imagem desta função é $[0,|S|]$, onde $|S|$ consiste no número de caracteres do estado. Uma premissa de $f_{EMISSÃO}$ é a de que um estado pode emitir um símbolo se todos os caracteres do estado aparecem na mesma seqüência no símbolo. Assim um estado hipotético NA poderia emitir os símbolos NAcional e aNtenA[17], pois todos caracteres do estado aparecem no símbolo na mesma ordem. O símbolo AgeNte não poderia ser emitido, pois os caracteres não aparecem na mesma ordem.

Formulamos uma hipótese, baseada na observação das bases de acrônimo estudadas. Esta hipótese é a de que os autores têm preferência por escolherem os primeiros caracteres dos símbolos para representá-los no acrônimo. Assim $f_{EMISSÃO}$ atribuirá um maior valor para NA emitir o símbolo NAcional do que emitir o símbolo aNtenA.

Podemos agora formalizar $f_{EMISSÃO}$

$$f_{EMISSÃO}(W,\ S)=\begin{cases}\sum_{j=0}^{|S|-1}\max_{0\le i\le|W|-1}(esc_{i,j}), & \max_{\arg_i Max(esc[i,j-1])+1\le i\le|W|-1}(esc_{i,j})\neq 0\ \forall j \\ 0, & caso\ contrário\end{cases} \quad (4.14)$$

, onde:

- Matriz $ESC=\{esc_{i,j}\}$;

[17] Os caracteres do estado alinhados no símbolo estão em maiúsculo.

- $esc[i,j] = \begin{cases} 0, & c_i \neq d_j \\ \frac{|W|-i}{|W|-j}, & c_i = d_j \end{cases}$ $0 \leq i \leq |W|-1, 0 \leq j \leq |S|-1$ e $|W| \geq |S|$;
- W é o símbolo e $|W|$ o número de caracteres de W;
- c_i o iésimo caractere de W; e
- d_j o jésimo caractere de S.

Na Figura 4.11 existem os valores de $f_{EMISSÃO}$ <Agencia, NA>[18] e para <Nacional, NA>.

f(Agencia, NA)

i	j	0	1
		N	A
0	A	0,00	
1	G	0,00	
2	E	0,00	
3	N	0,57	0,00
4	C	0,00	0,00
5	I	0,00	0,00
6	A	0,00	0,17
max(j)		0,57	0,17
f(Agencia, NA)			0,74

f(Nacional, NA)

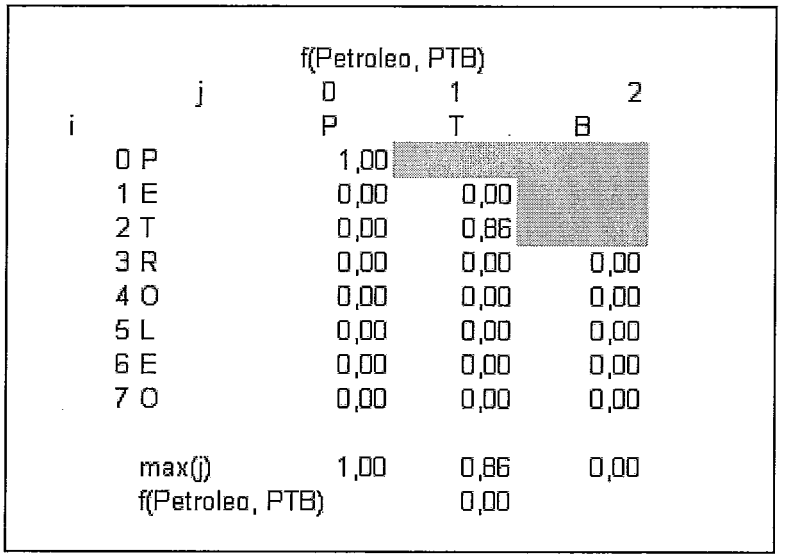
i	j	0	1
		N	A
0	N	1,00	0,00
1	A	0,00	1,00
2	C	0,00	0,00
3	I	0,00	0,00
4	O	0,00	0,00
5	N	0,38	0,00
6	A	0,25	0,29
7	L	0,00	0,00
max(j)		1,00	1,00
f(Nacional, NA)			2,00

Figura 4.11 - Exemplo de $f_{EMISSÃO}$ com valores maiores do que zero

Como esperado observa-se $f_{EMISSÃO}(Agencia, NA) = 0{,}74$ e $f_{EMISSÃO}(Nacional, NA) = 2{,}00$, valorizando o posicionamento inicial das letras representadas.

Na Figura 4.12 se exemplifica o valor de $f_{EMISSÃO}$ para <Petroleo, PTB>

[18] Na comparação substituem-se os caracteres acentuados pelos respectivos caracteres sem o acento uma vez que os caracteres não aparecem acentuados no acrônimo.



f(Petroleo, PTB)

i	j	0	1	2
		P	T	B
0	P	1,00		
1	E	0,00	0,00	
2	T	0,00	0,86	
3	R	0,00	0,00	0,00
4	O	0,00	0,00	0,00
5	L	0,00	0,00	0,00
6	E	0,00	0,00	0,00
7	O	0,00	0,00	0,00
	max(j)	1,00	0,86	0,00
	f(Petroleo, PTB)		0,00	

Figura 4.12 - Exemplo de $f_{EMISSÃO}$ com valor igual a zero

Também como esperado $f_{EMISSÃO}(Petroleo, PTB) = 0{,}00$, pois o caractere B não existe no símbolo Petroleo.

Uma grande vantagem de utilizar HMM consiste no fato de definir-se um template onde se pode modificar, substituir ou adicionar funções.

Existem caracteres ou grupos de caracteres que são utilizados pelos autores para a construção de um acrônimo. ZAHARIEV (2004) descreve este fato na sua regra (10) da Tabela 2.1 (associação simbólica). Um exemplo bastante utilizado consiste na seqüência de caracteres "[]" que simboliza "abraço". Para atender a esta regra criamos um conjunto de duplas $U = \{< W, S >\}$. Assim dada a dupla $d =< W^1, S^1 >$ qualquer e $d \in U$ substitui-se os caracteres de S^1 pelos caracteres de W^1, gerando um novo estado S^2. O novo estado mais o símbolo são então submetidos a $f_{EMISSÃO}$ como definido em 4.13.

Um exemplo aconteceria com o acrônimo "[]sA+" que consiste na forma reduzida de "Abraços e Até Mais"[19]. Seja o conjunto $U = \{< [], Abraço >, < +, mais >\}$. As funções $f_{EMISSÃO}$ para $f_{EMISSÃO}(abraço, abra\cos) = 6{,}00$, $f_{EMISSÃO}(A, ate) = 1{,}00$ $f_{EMISSÃO}(mais, mais) = 4{,}00$.

[19] Que não consiste num sintagma nominal. Esta expressão seria identificada pelo tamanho da janela definida na equação (4.5).

A matriz de emissão de símbolo seria montada enviando os índices de cada célula (as linhas com os estados e as colunas com os símbolos) para $f_{EMISSÃO}$. Após esta etapa, normaliza-se a matriz por linha, gerando a matriz de emissão de símbolos.

4.3.4.7 O vetor de inicialização

Para a definição completa do modelo de Markov definir o vetor de iniciação. Conforme MANNING (1999) iguala-se o estado-início ao estado-fim, atribuindo-se ao estado que contém o caractere ¶ o valor de 1.0. A probabilidade da cadeia começar por qualquer outro estado é 0.0.

4.3.4.8 O HMM para a resolução de acrônimos

Neste momento definiu-se para cada acrônimo candidato seu conjunto de estados *S*, seu conjunto de símbolos *W* (que consistem nos termos do sintagma nominal e da janela de busca), sua matriz de transição de estados, a sua matriz de emissão de símbolos e o seu vetor de inicialização.

Com isto o HMM está completamente especificado e para processarmos a função de identificação de acrônimo, descrita na função 4.6, se utiliza o algoritmo progressivo do HMM.

Se o valor retornado for positivo o acrônimo candidato ganha o status de "acrônimo", ou seja, uma extração positiva, com sua expansão sendo a expansão com a maior probabilidade de emergir do termo.

4.4 Testando a solução para a resolução de acrônimos

Antes de prosseguir com o experimento, a solução proposta neste capítulo foi testada com um banco de dados contendo mais de 600 acrônimos e suas respectivas expansões em cinco línguas distintas (português, espanhol, inglês, francês e alemão).

Um subconjunto com número variado de elementos (50, 100, 150 e 200) desta base de dados foi construído aleatoriamente. Deste subconjunto duas listas foram extraídas, sendo uma com os acrônimos e a outra com as expansões. O teste consistiu em encontrar a correta correspondência entre os elementos das duas listas.

A associação correta consistiu no resultado da função 4.6 onde para cada acrônimo da lista de acrônimos foi submetida a lista com todas as expansões, sendo seu desempenho independente da língua.

Capítulo 5 - O Experimento e os resultados

Neste capítulo será apresentado o sistema de extração de acrônimos. Na seção 5.1 encontram-se as descrições do sistema. Na seção 5.2 encontra-se uma descrição do desenvolvimento. Na seção 5.3 encontra-se a descrição do experimento e na seção 5.4 encontram-se os resultados alcançados.

5.1 A Descrição do sistema

Na Figura 5.1está descrito o processo de um sistema de extração de acrônimo. Este processo consiste numa adaptação daquele apresentado na seção Figura 2.2 para sistemas de extração de informação.

5.1.1 Conversão para o formato texto

Os documentos dos quais o sistema extrairá os acrônimos podem encontrar-se em formatos distintos, tais como: HTML, XML, PDF, .DOC e e-mail entre vários outros. Cada formato é definido por conjuntos de rótulos de marcação diferentes e, além disto, podem, em seu conteúdo, conter imagens, applets, programas em linguagem de script e outros elementos de onde, para o escopo desta dissertação[20], não há texto relevante à extração.

O objetivo da atividade "Conversão para Formato Texto" na Figura 5.2é filtrar os elementos não textuais existentes no documento, convertendo o documento original, que consiste no objeto de entrada desta atividade, para um formato padrão (o formato .txt), contendo apenas elementos textuais.

[20]As técnicas para reconhecimento de texto relevante em imagens e demais elementos não textuais não foram desenvolvidas nesta pesquisa.

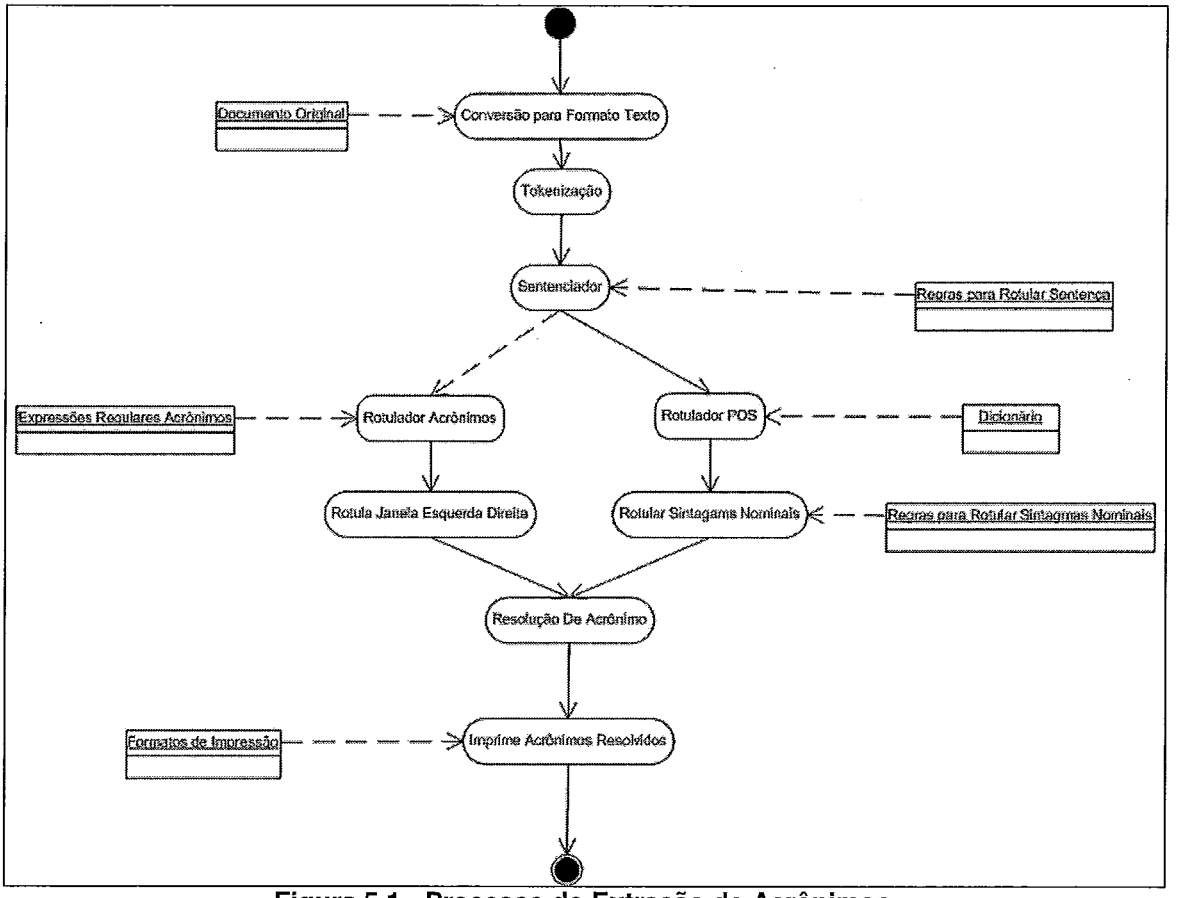


Figura 5.1 - Processo de Extração de Acrônimos

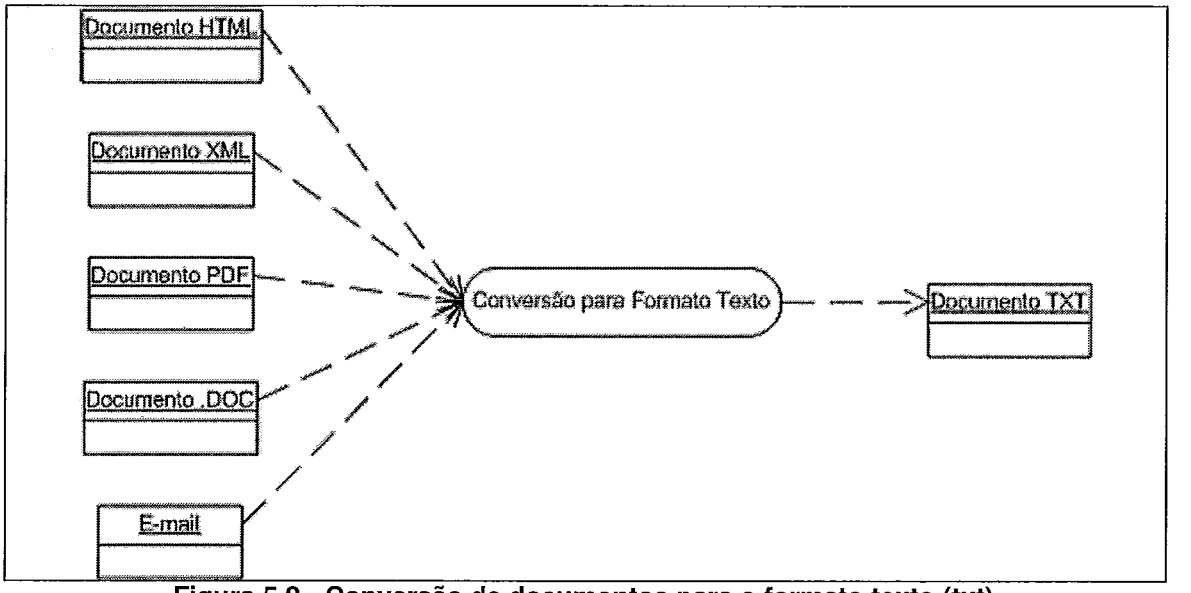


Figura 5.2 - Conversão de documentos para o formato texto (txt)

