Hilário Seibel Júnior

Recuperação de informações relevantes em documentos digitais baseada na resolução de anáforas

Vitória - ES, Brasil

16 de julho de 2007 (Data da defesa da Dissertação)

Hilário Seibel Júnior

Recuperação de informações relevantes em documentos digitais baseada na resolução de anáforas

Dissertação para obtenção do título de Mestre em Informática apresentada ao Programa de Pós-Graduação em Informática da Universidade Federal do Espírito Santo.

Orientador:
Sérgio Antônio Andrade de Freitas

Programa de Pós-Graduação em Informática
Departamento de Informática
Centro Tecnológico
Universidade Federal do Espírito Santo

Vitória - ES, Brasil

16 de julho de 2007 (Data da defesa da Dissertação)

Dados Internacionais de Catalogação-na-publicação (CIP)
(Biblioteca Central da Universidade Federal do Espírito Santo, ES, Brasil)

Seibel Júnior, Hilário, 1981-

S457r Recuperação de informações relevantes em documentos digitais baseada na resolução de anáforas / Hilário Seibel Júnior. – 2007.

94 f. : il.

Orientador: Sérgio Antônio Andrade de Freitas.

Dissertação (mestrado) – Universidade Federal do Espírito Santo, Centro Tecnológico.

1. Recuperação da informação. 2. Anáfora (Lingüística). 3. Busca em arquivo. 4. Processamento de Linguagem Natural (Computação). 5. Documentos digitais. I. Freitas, Sérgio Antônio Andrade de. II. Universidade Federal do Espírito Santo. Centro Tecnológico. IV. Título.

CDU: 004

Dissertação de Mestrado sob o título *"Recuperação de informações relevantes em documentos digitais baseada na resolução de anáforas"*, defendida por Hilário Seibel Júnior e aprovada em 16 de julho de 2007 (Data da defesa da Dissertação), em Vitória, Estado do Espírito Santo, pela banca examinadora constituída pelos professores:

Prof. Dr. Sérgio A. A. de Freitas
Orientador

Prof. Dr. Orivaldo de Lira Tavares
Universidade Federal do Espírito Santo

Dr. Emiliano Gomes Padilha
Universidade Federal do Rio Grande do Sul

Resumo

Os métodos de recuperação de informação (RI) tradicionais (Van RIJSBERGEN, 1979; BAEZA-YATES; RIBEIRO-NETO, 1999) se baseiam essencialmente na contagem da freqüência em que as palavras aparecem em um documento, sem apresentar soluções para que o conteúdo semântico do discurso seja interpretado. Por não interpretarem o documento analisado, tais métodos podem deixar de considerar informações importantes a seu respeito. Uma solução para contornar esse problema, citada em (SALTON; MCGILL, 1986), é utilizar o Processamento de Linguagem Natural (PLN) na recuperação de informação. Uma aplicação do PLN é o processamento de anáforas.

Anáfora (CARTER, 1987; BEAVER, 2004) é um fenômeno lingüístico no qual uma entidade introduzida *a priori* é referenciada posteriormente em outra frase através de alguma expressão lingüística, tal como em "Valentina nasceu em São Paulo. A menina é do signo de Peixes.". A resolução das anáforas identifica que o termo *menina* presente na segunda frase do texto referencia a entidade introduzida no discurso pelo termo *Valentina* da primeira frase. Isso permite afirmar que *Valentina* é mais relevante em relação ao texto do que se tal referência não ocorresse. Freitas propõe em (FREITAS, 2005) um método para resolver as anáforas de um documento através da criação de uma estrutura que permite acompanhar as entidades que se mantêm em evidência ao longo do discurso. Essa estrutura armazena informações que podem ser aproveitadas por um método de recuperação de informação.

Esta dissertação propõe uma metodologia computacional para recuperar informações relevantes a partir da resolução das anáforas de um documento, visando aumentar a qualidade dos resultados de uma *query*. A resolução de anáforas permite identificar exatamente a quantidade de vezes que cada entidade é referenciada em um discurso, expondo entidades e ligações que podiam estar obscuras no discurso original. Essa informação torna possível decidir se certa entidade é mais relevante que outra no documento, dando mais enfoque ao que o autor escreveu. Dessa forma, os documentos relevantes recuperados são classificados pela quantidade de informação que apresentam a respeito dos termos buscados, e não apenas pela localização e/ou quantidade de ocorrências de tais termos. Este trabalho também permite identificar, através da estrutura gerada pelo processamento de anáforas, os termos sinônimos (aqueles que referenciam uma mesma entidade). Se o documento indica que dois termos são sinônimos, a busca por um deles retorna o mesmo resultado que a busca pelo outro, aumentando ainda mais a qualidade dos resultados de uma *query*. Este trabalho apresenta os detalhes da metodologia proposta: as medidas utilizadas para calcular a relevância de um termo em relação ao documento interpretado pelo processamento de anáforas, os procedimentos necessários para a realização de uma *query*, o protótipo implementado e a análise de sua complexidade de tempo. Além disso, são avaliadas as características desta abordagem que a diferenciam dos métodos tradicionais em relação à qualidade dos resultados obtidos.

