
**CARLA FERNANDES DA SILVA**

# EXTRAÇÃO DE INFORMAÇÃO EM ARTIGOS CIENTÍFICOS APLICADA A ÁREA DE TESTE DE SOFTWARE

**LAVRAS - MG**
**2011**

**CARLA FERNANDES DA SILVA**

# EXTRAÇÃO DE INFORMAÇÃO EM ARTIGOS CIENTÍFICOS APLICADA A ÁREA DE TESTE DE SOFTWARE

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de informação para obtenção do título de Bacharel em Sistemas de Informação.

Orientadora:

Ms. Juliana Galvani Greghi

**LAVRAS - MG**
**2011**

**CARLA FERNANDES DA SILVA**

# EXTRAÇÃO DE INFORMAÇÃO EM ARTIGOS CIENTÍFICOS APLICADA A ÁREA DE TESTE DE SOFTWARE

Monografia de graduação apresentada ao Departamento de Ciência da Computação da Universidade Federal de Lavras como parte das exigências do curso de Sistemas de informação para obtenção do título de Bacharel em Sistemas de Informação.

APROVADA em 22 de novembro de 2011

Dra. Ana Paula Piovesan Melchiori UFLA

Dr. Tales Heimfarth UFLA

Msc. Juliana Galvani Greghi
Orientadora

**LAVRAS - MG**
**2011**

**AGRADECIMENTOS**

Agradeço a Deus porque sem ele nada seria possível. Aos meus pais pela assistência e paciência, por acreditarem em mim e me incentivarem sempre.

A professora Juliana por ter sido minha orientadora, e por ter me ajudado na execução deste trabalho.

Aos meus amigos pelos momentos felizes e pelas experiências vividas. Agradeço a todos.

**RESUMO**

Atualmente, o número de artigos científicos disponíveis em formato digital na rede mundial de computadores tem aumentado muito, tornando assim, as tarefas de organização e extração de conhecimento útil mais complexas. Desta forma, as técnicas de Processamento de Língua Natural vêm auxiliar na extração e processamento dessas informações. O presente trabalho apresenta o desenvolvimento de um protótipo para extrair informação de artigos científicos aplicada a artigos da área de teste de software, para verificar se estas informações podem servir para gerar conhecimento e entendimento sobre o processo de teste de software. Para a construção do protótipo, foi necessário a utilização de técnicas de PLN e Extração de Informação.

Palavras-chave**:** Processamento de Língua Natural. Extração de Informação. Algoritmo de Extração de Palavras-chave Baseado em Frequência de Padrões.

**ABSTRACT**

Currently, the number of papers available in digital format on the World Wide Web has greatly increased, thus making the task of organizing and extracting useful knowledge more complex. Thus, the techniques of Natural Language Processing are assisting in the extraction and processing of such information. This paper presents the development of a prototype to extract information from scientific papers applied to articles in the field of software testing, to verify that this information can serve to generate knowledge and understanding of the process of software testing. To construct the prototype, it was necessary to use techniques of NLP and Information Extraction.

Keywords: Natural Language processing. Information Extraction. Algorithm to Extract Keywords based on Frequency Standards.

## LISTA DE ILUSTRAÇÕES

## LISTA DE ABREVIATURAS E SIGLAS

| | |
|---|---|
| ASCII | American Standard Code for Information Interchange |
| EI | Extração de Informação |
| EPC-P | Extrator de Palavras-Chave por frequência de Padrões |
| MT | Machine Translation |
| MXPOST | Maximum Entropy POS Tagger |
| PDF | Portable Document Format |
| PLN | Processamento de Linguagem Natural |
| RI | Recuperação de Informação |
| SENTER | Sentence Splitter |
| UML | Unified Modeling Language |

**SUMÁRIO**

# 1 INTRODUÇÃO

Neste capítulo será contextualizado o assunto tratado, bem como exposta as motivações para o desenvolvimento deste trabalho, descrevendo-se os objetivos a serem alcançados e a estrutura do texto.

## 1.1 Contextualização

O Processamento de Língua Natural (PLN) visa simular as capacidades humanas de comunicação e interpretação de forma computacional, utilizando técnicas de representação de conhecimento, pode ser aplicado nas mais diversas áreas do conhecimento humano, e apresenta importantes contribuições de cunho social, econômico e educacional.

Dentre as aplicações mais relevantes do Processamento de Língua Natural está a extração de dados presentes em textos, tema em evidência por conta da grande quantidade de informação produzida diariamente por seres humanos e da necessidade de filtrar informação de forma mais eficiente.

Atualmente, o número de artigos científicos disponíveis em formato digital na rede mundial de computadores tem aumentado incessantemente, tornando assim, as tarefas de organização e extração de conhecimento útil mais complexa. Para auxiliar o usuário nessas tarefas, pode-se utilizar a técnica de Extração de Informações (EI), que pode aumentar a compreensibilidade do usuário no resultado final do processo. A Extração de Informação preocupa-se em localizar padrões específicos de dados e, por meio disso, extrair informação estruturada e relevante de dados não estruturados (Kushmerick e Thomas, 2003).

## 1.2 Motivação

Dentro da área Computação há muitas áreas de pesquisa que produzem muitas informações, muito conhecimento e a aplicação das técnicas e conhecimentos obtidos pelos estudos nessas áreas, quando aplicados a situações reais, pode ser insatisfatória pela manipulação equivocada.

Uma destas áreas é a Engenharia de Software, especificamente a área de teste de software, onde há um grande número de documentos relatando técnicas e métodos para a aplicação e melhoria do processo. Segundo Crespo et al. (2004), as empresas enfrentam uma grande dificuldade com a atividade de teste. Isto pode ser um reflexo da falta de profissionais especializados na área de teste de software ou mesmo da dificuldade em implantar um processo de teste utilizando as técnicas existentes na literatura.

Este trabalho objetiva utilizar a técnica de EI para extrair informação no domínio de teste de software visando analisar a qualidade da informação extraída e a possibilidade de uso destas para, por exemplo, facilitar a capacitação dos profissionais que possam vir a trabalhar na área.

## 1.3 Objetivos do Trabalho

O objetivo principal deste trabalho é desenvolver um protótipo capaz de extrair automaticamente informações de um conjunto de artigos científicos. Para fins de avaliação do protótipo, a aplicação será feita em artigos da área de teste de software, escritos em português do Brasil. Para atingir este objetivo, os seguintes objetivos específicos tiveram que ser alcançados:

- Realizar um estudo dos métodos a serem utilizados;
- Implementar o protótipo;
- Submeter os resultados a especialistas para análise.

## 1.4 Organização do Trabalho

O conteúdo deste trabalho está divido da seguinte maneira: na seção 2 serão apresentados conceitos de Processamento de Língua Natural, Extração de Informação, mostrando os principais conceitos e a importância para este trabalho. Na seção 3, apresenta-se a metodologia de desenvolvimento deste trabalho, as ferramentas utilizadas para desenvolvimento do protótipo assim como detalhes da implementação. Na seção 4 são apresentados os resultados. Na seção 5 é apresentada a discussão e trabalhos futuros e por último as referências bibliográficas.

# 2 REFERENCIAL TEÓRICO

Neste capitulo, será apresentado o referencial teórico que serviu como sustentação para este estudo. Nesse referencial foram reunidos os principais autores da área de Processamento de Língua Natural (PLN) e de Extração de Informação.

## 2.1 Processamento de Língua Natural

Processamento de Língua Natural (PLN) é um ramo da Inteligência Artificial que tem como finalidade interpretar e gerar texto em uma língua natural, com auxílio de recursos computacionais.

O PLN é uma área multidisciplinar que abrange áreas do conhecimento e da informação diversas e complexas, tais como: Ciência da Computação, Linguística e Ciências Cognitivas. Com o intuito de remover algumas barreiras entre a interação do homem com a máquina, como por exemplo, a incapacidade do computador compreender a língua natural, existe diversos estudos na área de PLN, que tentam solucionar esse problema. Entretanto esses estudos vêm caminhando lentamente, devido ao grande desafio que é fazer o computador compreender a língua natural.

