5
Desenvolvimento e Implementação

Nesse capítulo será descrito com detalhes a arquitetura e o modelo de pré-processamento de mineração de textos ora proposto. Ferramentas já implementadas no modelo *bag-of-words* podem ser encontradas em PreTexT (Matsubara, E. T. et al, 2003), Rainbow (MacCallum, A. K., 1996) e Ngram (Banerjee, S. e Pedersen, T., 2003). Diferentemente dessas ferramentas, o modelo proposto pretende aproveitar mais o conteúdo lingüístico do documento no tratamento das palavras (*words*) transformando o conteúdo em um modelo *bag-of-lexems*. Outra característica é a autonomia do sistema, de caráter pseudo-supervisionado, que tenta aproveitar as dicas presentes no próprio texto para o aprendizado automático de novos lexemas e novas classificações ontológicas, com o objetivo de minimizar esforços manuais. Gerenciar um léxico é um dos maiores esofrços de um modelo pré-processamento.

Nas seções seguintes falaremos um pouco sobre o modelo de aprendizado automático, a importância do léxico computacional e o desafio de mantê-lo atualizado, a estratégia linguística utilizada para esse objetivo, léxicos públicos em forma de tesauros, o modelo implementado, formato de representação, formato de armazenamento e exemplos de pré-processamento do modelo implementado.

5.1.
Aprendizado Automático

A maioria dos sistemas de aprendizado para mineração de textos são supervisionados, isto é, a partir de uma base de treinamento (gabarito), um algoritmo sintetiza um determinado conhecimento, seja de forma probabilística ou simbólica. Esse conhecimento passa a fazer parte de um programa extrator do tipo de informação selecionada para o aprendizado. Assim, o sistema será capaz de processar, em novos textos, as características desejadas.

A Figura 23 ilustra, em alto nível, o procedimento utilizado nos casos de aprendizado supervisidonado. Na etapa de mineração de dados propriamente dita (Figura 2) muitas abordagens seguem esse modelo, como será o caso exemplo nos últimos capítulos dessa tese.

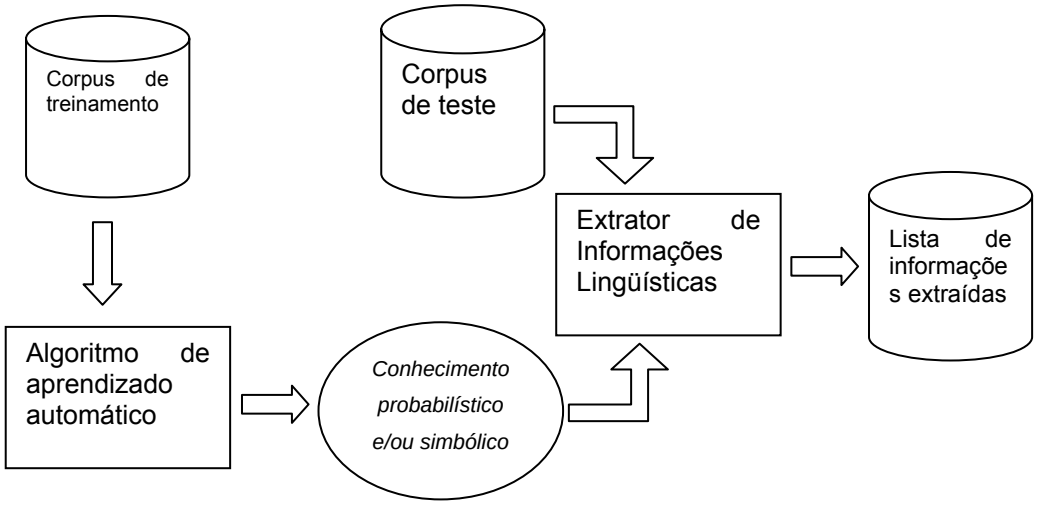

Figura 23 – Esquema geral de um extrator de informações lingüísticas.

O conhecimento probabilístico e/ou simbólico circulado na figura, no caso do pré-processamento, é armazenado em um léxico.

5.2. O Léxico Computacional

Um léxico é um repositório de palavras que pretende armazenar todas as palavras (lexemas) existentes em uma língua. É claro, que no caso limite, temos infinitas palavras em uma dada língua, na prática o léxico armazenará o número de palavras diferentes nos textos processados (corpus).

Além dos lexemas, o léxico faz a associação de cada palavra com outras informações pertinentes à aplicação. Nesse sentido ele tem um papel fundamental de apoio à percepção da palavra. Inspirado no léxico mental, que é o modelo cognitivo de como nosso cérebro armazena as palavras de uma língua, o léxico computacional procura cumprir o mesmo papel, mas já leva outro nome ("computacional") principalmente porque não carrega consigo nenhuma pretensão de ser fidedigno ao léxico mental. Isso significa dizer que a orientação da modelagem computacional desse trabalho não é necessariamente um problema cognitivo da mente humana.

5.2.1.
A Importância do Léxico

O léxico, cada vez mais, vem sendo reconhecido como um dos pontos-chave de programas que visam a lidar com PLN. Nesse âmbito, assumem importância fundamental questões relativas (i) à modelagem – como representar uma palavra e suas propriedades – e (ii) à aquisição lexical – como construir um léxico capaz de adquirir novas palavras automaticamente.

Quanto à modelagem lexical, o problema subdivide-se em como definir as informações que irão constituir os itens lexicais e em como construir o léxico, isto é, como organizar suas informações. No escopo do PLN, dadas as necessidades efetivas de processamento de textos, é interessante que o léxico possua informações relativas à morfossintaxe (como restrições de subcategorização), informações relativas à morfologia (como regras de derivação) e informações semânticas (como relações entre as palavras). Com relação à organização dessas informações em um léxico computacional, uma alternativa promissora, e que será utilizada nesse trabalho, é a Semântica Ontológica (Nirenburg, S. e Raskin, V., 2004) uma abordagem de PLN que usa uma ontologia - entendida como um modelo construído de mundo - como fonte principal para extração e representação do significado de textos, para o "raciocínio" sobre conhecimento derivado de textos e para a geração de textos em linguagem natural a partir das representações dos seus significados.

Diretamente relacionado à questão da modelagem está o problema da aquisição lexical: como preencher o léxico com as informações necessárias, definidas na fase de modelagem. Normalmente, esta tarefa é feita manualmente. Porém, acreditamos que o sucesso do pré-processamento de textos é altamente dependente de um léxico robusto, tanto em termos de qualidade quanto em termos de quantidade de informação. Veremos que executar esta tarefa manualmente é muito custoso, o que faz com que a aquisição lexical seja considerada um dos gargalos mais significativos do PLN (Borguraev, B. e Pustejovsky, J., 1996). Além disso, como o léxico é um conjunto com um número potencialmente infinito de elementos, um léxico computacional eficaz deve permitir o acréscimo de novas palavras sem comprometer ou modificar o sistema, o que se reflete no problema da escalabilidade.

5.3.
Percepção Lingüística

Sob um certo ponto de vista, o processo da formação da percepção por seres humanos pode envolver a construção mental de representação do conhecimento a partir da informação sensível captada por eles. A frase "percepção lingüística" se refere a formar uma percepção que melhore nossa capacidade de compreender e raciocinar com informação lingüística. Informação lingüística pode ser expressa de duas formas básicas: oral ou escrita. A informação escrita geralmente é mais estruturada e menos ruidosa que a oral. A informação oral inclui várias orações semanticamente incompletas (Michael W.Eysenck, 1990) que o interlocutor pode acompanhar devido ao alto grau de contextualização do tópico sendo discutido. O processamento da informação oral requer, além do processador de linguagem, processadores de som, cuja tecnologia está ainda mais distante de fornecer bons resultados práticos.