Abstract

Traditional information retrieval (IR) methods are essentially based on counting the frequency that words appear in a document, presenting no solutions for interpret the semantic content of the discourse (Van RIJSBERGEN, 1979; BAEZA-YATES; RIBEIRO-NETO, 1999). As they do not interpret the analyzed documents, such methods may not consider important information about them. For this reason, Natural Language Processing (NLP) is suggested to improve information retrieval. An application of NLP is anaphora processing.

Anaphora (CARTER, 1987; BEAVER, 2004) is a linguistic phenomenon that contains a coreference of one expression with its antecedent. The antecedent provides the information necessary for the expression's interpretation. This is often understood as an expression "referring" back to the antecedent, as in the following example: "Hilário tried to speak in English, but he has a strong foreign accent.". Anaphora resolution identifies that the pronoun *he* in the second phrase is a reference to the entity introduced in discourse by the term *Hilário*. Therefore it is possible to affirm that the term *Hilário* would be less relevant in relation to the text if such reference did not occur. Freitas proposes in (FREITAS, 2005) a method to resolve anaphora by creating a structure which allows us to recognize the entities that keep in evidence throughout a discourse. This structure stores some information that can be used by a method of information retrieval.

This thesis proposes a computational methodology to retrieve relevant information from the anaphora resolution of a document, in order to increase the quality of the results of a query. Anaphora resolution allows to accurately identify the amount of times that each entity is referred in a discourse, showing entities and linkings that could be obscure in the original text. This information makes it possible to decide if certain entity is more relevant than another one in the same document, analysing what the author really wanted to transmit in his writing. So, the relevant retrieved documents are classified by the amount of information about the searched terms that they present, and not only by the localization and/or amount of occurrences of such terms. This work also allows one to identify, in the structure generated by anaphora processing, the synonymous terms (those ones that are references to a same entity). If one document indicates that two terms are synonymous, then searching for one of them should return the same result of searching for the other one, increasing the quality of the results of a query.

This work presents details of the proposed methodology: the measures used to calculate the relevance of a term in a document (which was previously interpreted by anaphora resolution), how to execute a query, the prototype that was implemented and its time complexity analysis. Moreover, we analyse the characteristics of this approach that differentiate it from traditional methods of information retrieval in terms of quality of their results.

Dedicatória

Dedico este trabalho a toda a minha família (mamãe, papai, Veruska, Letícia, Valentina, avôs, avós, tios, tias, primos e primas), por tentarem sempre e incessantemente me fazer uma pessoa feliz (antes mesmo que eu pudesse aprender a gostar deles) e pela boa educação (moral e escolar) que sempre fizeram questão de me dar. À minha mãe, por me dar amor, paz e calma. Ao meu pai, por todo apoio e infra-estrutura, por nunca ter deixado nos faltar um bom jornal e uma revista. À Veruska, por me incentivar a querer sempre ir mais além e por ter me dado a sobrinha mais *gotosa* desse mundo. E à Letícia, por me fazer rir de coisas bobas até quando quero chorar. Cada um, a seu modo, deu um pouco de si para que eu chegasse até aqui.

"Educai as crianças e não será preciso castigar os homens."
Pitágoras

"Os pais somente podem dar bons conselhos e indicar bons caminhos,
mas a formação final do caráter de uma pessoa
está em suas próprias mãos."
Anne Frank

À Débora, minha namorada linda, por me fazer cada dia mais feliz. Por sempre me ajudar, não só neste trabalho, mas em todo o meu mestrado. E por estar comigo todos os dias, o dia todo, sem brigas e sempre com carinho, respeito e amor.

"The greatest thing you'll ever learn
is just to love and be loved in return."
Eden Ahbez

"Se eu não puder encontrar a felicidade sem fim
Quero ao menos você bem pertinho de mim."
Antonio Villeroy

"Fundamental é mesmo o amor, é impossível ser feliz sozinho."
Tom Jobim

Aos meus irmãos Rafael e Thiago, que me mostraram que posso fazer alguém gostar de mim *mesmo* sendo eu mesmo. Obrigado pelos *triálogos*, pelos açaís, pelas risadas, pelas viagens, pela amizade. Espero, de verdade, que a distância só nos aproxime cada vez mais.

