UNIVERSIDADE FEDERAL DO MARANHÃO
CENTRO DE CIÊNCIAS EXATAS E TECNOLÓGICAS
CURSO DE CIÊNCIA DA COMPUTAÇÃO

ROBINSON SANTOS CASTRO

# **EXTRAÇÃO DE INFORMAÇÃO**: conceitos, plataformas e sistemas

São Luís
2013

ROBINSON SANTOS CASTRO

# **EXTRAÇÃO DE INFORMAÇÃO**: conceitos, plataformas e sistemas

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Orientador: Prof. Ms. Ivo Serra.

São Luís

2013

Castro, Robinson Santos.

Extração de informação: conceitos, plataformas e sistemas / Robinson Santos Castro. – 2013.

53 f.

Impresso por computador (Fotocópia).

Orientador: Ivo Serra.

Monografia (graduação) – Universidade Federal do Maranhão, Curso de Ciência da Computação, 2013.

1. Extração de informação. 2. Arquitetura de sistemas. 3. Processamento textual. 4. Busca de informação.
I. Título. CDU 004.775

ROBINSON SANTOS CASTRO

# **EXTRAÇÃO DE INFORMAÇÃO**: conceitos, plataformas e sistemas

Monografia apresentada ao Curso de Ciência da Computação da Universidade Federal do Maranhão, como parte dos requisitos necessários para obtenção do grau de Bacharel em Ciência da Computação.

Aprovado em 11 de 12 de 2013

BANCA EXAMINADORA

Ivo José da Cunha Serra (Orientador)
Mestre em Informática
Universidade Federal do Maranhão

Carla Gomes de Faria Alves
Doutora em Informática
Instituto Federal de Educação, Ciência e Tecnologia do Maranhão

Carlos Eduardo Portela Serra de Castro
Mestre em Informática
Universidade Federal do Maranhão

À minha família

Aos meus amigos.

## RESUMO

A Extração de Informação é uma área de pesquisa cujo objeto é identificar informações relevantes a partir de textos, além de estruturar e armazenar essas informações, a fim de propiciar uma futura descoberta de relacionamentos interessantes entre as informações extraídas. Identifica-se ainda que a Extração de Informação pode ser usada como base para o desenvolvimento de sistemas mais robustos, como o povoamento de ontologias. Este trabalho descreve os principais conceitos da Extração de Informação como as tarefas, tipos de fontes, as abordagens para construção de sistemas que utilizam essa tecnologia, as arquiteturas baseadas em processamento da linguagem natural e *wrappers*, os obstáculos existentes e por fim, descreve sistemas que utilizam a Extração de Informação.



Palavras - chaves: Extração de informação, Processamento textual, Sistemas de Extração de Informação.

**ABSTRACT**

The Information Extraction is a research area whose purpose is to identify relevant information form texts, as well as organize and store this information in order to provide a future discovery of interesting relationships between the extracted information. Still identifies that the Information Extraction can be uses as a basis for the development of more robust systems like populating ontologies. This paper describes the key concepts such as Information Extraction tasks, types of sources, approaches to building systems that utilize this technology, architectures based on natural language processing and wrappers, existing barriers and finally, describes systems that use the Information Extraction.

Key-words: Information Extraction, text processing, Information Extraction systems.

## LISTA DE FIGURAS

## LISTA DE TABELAS

# ABREVIATURAS

| | |
|---|---|
| **ANNIE** | A Nearly-New Information Extraction System |
| **CREOLE** | Collection of Reusable Objects for Language Engineering |
| **DIAOP-Tool** | Tool for Automatic Ontology Population |
| **EI** | Extração de Informação |
| **GATE** | General Architecture for Text Engineering |
| **JAPE** | Java Annotation Patterns Engine |
| **LR** | Language Resources |
| **NER** | Named Entity Recognition |
| **PLN** | Processamento da Linguagem Natural |
| **POS** | Part of Speech |
| **REN** | Reconhecimento de Entidades Nomeadas |

# Sumário

# 1 INTRODUÇÃO

A busca por informações é um processo constante. A maior parte dessas informações é apresentada em forma de texto e a necessidade de se obter apenas a informação desejada dentre coleções de documentos é notória. Para o auxilio da atividade de busca de informação em textos utiliza-se buscadores automáticos. Porém esses buscadores automáticos, a partir de uma consulta, apenas recuperam um subconjunto de documentos a partir de uma coleção de documentos de diferentes domínios sem dar qualquer tipo de detalhes acerca do que está contido nos parágrafos desses textos. As informações recuperadas pelos buscadores são em sua maioria irrelevantes ao tema de interesse do usuário. Isso ocorre porque coexistem informações relevantes e uma considerável quantidade de informações irrelevantes.

Diante da imensa quantidade de informação disponível em formato textual, os seres humanos não são capazes de assimilar (ler) toda essa informação. Como solução para a extração automática do conteúdo dos documentos outro tipo de tecnologia é empregado: Extração de Informação (APPELT; ISRAEL, 1999). A Extração de Informação (EI) tem como objetivo localizar, estruturar e armazenar a informação relevante de um documento ou de um conjunto de documentos, a fim de propiciar uma futura descoberta de relacionamentos interessantes entre as informações extraídas.

A Extração de Informação serve ainda como base para construção de sistemas mais robustos. Por exemplo, as informações extraídas dos documentos selecionados poderão ser posteriormente utilizadas para povoar uma ontologia de domínio. Por sua vez, uma ontologia povoada poderá ser utilizada para auxiliar o usuário na tomada de decisões.

Um leitor adquire conhecimento a partir de um texto, facilmente e naturalmente, identificando as informações relevantes e memorizando-as. Porém, automatizar esse processo é algo complexo e depende do processamento da linguagem natural pelas maquinas. A busca do conhecimento a partir de bases textuais exige o entendimento dos dados contidos nos documentos, transformando esses dados em informação de forma automática (ZAMBENEDETTI, 2002).

Diante disto, é de grande utilidade para o usuário ter ao seu dispor sistemas capazes de extrair a informação relevante dos documentos e apresentar ao usuário de forma clara e objetiva. Os sistemas de Extração de Informação (EI), tratam o problema de analisar os documentos textuais selecionados em busca de dados relevantes para o usuário (COWIE, 1996). O objetivo geral de um sistema de Extração de Informação não é interpretar o documento inteiro, mas apenas extrair suas partes relevantes, armazenado-as de forma estruturada, geralmente em um Banco de Dados (ELMASRI, 1994).

A falta de informação estruturada nos documentos textuais dificulta o processamento desses documentos. Por exemplo, ao processar um documento, às vezes se faz necessário responder perguntas como: qual o título? Qual o autor? Onde foi publicado? Quais documentos são referenciados no documento? A Extração de Informação busca respostas automáticas para essas perguntas. Assim, além de dados estatísticos relativos às palavras que contém, o documento passa a conter também dados que descrevem a sua semântica, evitando que o usuário seja obrigado a ler coleções inteiras de documentos para encontrar determinada informação (GONÇALVES, 2010).

Para construção de sistemas onde a informação é extraída de forma automática, existem estudos de técnicas de processamento da linguagem natural aplicadas sobre corpus de fontes textuais. O objetivo do uso dessas técnicas no contexto de Extração de Informação é tentar compreender textos em alguma língua natural, a fim de encontrar informações relevantes a serem extraídas.

## 1.1 Objetivos

Tomando como base o cenário descrito acima, este trabalho de graduação tem como objetivo realizar uma apresentação geral sobre o tema Extração de Informação, mostrando conceitos, descrevendo arquiteturas, e sistemas de EI.

Os objetivos específicos deste trabalho são:

a) Apresentar os principais conceitos relacionados à Extração de Informação;

b) Apresentar as principais arquiteturas de sistemas de EI;

c) Discutir obstáculos ao desenvolvimento de um sistema de EI;

d) Apresentar plataformas disponíveis atualmente para desenvolvimento e execução de aplicações para Extração de Informação;

e) Apresentar sistemas de Extração de Informação.

## 1.2 Estrutura da monografia

Este trabalho está organizado da seguinte forma: o Capítulo 2 apresenta os principais conceitos relacionados à Extração de Informação, descreve-se as fontes textuais sobre as quais técnicas de EI serão aplicadas para obtenção de dados relevantes, mostra-se os tipos de sistemas de EI existentes, as abordagens para construção de sistemas, métricas de avaliação de sistemas e por fim os desafios enfrentados pelos desenvolvedores. O Capítulo 3 apresenta tecnologias disponíveis para o processamento da linguagem natural que auxiliam a construção de sistemas de Extração de Informação. O Capítulo 4 traz uma descrição de trabalhos estudados, apresentando também a metodologia dos sistemas e resultados obtidos. O Capítulo 5 conclui este documento com as considerações finais sobre a relativamente recente área de pesquisa EI.

# 2 FUNDAMENTOS DA EXTRAÇÃO DE INFORMAÇÃO

A Extração de Informação (APPELT; ISRAEL, 1999) é uma subárea da Ciência da Computação que se propõe a extrair automaticamente informações de fontes de informações estruturadas, semi-estruturadas ou não estruturadas. Os sistemas de Extração de Informação não têm o objetivo de compreender textos processados, e sim, realizar a extração de dados relevantes a partir de documentos textuais. A EI oferece um potencial de identificação e classificação de entidades e eventos contidos em um texto. Ela possibilita transformar um amplo material de informação bruta em informação refinada (estruturada). A Extração de Informação é vista como um processo que identifica, classifica e estrutura informações específicas, encontradas em fontes textuais. Esta estruturação dos dados permite uma maior facilidade para as tarefas de processamento de informação (MOENS, 2006).

Uma grande quantidade de fontes textuais torna a tarefa de recolhimento dos dados presentes nelas, trabalhosa ou impraticável para os humanos, pois não se tem como processar a grande quantidade de informação presente nas fontes textuais. A Extração de Informação vem facilitar a extração desses dados por meio de técnicas computacionais. Por exemplo, usando técnicas de EI os dados existentes em um documento de linguagem escrita poderiam ser utilizados para montar um banco de dados de maneira automatizada.

Extração de Informação não deve ser confundida com a área de Recuperação de Informação, a qual seleciona, de uma grande coleção, um subconjunto de documentos relevantes baseados em uma consulta do usuário. O objetivo da Recuperação de Informação é de recuperar documentos relevantes de uma coleção, enquanto Extração de Informação é de extrair informações relevantes dos documentos. Um exemplo de entrada e saída de um típico sistema de EI é apresentado na figura 2.1.

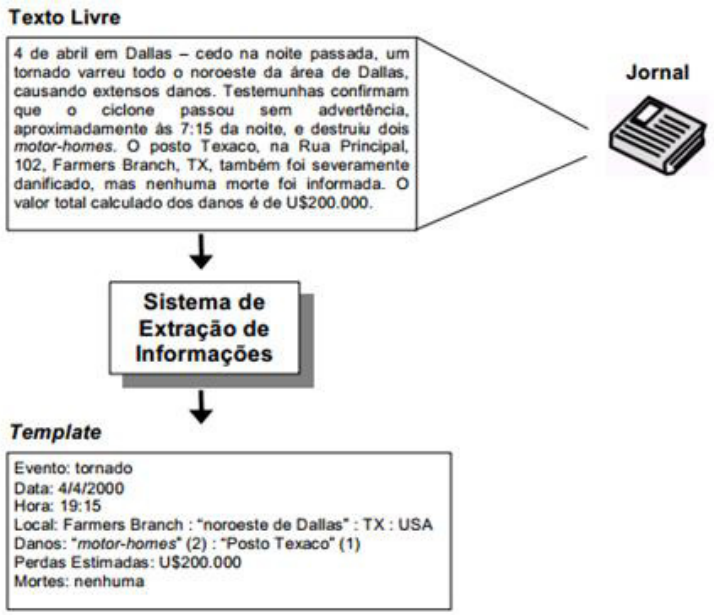


Figura 2.1: exemplo de entrada e saída de um sistema de EI (ZAMBENEDETTI, 2002)

Da forma ilustrada na figura 2.1 observa-se a estruturação dos dados obtidos de uma fonte textual. Essa estruturação permite a interpretação computacional dos dados extraídos para alimentar um banco de dados ou realizar inferências lógicas, por exemplo.