Nesta seção será exposto um breve histórico de como surgiu o Processamento de Língua Natural, uma breve introdução aos diferentes tipos de conhecimento linguístico para o tratamento da língua natural, arquiteturas que possibilitam a interpretação e geração de línguas naturais, e sobre a técnica de extração de informação.

### 2.1.1 Histórico

Desde a invenção dos computadores os cientistas se deparam com o desafio de criar programas capazes de interpretar mensagens codificadas em línguas naturais. Diante desde contexto, surgiram pesquisas em PLN para Extração de Informações. Segundo Dias-da-Silva et al (2007), o PLN apresenta-se como uma área de estudos bastante heterogênea e diversificada, acumulando uma vasta literatura e agregando pesquisadores de diversas especialidades.

No estudo da história do PLN é importante relatar que a tradução automática, segundo Santos (2001) e outros autores, foi a primeira área em que se trabalhou com PLN , sendo considerada o marco inicial do uso do computador para a investigação das línguas naturais.

No início dos anos 50, Weaver propõe a exploração automática do contexto dos termos, no intuito de solucionar problemas de ambiguidade semântica. Ele acreditava que os circuitos lógicos das calculadoras seriam capazes de resolver os elementos lógicos da linguagem. Em 1954, na Universidade de Georgetown, realizaram a primeira experiência bem-sucedida de tradução automática, entre russos e inglês, realizada por um computador (Garrão, 2004).

Somente a partir de meados da década de 70, depois de muitos casos mal sucedidos em relação ao PLN, é que os trabalhos de tradução automática foram retomados com mais força e maturidade, por exemplo, os sistemas de tradução automática TAUM-METEO, SYSTRAN, ATLAS II, EUROTRA e KBMT, desenvolvidos nesta época. Em 1970, Winograd, em sua tese de doutorado no Instituto de Tecnologia de Massachusetts (MIT), criou um sistema computacional que é citado como o marco dos estudos acadêmicos sobre o PLN: o sistema SHRDLU. A partir deste trabalho a comunidade científica pode

comprovar que era possível a interação homem-máquina através de línguas naturais (Dias-da-Silva et al., 2007).

De acordo com Insite (2010), talvez futuramente os computadores possam se igualar à capacidade humana de compreender e compor textos, mas atualmente essa capacidade ainda é muito limitada.

### 2.1.2 Área de Estudos Linguísticos

As línguas naturais são formadas por um conjunto infinito de frases, que possuem significado e representação sonora. As frases podem ser divididas em unidades menores de som e significado denominadas palavras. As palavras por sua vez podem ser divididas em unidades mínimas de som e significado (Barros e Robin, 2007).

As palavras podem ser caracterizadas de maneiras diferentes de acordo com o estatuto da descrição linguística.

> Podemos definir uma palavra conforme seu estatuto:
> 1. fonético-fonológico*:* quando se trata de depreender a identidade sonora dos elementos que constituem a palavra;
> 2. morfológico: quando as unidades mínimas dotadas de significado são isoladas para a compreensão do processo de formação e flexão das palavras;
> 3. sintático: quando a distribuição das palavras resulta em determinadas funções que elas desempenham na sentença;
> 4. semântico: quando o conteúdo significativo da palavra implica relações de natureza ontológica e referencial para a identificação dos objetos no mundo;
> 5. pragmático-discursivo: quando a força expressiva das palavras remete à identificação dos objetos do mundo em termos do seu contexto de enunciação e condições de produção discursiva (Dias-da-Silva et al., 2007, p.19).

### 2.1.3 Arquitetura de Sistemas de Interpretação de Língua Natural

Oliveira (2004) coloca que, para realizar a interpretação de língua natural, é necessário manter informações léxicas, sintáticas e semânticas em um dicionário, juntamente com palavras que o sistema compreenda. Abaixo, estão descritos os componentes.

#### 2.1.3.1 Analisador Léxico

O analisador léxico, de acordo com Oliveira (2004), irá identificar expressões ou palavras isoladas em uma sentença, tendo por auxiliares delimitadores como espaços em branco, pontuação, e as palavras são classificadas conforme sua categoria gramatical.

O analisador léxico é de suma importância na compreensão das frases, porque para a formação de uma estrutura coerente de uma sentença, é necessário compreender o significado de cada uma das palavras que formam a estrutura.

#### 2.1.3.2 Analisador Sintático

Segundo Dias-da-Silva et al. (2007, p.27), “este processo é responsável por construir (ou recuperar) uma estrutura sintática válida para a sentença de entrada, também chamada de estrutura profunda.”

Segundo Oliveira (2004), a análise sintática de uma oração em português deve considerar sintagmas como termos essenciais (sujeito e predicado), termos integrantes (complementos verbal e nominal) e termos acessórios (adjunto adverbial, adjunto adnominal e aposto). Já a análise do período deve considerar o tipo de período (simples ou composto), sua

composição (por subordinação, por coordenação) e a classificação das orações (absoluta, principal, coordenada ou subordinada).

#### 2.1.3.3 Analisador Semântico

A semântica estuda os significados das palavras e como elas se combinam para formar o significado nas frases. (Barros e Robin, 2007).

O analisador semântico tem como função verificar o sentido da estrutura das palavras que foram reagrupadas pelo analisador sintático, junto à árvore de derivação (árvore sintática), construída com as informações do analisador morfológico e sintático. Existem muitos morfemas que compõem uma palavra que podem mudar o sentido da frase, como exemplo, a ambiguidade (lavar, em “lavar a casa”, “lavar o carro” ou “lavar a roupa”) (Oliveira, 2004).

#### 2.1.3.4 Analisador de Discurso

Segundo Dias-da-Silva et al (2007), um discurso pode ser mono ou multi-sentencial. A análise de discurso em uma sentença pode depender ou não do significado da sentença que a antecede, influenciando no significado da sentença sucessora. Com isto, torna-se necessária a análise de todo o contexto em que a sentença em questão se insere.

Na frase o “O menino subiu no telhado. Ele foi buscar sua pipa”. É necessária uma análise do contexto inteiro para entender que “Ele” se refere ao menino citado na frase anterior.

#### 2.1.3.5 Analisador Pragmático

A análise pragmática se refere à obtenção do significado ‘não literal’ de uma sentença, ou seja, o significado completo, tal como o ser humano o percebe ao ler ou ouvir uma sentença. Além do conteúdo dito ‘literal’, há a necessidade de ligar as frases entre si, de modo a construir um todo coerente, e de interpretar a mensagem de acordo com a situação e com as condições do enunciado (Vieira e Lima, 2004).

A análise pragmática, conforme Dias-da-Silva et al. (2007), está mais ligada a contextualização e ao significado. Por exemplo, "Você sabe que horas são?" pode ser interpretada como um pedido de informação ou uma advertência pelo atraso.

## 2.2 Extração de Informação

Segundo Yangarber e Grishman (2000), Extração de Informação (EI) abrange uma gama de tarefas, incluindo a identificação de nome, classificação, rastreamento de entidade e captura de eventos. É um processo para identificar automaticamente tipos específicos de entidades, contidas em textos e armazenar as informações extraídas de uma forma estruturada.

De acordo com Gaizauskas e Wilks (1998), Extração de Informação é o termo que se dá a atividade de extrair automaticamente uma informação de um tipo de texto de língua natural. A informação extraída é determinada por um conjunto de padrões ou regras de extração específicas ao seu domínio: padrões que podem ser escolhidos de maneira manual, por algum especialista, ou de forma automatizada.

Para Riloff e Jones (1999), os sistemas de Extração de Informações têm como finalidade extrair informações específicas de texto em língua natural. Os

sistemas de EI sempre possuem dois domínios específicos: um dicionário de padrões de extração e um dicionário semântico. O dicionário de padrões pode ser gerado manualmente ou automaticamente, e o dicionário semântico quase sempre é construído manualmente por causa do seu vocabulário específico.