5.1.2 Tokenização[21]

A atividade de tonenização (do inglês "tokenization") trata, segundo MANNING (1999), da divisão do texto de entrada em unidades (chamadas de tokens) que consistem de palavras, números ou caracteres de pontuação. Não há um consenso na literatura pesquisada de como delimitar estes tokens, e, portanto este assunto não será tratado aqui. Uma regra intuitiva, e na prática muito utilizada para delimitar cada unidade, consiste em identificar um conjunto de caracteres contínuos entre espaços em branco (MANNING, 1999). Esta regra apresenta problemas tais como:

- Tratamento do ponto que delimita uma abreviação e não o fim de uma sentença;
- O hífen separando sílabas ou formando um termo composto; e
- O tratamento de palavras como "ponte aérea Rio-São Paulo" onde Rio-São não constitui uma palavra composta.

São estas unidades, ou seja, os tokens que as atividades subseqüentes rotularão para identificar os acrônimos e suas expansões.

5.1.3 Sentenciador

A atividade de segmentar um texto em sentenças, ou identificar o conjunto de tokens que a constituem, é realizada pelo sentenciador ou segmentador de sentença.

Normalmente sentenças começam com uma palavra com letra maiúscula e terminam com um ponto final. Esta tarefa, que em princípio não apresenta dificuldade, é complexa por um conjunto de razões. Uma delas está na utilização do ponto (o caractere ".") para delimitar abreviações. Outra está na utilização dos caracteres dois-pontos, ponto-e-vírgula e o travessão, delimitando ou não uma sentença. Uma outra razão está na utilização de sentenças aninhadas, ou seja, uma sentença no meio de uma outra. O desenvolvimento de muitos sentenciadores está baseado no algoritmo, extraído de MANNING(1999), apresentado na figura 5.3. Como baseiam-se em heurísticas dependem muito do conhecimento do desenvolvedor do contexto onde serão utilizados.

Nesta dissertação assumiu-se o princípio de que a expansão de um acrônimo encontra-se adjacente ou próximo a ele. Assim a principal função de utilizarmos o

[21] Trata-se de um anglicanismo, utilizado desta maneira na área de extração da informação e, por isto, aqui utilizado.

sentenciador é limitar o espaço de busca da expansão à sentença. Presumiu-se que o desempenho deste algoritmo é adequado à tarefa de extração de acrônimos.[22]

As regras de cada contexto são inseridas num banco de regras e este atua como um objeto para o sentenciador baseado em heurística.

1. Colocar o rótulo de fim da sentença (FDS) após caracteres que representam o ponto-final.
2. Se o caractere for sucedido por aspas mover o rótulo de FDS para após as aspas.[23]
3. Desconsiderar o ponto com delimitador de uma sentença nas seguintes situações:
    a. Se precedido por uma abreviação conhecida e esta abreviação for normalmente sucedida por um nome próprio (ex.: prof.); e
    b. Se precedido por uma abreviação e o próximo token não começar com letra maiúscula.
4. Desconsiderar o ponto de exclamação ou o de interrogação como FDS se:
    a. Sucedido por um token que inicia com letra minúscula; e
    b. Sucedido por com um nome próprio conhecido.
5. Considerar outros rótulos FDS como limitadores da sentença.

Figura 5.3 - Algoritmo de detecção de fim de sentença baseado em heurística

5.1.4 Rotulador de acrônimo

Esta atividade rotula os tokens como candidatos a acrônimo de acordo com os padrões descritos na Tabela 4.1. Basicamente estes padrões identificam se o acrônimo está entre parênteses ou a sua expansão é que está entre parêntese e, neste caso, o acrônimo é o token adjacente à esquerda do primeiro parênteses. O rotulador distingue um padrão do outro, baseado no número de tokens que estão entre os parênteses. Se o número de tokens entre parênteses for maior do que 2 é utilizado o rótulo referente ao padrão (4.2) da Tabela 4.1 (este é o caso em que a expansão está entre parênteses). Se

[22] No experimento esta eficiência se comprovou adequada.

[23] Na língua inglesa citações são delimitadas por duas aspas com o seu ponto final precedendo a segunda.

apenas um token estiver entre parênteses, recebe o rótulo referente ao padrão (4.1) (este é o caso em que o acrônimo está entre parênteses). Se o número de tokens for igual a 2, o conjunto recebe os dois rótulos, ou seja, aquele associado ao padrão (4.1) e aquele associado ao padrão (4.2) da Tabela 4.1.

Na Tabela 5.1 estão as 5 expressões regulares desenvolvidas em JAPE (linguagem descrita no capítulo 3, seção 3.4.4) para identificar os candidatos a acrônimo.

Tabela 5.1 - Expressões regulares em JAPE para rotular candidatos a acrônimo

Número	Regra em Jape	Descrição
(5.1)	Rule: Regra1 {Token.string == "("} ({SpaceToken.kind == "space"})* ((({Token.kind == word} ({Token.kind == number})?) ({Token.string == "/"}|{SpaceToken.kind == "space"})?) (({Token.kind == word} ({Token.kind == number})?)?)):rotulo {SpaceToken.kind == "space"} {Token.string == ")"} --> :rotulo.Acronym = {kind = "(1)"}	A **Regra1** rotula com o **padrão (4.1)** da tabela 4.1 strings que contenham um ou dois tokens (estes podem ser separados pelo caractere "espaço" ou "/") constituídos de letras e números.
(5.2)	Rule: Regra2 {Token.string == "("} ({SpaceToken.kind == "space"})* ({Token.orth == allCaps}):rotulo ({Token.string == ","}|{Token.string == ";"}|{SpaceToken.kind == "space"}) --> :rotulo.Acronym = {kind = "(1)"}	A **Regra2** rotula com o **padrão (4.1)** da tabela 4.1 strings que contenham apenas um token, constituído somente com letras maiúsculas e que terminem com os caracteres "," , ";" ou "espaço".
(5.3)	Rule: Regra3 {Token.string == "("} ({SpaceToken.kind == "space"})* ({Token.orth == mixedCaps} {SpaceToken.kind == "space"} {Token.kind == number}):rotulo --> :rotulo.Acronym = {kind = "(1)"}	A **Regra3** rotula com o **padrão (4.1)** da tabela 4.1 strings que contenham apenas um token, constituído somente com letras, sendo que a primeira minúscula, e que terminem com o caractere "espaço".

Número	Regra em Jape	Descrição
(5.4)	Rule: Regra4 {Token.string == "("} ({SpaceToken.kind == "space"})* ({Token.kind == "number"} ({Token.orth == mixedCaps}|{Token.orth == allCaps}) ({SpaceToken.kind == "space"})* ({Token.kind == number})*):rotulo -->:rotulo.Acronym = {kind = "(1)"}	A **Regra4** rotula com o **padrão (4.1)** da tabela 4.1 strings que contenham dois tokens separados pelo caractere "espaço", sendo o segundo token necessariamente um número.
(5.5)	Rule: Regra5 Priority: 1 (({Token.orth == allCaps}) ({Token.kind == number})?):rotulo ({SpaceToken.kind == "space"})* {Token.string == "("} ({SpaceToken.kind == "space"})* ({Token.orth == lowercase}|{Token.orth == allCaps}|{Token.orth == upperInitial})--> :rotulo.Acronym2 = {kind = "(2)"}	A **Regra5** rotula com o **padrão (4.2)** da Tabela 4.1 o termo a esquerda do parêntese de abetura.

As regras para identificar os padrões (4.3) e (4.4) da Tabela 4.1 não foram desenvolvidas, pois nos corpora utilizados não foram encontrados acrônimos e expansões nestes formatos.

5.1.5 Rotulador janela esquerda e janela direita

Esta atividade, baseada no rótulo do candidato a acrônimo, procura por sua expansão candidata na respectiva janela, cujo tamanho é determinado pela equação 4.5. Se o acrônimo estiver rotulado como "(1)" a janela consiste no número de tokens resultantes da equação 4.5 adjacentes à esquerda do primeiro parênteses. Se estiver rotulado com acrônimo do tipo "(2)" a janela consiste no número de tokens adjacente à direita do primeiro parênteses.

Se os dois rótulos estiverem associados ao candidato a acrônimo as duas janelas são recuperadas.

Antes de receberem o rótulo de expansão candidata o conjunto de tokens de cada janela passa por um conjunto de filtros que ao final garantem que o primeiro caractere do primeiro token da janela é um elemento do conjunto de caracteres do acrônimo[24].

Um exemplo seria o seguinte: na frase "The airway resistance (Raw) was determined with..." a atividade anterior (Rotulador de Acrônimo) rotularia "Raw" como (1) e esta atividade recuperaria "the airway resistance"[25], que, após a filtragem, será reduzida para "airway resistance", pois "Raw" não contém o caractere "t", mas contém o caractere "a" de "airway".

Este filtro pode dificultar a identificação de acrônimos como "xml", que reduz "extensible mark-up language". Neste caso o conteúdo da janela, após o processo de filtragem, seria reduzido para "mark-up language". Normalmente o conteúdo da janela é maior do que a expansão correta e estes filtros ajustam o início da janela para o início da expansão.

Uma vez filtrada, a expansão é inserida numa lista de expansões candidatas, e, por estar nesta lista, ganha o rótulo de "expansão candidata".

5.1.6 Rotulador POS

"Part of Speech (POS) Tagger", ou, em português, rotulador morfológico da palavra consiste na atividade de classificar ou rotular (ou etiquetar) cada token com a sua classe morfológica.

Os rotuladores POS consultam tabelas (dicionários) para obter a classe gramatical de uma palavra. Quando apenas uma classificação for possível o respectivo rótulo é atribuído à palavra.

Para um conjunto grande de palavras pode ser atribuída mais de uma classe morfológica. Segundo BORBA (2004) as classes morfológicas numeral, substantivo, adjetivo e conjunção são possíveis para a palavra "segundo".

Estes rotuladores se diferenciam entre determinísticos e probabilísticos de acordo com a forma que resolvem esta ambigüidade. Os determinísticos utilizam regras. Os possibilísticos utilizam a máxima verossimilhança de uma palavra receber

---

[24] Primeiro a filtragem procura na janela por um token que começa com primeira letra do acrônimo. Não encontrando procura pelo primeiro token que começa com qualquer letra do acrônimo.

[25] Para efeito da resolução dos acrônimos todas as letras são convertidas para minúsculo através de um filtro.

uma determinada classificação em um corpus. Ambos valem-se do contexto da palavra (as palavras que a antecedem e aquelas que a sucedem) para a desambigüação.

O rotulador POS é pré-requisito para a identificação dos sintagmas nominais. As ferramentas acessíveis para rotulação morfológica, BICK (2006), e extrair sintagmas nominais, SANTOS (2005), em português utilizam conjuntos de rótulos distintos para as classes morfológicas, necessitando o desenvolvimento de adaptadores para permitir o acoplamento entre ambas.

Por isto esta atividade foi objeto de pesquisa, mas, devido a sua complexidade, foi descartada por não ser possível seu desenvolvimento no tempo de duração deste trabalho[26].

5.1.7 Rotulador de sintagmas nominais

Uma vez rotulados e de posse das regras de agrupamento das palavras em torno de um substantivo, esta atividade agrupa os tokens em torno deste substantivo, estabelecendo o sintagma nominal. Como característico de sistemas baseados em regras, quanto maior o conhecimento do desenvolvedor sobre o contexto onde serão aplicados melhor serão os resultados.

Esta atividade foi objeto de estudo e descartada pelas mesmas razões apresentadas na seção 5.1.6 para o Rotulador POS.

Antes de serem inseridos na lista de expansões candidatas os sintagmas são filtrados com o mesmo conjunto de filtros descritos na seção Rotulador janela esquerda e janela direita (seção 5.1.5).

5.1.8 Resolução de acrônimos

Esta atividade tem como objetivo associar o acrônimo à sua expansão. Mais formalmente o objetivo desta atividade é identificar para cada acrônimo candidato identificado na atividade rotulador de acrônimo (seção 5.1.4) a sua expansão, selecionando-a de uma lista de expansões candidatas rotuladas nas atividades rotulador janela esquerda e janela direita (seção 5.1.5) e rotulador de sintagmas nominais (seção 5.1.7).

---

[26] Esta atividade consistirá em trabalho futuro. Nesta dissertação foi utilizado o recurso disponibilizado na plataforma GATE.

A resolução, ou seja, a definição da tupla <acrônimo, expansão> acontece da seguinte maneira:

Um módulo chamado de "Resolvedor de Acrônimo" é iniciado. Este resolve a tupla, utilizando a função descrita na equação 4.6.