As pesquisas da área de EI tiveram o seu desenvolvimento no inicio dos anos 80 e foram alavancadas a partir da criação das conferencias MUC (Message Understanding Conference) (GRISHMAN, 1995). Através dessas conferências foi formalizada a tarefa de extração de informação e foram definidas as métricas para a avaliação do desempenho dos sistemas de EI. Embora chamada de "conferência", a característica que distingue as MUCs não são as conferencias propriamente ditas, mas as avaliações às quais os sistemas dos participantes são submetidos (GRISHMAN, 1995). As MUCs eram competições que comparavam a efetividade das abordagens alternativas para um problema particular. Isto é, uma comparação entre sistemas. Foram realizadas sete MUCs, a primeira, a MUC-1, foi realizada em 1987, e a ultima, a MUC-7, em 1998.

Segundo Silva (SILVA, 2004), existem dois tipos de sistema de Extração de Informação: os que utilizam Processamento da Linguagem Natural (PLN) e os sistemas *wrappers* (programas extratores). Sistemas baseados em PLN são desenvolvidos para tratar textos livres, sem nenhuma ou pouca estruturação, ele

realizam um pré-processamento linguístico para extração dos dados. Os *wrappers* são mais utilizados para tratar textos estruturados, como documentos WEB, onde as regras do sistema são baseadas em informações da estrutura do texto, como formatação, delimitadores, freqüência estatística das palavras. A tabela 1 resume as principais características que diferenciam os sistemas de EI baseados em PLN e os sistemas *wrappers*:

| | **Sistemas *Wrappers*** | **Sistemas de EI baseados em PLN** |
|---|---|---|
| **Motivação** | Principalmente, extrair informações das diversas fontes na Web. | Extrair informações de textos em linguagem natural. |
| **Tipos de texto** | Geralmente estruturados e semi-estruturados, mas também textos livres, em alguns casos. | Apenas texto livre. |
| **Recursos usados para extração** | Informações de formatação do texto, marcadores presentes nos documentos, freqüência estatística das palavras e, em alguns casos, PLN. | Padrões lingüísticos baseados em PLN (uso intenso de PLN) |

Tabela 2.1: Quadro comparativo Sistemas *Wrappers* versus Sistemas de EI baseado em PLN

As aplicações de sistemas de extração de informação são vastas. Por exemplo, uma companhia que a deseja filtrar sugestões dos e-mails enviados pelos seus clientes através do SAC. Ou uma empresa que almeja construir um sistema de recomendação tendo como base o histórico de navegação de seus usuários. Algum órgão do governo poderia levantar um relatório de índices de criminalidade por bairros e períodos através de notícias na página policial do jornal. Pesquisas biométricas podem realizar uma investigação de medicamento versus doença analisando os laudos médicos. Enfim, as aplicações são ilimitadas e garantem uma poderosa ferramenta na manipulação da informação.

Entretanto, a tarefa de extrair informações de documentos textuais encontra muitas dificuldades devido à complexidade da linguagem natural. Por exemplo, algumas palavras possuem mais de uma semântica e a ambigüidade de sentenças impossibilita a total compreensão de textos de linguagem natural. Detalhes sobre os desafios da extração de informação serão apresentados na seção 2.7 deste capitulo.

## 2.1 Tarefas da Extração de Informação

A Extração de Informação objetiva reconhecer, rotular e extrair de elementos de informação das fontes textuais. Segundo Hamish (HAMISH, 2005), em 1998, na ultima MUC (*Message Understanding Conferences*), foi estabelecido tarefas típicas para um sistema de Extração de Informação. Essas cinco tarefas são:

- **Reconhecimento de Entidades Nomeadas (NER):** o reconhecimento de Entidades Nomeadas (*Named Entity Recognition – NER*) consiste no reconhecimento de nomes que se referem a objetos exclusivos do mundo (CIMIANO, 2006). O serviço de NER identifica objetos pertencentes a classes como nomes de pessoas, locais, data, números, nomes de empresas. A figura 2.2 mostra exemplo de NER:

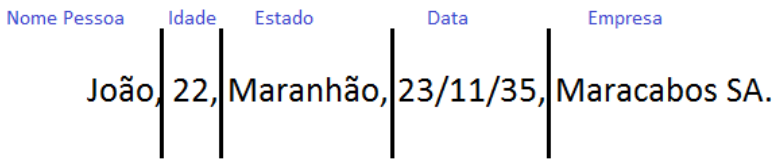


Figura 2.2: exemplos de NER

- **Resolução de co-referência (RC):** esta tarefa envolve a identificação de relações de identidade entre entidades em textos. Trata o problema de identificar quando uma nova frase nominal (normalmente, um pronome) se refere a outra já citada anteriormente. Por exemplo, na frase "***Maria*** *trabalhada na UFMA.* ***Ela*** *é professora do curso de Ciência da Computação*". A resolução de co-referência iria relacionar 'Maria' com 'Ela'.

  Este processo é menos relevante para os usuários do que outras tarefas de IE (ou seja, ao passo que as outras tarefas de EI produzem uma saída que é de utilidade óbvia para o usuário do sistema, essa tarefa é mais relevante para as necessidades do desenvolvedor do sistema). A principal importância desta tarefa, no entanto, é servir como um bloco de construção para as tarefas de TE e TC.

- **Construção de Template de Elemento (TE):** a tarefa de TE se baseia na NER e na resolução de co-referencia, associando informação descritiva com as entidades. Tal como a *Named Entity Recognition,* esta tarefa é fracamente dependente do domínio.
- **Construção de Template de Relação (TR):** a tarefa da Construção de Template de Relação exige a identificação de possíveis relações entre os templetes de elementos identificadas na TE. Este pode ser, por exemplo, uma relação entre um empregado e uma empresa, uma relação familiar entre duas pessoas, ou uma relação subsidiária entre duas empresas. No caso da MUC-6, a tarefa era reconhecer relações específicas entre pessoas que deixavam/assumiam cargos de gerência em empresas (KAUFMANN, 1995).
- **Construção de Template de Cenário (TC):** os templetes de Cenário visam unir as entidades extraídas na TE e as relações da TR para descrições de eventos. Esta é uma tarefa dependente do domínio.

Para exemplificar as tarefas de Extração de Informação, considere a seguinte sentença:

"*O livro de capa vermelha foi lançado na terça-feira. Ele foi escrito por Sr. João Silva. O Sr. João faz parte da equipe de jornalistas que trabalham no Jornal do Maranhão."*

**NER** descobre que as entidades presentes que são o livro, terça-feira, Sr. João e Jornal do Maranhão. **RC** descobre que 'ele' se refere ao 'livro'. **TE** descobre que o livro é vermelho e que foi idealizado por Sr. João. **TR** descobre que Sr. João trabalha para o Jornal do Maranhão. **TC** descobre que havia um evento de lançamento de livros em que as várias entidades foram envolvidas.

## 2.2 Fontes de informação

Técnicas de Extração de Informação são aplicadas em fontes textuais. O tipo de texto deve ser observado para a aplicação da estratégia mais apropriada. A homogeneidade do formato da fonte influência diretamente na eficácia da extração. As fontes de informações textuais são dividas em três categorias (SILVA, 2004):

**Fontes estruturadas**: possuem marcadores bem definidos e apresentam uma organização regular e previsível em sua estrutura. Por exemplo: um formulário preenchido.

**Fontes semi-estruturadas**: são documentos heterogêneos que mesclam alguma estrutura com texto livre. Os escopos bem definidos dessas fontes são explorados pelos extratores de informação para extração dos dados. Por exemplo: documentos HTML, anúncios de jornal, referências bibliográficas. A figura 2.3 apresenta um exemplo de anuncio onde o texto é semi-estruturado.

| |
|---|
| Aluga-se apartamento, 2 qt. 1 gar. elevador, 700 R$ |
| **MA** rua matos carvalho, 240, olho d'água. |

Figura 2.3: exemplo de texto semi-estruturado

**Fontes não-estruturadas**: são textos escritos em linguagem natural. Estão codificados sem nenhuma organização, o que dificulta sua interpretação. Por exempo: textos sem formatação, textos livres escritos português.

A maior limitação de EI geralmente está relacionada às fontes não-estruturadas, onde a máquina não tem nenhum apoio de marcação e delimitação da estrutura do texto, diferentemente das fontes estruturadas e semi-estruturadas. No caso dos documentos HTML, uma frase marcada com alguma tag *<head>*, indica maior relevância do que outra frase qualquer. Outro exemplo seria uma tabela que é identificada com marcadores *<table>, </table>*, facilitando todo o processo de análise e classificação dos dados dispostos dentro de um documento web.

A EI enfrenta problemas também nas fontes semi-estruturadas, como as referências bibliográficas de documentos. Uma referência bibliográfica pode ser definida como uma série arbitrária de campos em que cada transição entre eles ocorre após um separador (delimitador) específico. O problema que a Extração de Informação terá que lidar quando trabalha com referencias bibliográficas é o fato de que os delimitadores dos campos podem variar, assim como a ordem e que os campos aparecem, o formato das referências varia de documento para documento.

No capitulo 4 será apresentado o sistema BREx que trata esse problema de extração de referências bibliográficas.

## 2.3 Abordagens para a construção de sistemas de EI

Há duas principais abordagens para projetar um sistema de Extração de Informação: a abordagem baseada em Engenharia do Conhecimento e a abordagem baseada em Treinamento Automático (APPELT; ISRAEL, 1999).

Os sistemas de extração de informação aplicam regras para a extração da informação relevante. Essas regras são escritas manualmente ou com algum grau de automação. Para os sistemas construídos manualmente normalmente é adotada a abordagem de sistemas baseados em conhecimento, com uma base de conhecimento e algum mecanismo de inferência associado. Já os sistemas construídos automaticamente utilizam os mais diversos algoritmos de aprendizagem de maquina.