Extração de Informação não deve ser confundida com a área de Recuperação de Informação (RI), a qual seleciona, de uma grande coleção, um subconjunto de documentos relevantes baseados em uma consulta do usuário. O objetivo da RI é de recuperar documentos relevantes de uma coleção, enquanto EI é de extrair informações relevantes dos documentos (Gaizauskas e Wilks, 1998).

Os documentos dos quais são extraídas as informações de interesse podem apresentar algum nível de estruturação na apresentação dos dados, como também podem ser totalmente livres. Os tipos de texto podem ser definidos da seguinte maneira:

**Estruturado:** um texto é considerado estruturado quando apresenta regularidade no formato de apresentação das informações. Essa regularidade, facilmente capturada por sistemas para EI, permite que cada elemento de interesse seja identificado com base em regras uniformes, que consideram marcadores textuais tais como delimitadores e ordem de apresentação dos elementos. Como exemplo, pode-se citar um formulário preenchido.

**Semi-estruturado:** os textos semi-estruturados são aqueles que apresentam alguma regularidade na disposição dos dados. Alguns dados do texto podem apresentar uma formatação, enquanto outras informações aparecem de forma irregular. É o caso de anúncios de classificados em jornais que, em geral, não seguem formato rígido, permitindo variações na ordem e na maneira em que os dados são apresentados.

**Não-estruturado:** os textos não estruturados (livres) são aqueles que não exibem regularidade na apresentação dos dados. Como exemplo deste tipo de texto, pode-se citar uma página Web.

### 2.2.1 Abordagens para Extração de Informação

Segundo Matos (2010), apresentam-se vários tipos de abordagens para extração de informação: a abordagem baseada em dicionário, a abordagem baseada em regras e a abordagem baseada em aprendizagem de máquina. Que será explicado abaixo.

**Abordagem baseada em dicionário:** são bastante aplicadas por armazenar informações de um determinado domínio e a identificação de nomes. A abordagem baseada em dicionário utiliza uma lista de termos para identificar ocorrências no texto. O casamento de padrão, geralmente é utilizado entre as entradas contidas no dicionário e as palavras encontradas nas sentenças (Matos, 2010).

**Abordagem baseada em regras:** segundo Matos (2010), são utilizadas regras para a extração de informações, porém esta abordagem apresenta algumas desvantagens: prolonga significativamente a construção de sistemas, reduz a capacidade de adaptação de regras em outro sistema e exclui termos que não correspondem aos padrões predefinidos. Tem um desempenho bom, mas apresenta problemas de adaptação para novos domínios.

**Abordagem baseada em aprendizado de máquina:** de acordo com Álvarez (2007), esta abordagem pode ser utilizada para automatizar a aquisição das regras a serem usadas em um novo domínio. Os problemas são a necessidade de grandes quantidades de dados e necessidade de treinamento com a entrada de novos dados.

### 2.2.2 Arquitetura de Sistemas de Extração de Informação

O processo de extração de informação consiste em duas etapas principais: a extração de fatos (unidades de informação) do texto de um documento através da análise local do texto e a combinação desses fatos, produzindo fatos maiores ou novos fatos. Ao final, os fatos considerados relevantes ao domínio são estruturados para o padrão de saída.

Para estruturar as informações ao padrão de saída, as técnicas de EI baseadas em PLN utilizam modelos (*templates*) que são estruturas com campos (*slots*) a serem preenchidos pelas informações que devem ser extraídas de um texto.

Com base na arquitetura definida por Álvarez (2007) foram identificados seis módulos principais presentes em sistemas de EI baseados em PLN que será apresentado pela Figura 1 e explicado a seguir.

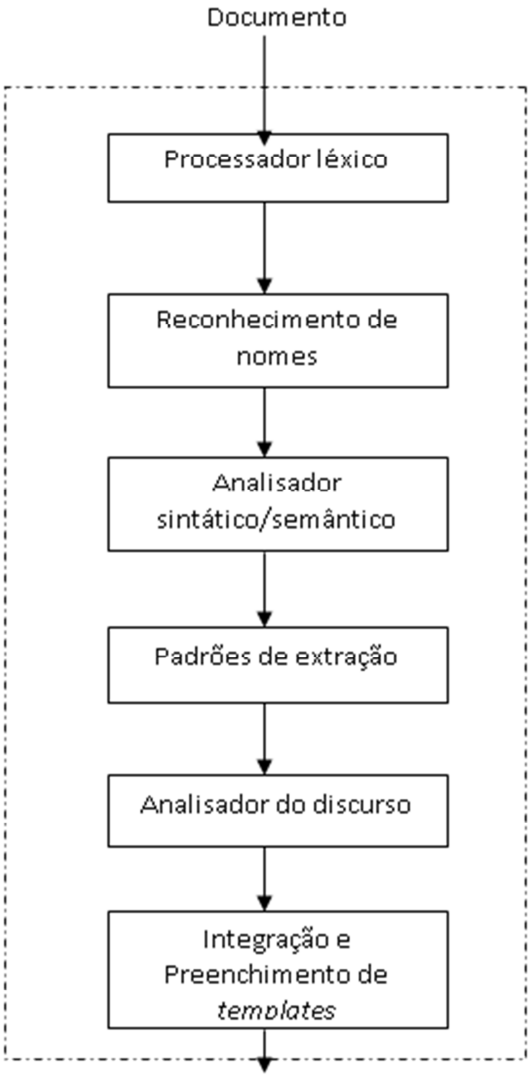



Figura 1 Estrutura de um sistema de Extração de Informação baseado em Processamento de Língua Natural (Álvarez, 2007).

A estrutura de um sistema de extração de informação inicia-se com a coleta de documento, depois o texto é dividido em sentenças e em termos. A

separação dos termos (*tokenization*) é realizada pelo reconhecimento de espaços em branco e outros sinais de pontuação que os delimitam. Após a separação, é feita uma análise léxica e morfológica dos termos para determinar a sua possível categoria morfossintática (substantivo, verbo, artigo, etc.), e as demais características (feminino, plural, etc). A próxima etapa do processamento, a de reconhecimento de nomes é responsável pela identificação de nomes próprios e outros itens que possam ter uma estrutura interna.

O módulo de análise sintática e semântica é responsável por construir uma estrutura sintática, juntamente com alguma informação semântica, para cada sentença do texto. A construção de regras ou padrões de extração consiste na indução de um conjunto de regras de extração específico para o domínio tratado.

A etapa de análise do discurso tem como objetivo relacionar diferentes elementos do texto. Esta fase considera o relacionamento entre as sentenças, ao contrário das anteriores. Caso seja necessário realizar algum processo de inferência sobre a informação, tornando-a explícita, esta tarefa pode ser realizada nesta etapa. E finalmente, as informações parciais são combinadas e os *templates*, definidos pela aplicação, são preenchidos com as informações relevantes ao domínio (Álvarez, 2007).

## 2.3 Trabalhos Relacionados

Esta sessão discute-se vários trabalhos que extraem informação em artigos científicos da língua portuguesa e que utilizam técnicas de PLN.

Dias (2004), realizou uma adaptação do algoritmo de extração automática de palavras-chave para a língua portuguesa. Esta adaptação consiste na utilização de um algoritmo de radicalização de palavras na língua portuguesa,

o qual foi aperfeiçoado neste estudo, e uma lista de *stopwords* da língua portuguesa.

Em Pereira et al. (2002), é apresentado a implementação e a análise de desempenho de dois algoritmos de extração de palavras-chave de textos em português baseados em métodos extrativos: o algoritmo ECP-P, um algoritmo de extração de palavras-chave baseada em frequência de padrões. E o algoritmo EPC-R, um algoritmo de extração de palavras-chave baseada em frequência de radicais.

Em Caseli e Nunes (2006), apresenta a ferramenta de análise morfossintática Anali, a qual foi desenvolvida no NILC (Núcleo Interinstitucional de Linguística Computacional) como resultado da união de outras duas ferramentas de PLN: o etiquetador MXPOST e a ferramenta de análise de corpus Unitex. Nesse sentido, anali representa um ganho em relação ao que é produzido pelas ferramentas citadas, em dois sentidos. Por um lado, enriquece a saída de MXPOST e, por outro, desambigua a saída de Unitex. Além disso, Anali pode operar em três modos distintos: etiquetação (com base apenas na saída de MXPOST), análise morfossintática (com base apenas na saída de Unitex) ou ambos.