A tupla contendo o par acrônimo/expansão é inserida na lista de acrônimos resolvidos, caso o módulo retorne uma identificação positiva. O acrônimo é inserido na lista de acrônimos não resolvidos no caso do "Resolvedor de Acrônimo" retornar uma identificação negativa.

Na Figura 5.4 está contido o algoritmo do módulo RESOLVEDOR-ACRONIMO, que recebe como argumentos o acrônimo candidato e a lista com as expansões candidatas.

```
RESOLVEDOR-ACRONIMO(acronimo, lista_expansao_candidata)

for cada expansao em lista_expansao_candidata{

        numero_caracteres_estados <- 1
        pesos[2], pesos[3], pesos[4] <- 0
        pesos[0], pesos[1] <- 1
        resolvido <- falso

        enquanto resolvido = falso {
                probabilidade <- 0
                enquanto nivel < tamanho(acronimo){
                    automato.inicializa(acronimo, nivel, pesos, lista_expansao_candidata)
                    probabilidade = automato.forward_procedure(expansao_candidata)
                    se probabilidade > 0
                                        então lista_expansao_candidata.insere(expansao_candidata, probabilidade)
                                        resolvido = verdadeiro
                                        numero_caracteres_estados <- tamanho(acronimo)
                    contrário
                     numero_caracteres_estados <- numero_caracteres_estados + 1
                }

                se resolvido = falso
                    então pesos[2], pesos[3], pesos[4] <- 1
                    numero_caracteres_estados <- 1
        }
}

se resolvido = falso
  então returna "Negativa"
contrário
  lista_expansao_candidata.decrescente()
  returna lista_expansao_candidata.recupera(elemento numero 1)
```

Figura 5.4 - Algoritmo para resolver acrônimo

Para cada expansão o primeiro passo é definir o nível de restrição da cadeia, que inicialmente é ajustado para o mais restritivo. A razão disto é que a maioria dos acrônimos segue a definição inicial de que é constituído com as iniciais das palavras da sua expansão.

O nível mais restritivo se configura, ajustando-se o número de caracteres por estado para 1, atribuindo-se pesos igual a 0 para as funções de transição de estado à esquerda (função 4.11), adjacente à esquerda (função 4.12) e auto-transição (função 4.13).

O peso das funções da transição à direita (função 4.9) e adjacente à direita (4.10) é ajustado para 1.

Inicia-se uma variável de controle para verificar se o acrônimo foi resolvido. Esta variável é inicialmente ajustada para falsa.

Enquanto o acrônimo não é resolvido o algoritmo prossegue diminuindo gradativamente o nível de restrição, primeiro aumentando o número de caracteres por estado e depois permitindo as transições à esquerda e adjacente à esquerda, ou seja, alterando seus respectivos pesos de 0 para 1. O objetivo destas transições é permitir o reconhecimento de acrônimos que não começam com a primeira letra da primeira palavra.

5.1.9 Impressão dos acrônimos resolvidos

Em um sistema do mundo real este módulo consistiria na impressão de dois relatórios. O primeiro contém a lista dos acrônimos encontrados e resolvidos, ou seja, uma lista dos acrônimos e das suas respectivas expansões.

O segundo contém uma lista dos acrônimos candidatos que não foram extraídos. A falha na extração pode ser resultante da expansão não constar no documento, do acrônimo rotulado como candidato não ser na verdade um acrônimo ou por falha no processo de extração.

Neste trabalho, porém, desenvolveram-se relatórios que permitem verificar o desempenho da solução em adição às informações contidas nos relatórios já mencionados. Os detalhes destes relatórios serão cobertos na seção 5.4 e aqueles gerados para o corpus de desenvolvimento e de teste encontram-se nos anexos A e B respectivamente.

5.1.10 O Algoritmo para extração de acrônimos

O algoritmo do sistema encontra-se na Figura 5.5. Basicamente cada documento original é convertido para o formato texto. Em seguida processa-se o Tokenizador e o Sentenciador. Para cada sentença identificam-se os acrônimos candidatos e os sintagmas nominais.

Para cada acrônimo candidato, recuperam-se o conteúdo da janela à sua esquerda ou à sua direita. Estes conteúdos, juntamente com os sintagmas nominais, são passados para o módulo de resolução de acrônimo que alimenta as devidas listas de acrônimos resolvidos ou não resolvidos dependendo da resposta da atividade de Resolução de Acrônimo.

```
EXTRAÇÃO-ACRONIMOS

Para cada DOCUMENTO_ORIGINAL em CORPUS
        DOCUMENTO <- Conversor_Para_Formto_Texto(DOCUMENTO_ORIGINAL)
        DOCUMENTO <- Tokenizador(DOCUMENTO)
        LISTA_SENTENCAS <- Sentenciador(DOCUMENTO)

        Para cada SENTENCA em LISTA_SENTENCAS{
                LISTA_ACRONIMO_CANDIDATO <- Rotulador_Acronimos(DOCUMENTO)
                LISTA_EXPANSAO_CANDIDATAS.adiciona(
                                Rotulador_POS(Rotulado_Sintagmas_Nominais(DOCUMENTO))

                Para cada acronimo_candidato em LISTA_ACRONIMO_CANDIDATO{
                        LISTA_EXPANSAO_CANDIDATAS.adiciona(
                                Rotulador_Janela_Esquerda_Direita(acronimo_candidato))
                        EXPANSAO <- Resolucao_Acronimo(
                                acronimo_candidato LISTA_EXPANSAO_CANDIDATAS)

                        se EXPANSAO = NEGATIVA
                                então LISTA_ACRONIMOS_NAO_RESOLVIDOS.adiciona(acronimo_candidato)
                        contrário
                                TUPLA <- acronimo_candidato,expansao)
                                LISTA_ACRONIMOS_RESOLVIDOS.adiciona(TUPLA)
                }

        }
```

Figura 5.5 - Algoritmo para a extração de acrônimos

5.2 O desenvolvimento

A figura 5.6 mostra a adaptação do processo de extração de acrônimo ajustado para operar dentro da plataforma GATE.

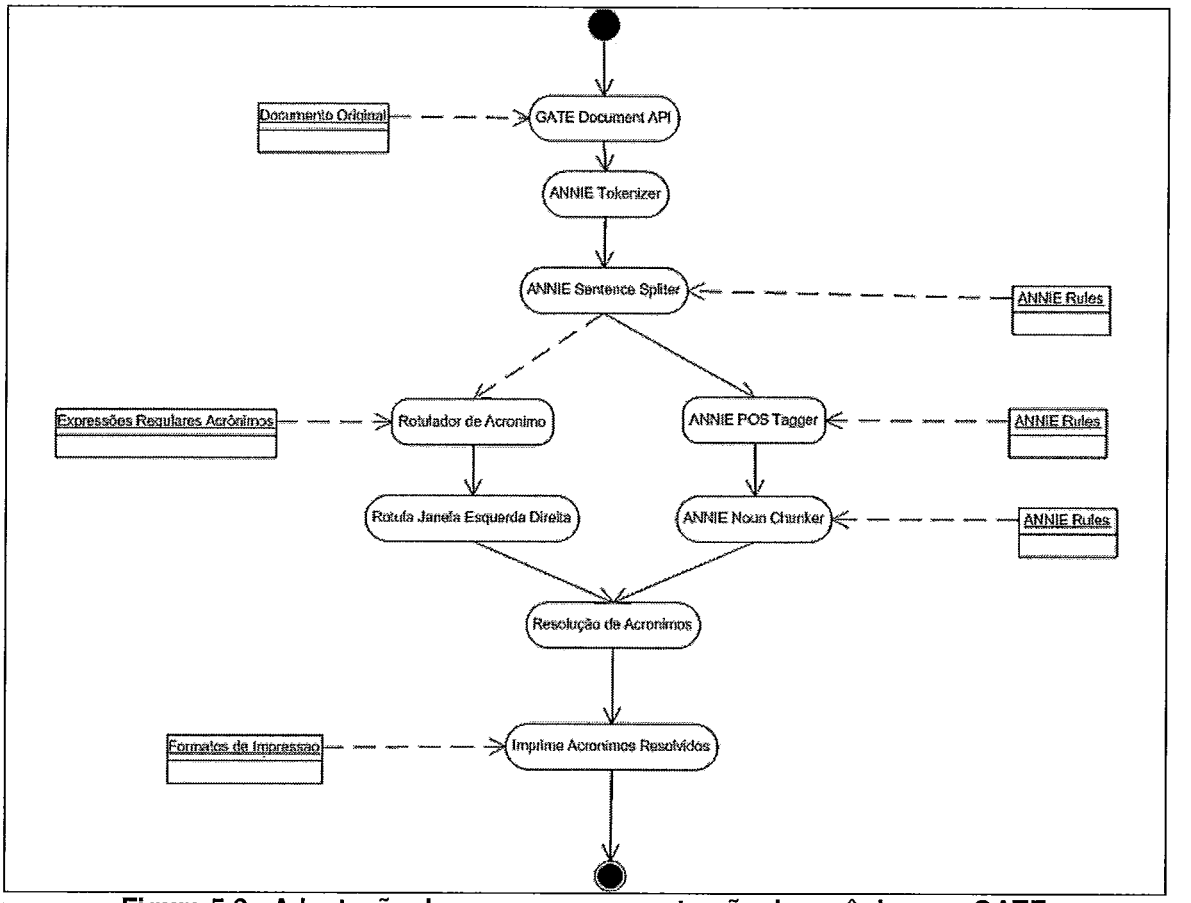


Figura 5.6 - Adaptação do processo para extração de acrônimo no GATE

A Figura 5.7 mostra a hierarquia das APIs utilizadas. Os módulos "MAQ HMM", "ROTULADOR ACRONIMO", "ROTULADOR JANELAS ESQ/DIR" e "Resolução de Acrônimo" foram desenvolvidos nesta dissertação.

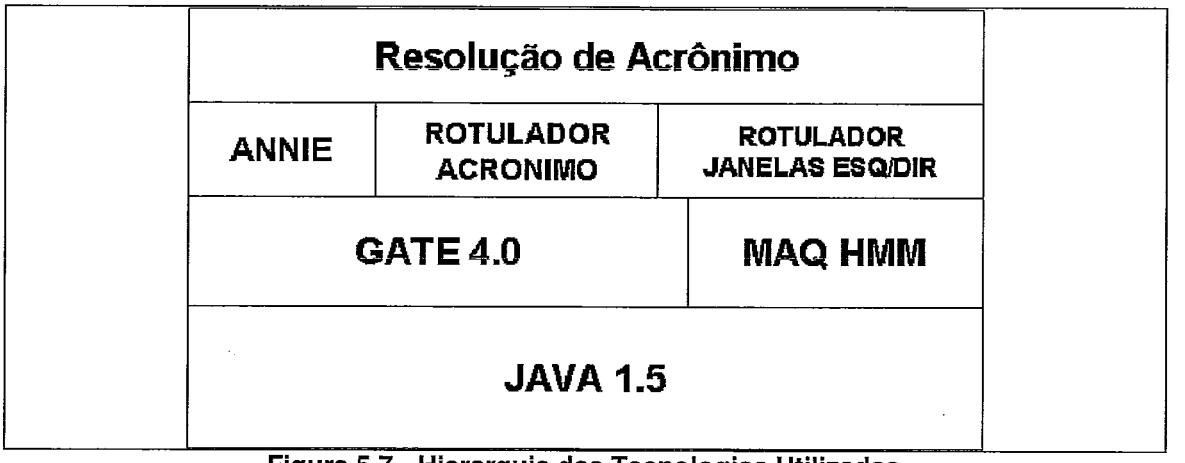


Figura 5.7 - Hierarquia das Tecnologias Utilizadas

Todo o sistema está desenvolvido em Java 1.5. Para utilizar o GATE foram importadas as API da plataforma para dentro do sistema de extração de acrônimos.

Foi desenvolvido um pacote, denominado de Máquina HMM, que implementa todas as classes necessárias para o processamento das funcionalidades de uma HMM conforme descritas no capítulo 3.

Reutilizaram-se alguns módulos do ANNIE (An Almost New Information Extraction System) que é o sistema de extração de informação residente no GATE.

Na Figura 5.6 observa-se que na atividade de conversão de texto utilizou-se a API do GATE pronta para converter documentos em diversos formatos (PDF, XML, HTML e texto) para o seu formato interno.

Utilizou-se o ANNIE English Tokenizer na atividade de tonenização e o ANNIE Sentence Splitter na atividade de segmentar o texto em sentenças.

Para o Rotulador POS e para o Rotulador de Sintagmas Nominais utilizou-se os módulos ANNIE POS Tagger e o ANNIE Noun Chunker respectivamente.

As expressões regulares desenvolvidas para a identificação dos acrônimos já foram descritas na Tabela 5.1. Através do GATE Transducer estas regras são convertidas em classes Java, que foram utilizadas no sistema.

As demais atividades foram desenvolvidas durante o desenvolvimento desta pesquisa e já descritas na seção 5.1.

5.3 O estudo do experimento

O objeto de estudo é avaliar a capacidade do sistema de extrair corretamente acrônimos de textos reais.

Seu objetivo é mostrar que um sistema probabilístico baseado em HMM tem desempenho, no mínimo, equivalente a de outros sistemas encontrados na literatura.

O foco de qualidade deste desempenho é a associação correta de acrônimos às suas respectivas expansões. Esta associação será medida através da precisão das associações e da cobertura de encontrar todas as tuplas <acrônimo, expansão> no texto.

Assim o estudo consistirá na **análise** da utilização do sistema para extração de acrônimos de textos reais; **com o propósito de** avaliar a sua capacidade de identificar corretamente as tuplas <acrônimo, expansão>; **referente** à precisão e à cobertura das tuplas extraídas **do ponto de vista da** comparação com outros sistemas encontrados na literatura, **no contexto de** textos na área biomédica.

5.3.1 O planejamento do experimento

A metodologia utilizada para avaliar o desempenho do sistema de extração de acrônimos é aquela descrita em PUSTEJOVSKY et al. (2001) e usual para a avaliação de sistemas de extração de informação conforme descrito em TURMO et. al.(2006).

Esta metodologia consiste na montagem de dois corpus com documentos selecionados aleatoriamente de um grande conjunto de documentos da área biomédica.