"A amizade é um amor que nunca morre."
Mário Quintana

"Aqueles que passam por nós, não vão sós, não nos deixam sós.
Deixam um pouco de si, levam um pouco de nós."
Antoine de Saint-Exupéry

E aos amigos que fiz na faculdade e que nunca mais sairão da minha vida: Mariella, minha parceira e afilhada, por sempre me lembrar de deixar os computadores e o trabalho um pouco mais de lado; Márcio, meu novo afilhado, por me mostrar que sempre posso ser menos preguiçoso e aprender um pouco mais; Jociel, que me mostra que sempre posso ser um pouco mais paciente e menos afobado; Alex, que me mostra que sempre posso arriscar um pouco mais na vida; Diogo, que me fez conhecer minha namorada e que me ensinou a programar e a gostar de computação; e Daniel, um grande amigo, parceiro, orientando e sabe-se lá o que mais, por todas as conversas intermináveis sobre nada de madrugada, pelos conselhos sempre meio malucos, pela lealdade, pela sinceridade e pela amizade 24 horas e 0-800.

"Tenho amigos para saber quem sou."
Oscar Wilde

"Good friends are hard to find,
harder to leave and impossible to forget."
Autor desconhecido

Amo todos vocês.

♡

Agradecimentos

Agradeço aos professores que me ajudaram a chegar até aqui, desde aqueles da época do Jardim até os da falculdade.

Ao tio Raul, meu primeiro (des)orientador, que sempre fez questão de ensinar muito mais do que sou capaz de absorver, que me incentivava a duvidar de tudo que já foi provado e que, mesmo longe, nunca negou ajuda quando gritei por socorro.

"A dúvida é o principio da sabedoria."

Aristóteles

"A parte que ignoramos é muito maior que tudo quanto sabemos."

Platão

E ao tio Sérgio, por ter me aturado todo esse ano, e por ter confiado a mim esse projeto, que eu sei que ele não entregaria nas mãos de qualquer um. Não sei ainda o motivo de tanta confiança nem como agradecê-lo, só espero não decepcioná-lo.

"O que a lagarta chama de fim de mundo, o mestre chama de borboleta."

Richard Bach

"A gratidão é o único tesouro dos humildes."

William Shakespeare

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1 Introdução

"Pessoas inteligentes falam sobre idéias.
Pessoas comuns falam sobre coisas.
Pessoas mesquinhas falam sobre pessoas."
Sócrates

Este capítulo apresenta a motivação e o objetivo deste trabalho, além de uma visão geral do que se encontra nesta dissertação.

1.1 Introdução

O volume de informações ao alcance das pessoas tem crescido rapidamente, tornando cada vez mais necessária a criação de ferramentas de busca por informações relevantes nos documentos disponíveis. Na *internet* existem diversos *sites* dedicados a esse tipo de busca (Google™, Yahoo!®, Altavista™ *etc*). Grandes empresas, que possuem vastos bancos de documentos digitais, também utilizam cada vez mais mecanismos que permitam a busca por certos documentos nesses bancos. O mesmo acontece com bibliotecas e seus acervos de livros que crescem constantemente.

Desde o seu surgimento, a Ciência da Informação vem estudando métodos para o tratamento automático da informação. A área que envolve a aplicação de métodos computacionais no tratamento e recuperação da informação é chamada de *Recuperação de Informação* (RI) (Van RIJSBERGEN, 1979; BAEZA-YATES; RIBEIRO-NETO, 1999). O termo em inglês *Information Retrieval* foi criado por Calvin Mooers em 1951. Tal termo ainda é bastante questionado pelo fato de que os sistemas não "recuperam" informação, e sim documentos ou referências cujo conteúdo poderá ser relevante para a necessidade de informação do usuário. Ferneda define em (FERNEDA, 2003) que o processo de recuperar informações consiste em identificar, no conjunto de documentos de um sistema, quais atendem às necessidades de informação dos usuários.

Os primeiros sistemas de RI se baseavam apenas na contagem da freqüência em que as palavras apareciam nos documentos e na eliminação de palavras que reconhecidamente apresentam pouca relevância (como *de*, *para*, *com*, *pois*). Esses sistemas tradicionais calculariam a mesma relevância para o termo *sabiá* em relação aos dois textos a seguir:

Texto 1.1 "Sabiá lá na gaiola fez um buraquinho.
Voou, voou, voou, voou.
E a menina que gostava tanto do bichinho
Chorou, chorou, chorou, chorou."
(Sabiá Lá na Gaiola, Mário Vieira/Hervê Cordovil)

Texto 1.2 "Minha terra tem palmeiras, onde canta o Sabiá;
As aves, que aqui gorjeiam, não gorjeiam como lá.
Nosso céu tem mais estrelas, nossas várzeas têm mais flores,
Nossos bosques têm mais vida, nossa vida mais amores."
(Canção do Exílio, Gonçalves Dias)