Matos (2010) propõem uma metodologia de pré-processamento textual, utilizando a combinação de três abordagens para extrair informação de artigos científicos do domínio biomédico: abordagem baseada em aprendizado de máquina, utilizada para classificar as sentenças em complicação, benefício e outros; abordagem baseada em dicionário, utilizada para identificar diretamente efeitos da anemia falciforme nas sentenças classificadas; e abordagem baseada em regras, utilizada para identificar padrões de extração de efeitos com expressões regulares.

Àlvarez (2007) apresenta um sistema de extração, denominado FIP, que analisa e extrai informações presentes no corpo dos artigos (títulos, autores,

resumos e referências). Este sistema apresenta uma abordagem baseada em regras para a extração de informação automática a partir de referências bibliográficas em trabalhos científicos.

No trabalho de Ribeiro Junior et al. (2005) é apresentado uma ferramenta que procura realizar a identificação automática das áreas de interesse de indivíduos a partir de seus currículos. Esta identificação é feita através da aplicação de técnicas de Descoberta de Conhecimento em Textos sobre as informações extraídas do arquivo XML gerado pela Plataforma Lattes.

Brant (2009) descreve a metodologia utilizada para o desenvolvimento de uma ferramenta capaz de extrair informações em blogs. E propõem o desenvolvimento de um sistema para a extração de informações em blogs indexados pela API Technorati.

# 3 METODOLOGIA

Neste capitulo, será apresentada a metodologia utilizada para desenvolver o protótipo. Será descrita as ferramentas utilizadas e apresentado detalhes sobre a implementação.

## 3.1 Tipo de Pesquisa

Os métodos de pesquisas utilizados neste trabalho foram a pesquisa bibliográfica e a pesquisa exploratória.

Segundo Wainer (2007), a pesquisa exploratória tem como objetivo descrever de forma objetiva e direta os fatos pesquisados. A pesquisa exploratória, além de descrever o fenômeno em estudo, fazer propostas de novas teorias ou novas observações tem como objetivo acrescentar algo de novo no que já foi proposto ou estudado.

Este trabalho tem como objetivo utilizar a Extração de Informação para identificar e extrair informações novas, úteis e interessantes em artigos científicos do domínio da área de teste de software. As informações a serem extraídas estão em artigos completos, disponíveis nas bases eletrônicas de pesquisa.

## 3.2 Ferramentas Utilizadas

Nesta seção são apresentadas as ferramentas utilizadas para auxiliar o desenvolvimento do trabalho aqui proposto. Todas essas ferramentas são de uso livre e são frequentemente utilizadas pela comunidade.

### 3.2.1 XPDF

XPDF é um software visualizador de arquivos no formato PDF, de código aberto. Ele fornece conversão em lote de arquivos *Adobe Acrobat* PDF para texto simples, oferece suporte à execução via linha de comando e será utilizado no pré-processamento do texto.

### 3.2.2 SENTER

SENTER (*SENtence splitTER*) é um segmentador sentencial automático para textos em português do Brasil. O objetivo do segmentador é dividir o texto em segmentos menores. Neste trabalho foi utilizado o SENTER (Pardo, 2006), desenvolvido no NILC, por ser uma ferramenta livre que pode ser obtida no endereço *http://www.icmc.usp.br/~taspardo/Senter.htm*.

O segmentador SENTER faz uso de várias regras para realizar a segmentação do texto, geralmente utilizando como caracteres delimitadores de sentença o ponto final, caracteres de exclamação e interrogação ou quando um marcador de nova linha é encontrado.

### 3.2.3 Tokenizador

A tarefa de *tokenization* consiste em identificar os *tokens* de uma sentença. *Token*, em computação, é definido como um item léxico de interesse para o processamento de língua natural ou mesmo de programação. Em compiladores, os *tokens* são identificadores, palavras reservadas, símbolos simples ou compostos e as constantes da linguagem. Em PLN, os itens léxicos são palavras simples e compostas, números indicando datas, telefone, dentre outros e os sinais de pontuação como vírgulas, ponto final, ponto e vírgula.

### 3.2.4 MXPOST

O etiquetador utilizado neste projeto é o MXPOST (Ratnaparkhi, 1996), um etiquetador morfossintático estatístico baseado na técnica de máxima entropia e aprendizado de máquina em um *corpus* de treinamento. Este etiquetador exige um texto sentenciado e separado em itens léxicos.

A tarefa de etiquetar consiste em adicionar as etiquetas morfossintáticas aos *tokens*. Essas etiquetas representam as categorias gramaticais da língua em questão, por exemplo, substantivo, adjetivo, verbo, nome, entre outros.

O etiquetador utilizado neste projeto foi treinado pelo NILC e apresenta uma precisão de 97%.

### *3.2.5* Algoritmo de *Stemming*

Em PLN, algoritmos de Radicalização, ou também denominados de *Stemming*, consistem em uma normalização linguística, em que as palavras e suas variantes são simplificadas a uma forma comum, em um “quase radical” ou *stem*, resultando na diminuição do número de palavras armazenadas na base de dados. É válido salientar que o radical resultante da normalização de uma palavra não é, necessariamente, igual a sua raiz linguística.

A maioria dos programas de extração de radicais (*stemming*) são baseados no algoritmo de Porter (Porter,1980). Seu funcionamento consiste na remoção de sufixos e/ou prefixos das palavras, o que o torna altamente dependente da linguagem de escrita dos textos utilizados.

O algoritmo desenvolvido para o português consiste de uma série de oito passos, executados de acordo com uma ordem pré-definida pelo algoritmo, de tal maneira que os sufixos mais extensos devem ser removidos primeiro (por exemplo, o sufixo de plural “es” seria removido antes do sufixo “s”). Esse

algoritmo foi desenvolvido com base nos sufixos mais comuns encontrados no português.

### 3.2.6 EPC-P (Algoritmo de Extração de Palavras-chave Baseado em Frequência de Padrões)

O EPC-P não trabalha sobre o texto original, mas sobre um texto etiquetado, onde todas as palavras aparecem associadas às suas categorias gramaticais.

O texto etiquetado é percorrido em sua totalidade e são construídas seis listas, cada uma contendo todas as palavras do texto que se encaixam em um dos seis padrões procurados (nome, nome adjetivo, nome preposição nome, nome adjetivo preposição nome, nome preposição nome adjetivo, nome adjetivo adjetivo), assim como o número de vezes que cada uma das palavras ocorre no texto.

Paralelamente à construção dessas seis listas, são construídas mais seis listas para cada padrão, sendo que nestas são inseridos somente os radicais das palavras. Estes radicais são obtidos através do algoritmo de Porter. Percorrido o texto todo, as seis listas de radicais são ordenadas em ordem decrescente do número de ocorrências de cada palavra no texto, enquanto que as seis listas que contêm as palavras originais são ordenadas alfabeticamente.

Tendo as doze listas construídas, podemos nos desprender do texto e iniciarmos a construção da lista de palavras-chave. Criamos uma lista que damos o nome de “lista1”. Preenchemos esta lista da seguinte forma: mantemos um ponteiro associado ao primeiro elemento de cada uma das seis listas de radicais; para cada um dos seis elementos analisados, dividimos o seu número de ocorrências pelo número de ocorrências do padrão em que ele se encontra (o que chamamos de frequência relativa); o radical que tiver a maior frequência relativa

e que ocorrer pelo menos duas vezes no texto será inserido na “lista1”, juntamente com um marcador para indicar de qual lista ele foi retirado; repete-se o processo até que a “lista1” possua 25 elementos.