Em seguida especialistas anotam manualmente em cada um dos corpus todas as tuplas <acrônimo, expansão>. O conjunto destas tuplas anotadas manualmente é denominado de **padrão-ouro**.

O sistema de extração de acrônimos é desenvolvido utilizando apenas um dos corpus anotados, que passa a ser denominado de corpus de desenvolvimento.

A avaliação do desempenho do sistema é feita no segundo, que passa a ser denominado de corpus de teste.

O desempenho do sistema é avaliada através das medidas de precisão e cobertura que são definidas nas equações 5.1 e 5.2 respectivamente, onde $\#TC$ significa o número de tuplas corretas, $\#TE$ o número de tuplas extraídas e $\#TT$ o total de tuplas anotadas manualmente no texto, ou seja, o número de tuplas do padrão-ouro. TURMO et. al.(2006) definem ainda a medida F, conforme a equação 5.3, como a média harmônica entre precisão e cobertura[27].

$$\text{Pr}\,ecisão = \frac{\#TC}{\#TE} \tag{5.6}$$

$$Cobertura = \frac{\#TC}{\#TT} \tag{5.7}$$

$$F = \frac{(\beta^2 + 1) * \text{Pr}\,ecisão * Cobertura}{\beta^2 * \text{Pr}\,ecisão + Cobertura} \tag{5.8}$$

Uma tupla extraída é dita correta se o seu elemento expansão for igual ao elemento expansão da tupla na base-ouro, dado o mesmo acrônimo.

[27] TURMO et al. (2006) fazem $\beta = 1$

5.3.2 A execução do experimento

Na execução do experimento resolveu-se utilizar os dois corpus desenvolvidos por PUSTEJOVSKY et al. (2001)[28], conforme a metodologia descrita na seção 5.1.3, denominado "Medstract Gold Standard Evaluation Corpus".

A razão da sua utilização foi permitir comparar a eficiência da solução proposta nesta dissertação com as soluções propostas e publicadas por outros autores que utilizaram o mesmo corpus.

O corpus de desenvolvimento consistiu num conjunto de 86 resumos extraídos aleatoriamente da base de artigos publicados nos anos de 1997 e 1998 da Medline. As tuplas <acrônimo, expansão> foram identificadas manualmente por um especialista e desta identificação construiu-se a base ouro de desenvolvimento com 123 tuplas.

O corpus de teste foi construído com um conjunto de 100 resumos extraído com uma máquina de busca, utilizando o termo "gene", de um conjunto pequeno de periódicos de alto impacto na área biomédica. Nesta base as tuplas <acrônimo, expansão> corretas também foram identificadas manualmente por um especialista e desta identificação construiu-se a base ouro de teste com 168 tuplas.

Seguindo a metodologia o sistema de extração de acrônimo foi desenvolvido utilizando o corpus de desenvolvimento. Verificou-se logo no início que o "ANNIE Noun Chunker" (o rotulador de sintagmas nominais que acompanha o GATE) não estava preparado para a identificação de sintagmas nominais específicos da área biomédica. Desta maneira foi descartado e o desenvolvimento prosseguiu apenas com o conteúdo das janelas à direita e à esquerda.

Segundo NADEAU (2005) "O principal interesse nesse corpus é que foi anotado por um biólogo utilizando uma definição informal de um par válido. Assim o corpus reflete a interpretação humana de um par acrônimo-expansão, tornando a identificação de um acrônimo desafiante para um processo automático."

Suspeita-se de que por isto todas as soluções, de acordo com os artigos publicados, foram desenvolvidas e testadas em corpus modificados. CHANG et al. (2002) e SCHWARTZ (2003) apenas mencionam que modificaram o corpus. PUSTEJOVSK et al. (2001) mencionaram que retiraram 11 elementos, sendo 6 do corpus de desenvolvimento, mas não especificaram quais. NADEAU (2005) escreve

[28] Disponíveis em http://www.medstract.org.

que removeu acrônimos aninhados e que removeu ou corrigiu erros óbvios, sem, também, especificá-los.

Antes do processamento do corpus, todas as tuplas foram analisadas manualmente. Aquelas onde não se encontrou nenhum alinhamento entre o acrônimo e a expansão e aquelas em que havia um alinhamento óbvio entre o acrônimo e a expansão permaneceram inalteradas. Nenhum alinhamento entre os elementos da tupla configura-se a relação apelido-nome, que PUSTEJOVSKY et al. (2001) caracterizam como uma generalização da relação acrônimo-expansão.

Para cada uma das tuplas cujo alinhamento entre seus elementos não ficou claro, pesquisou-se na literatura da área a correção da anotação. Esta permaneceu inalterada, se não constatada a divergência. Constatada a divergência a anotação do especialista somente foi modificada quando a expansão encontrada na literatura caracterizava um alinhamento óbvio com o acrônimo.

Exemplificando[29], o acrônimo DET foi anotado pelo biólogo com "**bilateral transfer of two embryos**". Na literatura pesquisada constatou-se duas expansões comuns para o acrônimo, sendo uma "double embryo transfer", cujo alinhamento com "DET" é obvio, e "transfer of two embryos". A modificação foi efetuada para "bilateral **transfer of two embryos**". É importante esclarecer que a palavra "bilateral" não foi removida do texto, apenas a tupla da base-ouro passou de <DET, bilateral transfer of two embryos> para <DET, transfer of two embryos>.

Outro exemplo foi para "**protein kinase Fused** (Fu)". Na literatura encontraram-se as seguintes associações: "serine-threonine kinase Fused (Fu)"[30]; "...a large protein complex consisting of the Fused (Fu), Costal2 (Cos2), and Cubitis interruptus (Ci) proteins..."[31]; e no próprio corpus existe o acrônimo "Su(Fu)", reduzindo "Suppressor of Fused". Estes exemplos, entre vários outros, onde existe uma óbvia associação de "Fused" com "Fu" levaram a modificar da anotação original para "protein kinase **Fused** (Fu).

---

[29] As anotações, tanto a original quanto a modificada, estão em negrito.

[30] http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=pubmed&dopt=AbstractPlus&list_uids=14523402

[31] http://www.ncbi.nlm.nih.gov/sites/entrez?cmd=Retrieve&db=pubmed&dopt=AbstractPlus&list_uids=11934882

Um exemplo em que a anotação não foi mudada após esta análise foi para LRP1 que reduz neste corpus "**Low-density lipoprotein receptor-related protein gene**". A suspeita de que a correta expansão seria "lipoprotein receptor-related protein gene" não foi comprovada na literatura.

Efetuadas as modificações processou-se o corpus de desenvolvimento. As regras de rotulação dos candidatos a acrônimos (aquelas que estão na Tabela 5.1 são resultantes desta fase) e a topologia da HMM (através dos ajustes nos pesos das transições entre estados e no número máximo de caracteres de cada um deles) foram ajustadas para maximizar, neste corpus, o número de acrônimos identificados.

Terminada esta etapa processou-se o corpus de teste, sem modificações nas regras e na topologia da HMM, concluindo o experimento.

5.4 Análise dos resultados do experimento

5.4.1 Análise Quantitativa do Experimento

Conforme descrita na fase de planejamento a avaliação do desempenho do sistema de extração de acrônimos é feita utilizando as medidas de precisão, cobertura e fator F conforme especificadas nas equações (5.6), (5,7) e (5.8) respectivamente.

Na Tabela 5.2 encontram-se os resultados alcançados nesta pesquisa tanto no corpus de desenvolvimento quanto no corpus de teste. No Apêndice A estão os relatórios resultantes da extração de onde os números das tabelas foram computados. Na Tabela 5.3 encontra-se o número de tuplas da base ouro, base extraída, tuplas corretas, tuplas incorretas, falso acrônimo e tuplas não identificadas.

Tuplas não identificadas são aquelas que constam da base ouro, porém não foram extraídas pelo sistema.

Falso acrônimo é aquele cuja tupla <acrônimo, expansão> foi extraída, mas que não constava na base ouro. Somente foi identificado um caso. No texto "The purpose of this study was to develop and evaluate a rapid microdose **14C-urea breath test (14C-UBT)** with a simplified protocol for detecting the infection of..." o sistema extraiu a tupla <14C-UBT, 14C-urea breath test> do conteúdo ressaltado em negrito.

Avaliando-se este caso à luz das regras conhecidas de geração de acrônimos pode-se concluir de que se trata de um acrônimo. Pesquisou-se na literatura se este caso

constitui uma redução comum para "14C-urea breath test", verificando-se que sim[32] . Concluiu-se que, apesar de ter sido tratado como um falso acrônimo, trata-se de um verdadeiro, exemplificando o desafio de identificar acrônimos tanto para um especialista quanto para um sistema automático.

Tabela 5.2 - Resumo dos resultados

Corpus de Desenvolvimento			Corpus de Teste		
Precisão	Cobertura	F	Precisão	Cobertura	F
92,00%	88,00%	89,96%	93,50%	85,70%	89,43%

Tabela 5.3 - Avaliação quantitativa dos resultados

	Base Desenvolvimento	Base de Teste
Base Ouro	126	168
Base Extraída	120	154
Corretos	111	144
Incorretos	8	10
Falso Acrônimo	1	0
Não Identificados	6	14

Para facilitar a análise dos resultados alcançados resolveu-se dividir as tuplas acrônimos em três grupos. O primeiro, denominado de APELIDO, encontram-se os casos onde o elemento acrônimo da tupla consiste num apelido da expansão. O segundo grupo, denominado de NÃO COMPLEXO, estão os acrônimos formados com a letra inicial das palavras que compõe a expansão (que consiste a regra (1) da Tabela 2.1). O terceiro, denominado de COMPLEXO, estão os demais casos. Um exemplo para o grupo APELIDO é < Spherix, Bacillus sphaericus B-101 serotype H5a 5b>[33]. Um exemplo para o segundo caso é <COPD, Chronic Obstructive Pulmonary Disease>. Um exemplo para o terceiro caso é <CIDR, Intravaginal Progesterone Device>.

---

[32] http://www.ncbi.nlm.nih.gov/sites/entrez?Db=pubmed&Cmd=ShowDetailView&TermToSearch=16372580&ordinalpos=4&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum

[33] O elemento expansão neste caso é "Bacillus sphaericus B-101, serotype H5a, 5b". As vírgulas foram retiradas para facilitar o leitor no reconhecimento dos elementos da tupla.

São 3 as razões desta classificação. A primeira é que a resolução das tuplas elementos do primeiro grupo estão além dos objetivos desta disseitação. Segundo PUSTEJOVSKY et al. (2001) "... são expressões do tipo apelido mais complexa do que pares acrônimo-significado e devem ser tratadas por uma estratégia diferente. Tal estratégia deverá utilizar informação adicional àquelas que constam simplesmente nas "strings...".

A segunda razão, decorrente da caracterização dos elementos deste grupo, é que o sistema de identificação deve resolver todas as tuplas que constituem este grupo. Um simples autômato pode comparar as letras iniciais de uma expansão candidata com cada uma das letras do acrônimo e identificar todos.

A terceira é que a resolução dos elementos do grupo COMPLEXO transcende a capacidade do autômato descrito na segunda razão, necessitando de muito mais regras do que aquela que simplesmente caracteriza o grupo NÃO COMPLEXO.

A figura 5.3 contém uma tabela com o número de tuplas classificadas em cada um destes grupos, informando o número de resolvidas e de não resolvidas.

Tabela 5.4 - Resultado da extração por complexidade do acrônimo

	Corpus de Desenvolvimento			Corpus de Teste		
	Apelido	Não Comple	Complexo	Apelido	Não Comp	Complexo
Corretos	0	57	56	1	54	91
Incorretos	4	0	9	12		10
Total	4	57	65	13	54	101

Nos dois corpora todas as tuplas classificadas como não complexas foram corretamente extraídas. Este resultado era o esperado, pois a resolução dos elementos deste grupo é simples.

As tuplas do grupo Apelido não foram resolvidas com a exceção de uma. Este também foi um resultado esperado uma vez que o sistema foi desenvolvido para extrair apenas acrônimos. A tupla extraída estava no seguinte texto: "Alpha-Tocopherol (vitamin E) is a lipid-soluble antioxidant...". Trata-se do começo de uma sentença e o conteúdo da janela à esquerda do acrônimo ficou reduzido à expressão "Alpha-Tocopherol". Com uma HMM pouco restritiva (permitindo transições à esquerda e adjacente à esquerda) o modelo foi de um estado inicial para o estado "A", transitando para o estado "T" e deste transitou para o estado final. A tupla corretamente extraída foi <vitamin E, alpha-Tocopherol>. Trata-se de uma situação muito particular, pois a

expansão está entre o início da sentença e o acrônimo e, ainda, a primeira palavra da expansão começa com uma letra do acrônimo (no caso a letra A).

No geral, considerando o corpora processado, um aspecto positivo a se destacar é que a queda do desempenho entre o processamento do corpus de desenvolvimento e do de teste não foi significativa. Enquanto houve uma queda da cobertura (de 88% para 86%) houve uma subida da precisão (de 92% para 94%), levando a medida F a permanecer a mesma.

5.4.2 Análise comparativa do experimento

A comparação entre as soluções não é, a rigor, possível por duas razões. A primeira é de que não dispomos das soluções dos demais autores para processá-las diversas vezes em corpus distintos para gerar um volume de resultados que permitam a comparação com alguma significância estatística. A segunda razão reside no fato dos autores terem modificado o corpus. Assim os experimentos foram realizados em corpus distintos.

Apesar disto elaborou-se o quadro da figura 5.10 baseado nos dados publicados.

Tabela 5.5 - Comparação de alguns algoritmos para resolução de acrônimos[34]

	Precisão	Cobertura	F1
PUSTEJOVSKY et al. (2001)	**98,00%**	72,00%	83,00%
CHANG et al. (2002)	80,00%	83,00%	81,50%
SCHWARTZ (2003)	96,00%	82,00%	88,40%
NADEAU(2005)	92,50%	84,40%	88,30%
Dissertação	93,50%	**85,70%**	**89,80%**

5.4.3 Análise qualitativa do experimento

A solução descrita nesta dissertação foi desenhada para atender as regras descritas no capítulo 2. Das 294 tuplas encontradas nos dois corpora o sistema resolveu corretamente 256, ou seja, 87%. Esta seção aborda basicamente os 38 casos em que falhou.