Apesar de o termo *sabiá* aparecer apenas na primeira frase de ambos os textos, pode-se perceber que o primeiro introduz mais informação sobre *sabiá* do que o segundo. Isso acontece porque o ser humano é capaz de interpretar no Texto 1.1 que o *sabiá*, além de ter feito um buraco na gaiola, também voou. Além disso, consegue perceber que o *bichinho* do qual a *menina* gostava no texto também se trata do *sabiá*. No Texto 1.2, *sabiá* aparece apenas numa breve descrição do ambiente em que se passará o restante do texto. A interpretação no trecho citado da canção de Mário Vieira e Hervê Cordovil é feita através do entendimento humano de que houve uma elipse do termo *sabiá* na segunda frase do texto e uma substituição do termo por outro com o mesmo valor semântico na terceira frase (o termo *bichinho*). Essa interpretação, entretanto, não é captada pelos mecanismos de buscas que se baseiam na contagem das palavras.

1.2 Motivação

Os métodos de RI tradicionais deixam de considerar informações importantes a respeito dos documentos analisados por não interpretarem-nos. Uma possível solução para esse problema, mostrada em (SALTON; MCGILL, 1986), é aplicar o Processamento de Linguagem Natural (PLN) na recuperação de informação, pois os objetos das busca são objetos lingüísticos. O PLN é um conjunto de técnicas computacionais para análise de textos, com o propósito de simular o processamento humano da língua. Entretanto, por apresentar alto custo computacional, costuma ser usado apenas na melhoria do desempenho de algumas tarefas da recuperação de informação tradicional. Uma aplicação do PLN é o processamento de anáforas.

Anáfora (CARTER, 1987; BEAVER, 2004; FREITAS, 1995) é um fenômeno lingüístico no qual uma **entidade** introduzida *a priori* é referenciada posteriormente em outra frase através de alguma expressão lingüística, tal como no texto:

Texto 1.3 a) *Valentina* acabou de aprender a andar,
b) mas **ela** já quer correr pela casa.

A resolução das anáforas identifica que o pronome **ela** presente na frase (b) do Texto 1.3 se refere ao substantivo *Valentina* da frase (a), ou seja, ambos referenciam uma mesma entidade. O fato de o termo *Valentina* ser referenciado no Texto 1.3 através do pronome **ela** indica que ele tem mais relevância em relação ao texto do que se essa referência não tivesse ocorrido.

O trabalho de Freitas em (FREITAS, 2005) propõe um método para resolver as anáforas de um documento através da criação de uma estrutura (FREITAS, 1992) que permite acompanhar as entidades que se mantêm em evidência ao longo do discurso. Essa estrutura contém informações que podem ser aproveitadas por um método de recuperação de informação.

1.3 Objetivo

O objetivo deste trabalho é aplicar o processamento de anáforas na recuperação de informação, através do uso da estrutura criada durante a interpretação de um documento, propondo uma metodologia computacional para recuperar informações em documentos digitais baseada na resolução de anáforas.

Quando uma *query*[1] é feita pelo usuário, a metodologia proposta neste trabalho encontra os documentos que são relevantes e classifica-os pela quantidade de informação que eles apresentam a respeito dos termos buscados, e não apenas pela disposição[2] dos termos nos documentos. Essa tentativa de interpretação do texto permite dar mais enfoque ao que o autor escreveu.

Retomando os Textos 1.1 e 1.2, a resolução das anáforas do primeiro texto permite identificar que é o termo *sabiá* que está sendo referenciado na elipse da segunda frase do texto, e que é esse mesmo termo que está sendo substituído por *bichinho* na terceira frase. Caso o usuário busque pelo termo *sabiá*, a metodologia consegue identificar que o Texto 1.1 introduz mais informações sobre o termo do que o Texto 1.2, classificando o primeiro como mais relevante do que o segundo. Essa característica não é encontrada nos métodos de recuperação de informação tradicionais.

1.4 Estrutura da dissertação

Os dois primeiros capítulos desta dissertação contêm a revisão bibliográfica necessária para o entendimento deste trabalho. O capítulo 2 apresenta uma visão geral das formas conhecidas de recuperação de informação em documentos digitais existentes na literatura. Além disso, apresenta as formas mais utilizadas para analisar a qualidade dos resultados

[1] Uma *query* é uma busca por determinada informação. Em geral, ela é definida por palavras-chave que são combinadas através de operadores *booleanos*.

[2] Este trabalho considera que a **disposição** de um termo no documento indica sua localização e/ou a quantidade de vezes que ele ocorre no discurso.

obtidos por tais métodos. Em seguida, o capítulo 3 explica o processo de resolução das anáforas de textos em linguagens naturais e apresenta a estrutura gerada pelo processo de resolução dessas anáforas.