Depois de construída a lista1, inicia-se a construção da lista final de palavras-chave, a “lista2”. A “lista2” é preenchida da seguinte maneira: pega-se o primeiro elemento da lista de radicais de nomes; pega-se a primeira ocorrência deste radical na “lista1” caso ele ocorra na mesma e insere-se o radical encontrado na “lista2”; repete-se o processo até que a “lista2” tenha 15 elementos (a escolha do preenchimento da “lista2” com 15 elementos baseia-se no que é proposto por Pereira et al. (2002)), ou que acabe a lista de radicais de nome, o que faz com que a lista de palavras-chave não esteja concluída. Finalmente, para cada elemento, deve ser percorrida a lista de palavras originais correspondente ao padrão em que ele se encaixa. O algoritmo tem uma faixa de acerto de 70.58 %. E foi implementado neste trabalho utilizando a linguagem Java, conforme proposto por Pereira et al. (2002).

## 3.3 Protótipo

O protótipo aqui proposto deve: converter artigos em formato .PDF para .TXT, separar os textos em sentenças e separar as sentenças em *tokens*. Após a separação em *tokens*, deve ocorrer a etiquetação, necessária para extração de palavras-chave (com a etiquetação é possível estabelecer os padrões utilizados pelo algoritmo o EPC-P), e por último a recuperação das frases.

O ECP-P utiliza o texto etiquetado para reconhecer os seus padrões. Para extrair as palavras frases é preciso uma preparação do texto para resolver problemas acarretados pela conversão do arquivo no formato de PDF para TXT, tais como: a presença de caracteres especiais (erros), surgimento de linhas em branco e frases sem pontuação final. Para resolver esses problemas foi preciso

percorrer todo o texto e retirar as linhas em branco, colocar pontuação em todo final de frase, retirar a linha de comando que fica nos textos depois de etiquetados e linhas que possuem apenas um numeral. Notou-se também que as referências bibliográficas atrapalhavam a extração de palavras-chave, pelo fato que as referências muitas vezes apresentam as palavras-chaves, porém não trazem nenhuma informação relevante. Optou-se por retirar as referências dos textos.

A aplicação foi desenvolvida em Java, uma escolha natural devido ao prévio conhecimento da autora deste trabalho. Além disso, como o projeto visa a facilidade de utilização e execução, o Java permite criar uma aplicação multiplataforma, podendo ser executada nos mais diversos sistemas operacionais. Outro ponto a favor é que há grande comunidade ativa de desenvolvedores desta tecnologia, que poderão colaborar com futuras melhorias e modificações.

Para o desenvolvimento deste protótipo, realizou-se uma modelagem simples em UML, produzindo assim o diagrama de caso de uso e diagrama de classes.

O protótipo foi desenvolvido para que apenas um usuário realize a sua tarefa, sem necessidade de autenticação de usuário. As tarefas realizadas pelo usuário estão relacionadas abaixo:

- Identificar o tamanho do grupo de artigos a serem processados (informar o número de artigos);
- Selecionar quais documentos deseja processar (documentos em PDF e não criptografado);
- Visualizar as palavras-chave extraídas.

Serão detalhadas, na seção seguinte, a interface do *software*, a implementação e organização do software.

## 3.4 Protótipo – Implementação

Esta seção tem como objetivo a descrição das classes desenvolvidas no protótipo. As classes foram modeladas utilizando UML e implementadas utilizando a linguagem Java.

. O diagrama de caso de uso é definido na seção 3.5 e são feitas as descrições dos casos de uso. O diagrama de classes é definido na seção 3.6 e são feitas as descrições do conteúdo de cada uma das classes.

## 3.5 Caso de Uso

O caso de uso tem como objetivo representar de forma usual os principais requisitos necessários para implementação do protótipo. Foram implementados cinco casos de uso na ferramenta, sendo eles apresentados na Figura 2.

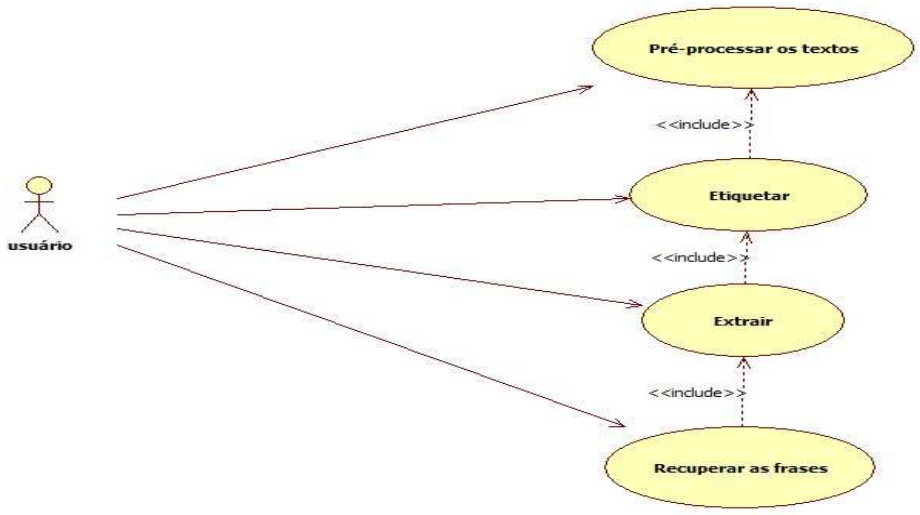



Figura 2 Diagrama de Caso de Uso.

Os casos de usos apresentados na Figura 2 realizam as tarefas Pré-processar os textos, Etiquetar, Extrair e Recuperar Frases que será explicado abaixo.

**Pré-processar os textos:** inicia o processo de extração, o caso de uso tem como dados de entrada os documentos em PDF que o usuário deseja extrair informações. Converte os arquivos em PDF para TXT e prepara os textos para a etiquetação;

**Etiquetar:** responsável pela etiquetação do texto. Este caso de uso depende do caso de uso Pré-processar os textos;

**Extrair:** realiza as ações necessárias para extrair as palavras-chave dos textos. Este caso de uso depende dos casos de uso Etiquetar.

**Recuperar Frases:** recupera as frases referentes a cada palavra-chave obtida no caso de uso Extrair.

## 3.6 Classes em UML

A Figura 3 mostra o diagrama de classes e abaixo será apresentado as classes.

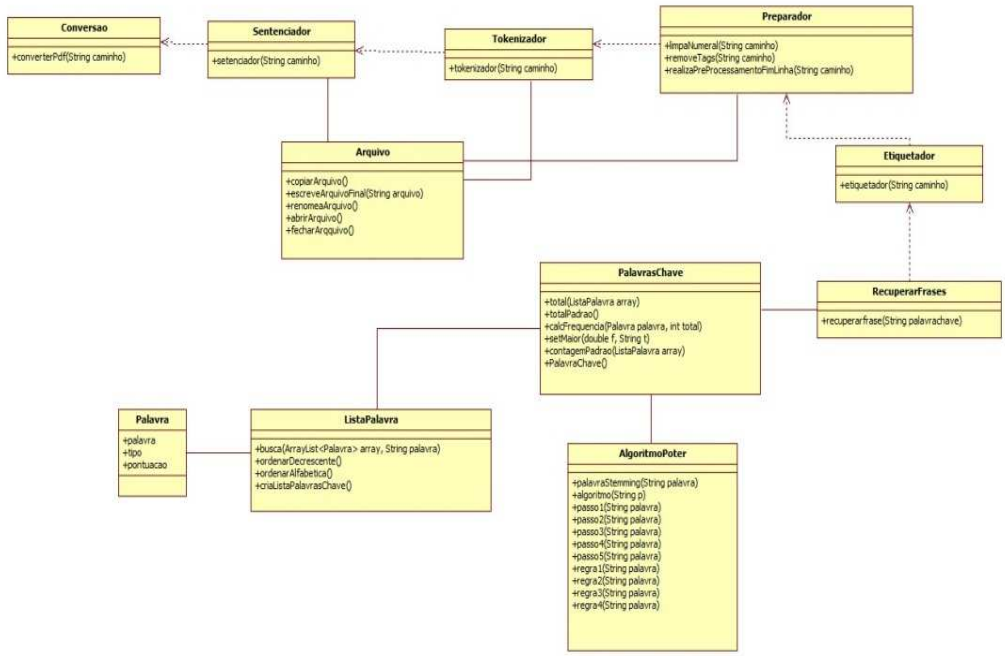



Figura 3 Diagrama de Classes.