Dos 38 casos, 16 (ou 42% dos 38) foram classificados como apelido e portanto estão fora do escopo da solução.

[34] Em negrito estão os maiores valores de cada medida.

Em 8 casos (ou 21%) o sistema identificou uma expansão maior do que a correta. A expansão correta consistia, portanto, numa "substring" da expansão extraída. Exemplos são:

- Para o acrônimo InsP3 foi extraída a expansão "Injection of inositol 1,4,5-triphosphate " quando o correto seria "inositol 1,4,5-triphosphate ";
- Para o acrônimo SKIPS foi extraída a expansão "screen to identify syk kinases-interacting proteins" quando o correto seria "syk kinases-interacting proteins"; e
- Para o acrônimo TSH foi extraído "target tissue for thyrotropin" quando a correta seria apenas "thyrotropin".

Nos dois primeiro casos o problema reside na identificação da primeira palavra que compõe a expansão. Quando submetidas as expansões corretas à HMM, foram retornadas probabilidades maiores do que as probabilidades das expansões extraídas. Supõe-se que um rotulador de sintagmas nominais ajustado para a área biomédica proporia à HMM, além das expansões das janelas, as expansões corretas, pois "inositol 1,4,5- triphosphate" e "syk kinases-interacting proteins" são sintagmas nominais.

TSH, no caso, é uma abreviação de "thyrotropin". O caractere S impediu que o estado da cadeia TSH emitisse o símbolo thyrotropin. O alinhamento foi para "**t**arget tissue for **th**yrotropin"[35]. A extração está errada, mas existe um alinhamento das expansão extraída com as regras de formação de acrônimo, descritas no capítulo 2.

Em 11 casos (ou 29%) o sistema identificou uma expansão menor do que a correta. A expansão extraída consistia, portanto, numa "substring" da expansão correta. Exemplos são:

- Para o acrônimo TH foi extraída a expansão "helper T " quando o correto seria "CD4 helper T ".
- Para o acrônimo TEMPS-I foi extraída a expansão "temperament interview" quando o correto seria "semi-structured affective temperament interview"; e
- Para o acrônimo JNK foi extraída "nh2-terminal kinase" quando a correta seria apenas "c-jun nh2-terminal kinase".

Os dois primeiros casos esbarram na limitação da solução. PUSTEJOVSKY et al. também extraíram a expressão "helper T", mas a consideram correta. Na opinião dos

[35] As letras alinhadas estão em negrito.

autores "não trata-se de um falso positivo, mas uma extração parcial de um híbrido apelido/acrônimo."

O caso de TEMPS-I pode ser classificado como um erro de anotação. A expansão que encontramos na literatura para este acrônimo foi "Temperament Assessment of Memphis, Pisa, Paris and San Diego-Interview"[36]. Não encontrou-se na outro texto com o acrônimo TEMPS-I reduzindo "semi-structured affective temperament interview" e, por isto, não modificou-se a anotação. No corpus o texto consiste no seguinte: "...by means of a semi-structured affective temperament interview (TEMPS-I) at T0 and T1 two years later.". Suspeita-se do erro de anotação pois encontramos em textos semelhantes o acrônimo TEMPS-A reduzindo "temperament auto-questionaire".

O caso do acrônimo JNK pode ser resolvido com uma melhoria no filtro de identificação da primeira palavra para tratar palavras compostas.

Nos últimos 3 casos (ou 8%) os acrônimos foram compostos segundo a regra (10) da Tabela 2.1 (Associação simbólica) que não foi implementada.

Com os resultados alcançados e comparando-os com os resultados alcançados por outras soluções, podemos concluir que o sistema de extração de acrônimos proposto nesta dissertação tem desempenho, no mínimo, equivalente as demais.

5.5 Tempo de processamento

O experimento foi realizado em uma máquina com processador Intel Pentium 4 (3,2 GHz) e com 1 GB de memória RAM, rodando Windows XP Professional.

Os 126 acrônimos do corpus de desenvolvimento foram processados em 53s (uma média de 2,38 acrônimos por segundo). Foram necessários 91s para processar os 168 acrônimos do corpus de desenvolvimento (uma média de 1,84 acrônimos por segundo). A redução do tempo médio da base de desenvolvimento para a base de teste é decorrente do processo de iniciação do GATE, que é independente do número de acrônimos ou do número médio de caracteres do mesmo.

---

36 http://www.ncbi.nlm.nih.gov/sites/entrez?Db=pubmed&Cmd=ShowDetailView&TermToSearch=15780672&ordinalpos=7&itool=EntrezSystem2.PEntrez.Pubmed.Pubmed_ResultsPanel.Pubmed_RVDocSum

O tempo de processamento deste algoritmo cresce linearmente em relação ao número de acrônimos e de forma quadrática em relação ao número de caracteres médio destes termos.

Capítulo 6 – Conclusão

6.1 Os resultados alcançados

Os resultados alcançados no corpus de teste de precisão com 93,5%, cobertura com 86,00% e fator F com 89,80% indicam que existe uma função estocástica que associa um acrônimo à sua expansão. Além disto, do conjunto não identificado, verificou-se que 42% são, na realidade, apelidos, que não possuem uma função de associação com a sua expansão e, para serem associados com as expressões que reduzem, precisam de informações adicionais do que aquelas contidas nas strings manipuladas (PUSTEJOVSKY et al., 2001). Retirando-se estes apelidos as três medidas (precisão, cobertura e fator F) teriam valor igual a 93,5%.

6.2 A solução

A extração automática de acrônimos de texto consiste num desafio pelo grande volume de novos termos que surgem nas diversas áreas de conhecimento, tornando a tarefa de manter manualmente um glossário atualizado dispendiosa.

O objetivo do trabalho aqui apresentado é extrair tuplas <acrônimo, expansão> de texto não estruturado escrito em linguagem natural, desde que estas tuplas estejam no formato restrito àqueles descrito na Tabela 4.1.

O problema de resolução de acrônimo é dividido nas fases de identificação dos elementos das tuplas no texto, da associação correta dos pares acrônimo-expansão identificados e da desambigüação que trata de encontra a correta expansão quando mais de uma for possível para um dado acrônimo.

Abordou-se a extração de acrônimos como um problema de extração de informação de texto. Das tarefas descritas no template da ACE (capítulo 2), foram apenas necessárias adaptar para esta trabalho a "Reconhecimento e Detecção de Entidades" e a "Detecção e Reconhecimento de Relação". Para a construção do sistema adaptou-se a arquitetura proposta por TURMO et al. (2006) também descrita no capítulo 2.

O sistema foi desenvolvido utilizando a biblioteca do GATE, conforme descrito no capítulo 3.

Viu-se como HMM são utilizadas na extração de informação e por isto a atividade de "Detecção e Reconhecimento de Relação" foi desenvolvida baseada neste modelo.

A solução está desenvolvida baseada na hipótese de que existe uma função estocástica que associa um acrônimo à sua expansão. O acrônimo constitui a cadeia e seus caracteres constituem os estados. O conjunto de símbolos é formado pelas palavras da expansão.

A função estocástica de associação (seção 4.3.3.2) foi desmembrada em dois conjuntos, sendo que no primeiro há cinco funções de transição de estado, ou seja, funções que definem de que forma é possível transitar pela cadeia, e duas funções de emissão de símbolo, que indicam a possibilidade ou não de uma palavra emergir de um caractere. Estas funções englobam a maioria das regras de formação de acrônimos proposta por ZAHARIEV (2004) .

A solução desenvolvida utiliza cinco expressões regulares para identificar candidatos a acrônimos, ou seja, a tarefa de "Reconhecimento e Detecção de Entidades". Baseadas no fato de que a maioria das expansões encontra-se adjacentes aos mesmos, duas expressões são extraídas, sendo uma com o conteúdo da janela à esquerda e a outra com o conteúdo da janela à direita.

O candidato a acrônimo e suas possíveis expansões são submetidas a uma atividade que calcula a probabilidade, utilizando a "Forward Procedure" da HMM descrita no capítulo 3, de cada uma das expressões emergirem do acrônimo. Aquela com a maior probabilidade é eleita a expansão que contém o significado do acrônimo, tornando positiva a identificação. No caso destas probabilidades serem nulas a identificação é considerada negativa.

6.3 As melhorias da solução

O sistema pode ser aperfeiçoado submetendo-se, em adição ao conteúdo das janelas, os sintagmas nominais, conforme proposto por PUSTEJOVSKY et al. (2001). Isto diminuiria o número dos falsos positivos, ou seja, aqueles cuja expansão identificada casava parcialmente com a expansão correta.

6.4 A principal contribuição

A principal contribuição desta dissertação está na proposta de que existe uma função estocástica que associa um acrônimo a sua expansão. Esta função,

implementada em HMM, foi dividida em 5 funções de transição de estado e duas de emissão de símbolo.

Com isto apenas 5 expressões regulares foram necessárias para identificar no texto candidatos a acrônimos e, adicionalmente, duas heurísticas para limitar o início da expansão à uma palavra que comece com uma letra do acrônimo.

Esta contribuição ganha relevância quando comparada com as demais soluções estudadas, que baseiam a identificação de um acrônimo em um conjunto de regras relativamente grande. Alguns autores como NADEAU (2005) e SCHUMANN (2005) utilizam técnicas de aprendizagem de máquina para reduzir este conjunto, que, mesmo depois de reduzido, possui mais de uma centena de elementos.

Os resultados alcançados também fortalecem a hipótese de que é viável o processamento de linguagem natural com técnicas estatísticas e corrobora com o fato destas precisarem de menor conhecimento do contexto para serem aplicadas.

6.5 Trabalhos futuros

Este trabalho foi desenvolvido com texto em inglês. Para o português faz-se necessário o desenvolvimento das seguintes ferramentas:

- Um tokenizador conforme descrito na seção 5.1.2;
- Um segmentador de sentenças conforme descrito na seção 5.1.3;
- Um rotulador morfológico de palavras conforme descrito na seção 5.1.6;
- Um rotulador morfológico de sintagmas nominais conforme descrito na seção 5.1.7; e
- Um dicionário para ser consultado pelo rotuladores acima, contendo as classes morfológicas das palavras.

Estas ferramentas podem então substituir aquelas específicas da língua inglesa para o tratamento do texto em português.

Nesta dissertação extraíram-se as expansões que estavam adjacentes ao acrônimo. Consiste num desenvolvimento futuro a extração da expansão não adjacente, ou seja, tuplas <acrônimo, expansão> que não estão nos formatos da Tabela 4.1.

A desambigüação, cujo objetivo é reconhecer o correto significado de um acrônimo quando este possuir mais de um, faz parte da resolução de acrônimos. Esta tarefa não foi tratada nesta pesquisa e constitui um trabalho futuro.

O sistema deve ser ampliado para extrair as tuplas <apelido, expansão> que, como já descrito, precisará de informação adicional além daquelas contidas nas strings manipuladas. Esta ampliação constitui outra atividade futura.

Os apelidos também não são abordados nesta dissertação e suas extrações também fazem parte do conjunto de tarefas futuras.

Bibliografia

AGICHTEIN E., CUCERZAN S., 2005, "Predicting accuracy of extracting information from unstructured text collections", *Proceedings of the 14th ACM international conference on Information and knowledge management*, SESSION: Paper session KM-4 (knowledge management): information extraction, PP. 413-420, Bremen, Germany.

BICK E., "Portuguese interactive syntax learning – palavras", http://visl.hum.ou.dk/visl/pt/, 2006.

BONTCHEVA K., TABLAN V., MAYNAR D., CUNNINGHAM H., 2004, "Evolving GATE to Meet New Challenges in Language Engineering", *Natural Language Engineering*. **10** (3/4), pp. 349-373

BORBA F. S. (Org.), 2004, *Dicionário UNESP do português contemporâneo*, Editora UNESP, ISBN 85-7139-576-4.

BRÜNINGHAUS S., ASHLEY K. D., 2001, "Improving the representation of legal case texts with information extraction methods", *Proceedings of the 8th international conference on Artificial intelligence and law*, pp 42 – 51, St. Louis, Missouri, United States.

BUENO T., WANGENHEIM C. G., MATTOS E. S., HOESCHL H. C., BARCIA R. M., 1999, "jurisConsulto: retrieval in jurisprudencial text bases using juridical terminology", *Proceedings of the 7th international conference on Artificial intelligence and law*, PP. 147-155, Oslo, Norway.

CALIFF M. E., MOONEY R. J., 2003, "Bottom-Up Relational Learning of Pattern Matching Rules for Information Extraction", *Journal of Machine Learning Research,* vol 4, pp. 177-210.

CARENINI G., NG R. T., ZWART E., 2005, "Extracting knowledge from evaluative text", *Proceedings of the 3rd international conference on Knowledge capture*. SESSION: Information extraction, PP. 11-18, Banff, Alberta, Canada.

CARVALHO V. R., COHEN W. W., 2004, "Learning to Extract Signature and Reply Lines from Email", in Proc. of the 2004 Conference on Email and Anti-Spam. Mountain View, California.

CHANG J. T., SCHÜTZE H., ALTMAN R. B., 2002, "Creating an Online Dictionary of Abbreviations from MEDLINE", *Journal of American Informatics Association* (JAMIA), 9(6):612-620.

CIARAMITA M., ALTUN Y., 2006, "Broad-Coverage Sense Disambiguation and Information Extraction with a Supersense Sequence Tagger", *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP).*

CIRAVEGNA F., DINGLI A., WILKS Y., PETRELLI D., 2002, "Amilcare: adaptive information extraction for document annotation", *Proceedings of the 25th annual international ACM SIGIR conference on Research and development in information retrieval*, POSTER SESSION: Poster session, pp. 367-368, Tampere, Finland.

CORMEN T. H., LEISERSON C.E., RIVEST R. L., STEIN C., 2002, *Algoritmos*, 2a. edição, Elsevier Editora Ltda., ISBN 85-352-0926-3.

COLLINS M., 1997, "Three Generative, Lexicalized Models for Statistical Parsing", *Proceedings of the Thirty-Fifth Annual Meeting of the {A}ssociation for {C}omputational {L}inguistics and Eighth Conference of the {E}uropean Chapter of the {A}ssociation for {C}omputational {L}inguistics,* Association for Computational Linguistics, pp. 16-23.

CULOTTA A., SORENSEN J., 2004, "Dependency tree kernels for relation extraction", *Proceedings of the 42nd Annual Meeting on Association for Computational Linguistics*, Article No. 423, Barcelona, Spain.