O capítulo 4 detalha a metodologia proposta neste trabalho para recuperar informações utilizando a resolução das anáforas de um documento. O protótipo feito a partir dessa metodologia é explicado no capítulo 5. Nesse capítulo também são apresentadas formas de se otimizar as *queries* e é feita uma análise da complexidade de tempo de processamento dos algoritmos implementados.

O capítulo 6 avalia a qualidade dos resultados obtidos pela recuperação de informações baseada na resolução de anáforas. Finalmente, o capítulo 7 apresenta as conclusões sobre este trabalho e algumas propostas de continuação.

"*640K é suficiente para qualquer um.*" Bill Gates, 1981

"*A Internet é apenas uma moda passageira.*" Bill Gates, 1994

2 *Recuperação de informação em documentos digitais*

"*Nenhum passo atrás*
que não seja para tomar impulso."
Che

Neste capítulo são descritos os métodos clássicos de recuperação de informação existentes na literatura, os conceitos básicos da área de RI e as formas mais aceitas para avaliar a qualidade dos resultados de um sistema de recuperação de informação em documentos digitais.

2.1 Modelos clássicos de recuperação de informação

Os sistemas de **Recuperação de Informação**[1] (RI – do termo em inglês *Information Retrieval*), apresentam uma complexidade indiscutível no processo de armazenamento e busca da informação, envolvendo uma série de aspectos que são interdependentes (Van RIJSBERGEN, 1979; GROSSMAN; FRIEDER, 1998; BAEZA-YATES; RIBEIRO-NETO, 1999). Dentre eles, Lopes destaca em (LOPES, 2002) os seguintes fatores:

- a tecnologia eletrônica conduz os usuários ao acesso democrático à informação, ampliando a busca de informações em bancos de dados geograficamente distantes; e
- o alcance da qualidade na informação recuperada requer o planejamento de estratégias de busca específicas para cada banco de dados.

O acesso aos grandes sistemas de recuperação de informação e, conseqüentemente, aos seus bancos de dados veio ampliar significativamente a qualidade das buscas bibliográficas, visto que esses bancos proporcionam diversificados pontos de acesso à informação.

A explosão documentária aumentou significantemente a dificuldade de recuperar informação em sistemas manuais. Segundo (TEIXEIRA; SCHIEL, 1997), o processo de recuperação de informação compreende basicamente três etapas: indexar, armazenar e recuperar. Com o advento da informática, essas etapas tornaram-se uma tarefa mais simples e eficiente, por haver recursos que permitem maior rapidez nestes três processos.

A recuperação de informação lida com o armazenamento de documentos e a recuperação automática de informação associada a eles (BAEZA-YATES; RIBEIRO-NETO, 1999). Os documentos são geralmente textos ou partes deles e o principal objetivo de um sistema de RI é recuperar informações (contidas nos documentos) que possam ser úteis ou relevantes para o usuário. Tais informações (de interesse do usuário) são normalmente chamadas de *necessidade de informação* do usuário. Infelizmente, caracterizar a necessidade de informação do usuário não é uma tarefa simples.

Para obter documentos de seu interesse, o usuário deve traduzir a informação que deseja obter em uma consulta, que pode ser expressa como um conjunto de palavras-chave. Tais palavras-chave devem resumir a informação desejada pelo usuário. Devido à riqueza e flexibilidade da linguagem natural, torna-se difícil para o usuário prever as palavras ou frases que aparecem nos textos de documentos relevantes e que ao mesmo tempo não

[1] Os sistemas de recuperação de informação também são denominados de bancos de dados/informações, no sentido de que são *repositórios de conhecimento*.

ocorrem nos documentos não relevantes. Em geral, uso de palavras-chave introduz uma diferença de semântica entre a intenção do usuário e o conjunto de documentos retornados. Essa diferença de semântica pode se tornar ainda maior devido à dificuldade em se lidar com textos em linguagem natural, que nem sempre são bem estruturados e podem ser semanticamente ambíguos (BAEZA-YATES; RIBEIRO-NETO, 1999).

A presença de documentos não relevantes entre os retornados por uma consulta é praticamente certa. Um sistema de RI deve tentar, então, recuperar o maior número possível de documentos relevantes e o menor número possível de documentos não relevantes. Uma forma simples de obter um conjunto de respostas para a consulta do usuário é determinar quais documentos em uma coleção contêm as palavras da consulta. Para ser eficaz, entretanto, um sistema de RI deve ordenar os documentos da coleção de acordo com o seu grau de relevância em relação à consulta do usuário.