A classe **AlgoritmoPoter** é responsável por implementar o algoritmo de Poter para versão português; a classe **RecuperarFrases** é responsável por recuperar as frases referentes a cada palavra-chave obtida; a classe **ListaPalavra** mantém uma lista de objetos do tipo Palavra e também métodos para armazenar e recuperar estes objetos; a classe **Palavra** mantém informações de palavras que serão etiquetadas no processo de etiquetagem, estas informações são: as possíveis etiquetas de uma determinada palavra, suas frequências de ocorrência no processo; a classe **Arquivo** realiza operações com os arquivos como abrir, fechar, copiar, criar um novo e obter caminho de um arquivo; a classe **Sentenciador** dividi o texto a ser etiquetado em sentenças; a classe **Tokenizador** dividi o texto a ser etiquetado em *tokens*; o gerenciamento do XPDF para a conversão dos artigos de PDF em TXT é executado pela classe

**Conversão;** a manipulação do etiquetador é feita pela classe **Etiquetador**; a classe **Preparador** é responsável preparar o texto para extrair as palavras-chave; e a classe **PalavrasChave** é responsável pela implementação do algoritmo ECP-P.

## 3.7 Protótipo – Interface

Esta seção tem como objetivo a descrição da interface desenvolvida no protótipo.

Ao executar o protótipo, o usuário se encontrará diante da tela principal do *software* (Figura 4). Essa é a tela que apresenta o *software* e oferece ao usuário duas opções iniciar o processamento ou **Sair**.

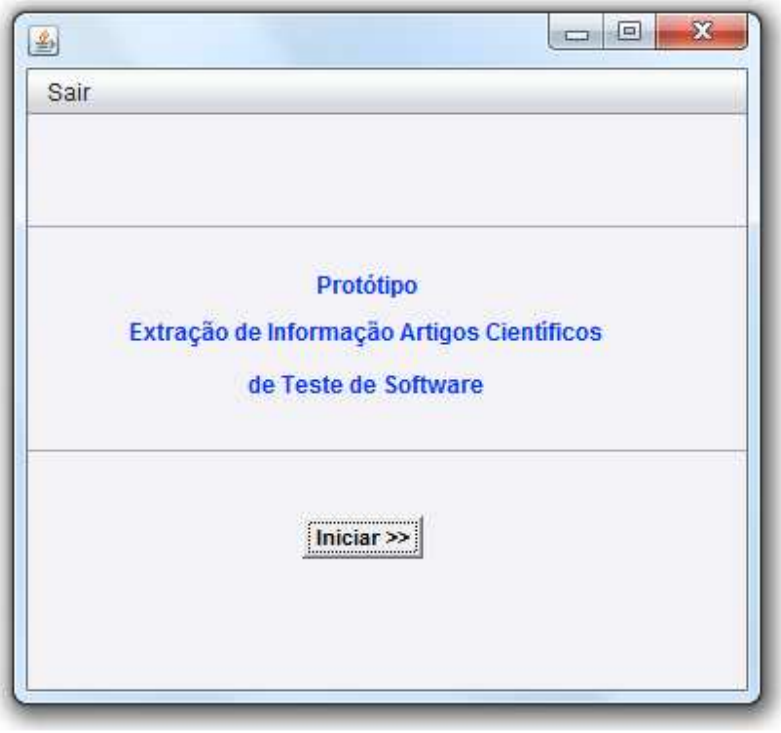



Figura 4 Tela inicial do protótipo.

A próxima tela do software é responsável por identificar o número de artigos que será processado. Uma restrição do software é que o número de artigos deve ser maior que um e no máximo cinco (Figura 5).

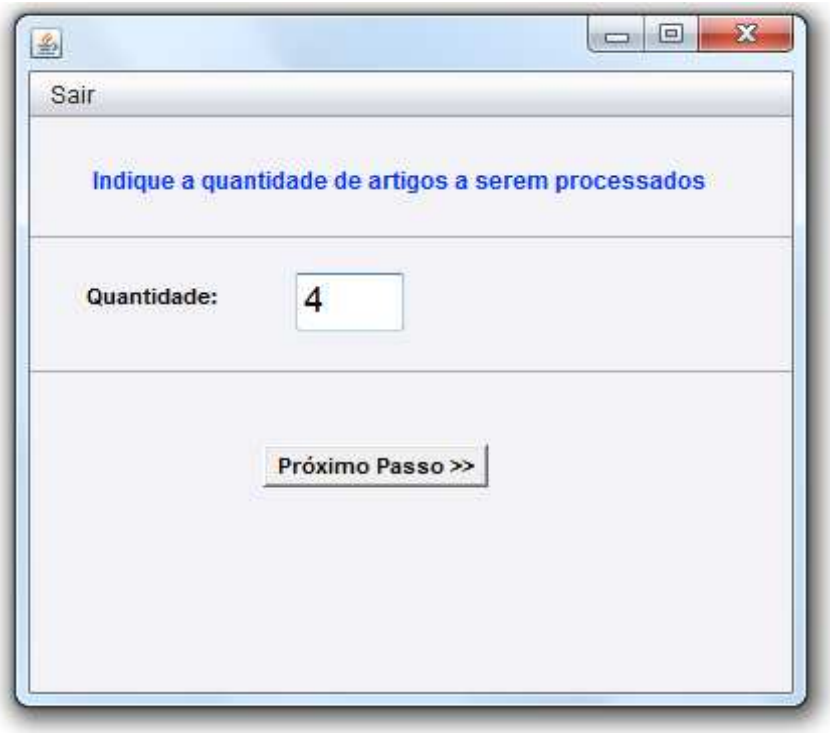



Figura 5 Tela quantidade de artigos a serem processados.

No próximo passo o usuário será requisitado a escolher os arquivos que ele deseja abrir. Os arquivos devem ser da extensão .PDF e não criptografado (Figura 6):

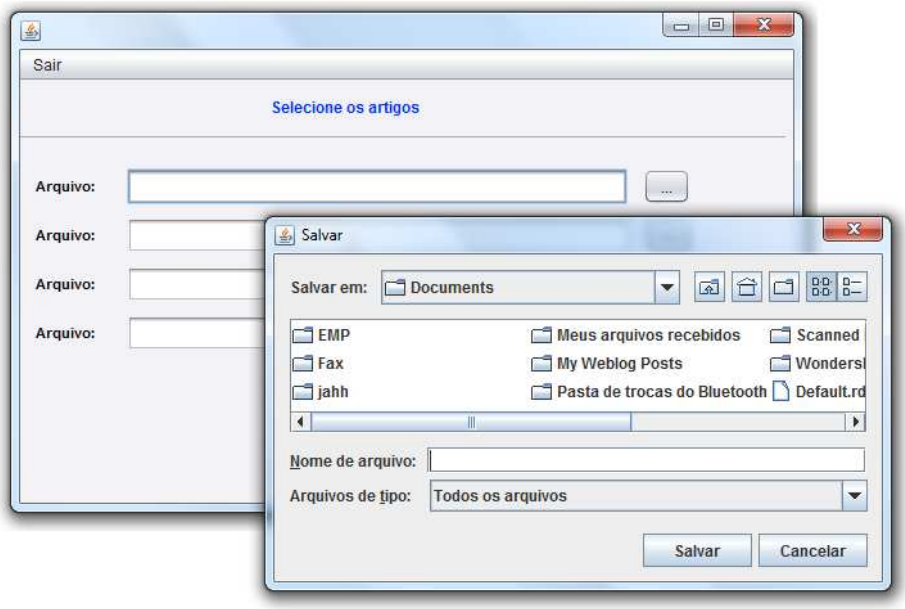



Figura 6 Tela mostrando os artigos selecionados.