Sendo assim, nos concentraremos apenas na forma escrita da língua natural. A semântica escrita é mais comportada, mas não por isso simples.

Em (Muradas, A., 1995) são descritas as etapas e os componentes da compreensão natural, mostrando que na maioria dos casos a compreensão da linguagem segue a seguinte ordem: analisador léxico, analisador sintático, analisador semântico e por fim o pragmático. No entanto, já aponta que esses módulos não precisam ser necessariamente distintos e muito menos utilizados nessa ordem, como iremos tentar mostrar também durante esse trabalho.

Para se ter uma idéia de como este problema pode ser difícil, o ato de entender uma frase em linguagem natural requer muitos conhecimentos prévios (*background knowledge*) e conceitos do assunto tratado. Por exemplo, a frase "A cadeira engoliu o tigre" está sintaticamente correta, porém semanticamente incorreta. Já a frase "A formiga engoliu o tigre" está sintática e semanticamente correta, mas pragmaticamente incorreta. Uma criança sem percepção sobre formigas e tigres não pode determinar o quão fraca é a frase pragmaticamente. Vemos então que é necessário formar uma percepção sobre o mundo animal como cadeiras, formigas e tigres para conseguir compreender a frase.

Em resumo, as seguintes interfaces de um sistema de linguagem vão auxiliar o processo de mineração de textos:

Sintaxe: estuda as regras de formação das frases a partir das palavras. Essas regras serão extraídas de acordo com a ordem com que as palavras aparecem no texto. Um modelo sintático pode implementar uma gramática para encadear as palavras. Um exemplo de uso na mineração de textos utilizando a sintaxe se encontra em (Silva, C. et al, 2003).

Semântica: estuda a relação lógica entre os significados das palavras do léxico. O conjunto dessas associações comporão a chamada rede semântica. Essas relações lógicas são representadas por assertivas como, por exemplo, "idéias não têm cores".

Morfologia: analisa a estrutura e a formação das palavras em termos dos seus elementos constitutivos (exemplo: prefixos e sufixos).

Conhecimento de Mundo: este seria um acervo de informações lógicas sobre o ambiente em que se atua. Contém todas as informações enciclopédicas de forma que possamos completar a histórias pela metade em nossa mente. Exemplo:

> "*Quando Lisa estava voltando da loja com um balão, ela tropeçou e o balão foi-se embora flutuando*". (Michael W.Eysenck, 1990)

Para se entender essa simples frase, utiliza-se uma considerável quantidade de conhecimentos. Considere todos os fatos que foram aceitos incondicionalmente e todas as conclusões plausíveis que foram feitas: de que Lisa é uma menina, de que ela comprou o balão na loja, de que Lisa tropeçou numa pedra, de que o balão estava preso pelo barbante, de que quando ela caiu ela largou o barbante e de que o balão subiu, e assim por diante. Quando se dá conta da extensão do conhecimento que foi utilizado no nosso dia-a-dia "sem pensar a respeito" é bastante surpreendente.

É importante frisar que as fronteiras entre as análises acima não são muito claras. De fato, esta partição é artificial e tem por fim representar os aspectos cognitivos envolvidos na compreensão da linguagem natural.

5.4. Tesaurus

A WordNet (Fellbaum, C., 1998) é um "léxico relacional" disponível gratuitamente para uso *online*. Nomes, verbos, adjetivos e advérbios estão organizados como sinônimos, cada um representando um conceito lexical. Esses termos (chamados *synsets*) são ligados através de diferentes relações, tais como sinonímia/antonímia, hiponímia/hiperonímia, meronímia e troponímia (para os verbos). Também existem relações derivacionais entre categorias sintáticas – por exemplo, entre um adjetivo e o nome de que ele se deriva (*cultural-cultura*) e entre advérbios e adjetivos (*rapidamente-rápido*).

Um dos projetos de desenvolvimento de léxicos computacionais mais significativos feitos para o português brasileiro é a WordNet.BR, construída nos moldes da WordNet. A WordNet.BR, desenvolvida pelo NILC (Núcleo de Lingüística Computacional da USP), toma por base o aplicativo Thesaurus Eletrônico para o Português do Brasil – TeP (Dias-da-Silva, B. C., 2003) e possui seus itens lexicais estruturados em função das relações de sinonímia e antonímia (Dias-da-Silva, B. C. et al, 2002); (Dias-da-Silva, B. C., 2003); (Dias-da-Silva, B. C. e Moraes, H. R., 2003). Atualmente, a WordNet.Br prevê a inclusão de estrutura argumental e de esquemas de subcategorização dos adjetivos na sua base de dados (Di Felippo, A. e Dias-da-Silva, B. C., 2004), informações que não existem na WordNet.

Bases como a WordNet e a WordNetBR são construídas a partir de fontes lexicais pré-existentes. As informações não disponíveis em tais fontes são acrescidas manualmente. No caso da WordNetBR, serviram de ponto de partida alguns dicionários, como Aurélio Eletrônico e Michaelis Eletrônico, três dicionários de sinônimos e antônimos, um dicionário analógico da língua portuguesa e um dicionário específico, com *frames* de categorização e papéis semânticos, o Dicionários de Verbos do Português, de Francisco Borba (Dias-da-Silva, B. C. et al, 2002).

A WordNet, por ser o resultado da compilação de uma vasta rede de conhecimentos léxico-semânticos, é utilizada por diversos sistemas que lidam com PLN e com a construção de ontologias (Crow, L. R. e Shadbolt, N. R., 2001). Sendo assim iremos considerá-la uma referência importante na nossa

investigação. Por outro lado, o fato de a WordNet e a sua versão brasileira serem construídas manualmente as distanciam dos nossos objetivos – investigar e implementar formas automáticas de aquisição de informação lexical para préprocessamento de dados textuais, dispensando (ou minimizando) o trabalho humano.

5.5. O Modelo

As seções seguintes irão mostrar as etapas e os desafios de um processo de aquisição da linguagem natural. Durante essa seção, utilizaremos bastante o conceito proveniente da cibernética de retroalimentação sistêmica (ou "feedback"). Para (Wiener, N., 1948), a retroalimentação é um retorno de aperfeiçoamento, de otimização do sistema, que caracteriza a reorganização progressiva contra a desordem e a tendência universal da entropia em todos os níveis.

Basicamente, para ser considerado um sistema inteligente capaz de aprender é necessário um módulo de percepção, responsável por captar as informações do meio e um de retorno, responsável por julgar os impactos dessas informações (Maturana, H. e VARELA, F. J., 1980). Para perceber, é necessário uma capacidade sensível, memória e uma capacidade de aprendizado com experiências anteriores armazenadas em sua memória. Para julgar as informações é necessário uma análise destas, incluindo comparações e projeções. Um ser inteligente então é aquele capaz de melhorar seu desempenho em uma determinada tarefa utilizando para isso seu aprendizado.

Em cada etapa são processadas características morfológicas ou semânticas, isto é, a forma como a palavra é escrita ou sua classe. O aprendizado inicialmente funciona por interação com o usuário, até atingir um status de maturidade onde o sistema consegue adquirir novos conhecimentos de forma autônoma. A partir desse ponto é viável o desenvolvimento de funcionalidades utilizando seu resultado, como veremos mais adiante.

5.5.1. Definições

Para as definições assumimos que na língua só existem significados literais, de maior ou menor força. Dessa maneira, uma metáfora será um significado fraco ou pouco freqüente.

Definimos aprendizado de uma língua como o processo de armazenamento da informação sentida ou percebida na memória. Memória como um objeto que armazena informações que não são nativas do código (*hardcoded*). Sensação como um processo de captar/sentir as informações do ambiente. Percepção como uma organização do sentido e comparação com a memória. Esses conceitos serão usados nas seções seguintes, ex. Figura 27.