Na *Abordagem baseada em engenharia do conhecimento* um especialista do domínio especifica manualmente as regras do sistema através de uma análise do corpus. Em seguida essas regras são aplicadas sobre o corpus e verifica-se a eficiência das regras. O refinamento das regras pode ser feito interativamente para aumentar o desempenho do sistema, esse é um trabalho de lapidação dos algoritmos até alcançar o resultado satisfatório. O sucesso de um sistema baseado em regras manuais está sujeito à habilidade do especialista em escrever tais regras. A figura 2.4 ilustra a Abordagem baseada na Engenharia do Conhecimento:

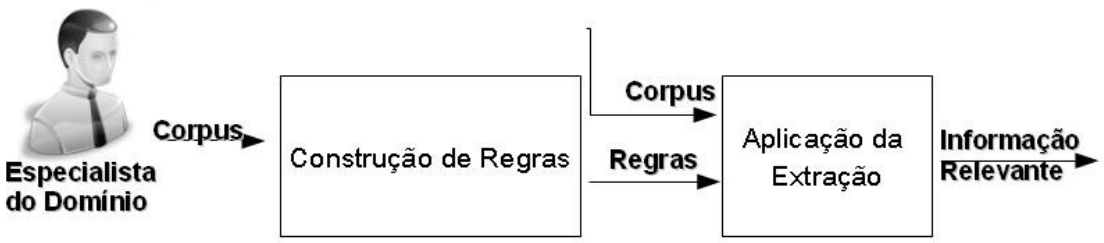


Figura 2.4: abordagem baseada na engenharia do conhecimento

Na *Abordagem baseada em treinamento automático*, o especialista, conhecedor do domínio anota um corpus de treinamento. Em seguida um algoritmo de aprendizado de maquina é executado, o resultado é informação que pode ser usada na análise dos textos de forma automática. O sistema aprende novas regras constantemente, refinando as mesmas a partir do corpus anotado e interação com o usuário. A figura 2.5 ilustra a Abordagem baseada na Treinamento Automático:

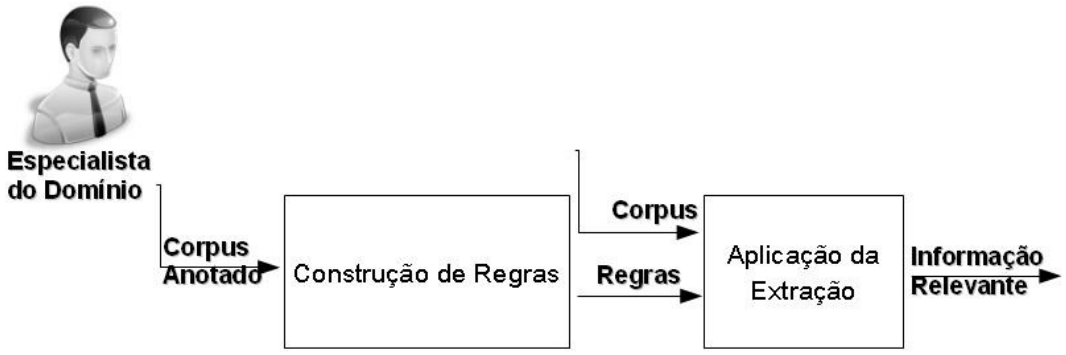


Figura 2.5: abordagem baseada em treinamento automático

Não há abordagem superior ou mais efetiva. Cada uma pode ser vantajosa na situação apropriada. Podendo existir até sistemas híbridos que utilizam os dois métodos de forma unificada. Na abordagem de engenharia do conhecimento as regras podem ser muito eficientes já que são escritas por um humano. Este método manual, em geral, são fáceis de implementar, depurar e manter já que não é necessário um conjunto de treinamento. Entretanto, o desenvolvimento baseado em conhecimento é caro e dispendioso, requer um especialista em lingüística e que entenda bem o domínio. Outra desvantagem é que o acoplamento à um domínio especifico é muito forte. No treinamento automático a portabilidade entre domínios é mais fácil. Porém, grande volume de corpus é necessário para efetuar um bom treinamento do algoritmo de aprendizagem. A inexistência ou alto custo para se obter um generoso numero de dados de treinamento podem inviabilizar o uso da abordagem por treinamento automático.

## 2.4 Processamento de linguagem natural em sistemas de EI

A compreensão refere-se à capacidade ou perspicácia de entender e

assimilar uma idéia. A compreensão é um ato de compreender uma forma de representação. O Processamento da Linguagem Natural (PLN) (DRAKE, 2003) (JIELIN, 2007) objetiva fazer a maquina compreender a linguagem natural falada ou escrita.

A PLN tenta reproduzir, nas máquinas, o mesmo comportamento humano na compreensão da língua (MARTIN, 2000). A combinação de palavras em sentenças modelada através de uma gramática dá um significado a uma oração. Utilizando os recursos computacionais tenta-se implementar esse processo de forma eficiente.

O processamento da linguagem natural consiste de uma aplicação seqüencial de diferentes componentes de análise em uma arquitetura *pipeline*. Um *pipeline* é um processo encadeado pelas suas saídas, ou seja, a saída processada de um recurso serve de entrada para outro.

A definição de etapas ou fases de processamento da linguagem natural se baseia nos conhecimentos lingüísticos necessários a compreensão da linguagem. Esse entendimento da linguagem natural pode se dá em vários níveis (DRAKE, 2003).

• Nível Fonético e fonológico: estuda a fisiologia ou produção motora dos sons da fala e sua produção acústica. O conhecimento fonológico de uma língua inclui conhecer as regras para combinar os fonemas desta língua. O domínio desse conhecimento é necessário ao desenvolvimento dos sistemas de reconhecimento e síntese de fala. O reconhecimento da fala envolve a interpretação de ondas sonoras e a associação destas com elementos de fala;

• Nível Léxico: determinação da relação das palavras com suas categorias gramaticais e seus significados. Diversos tipos de processamento contribuem para a compreensão ao nível lexical (de palavra). O primeiro passo geralmente é atribuir um rótulo de parte do discurso (classe gramatical) para cada palavra (nível morfológico). Quando aparece uma palavra que pode ser classificada com mais de um rótulo, é escolhido o rótulo com a maior probabilidade calculada com base nas características do documento em que a palavra ocorre;

• Nível Morfológico: lida com estrutura interna e a formação das palavras, é o estudo da constituição das palavras em elementos básicos. O morfema é a menor unidade significativa de uma língua. O significado de cada morfema permanece a mesma em

todas as palavras, por isso os seres humanos podem dividir uma palavra desconhecida em seus morfemas constituintes para compreender o seu significado (DRAKE, 2003). Por exemplo, a palavra *preregistration (pré-registro)* pode ser morfologicamente analisada em três morfemas separados: o prefixo *pre*, a raiz *registra*, e o sufixo *tion*;

• Nível Sintático: determinação da relação (papel) de um conjunto de palavras em uma sentença. Refere-se à estrutura das frases. Regras pelas quais palavras podem ser combinadas em frases gramaticalmente aceitáveis. O conhecimento sintático reflete-se na capacidade de um sistema de reconhecer frases que são estruturalmente ambíguas e sentenças que possuem o mesmo significado;

• Nível Semântico: determinação do significado e inter-relacionamento semântico das palavras. A semântica tem como objeto de estudo o significado das expressões da linguagem natural, ou seja, é o estudo do sentido das palavras de uma língua;

• Nível Discursivo: objetiva-se em determinar o significado de um conjunto de sentenças. A sintaxe e a semântica trabalham com unidades da sentença, enquanto que o nível de discurso do PLN trabalha com o texto como um todo, ou seja, não interpreta sentenças isoladamente. O nível de discurso faz conexões entre as frases componentes do texto. O nível de discurso trata do fato de que sentenças precedentes afetam a interpretação da próxima sentença;

• Nível Pragmático: Visa determinar o objetivo do uso da língua. Este nível está preocupado com o uso intencional da linguagem. Procura obter o significado não literal fazendo uso do conteúdo e contexto do texto, e de outros tipos de conhecimentos mais amplos para a compreensão da mensagem que está no documento. Busca a percepção da informação de fundo, necessária para transmitir a mensagem visada, o entendimento dos princípios cooperativos que estão por trás das trocas na conversação.

O desenvolvimento de modelos computacionais da língua natural permite um maior processamento de informações. A interação homem - máquina será completa e a PLN terá finalmente alcançado seu objetivo pleno quando os computadores passarem no desafio do teste de Turing. Esse teste proposto por Allan Turing visava descobrir se uma maquina poderia ou não pensar. O interrogador faria perguntas a

duas entidades. Sendo elas um humano e a outra uma maquina. O interrogador não saberia quem é quem a principio e deveria descobrir qual entidade era a maquina apenas analisando as respostas dadas. Caso no final do teste o interrogador não pudesse fazer a distinção então a maquina teria passado no teste pela coerência das respostas dadas corretamente.

As áreas de aplicação de PLN são numerosas. Alguns tipos de aplicações citados por Drake (2003) que usam PLN são:

a) Recuperação de informação (RI) – Sistemas de RI fornecem uma lista de documentos potencialmente relevantes em resposta a uma consulta do usuário.
b) Pergunta-resposta – Sistemas de perguntas e respostas fornecem ao usuário apenas o texto da própria resposta ou passagens de respostas disponíveis.
c) Resumo de texto – Os altos níveis de PLN, particularmente o nível de discurso, podem ser utilizados na implementação de aplicações que produzem a partir de um texto, outro menor constituído de uma abreviada representação narrativa do documento original.
d) Máquina de tradução – Talvez a mais antiga de todas as aplicações que usam PLN. Os vários níveis da PLN têm sido usados em tradutores, que vão desde as abordagens baseadas em palavras (nível léxico) até aplicações que incluem altos níveis de análise.
e) Correção Automática - Sistemas capazes de identificar o contexto em que uma determinada palavra ou frase se encontra e sugerir modificações, melhorias ou adequações às regras gramaticais vigentes. É um recurso comum nos processadores de texto atuais.

### 2.4.1 Arquitetura de um sistema de EI baseado em PLN

Técnicas de Processamento de Linguagem Natural são utilizadas para construção de sistemas EI que tratam textos com nenhuma ou pouca estruturação. A princípio aplica-se a PLN para análise do texto, estruturando a entrada. Em seguida o sistema integra os dados extraídos da etapa anterior produzindo novos

dados. Um Sistema de EI baseado em PLN basicamente segue uma mesma arquitetura de construção independente da tarefa que o sistema se propõe a realizar. Em geral, um sistema de extração de informação é composto por módulos de pré-processamento, parsing, análise de domínio e preenchimento de templetes. Sendo que o módulo de pré-processamento pode ser subdividido em módulos de tokenização, análise léxica e morfológica e resolução de co-referências. Esses módulos são necessários para se obter informações a um nível léxico, sintático e semântico. Dependendo do objetivo e dependendo do autor, algumas etapas de desenvolvimento podem ser descartadas ou agrupadas em um único modulo. De maneira genérica um sistema de EI baseado em PLN segue o modelo apresentado na figura 2.6. Nas seções de 2.4.1.1 a 2.4.1.7 comentaremos cada uma das etapas de PLN para sistemas de EI.