Após o usuário identificar quais artigos deseja processar. Inicia-se o processamento dos textos de acordos etapas do processo (exceto a extração das frases). A sequência das etapas será demostrada Figura 7 e explicada abaixo.

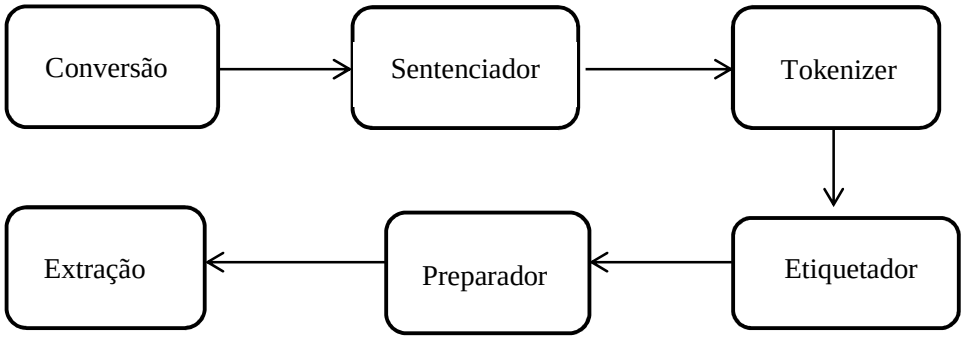



Figura 7 Sequência das etapas.

### 3.7.1 Conversão PDF em TXT

Esta é a fase inicial da aplicação. Ela é responsável por receber o grupo de artigos do qual será extraído as informações. Depois de relatado o número de artigos que serão processados, eles são convertidos para o formato TXT. A conversão é feita utilizando a ferramenta PDF2TXT através de linha de comando.

### 3.7.2 Sentenciador

Está fase é responsável por segmentar o texto em sentenças. Para esta fase foi utilizada a ferramenta SENTER um segmentador sentencial automático.

### 3.7.3 Tokenizer

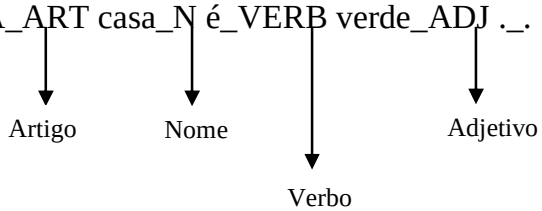
Separa os *tokens* com espaço para que o texto possa ser etiquetado pelo etiquetador morfossintático. Para isso, o texto é percorrido linha a linha e quando é encontrado um caracter especial é inserido um espaço em branco.

### 3.7.4 Etiquetador

Após o texto ter sido preparado, é possível o texto ser etiquetado pelo etiquetador MXPOST através de linha de comando. No qual o etiquetador etiqueta as palavras de acordo com a classe gramatical.

Exemplo texto etiquetado:

A_ART casa_N é_VERB verde_ADJ ._.



### 3.7.5 Preparador

Para extrair as palavras-chave das frases é preciso uma preparação do texto. Para realizar esta tarefa foi criado um filtro automático, que retira às linhas em branco, coloca ponto em todo final de frase, retira as referências dos artigos e retira as linhas de comando colocadas pelo etiquetador.

### 3.7.6 Extração

Nesta fase é feita a extração das palavras-chave. Foi implementado o algoritmo EPC-P como já foi descrito na seção 3.2.6.

A última tela apresenta ao usuário as palavras-chave extraídas depois do processamento cinco (Figura 8):

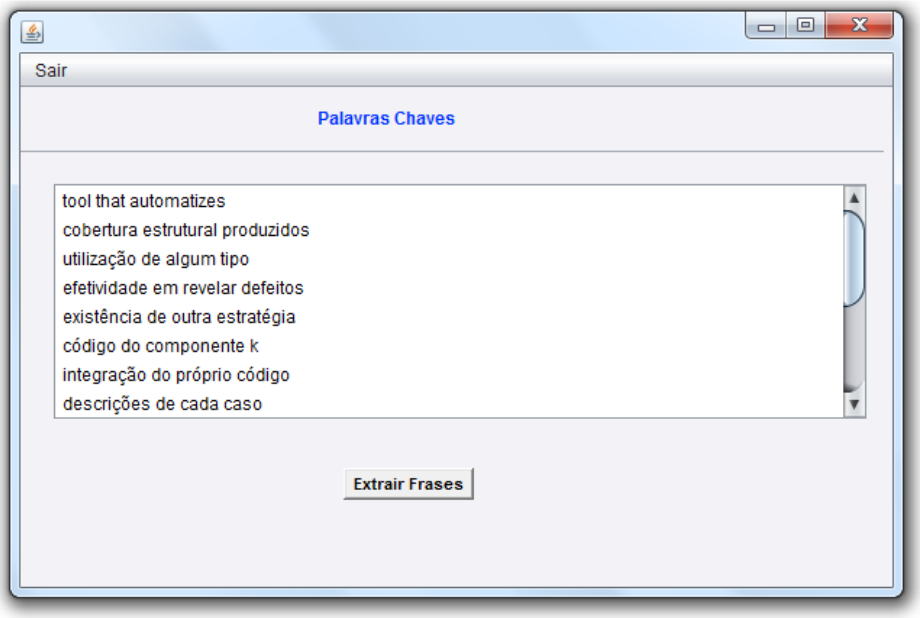



Figura 8 Tela palavras-chaves.

Ao clicar em Extrair Frases o protótipo recupera as frases que possuem as palavras-chave e gera um arquivo TXT. Dentro do diretório do protótipo, foi criada uma pasta **./frases** no qual o arquivo com as frases é salvo.

No diretório padrão, contém o arquivo senter.exe e as pastas xpd e MXPOST que são as ferramentas utilizadas para o processamento dos textos.

# 4 RESULTADOS

Para a avaliação do protótipo, inicialmente foram selecionados manualmente 8 documentos da área de teste de software para realizar a extração. Foram selecionados, inicialmente, artigos que tratam não somente de introdução à área de teste de software, mas também artigos que tratam assuntos mais específicos. O propósito desta seleção foi identificar qual seria a qualidade da informação extraída e se a mesma poderia ser facilmente utilizada. Este grupo de documentos foi dividido em dois conjuntos denominados I e II, cada conjunto contendo 4 documentos cada.

Para cada conjunto, foram extraídas automaticamente 15 palavras-chave e recuperadas todas as frases que apresentassem uma ou mais palavras-chave. Estas frases foram salvas em arquivos diferentes, denominados de arquivo I e II.

A avaliação subjetiva definida neste trabalho depende da colaboração de pessoas que estejam dispostas a ler os arquivos com as frases extraídas e avaliarem-se a informação de cada frase é relevante ou irrelevante. Estes arquivos foram entregue aos avaliadores, tendo em vista que apenas 4 avaliadores responderam, não foi possível obter uma base de dados com um total de avaliações que possa ser considerado uma amostra confiável.

Os resultados apresentados para avaliação devem ser considerados preliminares e será necessário realizar novos testes e ajustes no protótipo para a amostra e os resultados possam ser significativos.

A tabela 1 apresenta os resultados obtidos para o número de 3 avaliadores para o conjunto de documento I. Será apresentado o total de frases extraídas, o número de frases relevantes consideradas por cada avaliador e a precisão. Precisão é a fração de itens recuperados relevantes, em relação ao total de recuperados.

Tabela 1- Resultados das frases do arquivo I pelos avaliadores– Brasil – 2011

| | Total frases | Número de frases relevantes | Precisão (%) |
|---|---|---|---|
| Avaliador 1 | 11 | 7 | 63,63 |
| Avaliador 2 | 11 | 7 | 63,63 |
| Avaliador 3 | 11 | 5 | 45,45 |

A tabela 2 apresenta os resultados obtidos para o número de 1 avaliador para o conjunto de documento II.