CUNNINGHAM H., 1999, "A Definition and Short History of Language Engineering", *Journal of Natural Language Engineering*, pp 1-16, vol 5, 1999.

CUNNINGHAM H., MAYNAR D., BONTCHEVA K., TABLAN V., 2002, "GATE: A Framework and Graphical Development Environment for Robust NLP Tools and Applications", *Proceedings of the 40th Anniversary Meeting of the Association for Computational Linguistics (ACL'02)*. Philadelphia, July 2002.

FREITAG D., 1998, "Information Extraction from HTML: Application of a General Machine Learning Approach", In: Proceedings of the 15th Conference on Artificial Intelligence (AAAI-98). pp. 517-523.

FREITAG D., McCALLUM A., 1999, "Information extraction with HMM and shrinkage", In Proceedings of the AAAI Workshop on Machine Learning for Information Extraction.

FREITAG D., McCALLUM A., 2000, "Information extraction with HMM structures learned by stochastic optimization", *In Proceedings of the 17th. AAAI National Conference on Artificial Intelligence.*

FREITAG D., 2005, "Morphology Induction From Term Clusters", *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*, pp. 128–135, Ann Arbor, June.

FREITAS, M. C. ; GARRAO, M. U. ; OLIVEIRA, C. ; SANTOS, C. N. ; Silveira, 2005, "A anotação de um corpus para o aprendizado supervisionado de um modelo de SN". *In: III TIL - Workshop de Tecnologia da Informação e da Linguagem Humana*, 2005, São Leopoldo, RS. Anais do XXV Congresso da Sociedade Brasileira de Computação, 2005, 2005.

GUPTA S., KAIER G., NEISTADT D., GRIMM P., 2003, "DOMBased Content Extraction of HTML Documents", *In WWW2003 proceedings of the 12 Web Conference*, Budapest, Hungary, pp 207-214.

HAN H., MANAVOGLU E., ZHA H., TSIOUTSIOULIKLIS K., GILES C. L., ZHANG X., 2005, "Rule-based word clustering for document metadata extraction", *Proceedings of the 2005 ACM symposium on Applied computin*, SESSION: Information access and retrieval (IAR), pp. 1049-1053, Santa Fe, New Mexico.

HENRIQUES C. C., 2007, "Morfologia: Coleção Português na Prática", Editora Campus, ISBN 978-85-352-2277-7.

HOBBS J., 1993, "The Generic Information Extraction System", In Proceedings of the 5th. Message Understanding Conference (MUC 5).

IRESON N., CIRAVEGNA F., CALIFF M. E., FREITAG D., KUSHMERICK N., LAVELLI A., 2005, "Evaluating machine learning for information extraction", *Proceedings of the 22nd international conference on Machine learning*, pp. 345 – 352, Bonn, Germany.

JOHNSON-LAIRD P. N., 1988, *THE COMPUTER AND THE MIND: An introduction to cognitive science*, Harvard University Press, ISBN 0-674-15615-3.

KEPLER F. N., 2005, *Um Etiquetador Morfo-Sintático Baseado em Cadeias de Markov de Tamanho Variável*, Dissertação de Mestrado, Universidade de São Paulo, São Paulo, SP.

LAFFERTY J., MCCALLUM A., PEREIRA F., 2001, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data", *Proceedings 18th International Conference on Machine Learning*.

LARKEY L. S., 1996, "Combining Classifiers in Text Categorization Combining Classifiers in Text Categorization", *In Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval.*

LAWRENCE S., GILES C. L., BOLLACKER K., 1999, "Digital Libraries and Autonomous Citation Indexing", IEEE Computer, Volume 32, Number 6, pp. 67-71, 1999.

LI H., CAO Y., XU J., HU Y., LI S., MEYERSON D., 2005, "A new approach to intranet search based on information extraction", *Proceedings of the 14th ACM international conference on Information and knowledge management*, SESSION: Industry track session, pp. 460-468, Bremen, Germany.

LIN W., LAM W., 2000, "Learning to extract hierarchical information from semi-structured documents", *Proceedings of the ninth international conference on Information and knowledge management*, PP. 250-257, McLean, Virginia, United States.

LUGER G. F., 2004, Inteligência Artificial Estruturas e Estratégias para a Solução de Problemas Complexos, 4ª. Edição, Bookman, ISBN 85-363-0396-4.

MANNING C. D., SCHÜTZE H., 1999, *Foundations of Statistical Natural Language Processing*, The MIT Press, ISBN 0-262-13360-1.

MCCALLUM A., FREITAG, D., PEREIRA, F., 2000, "Maximum entropy Markov models for information extraction and segmentation", *In Proceedings of the 17th International Conference on Machine Learning (ICML).*

MCCALLUM A., WELLNER B., 2003, "Toward conditional models of identity uncertainty with application to proper noun coreference". *In Proceedings of the IJCAI-2003*, Workshop on Information Integration on the Web, pP 79–86, Acapulco, Mexico, August.

MCCALLUM A., 2005, "Distilling Structured Data From Unstructured Text", *ACM Queue*, vol. 3, no. 9 - November 2005.

MESQUITA R. M., 1996, *Gramática da língua portuguesa*, 5ª. edição, Editora Saraiva, ISBN 85-02-01423-4.

MINKOV E., WANG R., COHEN W. W., 2005, "Extracting personal names from email: applying named entity recognition to informal text", *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, PP. 443-450, Vancouver, British Columbia, Canada.

MUSLEA I., MINTON S., KNOBLOCK C., 1998, "Stalker: Learning extraction rules for semistructured, web-based information sources", *In AAAIWorkshop on AI and Information Integration*, 1998.

NG V., CARDIE C., 2002, "Improving Machine Learning Approaches to Coreference Resolution", *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

NIST ACE, 2007, *The ACE 2007 (ACE07) Evaluation Plan*, http://www.nist.gov/speech/tests/ace/2007/doc/ace07-evalplan.v1.3a.pdf.

NADEAU D., TURNEY P., 2005, "Supervised Learning Approach to Acronym Identification", *18th Canadian Conference on Artificial Intelligence (AI'2005),* LNAI 3501, 10 Pages, Victoria, BC, Canada, NRC 48121.

NAUGHTON M., CARTHY J., KUSHMERICK N., 2006, "Clustering sentences for discovering events in news articles", *Proceedings of the European Conference on Information Retrieval (ECIR-06).*

OLIVEIRA D. M., GIRÃO K. T., ARAÚJO F. F., COSTA M. P., PACHECO A. C., ARAÚJO-FILHO R., VIANA D. A., COSTA R. B., MAGGIONI R., 2007, "Mining Genes of WD-40 Superfamily in Leishmania through Hidden Markov Models and Natural Clustering", *VLDB '07, pp 23-28*, 2007, Vienna, Austria, ACM

PARK Y., BYRD R.J., 2001, "Hybrid text mining for finding abbreviations and their definitions", *Conference on Empirical Methods in Natural Language Processing (EMNLP).*

PARK Y., BYRD R., BOGURAEV B., 2002, "Automatic glossary extraction: beyond terminology identification", *Proceedings of the 19th international conference on Computational linguistics,* vol. 1, pp. 1-7, Taipei, Taiwan.

PINTO D., MCCALLUM A., WEI X., CROFT W. B., 2003, "Table extraction using conditional random fields", *Proceedings of the 2003 annual national conference on Digital government research*, vol. 130, pp. 1-4, Boston, MA.

PUSTEJOVSKY J., CASTAÑO J., COCHRAN B., KOTECKI M. , MORRELL M., 2001, "Automatic Extraction of Acronym-meaning Pairs from MEDLINE Databases", *Proceedings of the 10th World Congress on Medical Informatics*, Volume 84 / 2001, pp 371-375.

RABINER L. R., 1989, "A Tutorial on Hidden Markov Models and Selected Applications in Speech Recognition", *Proceedings of IEEE*, vol. 77, no. 2.

RAY S., CRAVEN M., 2001, "Representing Sentence Structure in Hidden Markov Models", In Proceedings of the 17th. International Joint Conference on Artificial Intelligence (IJCAI01).

RILOFF E., 1993, "Automatically Constructing a Dictionary for Information Extraction Tasks", *National Conference on Artificial Intelligence*, pp 811-816.

RILOFF E., LEHNERT W., 1994, "Information extraction as a basis for high-precision text classification", *ACM Transactions on Information Systems (TOIS)*, vol. 12, pp. 296-333, New York, NY, USA.

ROSENFELD D., FELDMAN R., AUMANN Y., 2002, "Structural extraction from visual layout of documents", *Proceedings of the eleventh international conference on Information and knowledge management*, SESSION: Information extraction and text segmentation, pp. 203-210, McLean, Virginia, USA.

ROTH D., YIH W., 2002, "Probabilistic reasoning for entity & relation recognition", *In The 20th International Conference on Computational Linguistics.*

RUSSEL S., NORVIG P., 2003, *Artificial Intelligence a Modern Approach*, 2nd Edition, Prentice Hall, ISBN 0-13-790395-2.

SANTOS C. N., 2005, *Aprendizado de máquina na identificação de sintagmas nominais: O caso do português brasileiro*, Dissertação de Mestrado, Instituto Militar de Engenharia, Rio de Janeiro, Brasil.

SARAWAGI S., COHEN W., 2004, "Semimarkov conditional random fields for information extraction", In Proceedings of ICML 2004.

SCHILDER F., 2004, "Extracting meaning from temporal nouns and temporal prepositions", *ACM Transactions on Asian Language Information Processing (TALIP)*, vol. 3, pp. 33-50, New York, NY, USA.

SCHWARTZ A. S., HEARST M. A., 2003, "A Simple Algorithm for Identifying Abbreviation Definitions in Biomedical Text", *Pacific Symposium on Biocomputing* 8:451-462.

SHEN D., ZHANG J., ZHOU G., SU J., TAN C., 2003, "Effective adaptation of a Hidden Markov Model-based named entity recognizer for biomedical domain", *Proceedings of the ACL 2003 workshop on Natural language processing in biomedicine*, vol 13, pp. 49-56, Sapporo, Japan.

SKOUNAKIS M, Craven M., SOUMYA R., 2003, "Hierarchical Hidden Markov Models for Information Extraction", *Proceedings of the 18th. International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, Morgan Kaufmann.

SÖDEBERG J. J., 2007, *Multidimension Hidden Markov Model Applied to Image and Video Analysis*, Tese de Doutorado, Institute Eurecom Sophia-Antipolis, France.

SODERLAND S., LEHNERT W., 1996, "Corpus-Driven Knowledge Acquisition for Discourse Analysis", *National Conference on Artificial Intelligence.*

SODERLAND S., 1997, "Learning to Extract Text-based Information from the World Wide Web", Proceedings of Third International Conference on Knowledge Discovery and Data Mining (KDD-97).

SUCHANEK F., M., IFRIN G., WEIKUM G., 2006, "Combining linguistic and statistical analysis to extract relations from web documents", *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, POSTER SESSION: Research track posters, pp. 712-717, Philadelphia, PA, USA.

TAGHVA K., GILBRETG J., 1999, "Recognizing acronyms and their definitions", International *Journal on Document Analysis and Recognition*, Springer-Verlag, vol 1, pp 191-198.

TURMO J., AGENO A., CATALÀ N., 2006, "Adaptive information extraction", *ACM Computing Surveys (CSUR)*, Volume 38 , Issue 2 Article No. 4.

YU H., KIM W., HATZIVASSILOGLOU V., WILBUR J., 2006, "A large scale, corpus-based approach for automatically disambiguating biomedical abbreviations", *ACM Transactions on Information Systems (TOIS)*, vol. 24, pp. 380-404, July.

WANG X., MOHANTY N., MCCALLUM A., 2005, "Group and topic discovery from relations and text", *Proceedings of the 3rd international workshop on link discovery*, pp. 28-35, Chicago, Illinois.

WINTERS-HILT S., 2006, "Hidden Markov Model Variants and Their Application", The Third Annual Conference on the MidSouth Computational Biology and Bioinformatics Society, Louisiana.

ZAHARIEV M., 2004, *A (ACRONYMS)*, Doctorial Thesis, Simon Fraser University, B.C. - Canada.