Definimos termo como o "átomo" da língua para este fim. A estrutura de um termo possui conteúdo, quantidade de espaços até o próximo termo, classe gramatical ou classe nativa do lexema, classe ontológica, identificador no léxico, identificador do grupo do termo no léxico, lista de relações de redundância semântica e uma lista de termos para construir as regras sintáticas.

5.5.2. Arquitetura

O modelo de armazenamento do conhecimento utilizado no algoritmo de pré-processamento está de acordo com a Figura 24. Uma tabela armazena todos os lexemas aprendidos (Léxico), outra tabela armazena as relações lexicais de redundância de informação, como acrônimos e flexões verbais. As classificações ontológicas são implementadas como um filtro ou uma visão do léxico para cada categoria.

A tabela Scan armazena as regras linguísticas utilizadas/aprendidas, *Rules* no modelo de classes da Figura 25. E, finalmente, os índices aumentam a performance dos acessos à bancos textuais.

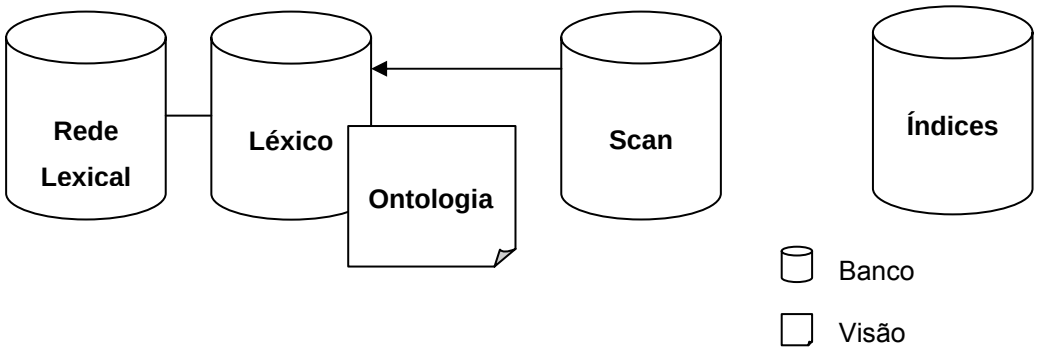

Figura 24 – Modelo de banco de dados.

O modelo de classes da Figura 25 apresenta a arquitetura orientada a objeto do modelo de pré-processamento proposto. A classe Common contém as informações e regras mais básicas das linguas ocidentais. As informações são constantes como os caracteres alfanuméricos, os separadores e diferença entre maiúscula e minúscula. As regras são processos de detecção de compostos e nomes. Para cada língua, então, essa classe é estendida de modo a atender às suas especificidades. Por exemplo, em inglês temos o formato de data MM/DD/AAA, já em português temos DD/MM/AAAA.

Todas essas constantes e regras são aplicadas pela classe *pipeline* para processar o texto. A classe *producer* é responsável por utilizar essas informações para o aprendizado automático.

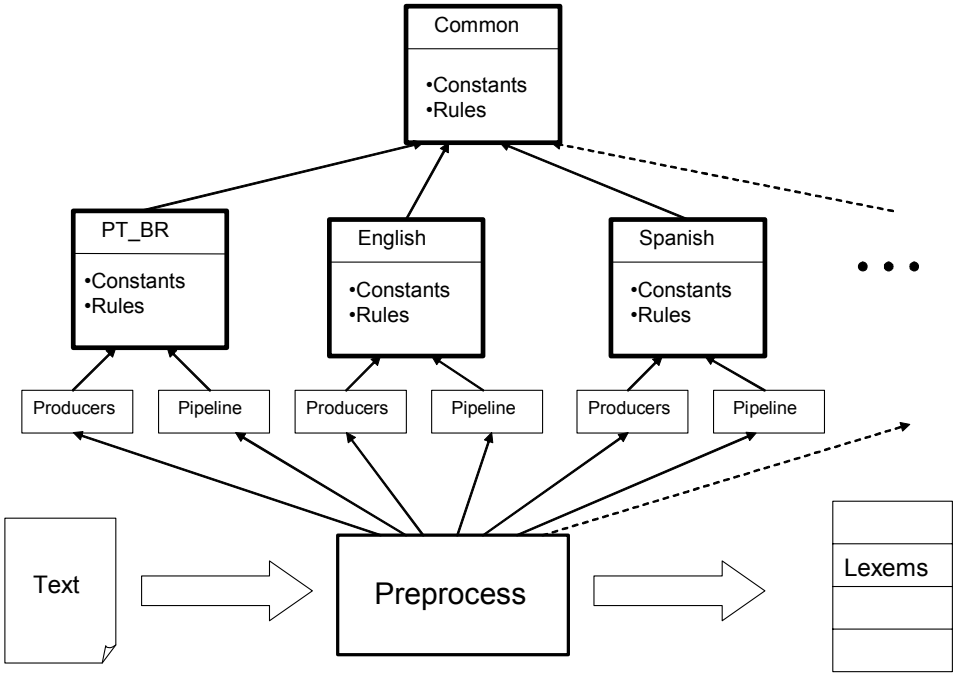

Figura 25 – Modelo de classes por orientação a objeto.

Finalmente, a classe *Preprocess* recebe o texto a ser processado como entrada e aciona o *Pipeline* e o *Producer* para pré-processamento e aprendizado respectivamente.

O modelo de classes da Figura 25 foi feito para ser extensível para outras línguas, sem impactos de arquitetura. Nessa tese, no entanto, implementamos apenas uma instância para o português.

O *pipeline* para o português é composto de sucessivos scans que vão sendo "plugados" e aplicados em linha. Cada um responsável por reconhecer uma classe de lexema diferente:

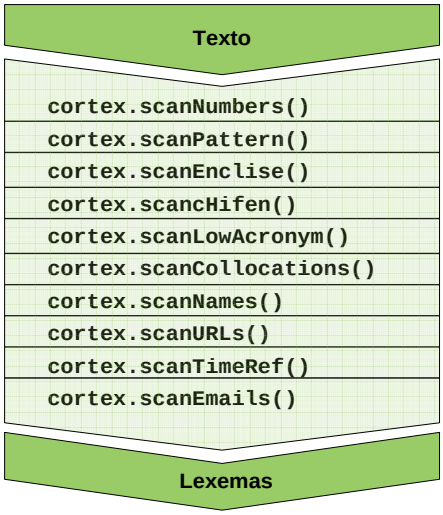

Figura 26 – (*Pipeline*) Sequência de procedimentos de reconhecimento de padrões e aprendizado de lexemas especializados em cada área do PLN.

Um texto escrito em uma determinada língua natural pode ser considerado como uma seqüência finita de símbolos arbitrários (Saussure, F., 1982). Estes símbolos são os átomos da língua, e computacionalmente são realmente indivisíveis e representados por caracteres. A seqüência de caracteres do texto contém padrões que acionam a percepção e que, por sua vez, agrupa seqüências desses símbolos formando um novo símbolo, remetendo a um outro significado.

Sob a perpectiva sistêmica, podemos diagramar o processo como na Figura 27. O texto passa primeiramente pelo tokenizador (módulo sensitivo) onde são gerados os atomos linguísticos, depois pelo reconhecimento de padrões que utiliza os conhecimentos prévios armazenados no banco de dados lexicais. Depois passa pelas regras comuns (*Common*), depois pelas regras específicas de língua e os

novos conhecimentos são inseridos no banco de dados. Os novos conhecimentos vão influenciar a percepção do próximo texto a ser processado.