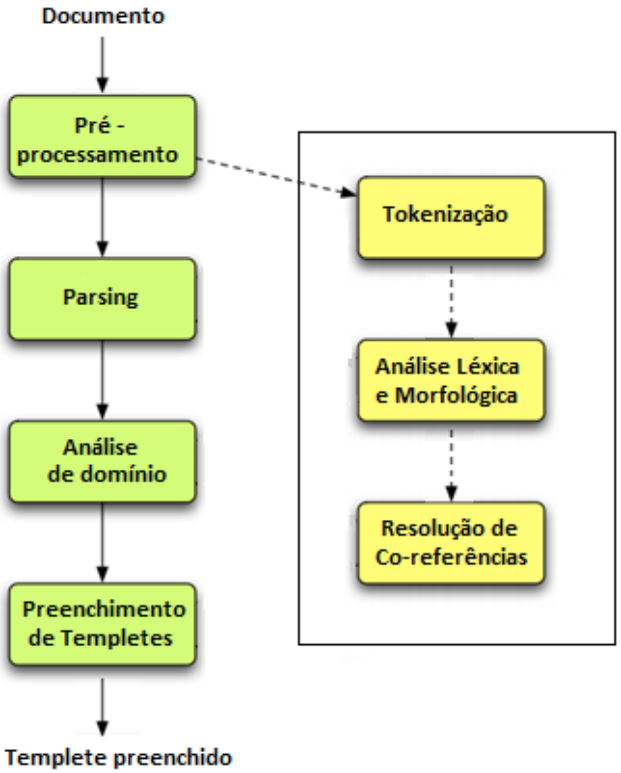


Figura 2.6: fases do processamento de um sistema de EI baseado em PLN

### 2.4.1.1 Tokenização

Tokenização é a fase responsável pela divisão do texto em sentenças e tokens. Em línguas como o português ou inglês é fácil de obter esses tokens, pois a estrutura das palavras nas frases está claramente delimitada por espaços em branco e pontuação. Por exemplo, na frase: "*O cachorro poodle late.*" Temos os seguintes tokens: *"O", "cachorro", "poodle", "late"* e "*.*". O mesmo processo não é tão trivial para línguas como o japonês ou chinês onde se faz necessário um módulo de segmentação de palavras para sistemas construídos para essas línguas. O problema nessa fase está em identificar as ambigüidades, principalmente com relação a pontuação. Por exemplo, um ponto (.) pode significar varias coisas. Ele pode ser um separador de casas decimais, indicar um fim de uma frase ou indicar uma abreviação. Observe: '3.5' e 'Mr. John', ambos possuem o token '.' no meio, porém indicando ações totalmente opostas.

### 2.4.1.2 Análise léxica e morfológica

Para compreender e gerar sentenças coerentes é necessário compreender cada palavra que forma a estrutura. Na análise léxica os tokens obtidos pela fase de tokenização são colocados em um dicionário para determinar seus possíveis part-of-speech (categoria gramatical). O POS Tagging (Part Of Speech Tagging) tem a tarefa de atribuir uma classe morfológica (verbo, substantivo, pronome, etc.) a cada palavra, identificar o caso, o numero e a pessoa para cada token. Por exemplo, na frase "*Tom is happy when He chases Jerry*" a classificação das palavras da frase é mostrada na Tabela 2.2.

| ***Token*** | ***Pos Tagging*** | ***Descrição*** |
|---|---|---|
| Tom | NNP | *Proper Noun* |
| Is | VBP | *Verb, sing. present* |
| Happy | JJ | *Adjective* |
| When | WRB | *WH-Adverb* |
| He | PRP | *Personal Pronoun* |

| | | |
|---|---|---|
| chases | VBZ | *Verb, 3th person sing. present* |
| Jerry | NNP | *Proper Noun* |

Tabela 2.2: Exemplo de *Pos Tagging*

É também nessa fase que são reconhecidos os tokens de nomes próprios e tokens que representam alguma estrutura interna como data, hora, profissões. Técnicas conhecidas como *Case Folding*, *Stemming*, *Lemmatization e remoçao de Stop Words* são empregadas nessa etapa de pré processamento para mais tarde realizar o *matching* (casamento) das palavras mesmo que elas não estejam na mesma forma morfológica exata. Essas técnicas são explicadas a seguir:

- *Case Folding:* todos os caracteres do documento são convertidos para o mesmo formato, ou seja, todos os caracteres são convertidos para a forma maiúscula ou minúscula. Pode remover acentos, pontos e números para a normalização do texto.
- *Stemming*: as palavras em um texto ocorrem em mais de uma forma. Por exemplo, as palavras 'sonhador' e 'sonho', tem um mesmo radical 'sonh' que exprime a idéia geral da palavra. Essas palavras de mesma família tem a raiz do radical igual que lhes dão uma base comum de significação. A função do *Steamming* é extrair esse radical das palavras, suprimindo afixos (prefixos e sufixos) que indicam formas verbais ou plurais.

- *Lemmatization:* A Lematização é um tipo de normalização morfológica que deriva o lema (forma lematizada) da palavra original. Assim, o lema de um verbo é sua forma infinitiva. O lema de uma palavra variável (que não seja verbo) é sua forma singular e (quando existe) masculina (CIMIANO, 2006). Por exemplo, 'computadores' seria reduzida para 'computador' e "dogs" seria reduzida para 'dog'. A função é reduzir a variação morfológica da palavra.

- *Remoçao de Stop Words:* algumas palavras não carregam significado para a maioria das situações, por exemplo artigo e pronomes. Essas palavras devem ser desconsideradas durante a análise para a compressão do texto e diminuição do índice gerado. A identificação das *StopWord* são feitas

comparando os tokens do documento com uma lista que contem palavras pouco relevantes chamada de *StopList*.

#### 2.4.1.3 Resolução de co-referências

Entidades relevantes serão referenciadas de diferentes formas ao longo do texto. Para o sucesso de um sistema de EI a fase de análise de co-referências deve abordar os problemas de redundância de ordem nominal e pronominal. A co-referencia nominal diz respeito aos nomes e suas variações comuns dentro do texto. Por exemplo, 'Marcos Machado' e 'Prof. Machado' são duas maneiras de referenciar uma mesma entidade. A co-referencia pronominal trata da relação que os pronomes (ex., 'ele', 'ela', 'eles') tem associado com entidades antecessoras. Na seguinte frase: 'José foi ao trabalho. **Ele** volta às 18hrs', está claro que o pronome pessoal 'ele' faz referencia a José.

#### 2.4.1.4 Parsing

Analise sintática tem como objetivo identificar a estrutura sintática válida da sentença analisada. A execução do processo de análise sintática é chamada de *Parsing* e é um importante passo para o entendimento da sentença. Sistemas de EI se interessam apenas por obter informações especificas no texto e ignora outras partes que não são relevantes no processo. Por isso a análise sintática deve considerar os termos essências da oração e a disposição das palavras nas frases.

Na processo de *Parsing* é construída uma árvore sintática para cada sentença, de modo a identificar as dependências sintáticas entre as palavras constituintes da sentença. As dependências sintáticas são relacionamentos a nível sintático entre as palavras. Por exemplo, se o sujeito de uma oração é o termo da oração que funciona como suporte de uma afirmação feita através do predicado e o predicado: é o termo da oração que, através de um verbo, projeta alguma afirmação sobre o sujeito. A análise sintática descobre a relação entre o sujeito e o predicado em uma sentença.

#### 2.4.1.6 Análise de domínio

Para a extração de fatos e eventos o sistema precisa de uma analise do domínio. Essa extração pode ser obtida manualmente (abordagem de engenharia do conhecimento) ou automaticamente (abordagem de treinamento automático). A parte do texto onde se encontra um determinado termo lingüístico é memorizada e a informação é extraída a partir de regras de extração para preencher o template. Existem duas abordagens para extrair essas informações: abordagem molecular e abordagem atômica (APPELT; ISRAEL, 1999). Na primeira ocorre o casamento dos argumentos em um evento com um termo simples. Essa abordagem começa com um numero pequeno de regras que capturam os casos mais comuns do domínio, ignorando os termos menos freqüentes. Ao longo do desenvolvimento essas regras são aumentadas para abranger os casos mais problemáticos. Por isso um sistema com essa abordagem inicia-se com baixos percentuais de recall e alto numero de precisão, invertendo esse quadro com o tempo. A abordagem atômica constrói um modulo de domínio que reconhece os argumentos em um evento e os combina dentro da estrutura de template. Qualquer entidade identificada é considerada relevante para o domínio obtendo altos valores de recall e baixos valores de precisão.

#### 2.4.1.7 Preenchimento de templetes

O objetivo da EI não é interpretar todo o documento que está sendo processado, mas apenas identificar os trechos desse documento que preenchem corretamente um dado formulário (template) de saída. Esse formulário define um conjunto de campos (slots) que determinam as informações que devem ser extraídas (SILVA, 2004). Nessa etapa acontece a atribuição de valores aos templates (lacunas a serem preenchidas pelas informações extraídas). Atribuições que representam data, hora, profissão, dentre outras, são padronizadas por regras de normatizações.

Os elementos de informação que foram tratados durante as fases anteriores devem ser agora instanciados para de se criar um template final. Nessa fase um algoritmo decide quais templates podem ser preenchidos. Podem ser utilizadas as duas abordagens de sistemas de EI (engenharia do conhecimento e treinamento automático) para se obter estratégias de combinação automáticas ou definidas por regras estáticas.

## 2.5 Sistemas *Wrappers* de Extração de Informação

Diferentemente dos sistemas para Extração de Informação baseados em PLN, os sistemas *wrappers* exploram a regularidade apresentada por textos estruturados ou semi-estruturados a fim localizar os dados relevantes.

Os wrappers, foram criados para extrair informações em textos estruturados ou semi-estruturados, onde um processamento linguístico é difícil de ser realizado. Um exemplo deste tipo de texto é uma tabela HTML com a listagem dos produtos de uma loja virtual. Os wrappers baseiam suas regras de extração em informações do texto como formatação, delimitadores, tipografia e frequência estatística das palavras.

A abordagem automática para construção de *wrappers* utiliza técnicas de aprendizagem de máquina, com o objetivo de aprender regras de extração a partir de um *corpus* de treinamento. Esta abordagem não requer praticamente nenhum esforço humano para escrever o código de um novo *wrapper*. No entanto, o comportamento das regras de extração geradas depende da seleção manual dos exemplos de treinamento.

A construção semi-automática de *wrappers* se faz com o auxílio de ferramentas que permitem que o usuário especifique a estrutura dos dados a serem extraídos e o contexto em que tais dados ocorrem no documento.

Construir manualmente um *wrapper* significa escrever todo o código do programa. É a técnica mais demorada e trabalhosa, porém é a que apresenta maior precisão nos dados extraídos. Trata-se de uma técnica simples, mas o nível de detalhes pode ser grande. Neste caso, normalmente as regras de extração são

regras de produção tradicionais, podendo-se utilizar uma máquina de inferência a fim de melhorar o desempenho do sistema (MELO, 2001).

## 2.6 Métricas de avaliação

Um sistema de extração de informação é avaliado por duas métricas de eficiência: ***recall*** e **precisão**. Essas métricas foram adaptadas das versões da área de Recuperação de Informação e medem o grau de habilidade de recuperação dos dados e de relevância dos dados em função do desejo do usuário. Em EI, o ***recall*** é uma medida da relação entre o total de informações corretas extraídas pelo sistema e o total de informações corretas presentes nos documentos processados. Já a **precisão** mede a relação entre a quantidade de informações corretas extraídas pelo sistema com o número total de informações extraídas (corretas + incorretas) (Silva, 2004). O ***recall*** e a **precisão** são formalmente definidos pelas equações (1) e (2).

$$\text{Precisão} = \frac{Qtd.\ de\ inf.\ corretamente\ \ extra\ídas}{total\ \ de\ informa\ ções\ extra\ídas} \qquad (1)$$

$$\textit{Recall} = \frac{Qtd.\ de\ inf.\ corretamente\ \ extra\ídas}{todal\ \ de\ informa\ ções\ a\ extrair} \qquad (2)$$

Para combinar essas duas métricas, outra medida usada é o ***f-measure*.** Ela é formamente definida na equação (3).

$$\textit{f-measure} = \frac{(\beta^2+1)*\text{Recall}\ *\text{Precis}\ \tilde{a}o}{\beta^2*\text{Precis}\ \tilde{a}o+\text{Recall}} \qquad (3)$$

O parâmetro β quantifica a preferência de recall sobre a precisão e geralmente é usado como valor 1, dando assim pesos iguais para o *recall* e a precisão na formula simplificada da equação (4).

$$\textit{f-measure} = \frac{2*\text{Recall}\ *\text{Precis}\ \tilde{a}o}{\text{Precis}\ \tilde{a}o+\text{Recall}} \qquad (4)$$

Um sistema perfeito que extraísse toda a informação desejada teria 100% de *recall*. Se esse mesmo sistema extraísse apenas a informação desejada e nada além dela teria 100% de precisão. Outro exemplo é um sistema com 60% de *recall*, isto significa que 40% de informações relevantes estão faltando. Já um sistema com 60% de precisão indica que a base de dados está com 40% de informações incorretas.