A **relevância** é um conceito fundamental em recuperação de informação e é um componente chave para se calcular o *ranking*[2] de documentos dentre o conjunto retornado por uma consulta. Para calculá-lo, o sistema de RI usualmente adota um modelo para representar os documentos e a consulta do usuário. Cardoso define em (CARDOSO, 2000) que os modelos clássicos utilizados no processo de recuperação de informação (*booleano*, *vetorial* e *probabilístico*) apresentam estratégias de busca de documentos relevantes para uma consulta (*query*). Estes modelos consideram que cada documento é descrito por um conjunto de palavras-chave, chamadas *termos de indexação* (BAEZA-YATES; RIBEIRO-NETO, 1999), que são termos que ajudam a identificar os assuntos principais dos documentos. Associa-se a cada termo de indexação t_i em um documento d_j um peso $w_{ij} \geq 0$, que quantifica a correlação entre os termos e o documento (CARDOSO, 2000). Os três modelos citados são explicados nas seções seguintes.

2.1.1 Modelo *booleano*

O modelo *booleano* é um modelo de recuperação de informação simples, baseado na teoria dos conjuntos e na álgebra *booleana* (BAEZA-YATES; RIBEIRO-NETO, 1999).

No modelo *booleano*, dada uma consulta q e um conjunto de documentos considerados relevantes para q, são atribuídos índices aos documentos indicando quais são mais relevantes que os demais, de forma que se estabeleça uma ordem de relevância entre eles. Esses índices são calculados com base na comparação entre a consulta e os documentos. No

[2] O *ranking* de um documento mede seu grau de relevância em relação ao conjunto de documentos retornados ao usuário.

modelo *booleano*, os documentos recuperados são aqueles que contêm os termos que satisfazem a expressão lógica da consulta. Uma consulta é caracterizada por uma expressão *booleana* convencional formada pelos conectivos lógicos *AND*, *OR* e *NOT*.

Em (SALTON, 1989), é sugerida uma maneira direta de implementar o modelo *booleano*: assuma a existência de uma lista invertida na qual cada entrada corresponde a um termo de indexação. A entrada t_i aponta para uma lista de documentos nos quais o termo t_i ocorre. O conjunto de documentos recuperados pode ser obtido pela intersecção das listas invertidas de documentos, dos termos que aparecem na consulta. Assim, somente documentos cujos termos de indexação satisfazem a consulta *booleana* são recuperados.

Os principais problemas do modelo são (BAEZA-YATES; RIBEIRO-NETO, 1999): (1) Sua estratégia de recuperação é baseada em um critério de decisão binário (i.e., um documento é ou não relevante), não admitindo graus de relevância. Por isso, o modelo *booleano* é, na verdade, mais um modelo de recuperação de dados do que de informações. (2) Nem sempre é simples traduzir uma necessidade de informação em uma expressão *booleana*. As expressões *booleanas* formuladas pelos usuários são, em geral, muito simples.

As vantagens desse modelo são a facilidade de implementação e a expressividade completa das expressões. Apesar de suas desvantagens, ele ainda é muito usado comercialmente e é um bom ponto de partida para quem é novo na área de RI.

2.1.2 Modelo de espaço vetorial

O modelo de espaço vetorial, ou simplesmente modelo vetorial, reconhece que o uso de pesos binários são muito limitantes, e propõe a utilização de pesos não-binários para indexar os termos nas *queries* e nos documentos. Tais pesos são usados para computar o grau de similaridade entre a *query* do usuário e cada documento armazenado no sistema, permitindo uma ordenação mais precisa dos documentos relevantes retornados ao usuário.

O modelo de espaço vetorial representa documentos e consultas como vetores de termos, que são ocorrências únicas nos documentos. O vetor resultado para uma consulta é montado através de um cálculo de similaridade. Aos termos das consultas e documentos, são atribuídos pesos que especificam o tamanho e a direção de seu vetor de representação. Ao ângulo formado por estes vetores dá-se o nome de θ. O $cos\ \theta$ determina a proximidade da ocorrência. Salton define em (SALTON, 1989) que o cálculo da similaridade é baseado nesse ângulo entre os vetores que representam o documento e a consulta, através da

seguinte fórmula:

$$sim(d,q) = \frac{\sum_{i=1}^{t}(W_{id} \times W_{iq})}{\sqrt{\sum_{i=1}^{t} W_{id}^2} \times \sqrt{\sum_{i=1}^{t} W_{iq}^2}} \tag{2.1}$$

Os pesos quantificam a relevância de cada termo para as consultas (W_{iq}) e para os documentos (W_{id}) no espaço vetorial. Para o cálculo dos pesos W_{iq} e W_{id}, utiliza-se uma técnica que faz o balanceamento entre as características do documento, utilizando o conceito de freqüência de um termo em um documento. Se uma coleção possui N documentos e n_{t_i} é a quantidade de documentos que possuem o termo t_i, então o inverso da freqüência do termo na coleção, ou *idf* (*inverse document frequency*), é dado por:

$$idf = log\frac{N}{n_i} \tag{2.2}$$

Este valor é usado para calcular o peso, a partir da fórmula $W_{id} = freq(t_i, d) \times idf_i$, ou seja, W_{id} é o produto da freqüência do termo no documento pelo inverso da freqüência do termo na coleção.