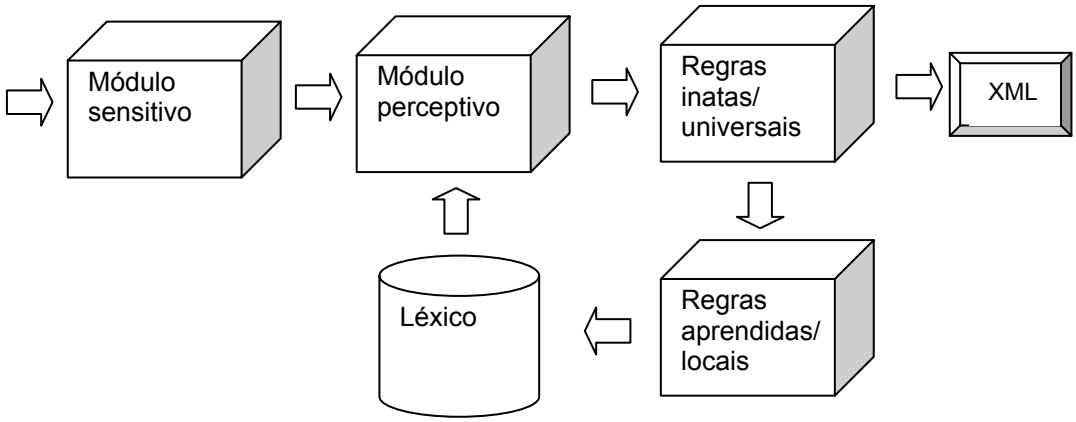

Figura 27 – Modelo de aprendizado autonomo e retroalimentado.

Continuando o processo de agrupamento, o resultado do agrupamento "primitivo" transforma esta seqüência inicial de símbolos em uma outra seqüência de símbolos, também arbitrários que podemos chamar de termo. Dentro do universo dos símbolos existem dois tipos: os alfanuméricos e os separadores. Os separadores atuam como delimitadores dos alfanuméricos. Dessa forma, os alfanuméricos são agrupados em um novo símbolo e os separadores continuam a consistir um símbolo atômico. Esta primeira etapa do processamento da linguagem natural foi definida como Sensação. Para muitos, como (Pinker, S., 2002), a sensação pode ser entendida como parte da percepção, ou uma percepção primitiva/nativa.

A função lógica deste primeiro aparato é alavancar o universo lingüístico e com isso a sua força de comunicação. Se antes tinha-se um alfabeto de 26 símbolos (26 significados possíveis para serem transmitidos), agora ter-se-ia logicamente infinitos significados para serem transmitidos ao outro, bastando para isso que se combinem, ou agrupem seqüências cada vez maiores de símbolos/caracteres. Observou-se, no entanto, que há um limite da capacidade de processamento de seqüências cada vez maiores, e, por uma questão de performance, observou-se que estas seqüências tem em média 13 caracteres de tamanho. Isso nos leva a concluir que a capacidade de comunicação de uma palavra não é infinita, mas algo da ordem de:

$$13^{26} = 9 \times 10^{28}$$

onde 13 é o tamanho da palavra e 26 é o numero de caracteres possíveis.

O processo de agrupamento continua em várias etapas. A primeira foi descrita como Sensação. As etapas estão ilustradas na Figura 28.

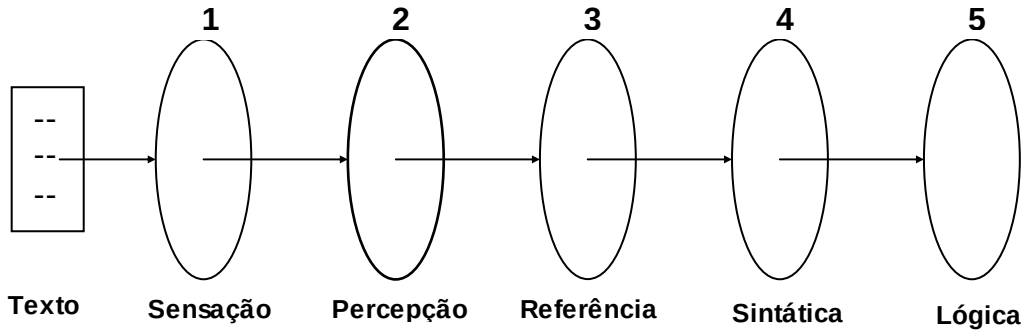

Figura 28 – Seqüências de etapas que compõe o processamento do texto.

Na segunda etapa, são agrupados símbolos em um segundo nível. A essa etapa deu-se o nome de Percepção porque é a partir daí que estes símbolos começam a ser atribuídos de traços semânticos e também, em alguns casos, começam a utilizar a memória.

Os traços semânticos são atribuídos por meio de padrões lingüísticos ainda bastante nativos. Na verdade existe uma graduação de nível perceptivo dos padrões. Essa graduação respeita a ontologia primitiva como na Figura 29. Quanto mais alto (global) o traço semântico, mais essencial e mais nativo ele é. Quanto mais baixa (local) mais elaborada e mais cultural ele é. As setas indicam os pré-requisitos procedurais para a determinação das classes.

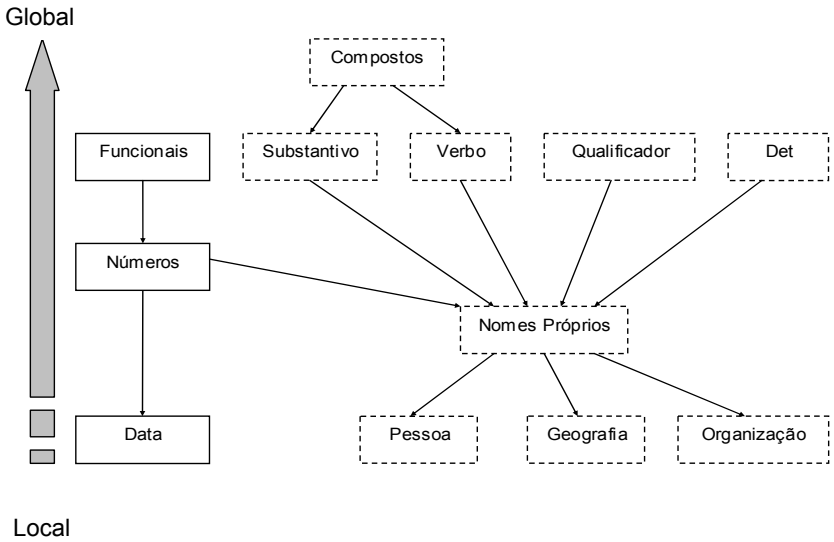

Figura 29 – Ontologia pré-definida para o processamento. As setas indicam os pré-requisitos (A→B = A é pré-requisito de B)**.**

As classes essenciais utilizam principalmente percepção de padrões morfológicos como, por exemplo, sufixos. A essas percepções é dada uma atribuição absoluta de um traço semântico para a palavra. No entanto, existem atribuições condicionadas à derivação das palavras. Vale ressaltar que no caso dos sufixos a atribuição é sempre absoluta, porém no caso da derivação pode ser tanto absoluta como relativa.

Ex.

Atribuição Absoluta

Se a palavra terminar em "oso" então existe um traço de qualificação (adjetivo).

Atribuição Absoluta Condicionada à Derivação

Se a derivação "+mente" existir então existe um traço de qualificação.

Atribuição Relativa Condicionada à Derivação

Se a derivação "+mente" existir então tem o mesmo traço que a palavra derivada.

Os procedimentos de identificação de compostos e extração de nomes próprios do texto merecem uma atenção especial e serão explorados adiante com mais detalhes.