Anexo A - Relatório da extração do corpus de desenvolvimento

Acrônimo	Expansão Base Ouro	Expansão Extraída	Status
p4	progesterone	progesterone	CORRETO
p4	progesterone	progesterone	CORRETO
abi	applied bio.	applied bio.	CORRETO
nm23	metastasis suppression gene	metastasis suppression gene	CORRETO
copd	chronic obstructive pulmonary disease	chronic obstructive pulmonary disease	CORRETO
spr	seasonal pregnancy rate	seasonal pregnancy rate	CORRETO
fg	fibrinogen	fibrinogen	CORRETO
fh	family history	family history	CORRETO
opu	ovum pick-up	ovum pick-up	CORRETO
sod1	superoxide dismutase	superoxide dismutase	CORRETO
l/m	loaded/multigenerational	loaded/multigenerational	CORRETO
pgf2 alpha	prostaglandin f2 alpha	prostaglandin f2 alpha	CORRETO
gaa	alpha-glucosidase	alpha-glucosidase	CORRETO
ox-ldl	oxidative modification ldl	oxidative modification ldl	CORRETO
trp	transition relevance place	transition relevance place	CORRETO
cfda	carboxifluorescein diacetate	carboxifluorescein diacetate	CORRETO
pi	inorganic phosphate	phosphate	INCORRETO
pi	propidium iodide	propidium iodide	CORRETO
m-ii	metaphase ii	metaphase ii	CORRETO
dyt5	dopa-responsive dystonia	dopa-responsive dystonia	CORRETO
bme	beta-mercaptoethanol	beta-mercaptoethanol	CORRETO
det	transfer of two embryos	transfer of two embryos	CORRETO
ivp	in vitro produced	in vitro produced	CORRETO
ivm	in vitro maturation	in vitro maturation	CORRETO
htlv-1	human t-lymphotropic virus type 1	human t-lymphotropic virus type 1	CORRETO
tsh	thyrotropin	target tissue for thyrotropin	INCORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	Status
eb	estradiol benzoate	estradiol benzoate	CORRETO
cml	chronic myeloid leukemia	chronic myeloid leukemia	CORRETO
asma	automatic sperm morphometry analysis system	automatic sperm morphometry analysis system	CORRETO
adhd	attention deficit hyperactivity disorder	attention deficit hyperactivity disorder	CORRETO
dd	d-dimer fragments	d-dimer fragments	CORRETO
mb	morulae and blastocysts	morulae and blastocysts	CORRETO
rut	rapid urease test	rapid urease test	CORRETO
bfgf	basic fibroblast growth factor	basic fibroblast growth factor	CORRETO
bpspb	bovine pregnancy-specific protein b	bovine pregnancy-specific protein b	CORRETO
mm	miyoshi myopathy	miyoshi myopathy	CORRETO
lrp1	low-density lipoprotein receptor-related protein g	lipoprotein receptor-related protein gene	INCORRETO
temps-i	semi-structured affective temperament interview	temperament interview	INCORRETO
rea	restriction endonuclese analysis	restriction endonuclese analysis	CORRETO
zsr	zeta sedimentation ratio	zeta sedimentation ratio	CORRETO
t	testosterone	testosterone	CORRETO
pef	peak flow	peak flow	CORRETO
bcs	body condition score	body condition score	CORRETO
lincl	late-infantile neuronal ceroid lipofuscinosis	late-infantile neuronal ceroid lipofuscinosis	CORRETO
iets	international embryo transfer society	international embryo transfer society	CORRETO
et	endothelin	endothelin	CORRETO
ivf	in vitro fertilization	in vitro fertilization	CORRETO
nfd	nonfertile or degenerated ova	nonfertile or degenerated ova	CORRETO
insp3	inositol 1,4,5-triphosphate	injection of inositol 1,4,5-triphosphate	INCORRETO
allo-pbsct	allogeneic peripheral blood stem cell transplantat	allogeneic peripheral blood stem cell transplantat	CORRETO
bpag 1	bovine pregnancy-associated glycoprotein 1	bovine pregnancy-associated glycoprotein 1	CORRETO
epf	early pregnancy factor	early pregnancy factor	CORRETO
gvhd	graft versus host disease	graft versus host disease	CORRETO
cham	computer-elicited hyperarticulate adaptation model	computer-elicited hyperarticulate adaptation model	CORRETO
apoe	apolipoprotein e	apolipoprotein e	CORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	Status
h-f	holstein-friesian	holstein-friesian	CORRETO
afs	ameloblastic fibrosarcoma	ameloblastic fibrosarcoma	CORRETO
te	trophectoderm	trophectoderm	CORRETO
te	trophectoderm	trophectoderm	CORRETO
gsdii	glycogen storage disease type ii	glycogen storage disease type ii	CORRETO
egf	epidermal growth factor	epidermal growth factor	CORRETO
bp	bipolarity	bipolarity	CORRETO
fmdv	foot and mouth disease virus	foot and mouth disease virus	CORRETO
tgf-beta s	transforming growth factor-beta s	transforming growth factor-beta s	CORRETO
sst	scrotal surface temperature	scrotal surface temperature	CORRETO
bl	blastocysts	blastocysts	CORRETO
coc's	cumulus-oocyte-complexes	cumulus-oocyte-complexes	CORRETO
cocs	cumulus-oocyte complexes	cumulus-oocyte complexes	CORRETO
pfge	pulse field gel electrophoresis	pulse field gel electrophoresis	CORRETO
pka	protein kinase a	protein kinase a	CORRETO
pmdd	prepubertal major depressive disorder	prepubertal major depressive disorder	CORRETO
sc	scrotal size	size	INCORRETO
pcr	polymerase chain reaction	polymerase chain reaction	CORRETO
pkc	protein kinase c	protein kinase c	CORRETO
pkc	protein kinase c	protein kinase c	CORRETO
das	dialog acts	dialog acts	CORRETO
e2	estradiol-17 beta	estradiol-17 beta	CORRETO
e2	estradiol	estradiol	CORRETO
cp	capsular polysaccharide	capsular polysaccharide	CORRETO
bvdv	bovine viral diarrhea virus	bovine viral diarrhea virus	CORRETO
cm	compact morulae	compact morulae	CORRETO
cl	corpus luteum	corpus luteum	CORRETO
cl	corpora lutea	corpora lutea	CORRETO
cl	corpus luteum	corpus luteum	CORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	Status
osr	ontario school record	ontario school record	CORRETO
odd	oppositional defiant disorder	oppositional defiant disorder	CORRETO
raw	airway resistance	airway resistance	CORRETO
hp	hilicobacter pylori	hilicobacter pylori	CORRETO
tgf-beta	transforming growth factor-beta	transforming growth factor-beta	CORRETO
set	transfer of single embryo	single embryo	INCORRETO
asr	automatic speech recognition	automatic speech recognition	CORRETO
icm	inner cell mass	inner cell mass	CORRETO
tunel	terminal deoxynucleotidyl transferase-mediated deo	terminal deoxynucleotidyl transferase-mediated deo	CORRETO
pao2	partial pressure of oxygen in arterial blood	pressure of oxygen in arterial blood	INCORRETO
pmns	polymorphonuclear neutrophils	polymorphonuclear neutrophils	CORRETO
ig	immunoglobulins	immunoglobulins	CORRETO
tgf-alpha	transforming growth factor-alpha	transforming growth factor-alpha	CORRETO
als	amyotrophic lateral sclerosis	amyotrophic lateral sclerosis	CORRETO
vwf	von willebrand factor	von willebrand factor	CORRETO
efsm	modified efs	solutions were compared: efs, modified efs	INCORRETO
hza	hemizona assay	hemizona assay	CORRETO
rit	rosette inhibition test	rosette inhibition test	CORRETO
fwdf	first-wave dominant follicle	first-wave dominant follicle	CORRETO
pcna	proliferating cell nuclear antigen	proliferating cell nuclear antigen	CORRETO
fcm	flow cytometry	flow cytometry	CORRETO
scsa	sperm chromatin structure assay	sperm chromatin structure assay	CORRETO
cbcl	child behavior checklist	child behavior checklist	CORRETO
fcs	fetal calf serum	fetal calf serum	CORRETO
cidr	intravaginal progesterone device	intravaginal progesterone device	CORRETO
brms	bech-rafaelsen mania scale	bech-rafaelsen mania scale	CORRETO
fitc-psa	fluorescein-conjugated pisum sativum agglutinin	fluorescein-conjugated pisum sativum agglutinin	CORRETO
ncsu	north carolina state university	north carolina state university	CORRETO
ai	artificial insemination	artificial insemination	CORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	Status
gmp-140	granule membrane protein-140	granule membrane protein-140	CORRETO
ad	alzheimer disease	alzheimer disease	CORRETO
mmc	melphalan, meccnu and cyclophosphomide	melphalan, meccnu and cyclophosphomide	CORRETO
hzi	hemizona index	hemizona index	CORRETO
smc	smooth muscle cells	smooth muscle cells	CORRETO
af	ameloblastic fibroma	ameloblastic fibroma	CORRETO

Anexo B - Resultados da extração do corpus de teste

Acrônimo	Expansão Base Ouro	Expansão Extraída	STATUS
saa	serum amyloid a	serum amyloid a	CORRETO
lps	lipopolysaccharide	selectively impede lipopolysaccharide	INCORRETO
saf	sas-binding factor	sas-binding factor	CORRETO
ttf-1	thyroid transcription factor 1	thyroid transcription factor 1	CORRETO
mefs	embryo fibroblasts	embryo fibroblasts	CORRETO
hot1	hot-spot	hot-spot	CORRETO
gfap	glial fibrillary acidic protein	glial fibrillary acidic protein	CORRETO
dpc	days postcoitum	days postcoitum	CORRETO
fob1	fork blocking	fork blocking	CORRETO
are	au-rich elements	au-rich elements	CORRETO
arc	arcuate nucleus	arcuate nucleus	CORRETO
npy	neuropeptide y	neuropeptide y	CORRETO
npy	neuropeptide y	neuropeptide y	CORRETO
hprt	hypoxanthine phosphoribosyltransferase gene	hypoxanthine phosphoribosyltransferase gene	CORRETO
vhl	von hippel-lindau	von hippel-lindau	CORRETO
hpv	human papillomavirus	human papillomavirus	CORRETO
tfiib	transcription factor iib	transcription factor iib	CORRETO
gbm	gliobastoma multiforme	gliobastoma multiforme	CORRETO
fu	fused	fused	CORRETO
drbms	dsrna binding motifs	dsrna binding motifs	CORRETO
fn	fibronectin	fibronectin	CORRETO
mhc	alpha-myosin heavy chain	heavy chain	INCORRETO
3utrs	3 untranslated regions	3 untranslated regions	CORRETO
wg	wingless	wingless	CORRETO
dpp	decapentaplegic	decapentaplegic	CORRETO
ampars	ampa receptors	some synapses lack functional ampa receptors	INCORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	STATUS
dhpg	dihydroxyphenylglycine	dihydroxyphenylglycine	CORRETO
hd	head direction	head direction	CORRETO
dsrna	double-stranded rna	double-stranded rna	CORRETO
jnk	c-jun nh2-terminal kinase	nh2-terminal kinase	INCORRETO
pf	prefrontal	prefrontal	CORRETO
sas	saa-activating sequence	saa-activating sequence	CORRETO
ph	periventricular heterotopia	periventricular heterotopia	CORRETO
ph	partial hepatectomy	partial hepatectomy	CORRETO
gr	glucocorticoid receptor	glucocorticoid receptor	CORRETO
csr	class switch recombination	class switch recombination	CORRETO
lfv	lassa fever virus	lassa fever virus	CORRETO
der	drosophila epidermal growth factor receptor	drosophila epidermal growth factor receptor	CORRETO
mglurs	metabotropic glutamate receptors	metabotropic glutamate receptors	CORRETO
dv	dorsoventral	dorsoventral	CORRETO
c2glcnact	core 2 beta n acetylglucosaminyltransferase	commonly synthesized with the golgi enzyme core 2	INCORRETO
psd-95	postsynaptic density-95	postsynaptic density-95	CORRETO
vg	vestigial	vestigial	CORRETO
epac	exchange protein directly activated by camp	exchange protein directly activated by camp	CORRETO
nfat	nuclear factor of activated t cells	nuclear factor of activated t cells	CORRETO
nes	nuclear export signal	nuclear export signal	CORRETO
egfr	epidermal growth factor receptor	epidermal growth factor receptor	CORRETO
rfb	replication fork blocking	replication fork blocking	CORRETO
smmhc	smooth-muscle myosin heavy chain	smooth-muscle myosin heavy chain	CORRETO
ercs	extrachromosomal rdna circles	extrachromosomal rdna circles	CORRETO
hcrs	highly conserved regions	highly conserved regions	CORRETO
fmri	functional magnetic resonance imaging	functional magnetic resonance imaging	CORRETO
pin1	pin-formed	pin-formed	CORRETO
sags	superantigens	superantigens	CORRETO
rtks	receptor tyrosine kinases	receptor tyrosine kinases	CORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	STATUS
map	mitogen-activated protein	mitogen-activated protein	CORRETO
socs-1	suppressor of cytokine signaling-1	suppressor of cytokine signaling-1	CORRETO
nmda	n methyl d aspartate	n methyl d aspartate	CORRETO
sufu	suppressor of fused	suppressor of fused	CORRETO
mam	meprin, a5, mu	meprin, a5, mu	CORRETO
hmg-coa	hydroxy methylglutaryl coenzyme a	hydroxy methylglutaryl coenzyme a	CORRETO
lmn	lateral mammillary nucleus	lateral mammillary nucleus	CORRETO
cebp	ccaat enhancer binding protein	ccaat enhancer binding protein	CORRETO
apr	acute-phase response	acute-phase response	CORRETO
sdic	sperm-specific dynein intermediate chain	sperm-specific dynein intermediate chain	CORRETO
rmp	rpb5-mediating protein	rpb5-mediating protein	CORRETO
vitamine	alpha tocopherol	alpha tocopherol	CORRETO
creb	camp response element binding protein	camp response element binding protein	CORRETO
pnr	purine-rich negative regulatory	purine-rich negative regulatory	CORRETO
haart	highly active antiretroviral therapy	highly active antiretroviral therapy	CORRETO
lmb	leptomycin b	leptomycin b	CORRETO
stat	signal transducer and activator of transcription	signal transducer and activator of transcription	CORRETO
alpha-dg	alpha-dystroglycan	alpha-dystroglycan	CORRETO
erf1	ethylene response factor1	ethylene response factor1	CORRETO
tlr4	toll-like receptor-4 gene	toll-like receptor-4 gene	CORRETO
tcr	t cell antigen receptor	t cell antigen receptor	CORRETO
tcr	t cell receptor	t cell receptor	CORRETO
tcr	t cell receptor	t cell receptor	CORRETO
tcr	t cell receptor	t cell receptor	CORRETO
tcr	t cell antigen receptor	t cell antigen receptor	CORRETO
hp-1	heterochromatin protein-1	heterochromatin protein-1	CORRETO
eya	eyes absent	eyes absent	CORRETO
afp	alpha-fetoprotein	alpha-fetoprotein	CORRETO
immuno-fish	immunofluorescence in situ hybridization	immunofluorescence in situ hybridization	CORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	STATUS
dcr3	decoy receptor 3	decoy receptor 3	CORRETO
ltp	long-term potentiation	long-term potentiation	CORRETO
ltp	long-term potentiation	long-term potentiation	CORRETO
dsbs	double-strand breaks	double-strand breaks	CORRETO
tk	thymidine kinase	thymidine kinase	CORRETO
bfnc	benign familial neonatal convulsions	benign familial neonatal convulsions	CORRETO
th	cd4 helper t	helper t	INCORRETO
tif2alpha	translation initiation factor 2alpha	translation initiation factor 2alpha	CORRETO
rpb5	rna polymerase ii subunit 5	rna polymerase ii subunit 5	CORRETO
ntn	neurturin	neurturin	CORRETO
gaba	gamma-aminobutyric acid	gamma-aminobutyric acid	CORRETO
mcr	melanocortin receptor	melanocortin receptor	CORRETO
3utr	3' untranslated region	3' untranslated region	CORRETO
gscs	germ-line stem cells	germ-line stem cells	CORRETO
p-sp	para-aortic splanchnopleural mesoderm	para-aortic splanchnopleural mesoderm	CORRETO
o-glycans	serine/threonine-linked oligosaccharides	oligosaccharides	INCORRETO
mch	melanin-concentrating hormone	melanin-concentrating hormone	CORRETO
pol i	rna polymerase i	of rna polymerase i	INCORRETO
psp	persephin	persephin	CORRETO
htl	heartless	heartless	CORRETO
dc	dendritic cell	dendritic cell	CORRETO
dc	dendritic cell	dendritic cell	CORRETO
cdic	cytoplasmic dynein intermediate chain	cytoplasmic dynein intermediate chain	CORRETO
pka	protein kinase a	protein kinase a	CORRETO
dmd	differentially methylated domain	differentially methylated domain	CORRETO
so	sine oculis	sine oculis	CORRETO
sd	scalloped	scalloped	CORRETO
hbx	hepatitis b virus x protein	hepatitis b virus x protein	CORRETO
cart	cocaine- and amphetamine-regulated transcript	cocaine- and amphetamine-regulated transcript	CORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	STATUS
camp-gefs	camp-regulated gefs	camp-regulated gefs	CORRETO
cbes	candidate boundary elements	candidate boundary elements	CORRETO
fasl	fas ligand	fas ligand	CORRETO
fasl	fas ligand	fas ligand	CORRETO
gmcsf	granulocyte macrophage colony stimulating factor g	granulocyte macrophage colony stimulating factor g	CORRETO
il-15ralpha	il-15 receptor alpha subunit	il-15 receptor alpha subunit	CORRETO
skips	syk kinases-interacting proteins	screen to identify syk kinases-interacting protein	INCORRETO
ci	cubitus interruptus	cubitus interruptus	CORRETO
gef	guanine-nucleotide-exchange factor	guanine-nucleotide-exchange factor	CORRETO
gef	guanine nucleotide exchange factor	guanine nucleotide exchange factor	CORRETO
ltd	long-term depression	long-term depression	CORRETO
ltd	long-term depression	long-term depression	CORRETO
scn	suprachiasmatic nucleus	suprachiasmatic nucleus	CORRETO
gdnf	glial cell line-derived neurotrophic factor	glial cell line-derived neurotrophic factor	CORRETO
rca	retrochiasmatic area	retrochiasmatic area	CORRETO
chr	chromosome	chromosome	CORRETO
hsf	heat shock transcription factor	heat shock transcription factor	CORRETO
tgf-beta	transforming growth factor-beta	transforming growth factor-beta	CORRETO
hh	hedgehog	hedgehog	CORRETO
rb	retinoblastoma	retinoblastoma	CORRETO
slfn	schlafen	schlafen	CORRETO
atn	anterior thalamus	anterior thalamus	CORRETO
fgf4	fibroblast growth factor 4	fibroblast growth factor 4	CORRETO
gro	groucho	groucho	CORRETO
cbfbeta	core binding factor beta	core binding factor beta	CORRETO
lcmv	lymphocytic choriomeningitis virus	lymphocytic choriomeningitis virus	CORRETO
clip	class ii-associated invariant chain peptide	chain peptide	INCORRETO
cns1	cyclophilin seven suppressor	cyclophilin seven suppressor	CORRETO
epscs	excitatory postsynaptic currents	excitatory postsynaptic currents	CORRETO