## 2.7 Desafios da Extração de Informação

A extração de informação enfrenta como desafio encontrar uma alta taxa nas métricas de avaliação. Problemas de portabilidade do sistema também são encontrados.

O ideal seria existir uma alta porcentagem para ambas as métricas de recall e precisao em um sistema de EI. Porém o que se observa na experiência prática são taxas conflitantes variando inversamente uma em relação à outra. Ou seja, permitindo-se que o sistema tenha um *Recall* maior, diminui-se a Precisão e permitindo-se que ele tenha uma *Precisão* maior, diminui-se o Recall.

Geralmente, em um sistema de EI, a taxa de precisão é maior do que a taxa de recall. Explica-se: se a *Precisão* é a razão entre o número de informação extraída corretamente e o número de informação extraída e o *Recall* é a razão entre o número de informação extraída corretamente e o número de informação no corpus, então para uma boa métrica de *Precisão* é preciso minimizar as informações incorretas extraídas. Sendo assim, é mais fácil detectar erros durante a depuração do sistema e mudar o modelo manualmente até que esses erros desapareçam. O problema para um alto *Recall* é que, sem um extenso corpus de treinamento, não é possível para o sistema conhecer os tipos de dados que se deseja extrair e ter uma boa eficácia na procura por entidades nas fontes não estruturadas.

O problema da portabilidade ocorre em um sistema que é construído e ajustado para uma determinada fonte de informação específica ou um domínio específico. Quando estas fontes mudam e/ou o domínio muda, o desafio para o sistema é detectar essas mudanças e se adaptar a elas.

## 2.8 Considerações finais

Neste capítulo foi apresentada a definição da Extração de Informação e descrito a visão geral dos principais conceitos relativos a essa área. Foram apresentadas tarefas gerais para a extração de informação. A descoberta de conhecimento em documentos textuais revela-se útil para a gestão de documentos. As tarefas de EI visão extrair dados e relacionamentos entre esses dados, dando aos usuários o poder de usufruir das informações coletadas automaticamente. O desenvolvimento de componentes de extração de informação não é uma tarefa trivial, para isso é importante conhecer a arquitetura de um típico sistema de EI. Foi mostrada a arquitetura de sistemas de extração de informação baseado em PLN e sistemas baseados em *wrappers*. Para a avaliação de eficácia desses sistemas recorremos a métricas de avaliações que são o recall e a precisão mostradas da seção 2.7. Por fim mostramos os obstáculos enfrentados no desenvolvimento de sistemas de extração de informação.

# 3 TECNOLOGIAS EM EXTRAÇÃO DE INFORMAÇÃO

Esse capítulo mostra plataformas de desenvolvimento para aplicações de Extração de Informação que utilizam PLN. Serão apresentados a ferramenta GATE[1] e o NLTK[2]. Essas ferramentas auxiliam o desenvolvedor na construção de aplicações de Extração de Informação, pois possuem módulos já prontos para as principais tarefas de PLN.

## 3.1 Plataforma GATE

O GATE (General Architecture for Text Engineering – Arquitetura Genérica para Engenharia de Texto) foi desenvolvido pela Universidade de Sheffield para lidar com tarefas de PLN. É uma infraestrutura para desenvolvimento e implantação de componentes de software que processam a linguagem natural. Possui um ambiente gráfico (figura 3.1) de desenvolvimento integrado, que permite a criação de componentes para PLN, possibilitando assim o desenvolvimento de sistema de extração de informação. Como um ambiente de desenvolvimento, auxilia os desenvolvedores a construir aplicações LE (Language Engineering), minimizando o tempo gasto para criação e alteração desses sistemas.

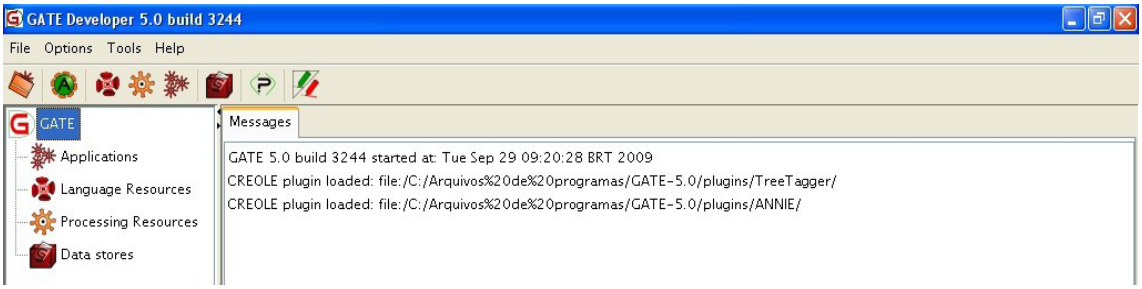


Figura 3.1: tela principal do ambiente gráfico do GATE

---

[1] http://gate.ac.uk/

[2] http://nltk.org/

Esta plataforma fornece facilidades para o processamento e visualização de recursos, incluindo a representação, importação e exportação de dados. Os módulos reutilizáveis são capazes de executar tarefas básicas de processamento de língua natural. Como são reutilizáveis, os utilizadores podem aproveitar esses módulos na criação de outros.

### 3.1.1 Arquitetura

A arquitetura do sistema GATE pode ser visualizada na figura 3.2 e é dividia em três componentes distintos: *Languages Resources* (LR), *Processing Resources* (PR) e *Visual Resources* (VR).

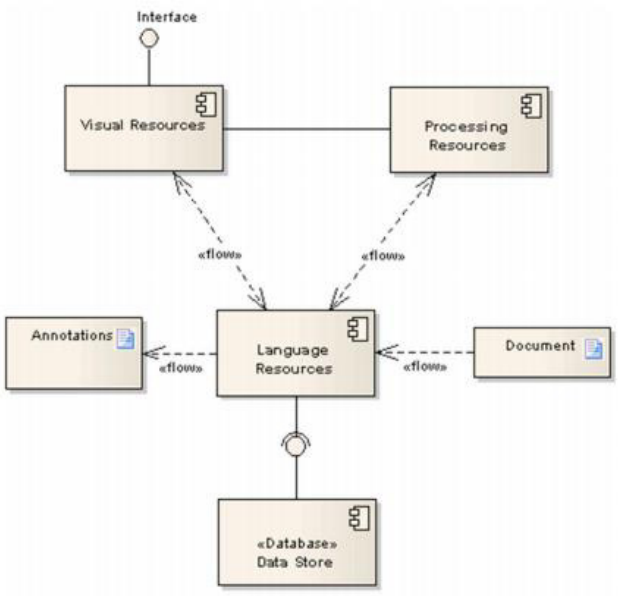


Figura 3.2: arquitetura do GATE (Gonçalves, 2010)

O LR representam entidades, tais como: léxicos (acervo de palavras usadas em um domínio especifico), corpus (conjunto de textos) e ontologias. O PR são os serviços disponibilizados pelo sistema ANNIE (A nearly-New IE sysmte). Estes serviços são módulos reutilizáveis (*tokeniser, gazetter, sentencesplitter, named entity transducer* e *Orthographic Coreference*). Por últimos, os VR são os recursos gráficos utilizados pelo GATE.

Todo esse conjunto de recursos integrados ao GATE é chamado CREOLE (*Collection of Reusable Objects for Language Engineering – Coleção de Objetos de Engenharia da Linguagem Reusáveis*). Quando uma aplicação é desenvolvida no ambiente gráfico GATE, o utilizador escolhe quais os recursos de processamento vai usar e qual a ordem que eles serão executados, assim como os dados a processar. Desta forma pode-se realizar uma comparação de resultados, por exemplo, executando o mesmo módulo em diferentes coleções de texto e analisar as diferenças.

### 3.1.2 ANNIE

O GATE é distribuído com uma aplicação padrão para EI chamada ANNIE ( A nearly-New IE sysmte). ANNIE baseia-se em algoritmos de estado finito e na linguagem JAPE (Java Annotation Patterns Engine) que permite reconhecer expressões regulares em documentos anotados. Os recursos disponibilizados pelo sistema ANNIE (Figura 3.3) são vários módulos reutilizáveis que além de poderem ser usados individualmente, podem também ser utilizados em conjunto. São eles:

a) *Document reset*: retorna o documento ao seu estado original. É útil para o processo de depuração do código, já que durante o desenvolvimento é possível manter ou não as anotações durante a análise do documento.

b) *Tokeniser*: divide o texto em unidades menores, tokens simples, como palavras, números, pontuações.

c) Gazetter: é um conjunto de listas previamente definidas representando entidades como cidades, pessoas, organizações, dias da semana, etc. O GATE utiliza essa lista para a anotação de palavras.

d) *Sentencesplitter*: segmenta o texto em frases. Este modulo é necessário para o *tagger.* O sentence Splitter é independente de aplicação ou domínio.

e) POS *tagger*: atribui uma *tag* correspondente à categoria gramatical de casa palavra.

f) *NE Transducer*: utilizando um conjunto de regras que atuam em anotações criadas por componentes anteriores, produz novas anotações sobre entidades nomeadas.

g) *Orthographic Coreference*: Adiciona relações de identidade entre as entidades nomeadas encontradas pelo NE Transducter a fim de realizar co-referência entre as entidades utilizando as suas ligações ortograficas. Não encontra novas entidades nomeadas. Mas, pode classificar corretamente uma entidade não classificada.

A correspondência entre os recursos GATE e as tarefas de PLN é mostrada na Tabela 3.1.