As principais vantagens do modelo vetorial são a sua simplicidade, sua facilidade de computar similaridades com eficiência e o fato de se comportar bem com coleções genéricas. Segundo (BAEZA-YATES; RIBEIRO-NETO, 1999), o modelo é muito popular atualmente por ser simples e rápido.

2.1.3 Modelo Probabilístico

O modelo probabilístico descreve documentos considerando pesos binários que representam a presença ou a ausência de termos. O vetor resultado gerado pelo modelo tem como base o cálculo da probabilidade de um documento ser relevante para uma consulta. A principal ferramenta matemática do modelo probabilístico é o teorema de *Bayes* (Van RIJSBERGEN, 1979).

O princípio probabilístico de ordenação estabelece que o modelo pode ser usado de forma ótima. Esse princípio é baseado na hipótese de que a relevância de um documento para uma determinada consulta é independente de outros documentos. O princípio probabilístico de ordenação, do inglês *Probability Ranking Principle*, pode ser definido como: "*Se a resposta de um sistema de recuperação de referência a cada requisição, é uma ordem de documentos classificada de forma decrescente pela probabilidade de relevância para o usuário que submeteu a requisição, onde as probabilidades são estimadas com a melhor precisão com base nos dados disponíveis, então a efetividade geral do sistema para o seu*

usuário, será a melhor que pode ser obtida com base naqueles dados".

O modelo considera um processo iterativo de estimativas da probabilidade de relevância, no qual devem ser calculados: $P(+R_q|d)$, que indica a probabilidade de que um documento d seja relevante para uma consulta q, e $P(-R_q|d)$, indicando a probabilidade de que um documento d não seja relevante para uma consulta q.

O documento d é considerado relevante para a consulta q se $P(+R_q|d) > P(-R_q|d)$, e o vetor resultado é decidido com base em um fator $W_{d|q}$, definido por:

$$W_{d|q} = \frac{P(+R_q|d)}{P(-R_q|d)}. \tag{2.3}$$

Este fator minimiza a média do erro probabilístico. Através do teorema de *Bayes* e estimativas de relevância baseadas nos termos da consulta, pode-se chegar à seguinte equação:

$$sim(d,q) = W_{d|q} = \sum_{i=1}^{t} (x_i \times W_{qi}), \tag{2.4}$$

onde:

- $x_i \in \{0,1\}$;
- $W_{qi} = log\ r_{qi}(1 - s_{qi})/s_{qi}(1 - r_{qi})$;
- r_{qi} é a probabilidade de que um termo de indexação i ocorra no documento, dado que o documento é relevante para a consulta q; e
- s_{qi} é a probabilidade de que um termo de indexação i ocorra no documento, dado que o documento não é relevante para a consulta q.

O modelo probabilístico tem como vantagem, além do bom desempenho prático, o princípio probabilístico de ordenação, que uma vez garantido, resulta em um comportamento ótimo do método. Entretanto, a desvantagem é que este comportamento depende da precisão das estimativas de probabilidade. Além disso, o método não explora a freqüência do termo no documento e ignora o problema de filtragem de informação.

Em (GROSSMAN; FRIEDER, 1998) são mostrados exemplos de cada um dos modelos clássicos descritos nesta dissertação. Esses modelos (*booleano*, *vetorial* e *probabilístico*) se baseiam, essencialmente, na disposição das palavras nos documentos. Entretanto, nos documentos escritos em linguagens naturais, uma entidade pode ser referenciada em

determinado documento através de mais de uma palavra. Por isso, a qualidade dos resultados de um método de recuperação de informação em linguagem natural seria melhor se houvesse algum tipo de interpretação *a priori* dos documentos, de forma que fossem identificadas todas as referências a cada entidade existente neles.

Este trabalho propõe que a resolução das anáforas seja usada na recuperação de informação em documentos digitais. A resolução de anáforas permite identificar, por exemplo, que no texto "Débora quer *uma cachorrinha*. **O pêlo** pode ser dourado.", os termos *cachorrinha* e **pêlo** referenciam a mesma entidade. A forma como isso é feito é detalhada no capítulo 3.