Acrônimo	Expansão Base Ouro	Expansão Extraída	STATUS
mapk	mitogen-associated protein kinase	mitogen-associated protein kinase	CORRETO
mapk	mitogen-activated protein kinase	mitogen-activated protein kinase	CORRETO
dlgn	dorsal lateral geniculate nucleus	dorsal lateral geniculate nucleus	CORRETO
annx	annexin x	annexin x	CORRETO
fln1	filamin 1	filamin 1	CORRETO
pab1p	poly a binding protein	poly a binding protein	CORRETO
seb	staphylococcal enterotoxin b	staphylococcal enterotoxin b	CORRETO
seb	staphylococcal enterotoxin b	staphylococcal enterotoxin b	CORRETO
pbp1	pab1p-binding protein	pab1p-binding protein	CORRETO
prp	prion protein	prion protein	CORRETO
dsf	dissatisfaction	dissatisfaction	CORRETO
pomc	proopiomelanocortin	proopiomelanocortin	CORRETO

Anexo C - Exemplos de extração de acrônimos de texto não estruturado

A Figura C.1 contém uma amostra de texto não estruturado extraída do corpus de desenvolvimento. Esta amostra consiste na entrada do processo de resolução de acrônimos descrito na Figura 4.2.

Lack of association between apolipoprotein E genotype and sporadic amyotrophic lateral sclerosis [In Process Citation]

T. Siddique, M. A. Pericak-Vance, J. Caliendo, S. T. Hong, W. Y. Hung, J. Kaplan, D. McKenna-Yasek, J. B. Rimmler, P. Sapp, A. M. Saunders, W. K. Scott, N. Siddique, J. L. Haines and R. H. Brown

Neurogenetics

1

213-6

1998

Amyotrophic lateral sclerosis (ALS) is a neuro-degenerative disorder with both sporadic and familial forms. Approximately 20% of autosomal dominant ALS is caused by mutations in the Cu/Zn superoxide dismutase (SOD1) gene. The causes of the remaining forms of ALS are unknown. The apolipoprotein E (APOE) gene is a known genetic risk factor for Alzheimer disease (AD) , another neuro-degenerative disease. The APOE-4 allele increases risk and decreases age at onset in AD. Studies examining ALS and APOE have failed to show a significant effect of APOE on overall risk in ALS. Studies examining the effect of APOE-4 on site of onset in ALS (bulbar or limb) have been contradictory, with some studies showing an APOE association with bulbar onset and others showing no effect. Sample size was limited in these previous reports, particularly with respect to the number of bulbar onset cases (n = 33, 34 and 53). The present study examines a large collaborative data set of ALS patients (n = 363; 95 with bulbar onset) and age-matched neurologically normal controls. The results for these data showed no significant differences in the percentage of subjects with the APOE-4/4 and APOE-4/X genotypes (X = APOE-2 or APOE-3) when comparing cases and controls in both the overall data set or in the data set stratified by site of onset. Similarly, logistic regression analysis in the overall and stratified data set while controlling for sex showed no increase or decrease in risk of ALS associated with the APOE-4 allele. In addition, there were no significant differences in age at onset between patients with APOE-X/X, and APOE-4/4 or APOE-4/X genotypes, overall or stratified by site of onset. We conclude based on these data that the APOE gene is not a major genetic risk factor for site of onset in ALS.

Figura C.1 - **Exemplo de texto não estruturado com acrônimos para extração**

A Figura C.2 mostra o texto após ter sido processada a atividade de "Identificação de Acrônimos Candidatos". A lista resultante deste processamento contém os seguintes elementos: ALS, SOD1, APOE, AD, ALS, cases, patients e genotype. ALS aparece duas vezes, sendo que na primeira identificação o acrônimo candidato está alinhado com o padrão (4.1) da Tabela 4.1 e na segunda identificação está alinhado com o padrão (4.2).

Na Figura C.3 mostra o mesmo texto com os conteúdos das janelas direita e esquerda identificados para cada acrônimo. O número de palavras em cada janela é calculado com a fórmula (4.5). No caso dos acrônimos candidatos identificados com o padrão (4.2) da Tabela 4.1 somente a janela à direita.

Para o primeiro ALS a janela à esquerda é composta pela expressão "Amyotrophic lateral sclerosis" enquanto a outra contém a expressão "is a neuro-degenerative disorder with both".

O segundo ALS tem somente uma janela e seu conteúdo é "bulbar or limb".

Lack of association between apolipoprotein E genotype and sporadic amyotrophic lateral sclerosis [In Process Citation]

T. Siddique, M. A. Pericak-Vance, J. Caliendo, S. T. Hong, W. Y. Hung, J. Kaplan, D. McKenna-Yasek, J. B. Rimmler, P. Sapp, A. M. Saunders, W. K. Scott, N. Siddique, J. L. Haines and R. H. Brown

Neurogenetics

1

213-6

1998

Amyotrophic lateral sclerosis (ALS) is a neuro-degenerative disorder with both sporadic and familial forms. Approximately 20% of autosomal dominant ALS is caused by mutations in the Cu/Zn superoxide dismutase (SOD1) gene. The causes of the remaining forms of ALS are unknown. The apolipoprotein E (APOE) gene is a known genetic risk factor for Alzheimer disease (AD) , another neuro-degenerative disease. The APOE-4 allele increases risk and decreases age at onset in AD. Studies examining ALS and APOE have failed to show a significant effect of APOE on overall risk in ALS. Studies examining the effect of APOE-4 on site of onset in ALS (bulbar or limb) have been contradictory, with some studies showing an APOE association with bulbar onset and others showing no effect. Sample size was limited in these previous reports, particularly with respect to the number of bulbar onset cases (n = 33, 34 and 53). The present study examines a large collaborative data set of ALS patients (n = 363; 95 with bulbar onset) and age-matched neurologically normal controls. The results for these data showed no significant differences in the percentage of subjects with the APOE-4/4 and APOE-4/X genotypes (X = APOE-2 or APOE-3) when comparing cases and controls in both the overall data set or in the data set stratified by site of onset. Similarly, logistic regression analysis in the overall and stratified data set while controlling for sex showed no increase or decrease in risk of ALS associated with the APOE-4 allele. In addition, there were no significant differences in age at onset between patients with APOE-X/X, and APOE-4/4 or APOE-4/X genotypes, overall or stratified by site of onset. We conclude based on these data that the APOE gene is not a major genetic risk factor for site of onset in ALS.

Figura C.2 - **Exemplo de texto com os acrônimos candidatos identificados**

Lack of association between apolipoprotein E genotype and sporadic amyotrophic lateral sclerosis [In Process Citation]

T. Siddique, M. A. Pericak-Vance, J. Caliendo, S. T. Hong, W. Y. Hung, J. Kaplan, D. McKenna-Yasek, J. B. Rimmler, P. Sapp, A. M. Saunders, W. K. Scott, N. Siddique, J. L. Haines and R. H. Brown

Neurogenetics

1

213-6

1998

Amyotrophic lateral sclerosis (ALS) is a neuro-degenerative disorder with both sporadic and familial forms. Approximately 20% of autosomal dominant ALS is caused by mutations in the Cu/Zn superoxide dismutase (SOD1) gene. The causes of the remaining forms of ALS are unknown. The apolipoprotein E (APOE) gene is a known genetic risk factor for Alzheimer disease (AD) , another neuro-degenerative disease. The APOE-4 allele increases risk and decreases age at onset in AD. Studies examining ALS and APOE have failed to show a significant effect of APOE on overall risk in ALS. Studies examining the effect of APOE-4 on site of onset in ALS (bulbar or limb) have been contradictory, with some studies showing an APOE association with bulbar onset and others showing no effect. Sample size was limited in these previous reports, particularly with respect to the number of bulbar onset cases (n = 33, 34 and 53). The present study examines a large collaborative data set of ALS patients (n = 363; 95 with bulbar onset) and age-matched neurologically normal controls. The results for these data showed no significant differences in the percentage of subjects with the APOE-4/4 and APOE-4/X genotypes (X = APOE-2 or APOE-3) when comparing cases and controls in both the overall data set or in the data set stratified by site of onset. Similarly, logistic regression analysis in the overall and stratified data set while controlling for sex showed no increase or decrease in risk of ALS associated with the APOE-4 allele. In addition, there were no significant differences in age at onset between patients with APOE-X/X, and APOE-4/4 or APOE-4/X genotypes, overall or stratified by site of onset. We conclude based on these data that the APOE gene is not a major genetic risk factor for site of onset in ALS.

Figura C.3 - **Exemplo com a identificação das expansões candidatas identificadas**

A atividade "Resolver Tupla <Acrônimo, Expansão>" da Figura 4.2 é a responsável pela identificação dos termos que são acrônimos e seus respectivos significados. No exemplo deste anexo as expressões possíveis para o segundo ALS e as palavras "cases", "patients" e "genotype" obtiveram probabilidade nula de emergirem dos termos. O sistema, conseqüentemente, não os reconheceu como acrônimos passíveis de serem extraídos. O resultado do processamento desta atividade está exemplificado na Figura C.4.

No Anexo A está o relatório do processamento que extraiu todos os termos do corpus de desenvolvimento reconhecidos como acrônimos, incluindo os deste exemplo.

Lack of association between apolipoprotein E genotype and sporadic amyotrophic lateral sclerosis [In Process Citation]

T. Siddique, M. A. Pericak-Vance, J. Caliendo, S. T. Hong, W. Y. Hung, J. Kaplan, D. McKenna-Yasek, J. B. Rimmler, P. Sapp, A. M. Saunders, W. K. Scott, N. Siddique, J. L. Haines and R. H. Brown

Neurogenetics

1

213-6

1998

Amyotrophic lateral sclerosis (ALS) is a neuro-degenerative disorder with both sporadic and familial forms. Approximately 20% of autosomal dominant ALS is caused by mutations in the Cu/Zn superoxide dismutase (SOD1) gene. The causes of the remaining forms of ALS are unknown. The apolipoprotein E (APOE) gene is a known genetic risk factor for Alzheimer disease (AD) , another neuro-degenerative disease. The APOE-4 allele increases risk and decreases age at onset in AD. Studies examining ALS and APOE have failed to show a significant effect of APOE on overall risk in ALS. Studies examining the effect of APOE-4 on site of onset in ALS (bulbar or limb) have been contradictory, with some studies showing an APOE association with bulbar onset and others showing no effect. Sample size was limited in these previous reports, particularly with respect to the number of bulbar onset cases (n = 33, 34 and 53). The present study examines a large collaborative data set of ALS patients (n = 363; 95 with bulbar onset) and age-matched neurologically normal controls. The results for these data showed no significant differences in the percentage of subjects with the APOE-4/4 and APOE-4/X genotypes (X = APOE-2 or APOE-3) when comparing cases and controls in both the overall data set or in the data set stratified by site of onset. Similarly, logistic regression analysis in the overall and stratified data set while controlling for sex showed no increase or decrease in risk of ALS associated with the APOE-4 allele. In addition, there were no significant differences in age at onset between patients with APOE-X/X, and APOE-4/4 or APOE-4/X genotypes, overall or stratified by site of onset. We conclude based on these data that the APOE gene is not a major genetic risk factor for site of onset in ALS.

Figura C.4 **- Resultado da extração de acrônimo em texto não estruturado**