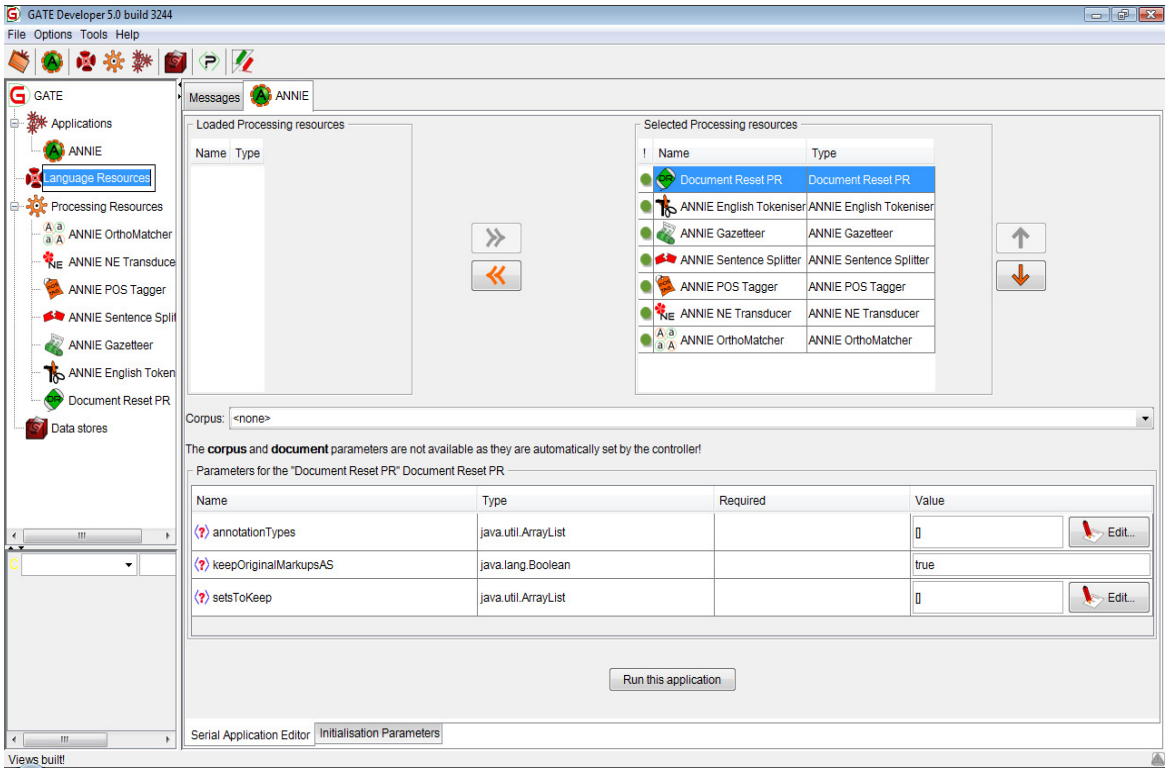


Figura 3.3: aplicação padrão ANNIE

| ***Tarefas de PLN*** | ***Recurso GATE*** |
|---|---|
| *Tokenização* | *ANNIE English Tokeniser* |
| *Normalização* | *ANNIE Gazetter* |
| *Divisão em Sentenças* | *Sentence Splitter* |

| | |
|---|---|
| *POS Tagging* | *ANNIE POS Tagger* |
| *Lematização* | *GATE Morphological Analyser* |
| *Stemming* | *Stemmer* |
| *Reconhecimento de Entidades Nomeadas* | *NE Transducer* |
| *Co-referência entre Entidades Nomeadas* | *ANNIE OrthoMatcher* |
| *Co-referencia pronominal* | *Pronominal Coreference* |
| *Chunking* | *Verb Group Chunker e Noun Phrase Chunker* |
| *Parsing* | *StanfordParser* |

Tabela 3.1 – Correspondencia entre tarefas de PLN e Plugins do GATE

### 3.1.3 JAPE

O JAPE (Java Annotation Patterns Engine) é uma linguagem que permite reconhecer expressões regulares em um documento anotado. Uma gramática JAPE consiste em um conjunto de fases, estas por sua vez consistem num conjunto de regras. O JAPE possui dois tipo de regras: as left-hand-side (LHS) e as right-hand-side (RHS). Enquanto as LSH são constituídas por padrões de anotações que podem conter operadores de expressões regulares tais como “*” (zero ou mais ocorrências), “?” (zero ou uma ocorrência), “|” (ou) ou “+” (uma ou mais ocorrencias). Já as RHS são compostas por instruções que permitem manipular as anotações, definem as ações que devem ser executadas quando as regras “à esquerda” forem satisfeitas. Pode conter blocos de código Java válido. Na figura 3.5 temos um exemplo da gramática JAPE para identificação de universidades federais em um texto:

```
1  Phase:Fase1
2  Input:Token
3  Options: control = appelt
4
5  Rule:IdentificaUniversidades
6  Priority: 25
7
8  (
9      {Token.string == "Universidade"}
10     ( {Token.string == "Federal"} )?
11     ( {Token.string == "de"} |
12       {Token.string == "da"} |
13       {Token.string == "do"}  )
14     ( {Token.category == "NNP"} )+
15
16 ):nomeOrg
17
18 --> :nomeOrg.Organizacao = {regra="IdentificaUniversidades"}
```

Figura 3.4: exemplo de gramática JAPE para identificação de universidades federais

A regra “à esquerda” (linhas 8 a 16 que antecede o ‘->’) contém a expressão regular que será executada nas anotações de entrada. A regra especifica a sequencia de tokens *Universidade*, seguida opcionalmente(“?”) por uma ocorrencia da palavra *Federal*, seguida pelos tokens ‘*de*’ ou ‘*da*’ ou ‘*do*’ e de um ou mais (“+”) substantivos próprios (NNP).

A regra “à direita” (linha 18) será executada e irá criar uma nova anotação. Por exemplo, se a regra aplicada na sequencia de tokens “*Universadade*”, “*do*”, “*Maranhão*”, uma nova anotação será criada e se estenderá pela sequencia “*Universidade do Maranhão*”. Essa nova anotação receberá o valor IdentificaUniverdades, para caracterizar a regra usada em sua identificação e criação.

### 3.1.4 Avaliação

O GATE possui dois mecanismo para avaliação de um sistema de extração de informação: o AnnotatioDiff Tool e o Benchmarking Tool.

O AnnotatioDiff Tool permite a medição de desempenho e visualização de resultados. A ferramenta intercepta automaticamente todos os tipos de anotações e

para cada uma é gerado números de *recall*, *precisão, f-measure* e falsos positivos. Desta forma é possível fazer uma comparação entre elas ou a comparação entre duas versões do sistema.

O Benchmarking Tool difere da ferramenta anterior na medida em que permite a avaliação ao longo de um corpus inteiro ao invés de um único documento. Ela também permite o acompanhamento da performance do sistema. São mostradas estatísticas de desempenho para cada texto no corpus e médias para um corpus em comparação a um corpus de referencia.

## 3.2 Natural Laguage ToolKit – NLTK

O NLTK (Natural Language ToolKit – Ferramenta de Linguagem Natural) é uma infra-estrutura para processar a linguagem natural. Ela foi desenvolvida em 2001 na Universidade da Pennsylvania, em conjunto com um curso de Linguística Computacional. O NLTK foi desenvolvido em linguagem Phyton. O kit se destina a apoiar a pesquisa e o ensino em PNL ou áreas afins, incluindo a linguística empírica, inteligência artificial, recuperação de informação, aprendizagem de máquina e extração de informação.

O projeto do NLTK foi feito de forma a arranjar um grande numero de módulos minimamente dependentes entre si. A idéia principal é facilitar o reuso de pequenas partes da suíte, nas mais diversas aplicações onde isso se fizer necessário. Existe, entretanto, um conjunto de módulos centrais, que definem os principais tipos de dados utilizados nas tarefas de PLN. A correspondência dos recursos do NLTK e as tarefas de PLN são mostrados na Tabela 3.2.

| **Tarefas de PLN** | **Recurso NLTK** |
|---|---|
| *Tokenização* | *NLTK tokeniser* |
| *POS Tagging* | *NLTK tag* |
| *Stemming* | *NLTK stem* |
| *Parsing* | *NLTK parse* |

Tabela 3.2: correspondência entre tarefas de PLN e os módulos do NLTK

O NLTK tem sido usado com sucesso como uma ferramenta de ensino, como uma plataforma para a criação de protótipos e construção de sistemas de pesquisa.

# 4 SISTEMAS DE EXTRAÇÃO DE INFORMAÇÃO

Neste capitulo são apresentados dois sistemas de Extração de Informação. O sistema BREx que trata o problema da identificação e extração de referências bibliográficas e a ferramenta DIAOP-tool que utiliza a extração de informação como base para identificação de instâncias em um corpus anotado e usa essas instâncias para povoar ontologias. Serão analisados aspectos de cada sistema relativo ao tipo de domínio que eles atuam, tipo de texto, tipo de abordagem e resultados obtidos. Por fim, é realizada uma breve discussão dos sistemas mostrados em relação aos conceitos de EI apresentados nos capítulos 2 e 3.

## 4.1 BREx – Extração de Referencias Bibliográficas

O BREx (Gonçalves, 2010) é um sistema de Extração de Informação que foi projetado para tratar o problema da extração e classificação de referencias bibliográficas a partir de texto livre. O sistema localiza as referencias e classifica-as segundo um conjunto pré definido de categorias. O BREx atua em dois níveis de granularidade para procurar as referências. Num primeiro nível, de maior granularidade, a categoria que é identificada é a referência bibliográfica completa. No segundo nível, de menor granularidade, identificam-se os campos que constituem as referências bibliográficas. Esses campos possuem dados do tipo NER como mostrado na figura 4.1.

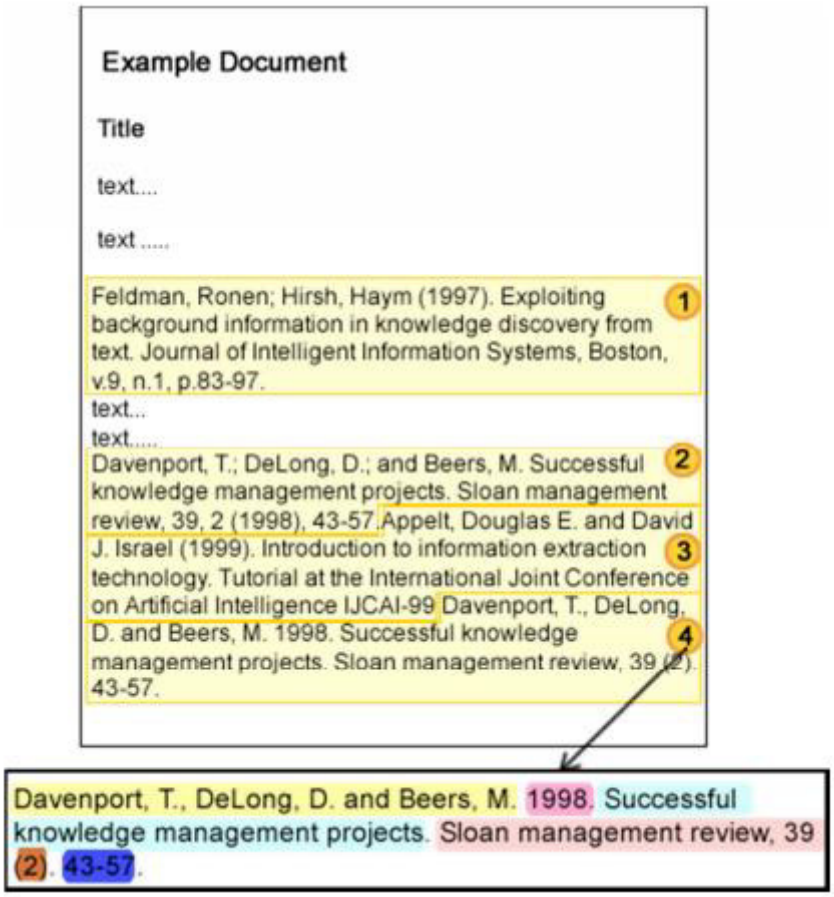


Figura 4.1: exemplo de NER em referencia bibliográficas (GONÇALVES, 2010)

Os serviços de NER – *Named Entity Recognition,* é a tarefa de localizar elementos atômicos em texto e classificá-los, tais como: datas, localização, nomes próprios, organizações e outros. Esses elementos precisam ser identificados nas referências para futura classificação das mesmas. Os serviços de NER do sistema BREx possue uma solução proposta chamada GATE Extractor, usando a plataforma GATE, explicada a seguir.