Continuando o processo de agrupamento, ao final dessa etapa, temos um conjunto de lexemas dotados de um traço semântico segundo a ontologia estabelecida. A este conjunto daremos o nome de conhecimento local. Esses lexemas representam a memória fresca, recente e de fácil acesso. A última tarefa da percepção é persistir o conhecimento local de forma a acumular conhecimentos que poderão ser usados futuramente, isto é, aprender.

Seu aprendizado, então, ocorre da seguinte forma: se um lexema é desconhecido então ele é adicionado na memória junto com seu respectivo traço semântico (campo class do léxico); se ele já é conhecido (já existe na memória), então sua importância na língua é reforçada. Essa "importância" é a distância de sua posição para a porta de acesso externo. Isso significa que quanto maior a importância de um lexema mais próximo à porta, e quanto menor a importância mais longe da porta e mais difícil será o seu acesso. Para uma melhor compreensão, podemos fazer uma analogia desse acesso à nossa capacidade de

lembrar palavras. Se uma palavra é muito usada, ela vem rapidamente a nossa mente. Se é rara, então temos de nos esforçar para lembrar.

Na terceira etapa, o objetivo é organizar semanticamente os lexemas de forma a obedecer ao princípio da economia [Navalha de Ockwan]. Essa tarefa se traduz em perceber os termos que contêm informação redundante entre si e assim fazer sua associação, junção ou referência. A **Figura 30** ilustra o resultado desse processo que acaba por compactar a informação percebida. Os dois objetos associados não são exatamente iguais, mas têm um percentual de informação redundante que torna mais econômico o processo de representação do conhecimento.

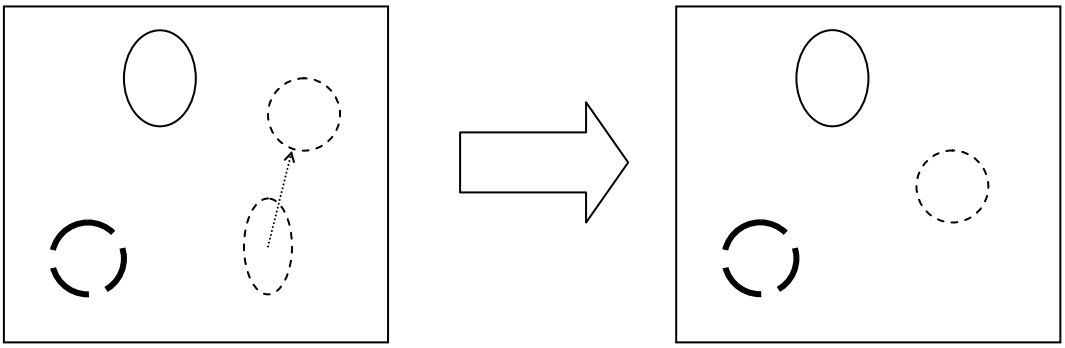

Figura 30 – Cada círculo representa um lexema percebido. A borda de cada um deles indica um traço semântico. O resultado das referências agrupa as redundâncias semânticas.

Dentre as correferências existentes, as três principais são acrônimos, truncamento de nomes e anáfora.

Acrônimo	Copom = Comitê de Política Monetária
Truncamento	Bush = George W. Bush
Anáfora	Ele = Fernando Santos

Para os acrônimos é usado um padrão sintático de proximidade e parênteses que aciona uma comparação entre as letras e o nome. Ao perceber uma semelhança os dois lexemas são ligados como referências ao mesmo significado. O processo de aprendizado é parecido: a cada acrônimo reconhecido este é memorizado. Dessa forma, se o acrônimo for usado em um texto futuro que não

contém seu significado, esse conhecimento pode ser usado como conhecimento enciclopédico. Porém, se o próprio texto indica a ligação no contexto, ela tem prioridade ao conhecimento enciclopédico.

No caso do truncamento de palavras, muito usado em nome de pessoas, os critérios de ligação são o reconhecimento de completude, onde um lexema estende o outro adicionados exatamente À restrição de traço semântico de pessoa proveniente da percepção. Além disso, a capacidade de contextualização depende dos termos lidos mais recentemente, isso significa que em um caso de ambigüidade, o mais próximo deve ter prioridade.

Finalmente, a anáfora simples procura ligar os pronomes a seus respectivos nomes ou substantivos. Essa ligação é feita através da lógica da sintaxe; então será necessário uma ajuda do módulo sintático para resolver esta referência.

A quarta etapa atua em um nível ainda superior. Ela procura por padrões na ordem em que os lexemas aparecem no texto. Estes padrões vão indicar como cada lexema está relacionado com o outro para construir o significado da frase. Com esse mapa de relacionamentos, a sintaxe contribui em muito para a extração da informação de contexto. Essa informação incorporada às etapas anteriores poderá reavaliar um agrupamento ou traço semântico. Além disso, a informação de contexto será importante também para resolver os problemas de ambigüidade lexical. Essa etapa é tão importante que a informação semântica só começa a existir a partir dela. Nesse trabalho consideraremos esta etapa importante, porém complementar. Seu processamento se baseia em estruturas computacionais como BNF (*Backus-Naur Form*) e procedimentos de análise LR (*Left-Right parsing*) exemplificado na **Figura 31**. Esses modelos computacionais podem ser usados para implementar regras gramaticais da língua. No caso do português temos o exemplo da Figura 31.

Uma gramática bem especificada, no entanto, necessita de uma quantidade de bem maior que o exemplo da Figura 31. Uma gramática totalmente especificada tende a um número infinito de regras. Porém, tirou-se proveito apenas de parte das relações sintáticas para os casos das aplicações de extração de informações.

```
frase --> sintagmaNominal, predicado, ['.'].
sintagmaNominal --> artigo, substantivo.
predicado --> verbo, sintagmaNominal.
predicado --> verbo.
artigo --> [o]; [a].
substantivo --> [menino] ; [menina] ; [cachorro] ; [gato].
verbo --> [viu] ; [chamou] ; [dormiu] ; [mordeu].
```

Figura 31 – Exemplo de uma gramática escrita sobre a de especificação de Backus-Norm-Form.

Essas regras vão ligando seqüências de termos criando novos termos com categorias semânticas instantâneas, especificamente para aquele contexto. Estas categorias contextuais são chamadas de POS (*Part of Speech*) e servem à estrutura sintática. Os padrões têm forças diferentes que fazem um ter prioridade sobre o outro quando os dois são encontrados simultaneamente. Além disso, os padrões mais longos são procurados primeiro; isso indica que eles também são mais fortes e recebem mais prioridade sobre os outros. O resultado disso é uma árvore de sucessivos agrupamentos que pretendem indicar quem está ligado a quem e qual o teor dessa ligação.

Como já foi dito anteriormente, a função da sintaxe é elaborar o contexto e agrega muita informação que pode ser utilizada pelos outros módulos nas outras etapas. Dessa forma, além do módulo de referência se utilizar dela para resolver a anáfora, o módulo perceptivo também faz uso para a classificação semântica e detecção de padrões como data, número e moeda. A necessidade desse intercambio entre as etapas/módulos já foi mencionada em (Nascimento, M. R., 1993) no módulo *early evaluat*ion. Outra influência da sintaxe está também na desambigüização dos termos, onde uma informação sintática pode alterar um traço semântico de um termo.

Com este suporte da etapa sintática às etapas anteriores foi redesenhado o diagrama inicial conforme ilustrado na Figura 22.