Tabela 2- Resultados das frases do arquivo II pelos avaliadores – Brasil – 2011

| | Total frases | Número de frases relevantes | Precisão (%) |
|---|---|---|---|
| Avaliador 1 | 24 | 10 | 41,67 |

Com relação à precisão obtida nas frases, cabe ressaltar que: na extração das frases pode ser observado que forma selecionadas muitas sentenças com ruídos. Isso pode ser devido ao fato de a sentença ter sido selecionada por conter uma ou mais palavras-chave, porém não apresentar nenhuma informação relevante. A frase abaixo demonstra um exemplo:

"A Tabela 1 apresenta um resumo das principais ..."

Para solucionar o problema citado acima, é necessário um melhoramento na fase de preparamento do texto, retirando-se todas as frases que representam títulos de tabelas e figuras.

Foram apresentados alguns problemas como informações incompletas ou imprecisas. Algumas dessas informações extraídas poderiam ser consideradas

importantes se tivessem sido extraídas junto com o seu contexto. Porém como o processo de extração se resume em extrair apenas frases com a palavra-chave, a informação sozinha não apresenta nenhum conhecimento.

# 5 DISCUSSÃO E TRABALHOS FUTUROS

Analisando os objetivos propostos e os resultados alcançados, pode-se concluir que a taxa aproximada de 50% não invalida o protótipo, mas requer muitos refinamentos para que a precisão seja maior.

O protótipo ainda precisa de ajustes em relação à extração de palavras-chaves, pois algumas palavras-chave extraem informações que não são relevantes. Para resolver este problema será necessário realizar testes em relação ao número de palavras-chave extraídas e até mesmo a utilização de outro algoritmo.

Neste trabalho pode-se destacar que um dos maiores desafios encontrados foi o pré-processamento automático do texto a ser extraído (geralmente os textos a serem extraídos são preprocessados manualmente). Este processo precisa ser melhorado para evitar a extração de frases com ruídos.

Trabalho futuros poderão acrescentar novas funcionalidades ao protótipo tais como: a definição do número de palavras-chaves, assim como a edição das palavras selecionadas, novos métodos de pré-processamento e métodos de recuperação de informação.

# 6 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.

BARROS, F.A; ROBIN, J. **Processamento de Linguagem Natural**. In: Congresso da Sociedade Brasileira de Computação, Recife, 1997.

BRANT, L. **Extração de Informação Blogs**. 2009. 54 p. Monografia (Bacharelado em Sistemas de Informação) – Centro Universitário de Feevale, Nova Hamburgo, 2009.

CRESPO, A.N., Silva, O.J., Borges, C.A., Salviano, C.F., Argolo, M.T., Jino, M.. **Uma Metodologia para Teste de *Software* no Contexto da Melhoria de Processo**. Simpósio Brasileiro de Qualidade de *Software*, p. 271-285, Maio 2004.

CASELI, H.M.; NUNES, M.G.V. **Anali: Uma ferramenta morfossintática**. São Carlos: NILC, Série de Relatórios do NILC, NILC-TR-06-09. São Carlos, 2006, 48p.

COWIE, J.; Lehnert.W (1996). **Information extraction**. Communications of the ACM, v.39, n.1, 1996.

DIAS-DA-SILVA, B.C. et al. **Introdução das Línguas Naturais e Aplicações**. São Carlos: NILC, Série de Relatórios do NILC, NILC-TR-07-10. São Carlos, 2007, 121p.

DIAS, M. A. L. **Extração Automática de Palavras-Chave na Língua Portuguesa Aplicada a Dissertações e Teses da Área das Engenharias**. Dissertação de Mestrado. Campinas: FEEC-UNICAMP, 2004

GARRÃO, M. U. **Tradução Automática: ainda um enigma multidisciplinar.** Rio de Janeiro, [entre 2000 e2004]. Disponível em: <http://www.filologia.org.br/vcnlf/anais%20v/civ11_05.htm > Acesso em: 29/11/2011.

KUSHMERICK, N.; THOMAS, B. **Adaptive information extraction: Core technologies for information agents**, Lecture Notes in Computer Science, Springer, v. 2586, p.79-103, 2003.

MATOS, P.F. **Metodologia de Pré-processamento Textual para Extração de Informação sobre efeitos de Doenças em Artigos Científicos do Domínio Biomédico**. 2010. 161 p. Dissertação (Mestrado em Ciência da Computação) – Programa de Pós-Graduação em Ciência da Computação, Universidade Federal São Carlos, São Carlos, 2010.

MIRANDA, E. M. **Uma ferramenta de apoio ao processo de aprendizagem de algoritmos**. 2004. 128 p. Dissertação (Mestrado em Ciência da Computação) - Programa de Pós-Graduação em Ciência da Computação, Universidade Federal de Santa Catarina, Florianópolis, 2004.

MIRANDA, E. M. et al. **Utilização de Processamento de Linguagem Natural para auxiliar na avaliação de algoritmos**. Santa Catarina: UNIVALI / Laboratório de Inteligência, 2004. 11p. Relatório.

NILCTAGGERS**. Núcleo Interinstitucional de Linguística Computacional**. Disponível em: [http://www.nilc.icmc.usp.br/nilc/tools/nilctaggers.html
Acessado em: 17 de maio de 2011].

NUPILL**. Núcleo de Pesquisa em Informática, Literatura e Linguística**. Disponível em: [http://www.cce.ufsc.br/~nupill/
Acessado em: 17 de junho de 2010].

OLIVEIRA, F.A.D**. Processamento de linguagem natural: princípios básicos e a implementação de um analisador sintático de sentenças da língua portuguesa**. Rio

Grande do Sul, 2004. Disponível em: [http://www.inf.ufrgs.br/gppd/disc/cmp135/trabs/992/Parser/parser.html. Acessado em: 07 de junho de 2010].

PARDO, T. A. S. (2006). **Senter: Um segmentador sentencial automático para o português do brasil**. Relatório técnico, ICMC-USP, São Carlos, SP.

PEREIRA, M.B., SOUZA, C.F.R., NUNES, M.G.V.: **Algoritmos de Extração de Palavras-Chave de Textos em Português**. Revista Eletrônica de Iniciação Científica. Ano II, Volume II, Número I (2002).

PUCRS**. Processamento de linguagem natural**. Disponível em: [http://www.inf.pucrs.br/~linatural/ Acessado em: 17 de junho de 2010].

PTSTERMMER. Ptstermmer. Disponível em: [http://code.google.com/p/ptstemmer/downloads/detail?name=PTStemmer-2.0-Java.zip Acessado em: 17 de junho de 2011].

RIBEIRO JUNIOR, L.C. et al. **Identificação de Áreas de Interesses a partir da extração de Informações de Currículos Lattes/XML**. In: Escola Regional de Banco de Dados, 2005, Porto Alegre. Escola Regional de Banco de Dados – ERBD, 2005.

RILOFF, E; JONES, R. **Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping**. In *Proceedings of AAAI-99,1999.*

SANTOS, D. **Introdução ao Processamento de Linguagem Natural através das aplicações.** Lisboa, 2001. Disponível em: [http://193.136.2.105/Diana/download/Santos2001Aplicacoes.pdf Acesso em: 29/11/2011].

VIEIRA, R; LIMA, V.L.S. **Lingüística Computacional: princípios e aplicações.** Rio Grande do Sul, 2004. Disponível em:

<http://www.inf.unisinos.br/~renata/laboratorio/publicacoes/jaia12-vf.pdf> Acesso em 29/11/2011.

XPDF**. XPDF.** Disponível em: [http://foolabs.com/xPDF/about.html. Acessado em: 28 de abril de 2011].

ZAVAGLIA, C. et al. **Avaliação de Métodos de Extração Automática de Termos para a Construção de Ontologias**. São Carlos: NILC, Série de Relatórios do NILC, NILC-TR-05-01. São Carlos, 2005, 13p.

YANGARBER, R.; GRISHMAN, R. **Extraction Pattern Discovery through Corpus Analysis**. TR- 00-143, The Proteus Project, New York University. In: Proceedings of the Workshop Information Extraction meets Corpus Linguistics, Second International Conference on Language Resources and evaluation (LREC 2000), Athens, Greece, 2000.