O componente do sistema BREx chamado GATE extractor, utiliza recursos de processamento que o GATE fornece. Estes recusos são utilizados de forma sequencial e em cascata, onde a saída processada de um recurso serve de entrada para outro.

Na primeira fase o sistema inicia dividindo o texto em *tokens* simples, recorrendo ao recurso do GATE chamado Tokeniser. Em seguida, depois que os *tokens* estão anotados, aplica-se outro recurso chamado Gazetteer. Este é responsável por extrair categorias pré-definidas, tais como, nomes de cidades,

organizações, dias da semana. Logo depois mais outro recurso do GATE, o *Sentence Splitter*, é aplicado para segmentar o texto em frases.

Após esse pré-processamento pelos recursos do GATE o documento é submetido ao *Semantic Tagger* que foi desenvolvido recorrendo à linguagem JAPE. Criou-se uma gramática que consiste num conjunto de regras para a extração das referencias. Essas gramáticas são dividias em três fases. Na primeira fase elabora-se regras com elevado grau de certeza, aqui se identificam componentes como: data, volume e numero. A segunda apresenta regras com maior ambiguidade com base nas anotações anteriores, por exemplos os nomes de autores são extraídos nessa fase. E na terceira fase as regras avaliam sequencia de anotações feitas anteriormente, tentam validar se se tratam de referencia bibliográfica, ou não. As regras devem abranger vários estilos de referencias, por isso durante a escrita dessas regras é prudente desenvolver um código que aceite vários estilos como entrada. Por exemplo, a figura 4.2 ilustra uma regra para anotar as possíveis datas de publicação.

```
Rule:  DateComplete04
Priority: 90
// May 12, 1999
// May 12 1999
((({SpaceToken.kind == space}{Token.kind == punctuation})|({Token.kind == punctuation}{SpaceToken.kind == space}))|{SpaceToken.kind == space})
((({Lookup.majorType == year})| (FOUR_DIGIT))
      ({Token.string =~ "[a-z]"})?)
      (
            (
                  ({SpaceToken.kind == space}{Token.kind == punctuation})|
                  ({Token.kind == punctuation}{SpaceToken.kind == space}))|
            {SpaceToken.kind == space})
      (
            ((({Token.string == "early"}|{Token.string == "late"}|{Token.string == "mid"}|{Token.string == "mid-"}|{Token.string == "end"})?
            {Lookup.minorType == month})
                  ({SpaceToken.kind == space})?
                  ({Token.string == "/"}|{Token.string == "-"}))?
            (({Token.string == "early"}|{Token.string == "late"}|{Token.string == "mid"}|{Token.string == "mid-"}|{Token.string == "end"})?
                  {Lookup.minorType == month}))
      (
            (
                  ({SpaceToken.kind == space}{Token.kind == punctuation})|
                  ({Token.kind == punctuation}{SpaceToken.kind == space}))
            |{SpaceToken.kind == space})
      (
            (
                  ({Lookup.majorType == year})| (FOUR_DIGIT))({Token.string =~ "[a-z]"})?)(({SpaceToken.kind == space})?
                  ({Token.string == "/"}|{Token.string == "-"})
                  ((
                        ({Lookup.majorType == year})| (FOUR_DIGIT))
                        ({Token.string =~ "[a-z]"})? )|
                        ({Token.kind == number, Token.length == "2"}))?
            )
      ):data
-->
{
      AnnotationSet matchedAnnotations = (AnnotationSet) bindings.get("data");

      gate.AnnotationSet otherAnnotations = inputAS.get(matchedAnnotations.firstNode().getOffset(), matchedAnnotations.lastNode().getOffset());
      inputAS.removeAll(otherAnnotations);

      gate.FeatureMap features = Factory.newFeatureMap();
      features.put("rule ", "DateComplete04");
      outputAS.add(matchedAnnotations.firstNode(),matchedAnnotations.lastNode(),"date", features);
}
```

Figura 4.2: exemplo de regra JAPE do sistema BREx (GONÇALVES, 2010)

Ao final, aplicando as ferramentas oferecidas pelo GATE, juntamente com as regras JAPE foi possível anotar todos os elementos desejados contidos no texto de entrada, dando-lhes uma estrutura que será interpretada pela maquina para a extração das referencias bibliográficas.

A avaliação dos resultados obtidos foi feita testando uma coleção de documentos contendo 300 referencias bibliográficas. O sistema foi capaz de identificar 287 referencias, sendo que apenas 202 estavam corretas. As métricas de avaliação *precisão, recall e f-measure* são expostas abaixo:

Precisão = $\frac{202}{287}$ = 0,703

Recall = $\frac{202}{300}$ = 0,673

F-measure = $\frac{2\ x\ 0{,}703\ x\ 0{,}673}{0{,}703 + 0{,}673}$ = 0,68

O resultado do valor de *precisão* é considerado razoável. Apresenta uma boa resposta para a relação entre o numero de referencias extraídas corretamente e o numero total de referencias extraídas. O valor de *recall,* um pouco mais baixo é devido a menor abrangência do sistema, já que de 300 referencias só foram identificadas 202. É muito complexo prever todas as ocorrências das referências, uma vez que essa tarefa está vinculada ao estilo das referencias. Por exemplo, se existir estilos não conhecidos pela gramática, então o sistema deixa de reconhecer a referencia escrita em um estilo que foge do padrão. A media f-measure com valor de 0,68 ainda assim é considerada satisfatória.

### 4.1.2 Considerações

No sistema BREx (Gonçalves, 2010), a escrita das regras utilizando a plataforma GATE segue a abordagem baseada em engenharia do conhecimento (abordagem manual), desta maneira foi exigido muito tempo e esforço no desenvolvimento da gramática. Levando em consideração que o sistema trabalha com fonte de informações semi-estruturadas, elas mesclam certa estrutura com texto livre, por isso o sistema deve ser capaz de lidar com variados tipos de referencias bibliográficas (vários estilos), isto aliado ao fato da falta de rigor de como são feitas

as referencias em cada texto, tornam a tarefa de escrever regras que satisfazem a extração de todos os estilos de referencias bibliográficas bastante complicada. Por exemplo, temos um conflito na referência da figura 4.3:

> **Bodenreider, Olivier; Zweigenbaum, P. 2000. Identifying Proper Names in Parallel Medical Terminologies. Stud Health Technol Inform 77.443-447, Amsterdam: IOS Press.**

Figura 4.3: exemplo de conflito em uma referência bibliográfica (GONÇALVES, 2010)

Analisando a referência bibliográfica da figura 4.3, percebe-se que 2000 refere-se ao ano de publicação. No entanto, o nome do autor abreviado é 'P.' e por coincidência vem logo antes do *token* 2000. Isso faz o sistema errar, pois ele entende que P.2000 refere-se ao numero de páginas do documento. Para resolver esse conflito é necessário criar regras específicas. Neste exemplo em particular, deverá ser criada uma regra que leve em consideração o contexto, ou seja, tentar perceber que P. poderá, ou não fazer parte do um nome de autor. Nem sempre foi possível eliminar todos os conflitos devido ao elevado número de regras. Os sistemas de extração de informação baseados em regras manuais apresentam essa desvantagem, pois para escrever tantas regras é necessário um esforço massivo e muito tempo.

## 4.2 DIAOP-Tool Ferramenta para o Povoamento Automático de Ontologias

O DIAOP-Tool (FARIA, 2013) se trata de uma ferramenta para o problema do Povoamento Automático de Ontologias a partir de fontes textuais, que aplica técnicas de processamento da linguagem natural e extração de informação para extrair e classificar instâncias de ontologias.

O povoamento de ontologias constitui uma abordagem para automatizar ou semi-automatizar a instanciação de propriedades e relacionamentos não taxonômicos de classes de ontologias com conhecimento descoberto em diferentes fontes de dados, como documentos textuais. Por exemplo, no domínio do direito da família e considerando o conceito da ontologia "Mother", o povoamento poderia

associar o termo “Maria” a esse conceito. A ontologia povoada poderá ser utilizada na execução de  sistemas baseados em conhecimento para auxiliar a tomada de decisões.

Entre as várias definições de ontologia existentes, uma é a apresentada por Fensel (2001) provenientede de T. R. Gruber (GRUBER, 1993), em seu artigo “A Translation Approach to Portable Ontology Specification”:

“*Uma ontologia é uma especificação formal explicita de uma conceitualização compartilhada.”*

Nessa definição é importante explicitar-se o significado de algumas das palavras utilizadas. A palavra “conceitualização” refere-se a um modelo abstrato de algum fenômeno que identifique conceitos relevantes desse fenômeno. A palavra “explicita” significa que os tipos de conceitos usados e as limitações do uso desses conceitos devem ser definidos de forma explícita. A palavra “formal” refere-se que a ontologia deve ser passível de ser processada por uma máquina. Por fim “compartilhada” reflete a noção de que a ontologia captura um conhecimento consensual, isto é, esse conhecimento não deve ser restrito a alguns indivíduos, mas aceito por um grupo de pessoas (FENSEL, 2001).

O povoamento de ontologias é realizado através de três fases (FARIA, 2013): identificação de instâncias candidatas, construção de um classificador e classificação de instâncias. A identificação de instâncias candidatas ocorre através da aplicação de técnicas de PLN e/ou técnicas estatísticas. A construção de um classificador e a classificação de instâncias são realizadas através da aplicação de técnicas de EI e serão apresentadas a seguir.

A fase “Identificação de Instâncias Candidatas” consiste de três tarefas: “Análise Léxica e Morfológica”, “Reconhecimento de Entidades Nomeadas” e “Identificação de Co-Referências”.  Esta fase tem como entrada o corpus e como produto o corpus anotado. Ela realiza a marcação dos dados no corpus de entrada e identifica as instancias que serão utilizadas para povoar a ontologia. A fase ‘construção de um classificador’ tem como objetivo a criação automática de um classificador baseado em regras. E a ultima fase chamada ‘classificação de instâncias’ aplica as regras geradas, ou seja, roda o código gerado de forma

automática e associa as instâncias identificadas às suas respectivas classes da ontologia.

Para a construção do classificador a autora (FARIA, 2013), desenvolve uma ferramenta automática. O classificador é um conjunto de regras escritas em linguagem JAPE, como descrito anteriormente uma regra escrita nessa linguagem possui dois lados. O lado esquerdo da regra contém uma expressão regular a ser detectada no conjunto de documentos. O lado direito descreve a ação a ser tomada sobre a expressão regular detectada. Em outras palavras: no lado esquerdo da regra estão as condições compostas de um conjunto de atributos e no lado direito da regra estão as classes da ontologia à qual a instância pertence.