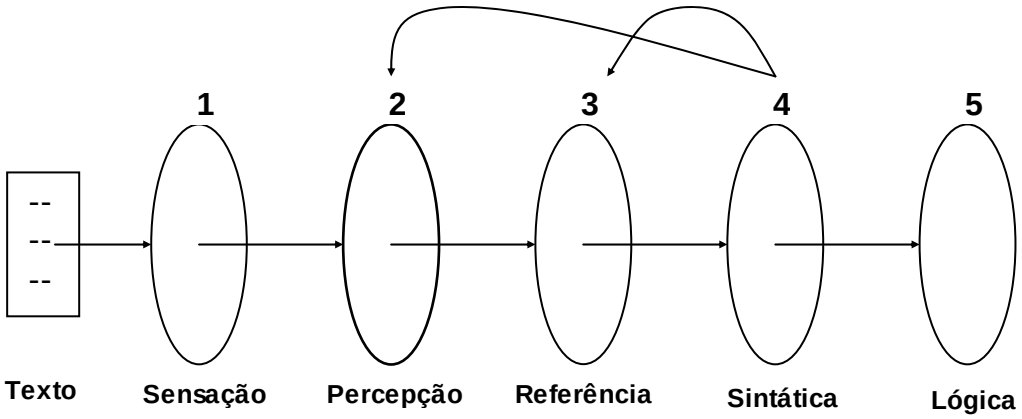

Figura 32 – Suporte da interface sintática forçando a reavaliação da percepção e referência.

A etapa sintática transforma o texto em uma lista de relacionamentos entre os elementos do texto. A maioria desses relacionamentos são feitos por termos funcionais de cunho lógico. Cada um desses termos contém uma semântica lógica específica que permitirá ao módulo de Lógica fazer inferências sobre a teia de relacionamentos. Esse conceito também conhecido como *Reasoning* (Frege, G., 1975) é bastante próximo à compreensão computacional de um texto. Isso significa que se tornará possível a verificação de coerência e coesão. Para exemplificar como isso pode acontecer, seguimos um exemplo.

"As ações da Ambev subiram com força, influenciada por rumores de que a empresa estaria prestes a fechar um acordo de produção e distribuição com a cervejaria belga Interbrew, ampliando sua atuação internacional."

Lista de assertivas lógicas:

As ações da Ambev subiram com força. **(conhecimento local)**
As ações da Ambev foram influenciadas por Z. **(conhecimento local)**
Z = "rumores de que X" **(conhecimento local)**
X = "a empresa estaria prestes a fechar um acordo com a Interbrew". **(conhec. local)**
O acordo é de produção e distribuição. **(conhecimento local)**
a Interbrew é uma cervejaria belga. **(conhecimento local)**
a Ambev é uma empresa. **(memória)**

empresa = Ambev **(anáfora)**
X = "a Ambev estaria prestes a fechar um acordo com a Interbrew". **(conhec. local)**

A seguir falaremos sobre como o sistema identifica, aprende e reconhece compostos da língua. Os compostos representam um novo paradigma de processamento de língua, onde as fronteiras dos objetos linguísticos são mais difusas.

5.5.3. Compostos

Nesse modelo partimos de uma teoria composicional do significado, onde o significado de uma seqüência de palavras é formado pelas palavras que a constituem. No entanto, esta não é sempre uma verdade. Existem seqüências de palavras que nos remetem a um significado distinto do significado das partes. Essas seqüências surgem devido ao uso intenso dessas palavras conjuntamente formando uma associação forte entre elas e cristalizando seu significado. Com tempo o significado de cada parte pode flutuar e se descolar do significado conjunto cristalizado. (Saeed, J. L., 1997)

A identificação de compostos foi feita usando como base um corpus jornalístico que contém aproximadamente 8 milhões de palavras (é um corpus dinâmico que cresce em média 300.000 palavras por dia). O sistema filtra combinações que tenham como parte do bigrama pronomes pessoais, pronomes demonstrativos e nomes próprios.

Depois de separar as palavras do texto, o sistema vai rankear as combinações de acordo com um critério estatístico inspirado no tradicional teste de hipótese de *t-student*. O objetivo é selecionar seqüências de palavras em que a ocorrência de uma palavra seja condicionada à ocorrência da outra palavra, criando assim o composto. (Oliveira, M. F. et al, 2004)

O princípio adjacente ao teste de hipótese vem da comparação (contraste) entre a freqüência de ocorrência de uma das palavras do composto e a freqüência esperada de ocorrência da mesma palavra. Esse contraste é resolvido através do cálculo da distância entre esses dois valores. Existe um fator limite da distância para garantir uma certeza de 95%.

$$\frac{\bar{x}}{\sqrt{\frac{s^2}{N}}} - \frac{\mu}{\sqrt{\frac{s^2}{N}}}$$

onde $\bar{x}$ é o valor real e μ é o valor esperado, N é o tamanho da amostra e s é a covariância.da amostra.

No caso do composto bigrama, temos uma estrutura do tipo A B (ex. impressão digital, impressão=A e digital=B) o objetivo é saber se a associação entre as duas palavras do composto é significante. Fazemos o teste de hipótese para saber se a probabilidade de a palavra A ocorrer conjuntamente com a palavra B é a mesma que a probabilidade de a palavra A ocorrer. É o mesmo conceito de testar a dependência entre duas variáveis.

$$P(A/B) > P(A) \text{ ou}$$
$$P(B/A) > P(A)$$

De acordo com nosso modelo, existe um agente responsável por extrair os elementos do texto e alimentar o léxico. O léxico armazena os itens lexicais já lidos e duas freqüências. Até a data de 15/09/2005 já haviam 16.677.358 palavras provenientes de textos processados.

Existe um outro agente responsável pelo calculo dos compostos. Esse agente serializa o corpus (de acordo com os itens lexicais) e armazena eles em outra base, que é uma sub-amostra do total lido pelo sistema. As freqüências dos compostos candidatos são calculados sobre essa sub-amostra. Fizemos isso porque a tarefa demanda uma grande capacidade de armazenamento e processamento de máquina que cresce de forma exponencial.

De qualquer maneira, como a aplicação de um teste de hipótese demanda um grande esforço computacional e mesmo assim não oferece garantia de 100%, um outra solução foi encontrada de forma à simplificar os cálculos.

Na maioria dos trabalhos, apenas o conceito de teste de hipótese é usado para calcular as distâncias e ordenar os candidatos a compostos (Manning, C. e Schutze, H., 1999). Na prática, analisando esses resultados vemos que podemos reduzir os cálculos com atalhos que aproximam o resultado final de um teste de hipótese completo. Dessa forma podemos aumentar a performance computacional e viabilizar esse cálculo para grandes bases de dados.

O cálculo se reduz para:

$$\frac{f_c}{f_{n1} + f_{n2}} \quad f > 0$$

onde f_c é a freqüência de coocorrência e f_{ni} is a freqüência de ocorrência of i.

Adicionalmente, uma segunda aproximação consiste na consideração no calculo das probabilidades marginais f_{ni} não da sub-amostra, mas de todo o corpus (informação pré armazenada no léxico). Vale lembrar que a conjunta f_c continua a ser calculada sobre a sub-amostra. Os resultados mostram que, apenas para ordenar os compostos candidatos não se perde muita informação importante. O problema passa a ser determinar o ponto crítico do teste, que é alterado pelas simplificações. Para isso, utilizou-se para a expertise humana de juízes que manualmente classificaram os compostos. Quando a quantidade de compostos errados atingiam 10% do total, arbitrariamente fixou-se o ponto crítico (análogo à precisão de 90% de um teste de hipótese).

O resultado de toda essa análise é o agrupamento destas palavras em lexemas distintos, criando assim uma nova entrada lexical para ela. Veja o exemplo da tabela:

Identificação	Expressão linguística
ID1	Impressão
ID2	Digital
ID3	Impressão digital

Uma vez agrupado um composto como uma nova entrada no léxico ele passa a se comportar como um item lexical igualmente aos outros. Dessa forma, o aprendizado dinâmico permite que construções com n palavras sejam geradas, mais detalhes se encontram em (Aranha, C. et al, 2005). Felizmente o uso regular de uma seqüência de palavras decresce com a quantidade de palavras. Dessa forma as associações mais fortes são de 2 ou 3 palavras.

5.5.4.
Nomes Próprios

Embora a identificação dos nomes próprios tenha como principal evidência a ocorrência da letra maiúscula no início da palavra, essa tarefa está longe de ser a mais fácil. O principal fator é que essa classe concentra 50% da novidade de um texto, isto quer dizer que não é possível fazer uma lista dos nomes próprios existentes, ao contrário das outras classes. Em (Borguraev, B. e Pustejovsky, J., 1996) podemos encontrar uma descrição de diversos problemas que são encontrados normalmente no processamento de nomes próprios, assim como algumas dicas de como resolver alguns casos. Abaixo encontram-se alguns problemas com resultados bastante satisfatórios (Freitas, M. C. et al, 2005):

Letra maiúscula no início da frase: “Governador faz viagem internacional.”
Abreviação no meio do nome: “Philip B. Morris”
Conexão por palavras funcionais: “Juiz Nicolau dos Santos Neto”
Titularidade: “Presidente da Câmara dos Vereadores Alcides Barroso”

Depois de reconhecido todos os nomes próprios e resolvidas as correferências, seguimos a ontologia estabelecida e passamos para uma segunda etapa de sub-categorização desses NPs nas categorias Pessoas, Lugares e Organizações. Para a classificação em cada uma dessas categorias foram utilizados recursos das interfaces de percepção e sintática.

5.5.5.
Sintaxe

A resolução da sintaxe se apresenta de duas formas, cada uma delas em seqüências de processos paralelos. Um processo é capaz de lidar com grandes volumes de dados e orientado para a extração de conhecimento e aprendizado. O outro é orientado à performance (processa em tempo real um pequeno volume de texto) e não está conectado com a interface de aprendizado.

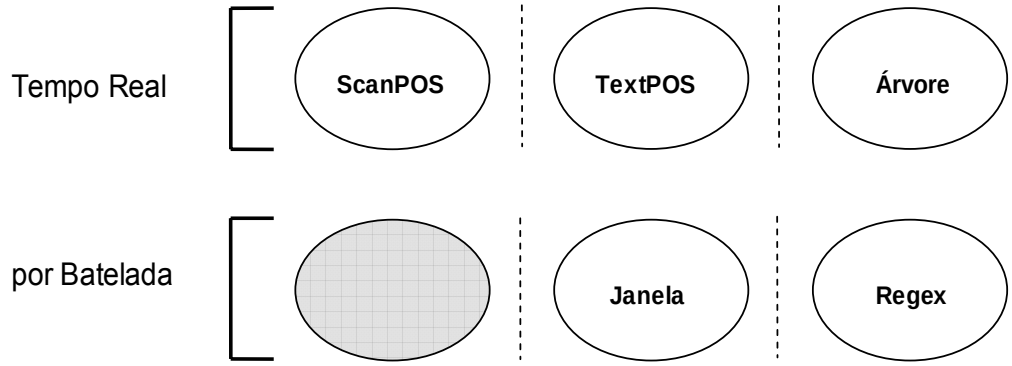

Figura 33 – Diferentes módulos para resolver a sintaxe. Estão separados por fatores de performance e forma de execução.

O processo em tempo real contém os módulos de ScanPOS que auxilia na percepção de todas as categorias através de metalinguagem. O módulo TextPOS que insere a palavra no contexto sintático, podendo recategorizar alguma categoria dada a priori. E finalmente há o módulo de geração da árvore sintática, onde o sistema traça todos os relacionamentos entre todos os objetos do texto.

O processo por batelada está associado ao aprendizado do sistema. No módulo janela Scripts de SQL são acionados periodicamente para aprender a classificação de novas palavras assim como a metalinguagem associada. No módulo de Regex, o sistema usa os bem conhecidos motores de expressões regulares para extrair informações específicas dos textos como cargos de pessoas em empresas.

O aprendizado se faz de forma síncrona entre metalinguagem e termos, como em um processo de otimização de duas varáveis por derivada parcial. Aprendemos a variável A em função de um condição inicial de B, com o resultado fixamos A e melhoramos B e assim por diante. Esse processo cibernético serve de apoio ao sistema em tempo real, como se fosse uma manutenção continuada ao sistema.

5.6. Representação do Documento

Após o pré-processamento dos textos, é gerada uma lista de objetos lingüísticos (lexemas). Cada documento é composto de uma seqüência de

parágrafos, que por si são compostos de uma seqüência de frases, que por si, ordenam os objetos e as entidades. Essa lista inclui as entidades mencionadas correferenciadas. Um histograma dessa lista transforma a representação em uma estrutura *bag-of-lexems* normalizada.

5.6.1. Formato de Armazenamento

Ao longo dessa seção discutiremos metodologias que permitem o enriquecimento de dados textuais através da criação e uso de metadados agregados aos documentos. O modelo mais usado hoje é o XML (eXtended Markup Language). As vantagens para este formato são inúmeras para a mineração de dados (Reis, M., 2005).

Estes metadados são também denominados "*tags*", e estão relacionados a um documento ou a trechos do texto contidos neste. Ilustraremos especialmente as vantagens do uso de *tags* para a recuperação de informações em grandes repositórios de documentos, quando comparado às técnicas mais comuns de busca.

Existem muitos tipos de *tags* (ou metadados) que podem ser usadas para qualificar documentos ou partículas de texto. A seguir apresentamos alguns tipos de tags úteis no auxílio à análise do conteúdo de grandes repositórios de documentos.

Em uma definição simplista, metadados são "dados sobre os dados". São comumente usados para enriquecer dados, provendo informações específicas sobre os registros, colunas ou células de um repositório de dados, como por exemplo:

- A origem do dado
- Regras de transformação para o dado
- O formato usado para representar o dado

5.6.1.1. Tags de Categoria

As *tags* de categoria são associadas a um documento e fornecem informações quanto ao tipo do documento. Categorias são geralmente atribuídas

ao documento levando em conta seu local de origem, seu tipo, formato, escopo, etc. Alguns exemplos de categorias são mostrados na Figura 34.

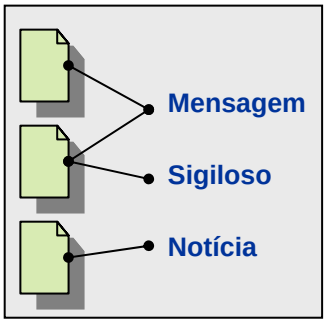

Figura 34 – Exemplos de tags de categoria

Um mesmo documento pode conter mais de uma *tag* de categoria. Por exemplo, ser qualificado como um e-mail e também como um documento sigiloso. *Tags* de categoria são especialmente úteis para limitar ou filtrar buscas. Torna-se possível, por exemplo, realizar uma busca limitada aos documentos não sigilosos da organização, ou fazer uma busca somente por documentos produzidos pelo departamento jurídico de uma organização.

5.6.1.2. Tags de Contexto

Tags de contexto são de um tipo semelhante às *tags* de categoria, e qualificam um documento. A diferença é que as *tags* de contexto são utilizadas para indicar um ou mais assuntos aos quais o documento se refere, como por exemplo “política”, “esportes”, “informática” ou “desenvolvimento sustentável”. Embora pareça um trabalho que só possa ser realizado por seres humanos capazes de entender e qualificar os assuntos tratados por um texto, começam a surgir agentes inteligentes capazes de realizar esta tarefa automaticamente com uma acurácia razoável.