A Figura 4.4 ilustra um exemplo de regra gerada para o relacionamento não taxonômico “wife” da classe “marriage” em uma ontologia específica para tratar do direito de família. Os relacionamentos não taxonômicos são associações entre as classes. Por exemplo, o relacionamento não taxonômico “wife” ocorre entre a classe “marriage” e a classe “person”. O lado esquerdo da regra representa a expressão regular a ser detectada no corpus sugerindo a instância do relacionamento não taxonômico “wife”. O lado direito da regra estabelece a ação a ser executada que é classificar “Nomes Próprios” como instâncias do relacionamento não taxonômico “wife” da classe “marriage”.

```
Rule: Marriage_wife0
Priority: 50
(
       {Token.string == "wife" }
):Marriage_wife
-->
:Marriage_wife.Marriage_wife = { rule = "Marriage_wife0" ,findCategory="NNP",InterText="True",
RuleType="wife", owlPropName="wife_member", owlClassName="Marriage", owlRanger="#Person" }
```

Figura 4.4: exemplo de regra gerada na linguagem JAPE para o relacionamento não taxonômico “wife” da classe “Marriage”

Observamos como a EI é usada nesse exato momento na engrenagem desse sistema. O objetivo de um sistema de EI é modelar uma função do tipo preencheFormulario(Documento)=>formularioPreenchido, que recebe um documento de entrada e retorna o formulário de saída com seus campos preenchidos. No caso do sistema DIAOP-Tool (FARIA, 2013), na fase construção de um classificador é construído/gerado a função do tipo InstanciarOntologia e na fase

Classificação de Instancias o produto é a ontologia povoada. O processo de povoamento da ontologia é mostrado na figura 4.5:

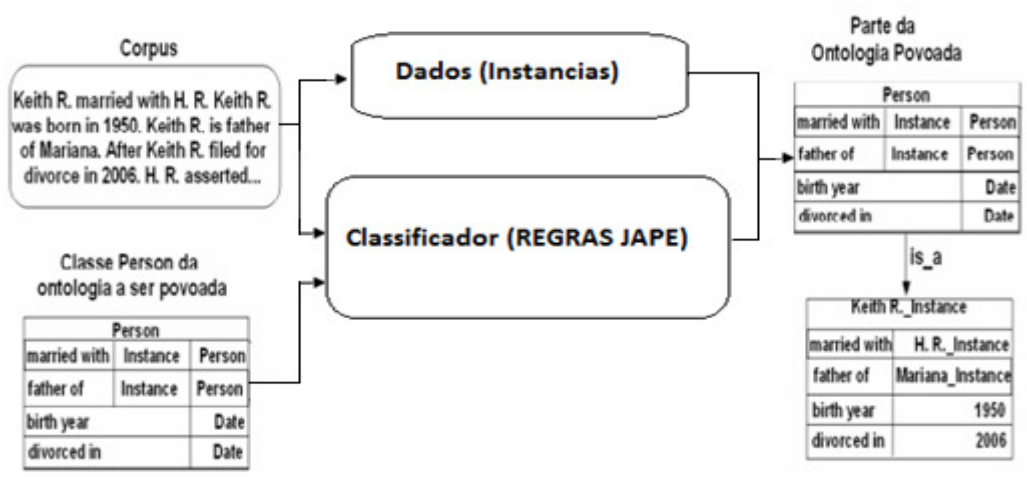


Figura 4.5: povoamento de uma ontologia (FARIA, 2013)

Resumindo todo o processo, a ferramenta recebe um corpus e uma ontologia como entrada e como saída tem as instâncias identificadas às suas respectivas classes da ontologia. A função InstanciarOntologia=>OntologiaInstanciada é análoga a função preencheFormulario(Documento)=>formularioPreenchido.

### 4.2.1 Considerações

O sistema DIAOP-tool utiliza a extração de informação para adquirir e classificar instâncias de ontologias. A fase “Construção de um Classificador” aplica técnicas de EI para construir um classificador baseado em um conjunto de regras lingüísticas. Essas regras são geradas automaticamente através de códigos escritos em JAPE e com o suporte da linguagem JAVA, pois para operações mais complexas o JAPE aceita blocos de código JAVA no RHS (right-hand-side). Desta forma temos um código gerando outro código. Essa arquitetura segue a *Abordagem baseada em treinamento automático,* onde novas regras são aprendidas automaticamente. O classificador é gerado através de um algoritmo de aprendizagem. A geração do classificador inicia-se com a escolha de um conjunto de exemplos de treinamento e exemplos de teste. Os exemplos de treinamento são submetidos ao algoritmo de

aprendizado, que, por sua vez, constrói um modelo para este conjunto de treinamento. Tal modelo será capaz de novos exemplos de instanciação de elementos na ontologia.

# 5 CONCLUSÃO

A busca de informação útil em documentos textuais é uma tarefa complexa e que requer tempo, sendo na maioria dos casos impraticável para um ser humano realizar sem o auxílio de ferramentas automatizadas. Com a enorme quantidade de informação presentes em documentos textuais, enfrenta-se um problema de gestão de todas essas informações. A tecnologia da Extração de informação visa resolver esse problema criando mecanismos que facilitem o rápido acesso ao conhecimento presente nos textos sem a necessidade de efetivamente ler um documento inteiro. Essa área de pesquisa é uma subárea de um campo mais abrangente chamado Processamento da Linguagem natural (PLN), que estuda os idiomas humanos sob uma perspectiva computacional.

O objetivo deste trabalho foi realizar uma apresentação geral sobre o tema Extração de Informação. Este trabalho mostrou os principais tópicos acerca desse tema. Foi realizado um levantamento das técnicas e sistemas de extração de informação, identificando os principais processos descritos na literatura da área. Para isso, foi realizado uma descrição das principais técnicas utilizadas pelos sistemas típicos de EI baseados em PLN e wrappers para extração de informação, e foram analisados dois sistemas concretos que utilizam essas técnicas para exemplificar como, de fato, se realiza a extração de dados relevantes de um documento.

No Capítulo 2 foi apresentada a definição da Extração de Informação. Foi descrito a visão geral dos principais conceitos relativos a essa área, as tarefas definidas para EI que visam combinar dados extraídos para a obtenção de conhecimento, as fontes textuais onde as técnicas de EI são empregadas. Foram apresentados ainda os tipos de abordagens para construção de sistemas, tipos de arquiteturas baseada em PLN e *Wrappers*, métricas para avaliar sistemas, e também mostramos os desafios mais sobressalentes para os sistemas de EI.

As tecnologias apresentadas para execução e desenvolvimento de sistemas de extração de informação foram o GATE e o NLTK, elas reduzem o esforço na produção de aplicações de extração de informação onde necessite de

processamento da linguagem natural, pois essas ferramentas contém módulos pré-definidos para as principais tarefas de PLN mostradas na seção 2.4 do capitulo 2.

Por fim, foram descritos dois sistemas concretos que utilizam EI: o BREx e o DIAOP-tool. O primeiro é um sistema foi projetado para tratar o problema da extração e classificação de referencias bibliográficas a partir de texto livre. E o segundo sistema apresentado, o DIAOP-tool, utiliza técnicas de extração de informação em sua execução para localizar e rotular instâncias que serão usadas para o povoamento de ontologias. Assim podemos verificar a utilidade da Extração de informação que não só serve para a extração puramente de elementos do texto mas também é utilizada como base complementar em sistemas maiores.

A principal contribuição desse trabalho foi reunir informação referente ao tema de Extração de Informação. Por ser uma área relativamente recente, a maioria dos avanços tem surgido a nível acadêmico. O referencial teórico é a base que sustenta qualquer pesquisa científica. Antes de avançar é necessário saber o que já foi desenvolvido por outros autores.

## 5.1 Limitações e trabalhos futuros

O trabalho apresentado expôs o tema Extração de Informação. O foco foi os sistemas de extração baseados em processamento da linguagem natural. Os sistemas *wrappers* também foram apresentados, porém com menor destaque. Para trabalhos futuros pode-se discorrer mais sobre esse tipo particular de sistema de Extração de Informação. Os *wrappers* tratam de problemas que os sistemas baseados em PLN não são capazes de resolver, onde o processamento lingüístico é difícil de ser realizado, por exemplo, extrair dados de uma tabela HTLM. As principais técnicas ultilizadas pelos *wrappers* para realizar a extração de um texto de entrada são (SILVA, 2004): autômatos finitos, casamentos de padrões, classificadores e modelos de markov escondidos. Cada uma dessas técnicas poderia ser detalha em trabalhos futuros.

# REFERÊNCIAS

ÁLVAREZ, A.C. **Extração de Informação de Artigos Científicos: uma abordagem baseada na indução de regras de etiquetagem**. 2007. 131 p. Dissertação (Mestrado em Ciência da Computação e Matemática Computacional) – Instituto de Ciências de Computação e Matemática Computacional, Universidade São Paulo, São Carlos, 2007

APPELT, D. E., ISRAEL, D. J. **Introduction to Information Extraction Technology: a Tutorial.**. In proceeding of: 16th International Joint Conference on Artificial Intelligence (IJCAI'99)16th International Joint Conference on Artificial Intelligence, Stockholm, Sweden. 1999.

CIMIANO, P. **Ontology Learning and Population from Text: Algorithms, Evaluation and Applications**. Springer-Verlag New York, Inc. Secaucus, NJ, USA, 2006.

COWIE, J.; LEHNERT, W. **Information Extraction.** Communications of the ACM, January 1996.

DRAKE, Miriam A. **Encyclopedia of Library and Information Science: Lib-Pub**. CRC Press, 2003.

ELMASRI,R., NAYATHE, S. B. **Fundamentals of Database Systems**. Addison-Wesley, Menlo Park, CA, 2 edition, 1994.

FENSEL, D.; VAN HARMELEN, F.; HORROCKS, I.; MCGUINNES, D.L.; PATEL-SCHNEIDER, P.F. **OIL: An ontology infrastructure for the Semantic Web.** IEEE Intelligent Systems & their appplications, 2001.

GONÇALVES, Ricardo. **Extracção de referências bibliográficas**. 86 f. Dissertação (Mestrado). Instituto superior técnico, Universidade Técnica de Lisboa. 2010.

GRISHMAN, R. **Design of the MUC-6 Evaluation**. In: MESSAGE UNDERSTANDING CONFERENCE, 6., 1995.

GRISHMAN, R. e SUNDHEIM, B., **Message understanding conference - 6: A brief history.** In Proceedings of the 16th International Conference on Computational Linguistics, Copenhagen, June 1996

GRUBER, TR. **A translation approach to portable ontology specification**. Knowledge Acquisition. 1993.

HAMISH, Cunningham. **Information Extraction, Automatic.** Encyclopedia of Language and Linguistics, Department of Computer Science, University of Sheffield, Second Edition, 2005. Portobello Street, Sheffield S1 4DP, UK. 2005.

JIELIN, D. **Network Dictionary**. Javvin Technologies, 2007. Disponivel em: < http://books.google.com.br/books?id=On_Hh23IXDUC&printsec=frontcover&dq=Network+Dictionary#v=onepage&q=&f=false >. Acesso em: dez. 2013.

MELO, t., m., l. **Um Sistema Especialista para Extração e Classificação de Receitas Culinárias em Páginas Eletrônicas.** Monografia. Universidade Federal de Pernanbuco. 2001.

MARTIN, J. H. **Speech and Language Processing**, prentice-hall. 2000

MOENS, M. F. **Information Extraction: algorithms and prospects in a retrieval context**. Springer-verlag. 2006.

KAUFMANN, Morgan. **Proccedings of the Sixth Message Understanding Conference (MUC-6).** 1995.

SILVA, Eduardo fraga do Amaral. **Um sistema para extração de informação em referências bibliográficas baseado em aprendizagem de máquina**. Dissertação (Mestrado). Universidade Federal de Pernambuco, Recife, 2004

ZAMBENEDETTI, Christian. **Extração de Informação sobre bases de dados textuais.** 142 f. Dissertação (Mestrado). Universidade Federal do Rio grande do Sul, Porto Alegre, 2002.