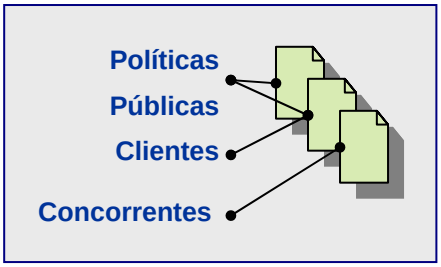

Figura 35 – Exemplo de Tags de Contexto.

Tags de Contexto são muito úteis na filtragem de grandes quantidades de documento. Imagine por exemplo uma organização que atue na extração e refino de petróleo. Seria possível desenvolver um sistema capaz de baixar automaticamente notícias das principais mídias disponíveis na web e separar aquelas que dizem respeito ao mercado de petróleo, diminuindo dramaticamente a quantidade de material a ser lido diariamente pelos tomadores de decisão. Seria possível ainda separar as notícias que falam sobre extração, as que tratam de refino e as que tratam de políticas públicas regulatórias, direcionando conteúdo específico para departamentos distintos.

5.6.1.3. Tags de Função

Diferente dos tipos de *tags* citados anteriormente, *tags* de função qualificam elementos internos de um documento. Eles informam o papel de uma partícula de texto dentro do documento. Por exemplo, um parágrafo de um artigo pode ser qualificado como sendo seu abstract, uma linha pode ser qualificada como o título, e uma outra partícula pode indicar o autor do documento.

Tags de Função possibilitam especificar melhor o escopo de uma busca. Assim, em um cenário onde antes se faria uma busca por palavra-chave ao longo de todo o conteúdo de cada documento, pode-se agora restringir a busca por um campo específico. Por exemplo, em uma base literária, pode-se buscar por referências a “Machado de Assis”, somente onde este termo aparece como autor da obra. Buscas por escopos restritos aceleram muito o tempo de resposta ao usuário.

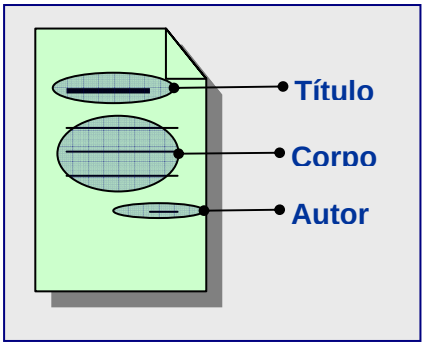

Figura 36 – Exemplos de Tags de Função.

A utilização de heurísticas para normalização de termos também pode possibilitar a migração de bases desestruturadas de documentos para um modelo relacional. Este tipo de heurística associa termos que na verdade têm o mesmo significado como ocorrências de um mesmo termo. Por exemplo, o termo "Assis, Machado de" e "Machado de Assis" podem ser reconhecidos como referências a um mesmo autor. Isso torna viável e eficiente recuperar, por exemplo, todas as obras escritas por um mesmo autor.

Um tipo específico de *tags* de função são as *tags* de resumo. Estas *tags* atribuem resumos a documentos extensos. Estas *tags* podem ser estáticas ou dinâmicas. Resumos estáticos utilizam diversas técnicas para tentar compor uma partícula de texto mais semanticamente completo em relação ao texto original. Esta partícula também deve ser construída suficientemente de acordo com regras da língua, de forma que as construções façam sentido e o resumo seja de leitura fácil.

Resumos gerados dinamicamente podem levar em consideração critérios e parâmetros variáveis. Por exemplo, na apresentação de resultados de uma busca, pode ser mostrado junto a cada ocorrência um resumo que depende dos termos sendo pesquisados. *Tags* de Resumo melhoram a manuseabilidade de uma quantidade grande de documentos, ou de documentos muito extensos.

5.6.1.4. Tags de Estrutura

Dentro dos blocos de texto especificados pelas tags de função, temos as tags de estrutura. Essas tags organizam as frases dentro do documento, mostrando como elas estão relacionadas. Para isso, as tags de estrutura estão atenta aos

separadores e indicam quando é um início de parágrafo, quando é o fim da frase, quando é um ponto final, quando temos uma citação entre aspas, ou quando temos uma correferência.

Essa estrutura pode ser vista também na Figura 38 e Figura 39, onde |PS| é o início do parágrafo, |FS| é o início da frase, |QS| é o início de declaração em aspas. |PE|, |FE| e |QE| são os respectivos fins.

5.6.1.5. Tags Descritivas

Tags descritivas são o tipo mais complexo de metadado usado em documentos texto. Estas *tags* fornecem informações quanto à semântica (significado) e tipo de palavras, termos ou frases no texto. O Conjunto de *tags* descritivas a ser usado varia de acordo com o domínio em que estamos trabalhando. Por exemplo, em um domínio de negócios, "companhia", "indústria", "executivo", "produto" são exemplos de *tags* que devem ser úteis. No domínio de uma aplicação médica, exemplos de *tags* úteis podem ser "medicamento", "doença", "sintoma" ou "órgão". Justamente por esse motivo, o dicionário de regras usadas para identificar as *tags* devem ser diferentes para cada domínio.

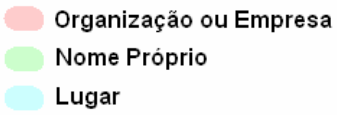

Figura 37 – Exemplos de Tags Descritivas.

O uso de *tags* descritivas permite buscas mais precisas no aspecto semântico. Por exemplo, pode-se buscar por todas as ocorrências de termos do tipo "executivo" que aparecem em documentos onde o termo "Petrobras" é citado. Obter-se-á uma relação aproximada de executivos que participaram ou que de alguma forma estiveram relacionados com a empresa. Sem o uso de *tags* descritivas, seria muito difícil obter um resultado deste tipo.

Observação final: Essa seção já se tornou um artigo que será submetido em breve.

5.6.2. Exemplos

Uma amostra do resultado do módulo de reconhecimento de entidades utilizando representação por tags pode ser visto na Figura 38 e na Figura 39. As cores são marcações das diferentes rótulos das *tags* descritivas.

|PS||FS|A mediana das projeções de mercado para a inflação do IPCA em 2004 voltou a subir, na semana passada, fechando a sexta-feira em 6,14%, segundo pesquisa divulgada ontem pelo Banco Central. |FE||FS|Esse era um patamar que não se via desde novembro do ano passado. |FE||FS|O mercado também ajustou, de 15,5% para 15,75% ao ano, a sua previsão para o nível da taxa básica de juros no final de maio. |FE||FS|A expectativa, portanto, é de que, em maio, o BC voltará a reduzir a Selic em 0,25 ponto percentual.|FE| |PE|

Figura 38 – Exemplo de reconhecimento de entidades. Instituições em vermelho, índice em verde e tempo em roxo.

|PS||FS| Antecipando-se à viagem do presidente Luiz Inácio Lula da Silva à China, em maio, uma delegação com alguns dos mais importantes dirigentes de empresas chinesas desembarca amanhã, no Brasil, para maratona de visitas a empresas brasileiras e encontros com autoridades, entre elas o próprio Lula. |FE||FS||QS|"|FS|É uma visita de representantes das empresas chinesas mais importantes, não só para aprofundar as relações comerciais e de cooperação, mas para discutir aumento de investimentos na China e no Brasil|FE|"|QE|, resume o secretário comercial da embaixada chinesa no Brasil, Ki Lin Fa.|FE| |PE|

Figura 39 – Exemplo de reconhecimento de entidades. Nomes de pessoas em laranja e lugares em azul.

A identificação e classificação automática de NEs, embora ainda sem resultados expressivos para o português brasileiro, tem se desenvolvido nos últimos anos como *Automatic Content Extraction* – ACE (para a língua inglesa) e HAREM (para a língua portuguesa).

O reconhecimento das EM do texto compõe a etapa de pré-processamento dos dados. A seguir veremos algumas técnicas freqüentemente usadas de mineração de texto.