2.2 Avaliação da qualidade de um modelo

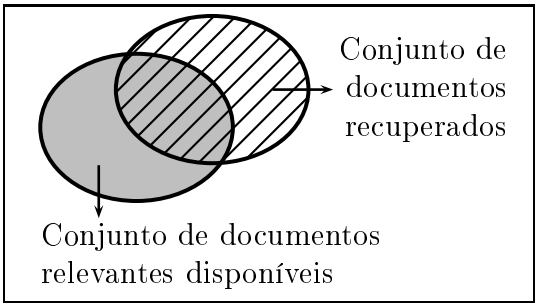
Devido à dificuldade de realizar buscas por informações nos textos escritos em linguagens naturais, os sistemas conhecidos de RI não garantem que todos os documentos relevantes são retornados e que todos os não relevantes são descartados. Torna-se necessário, então, avaliar a qualidade desses sistemas.

Existem várias formas de se medir a qualidade de um sistema de recuperação de informação. Para isso, é necessária uma coleção de documentos e uma *query* cuja relevância em relação aos documentos da coleção seja conhecida. As medidas mais comuns assumem uma relevância binária (CLEVERDON, 1997): ou o documento é relevante ou ele é completamente irrelevante. Na prática, *queries* costumam ser mal formuladas e a relevância pode depender do que a pessoa realmente quer buscar.

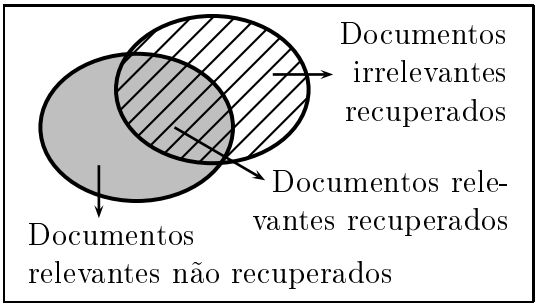
Van Rijsbergen explica em (Van RIJSBERGEN, 1979) que, atualmente, as duas medidas mais utilizadas para avaliar os sistemas são *precision* e *recall*, que são baseadas na noção de documentos relevantes de acordo com uma determinada **necessidade de informação**. *Recall* é a proporção de documentos relevantes de uma coleção que foram recuperados (i.e. o número de acertos em relação ao número de casos existentes) e *precision* é a proporção dos documentos recuperados em uma busca que são relevantes (ou seja, o número de acertos em relação ao número de casos tratados) (YANG; LIU, 1999; GROSSMAN; FRIEDER, 1998). Em geral, os valores de *precision* e *recall* são calculados usando uma coleção de consultas, documentos e julgamentos de relevâncias conhecidos (BAEZA-YATES; RIBEIRO-NETO, 1999).

Como pode ser visto na Figura 2.1 (JIZBA, 2000), essas medidas assumem que:

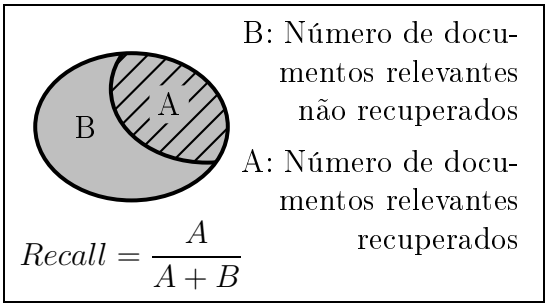
1. há um conjunto de documentos na base de dados que são relevantes à consulta;
2. documentos são relevantes ou irrelevantes (não são admitidos graus de relevância);
3. o conjunto recuperado não costuma ser perfeitamente igual ao conjunto de documentos relevantes.



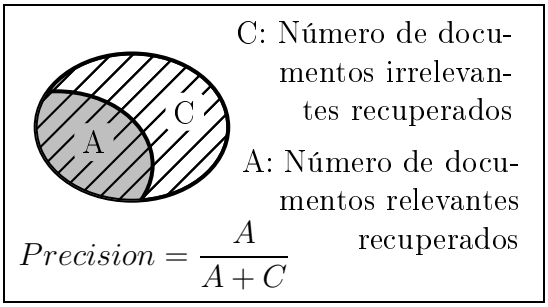
(a) Documentos relevantes e documentos recuperados



(b) Documentos relevantes não recuperados, documentos relevantes recuperados e documentos irrelevantes recuperados



(c) Cálculo do valor de *recall*



(d) Cálculo do valor de *precision*

Figura 2.1: Cálculo de *recall* e *precision*

De acordo com a Figura 2.1, *recall* é a razão entre o número de documentos relevantes recuperados e o total de documentos relevantes no conjunto de dados. Já o valor de *precision* é calculado pela razão entre o número de documentos relevantes recuperados e o total de documentos relevantes e irrelevantes recuperados. A Figura 2.2 mostra o conjunto R com os documentos relevantes de uma consulta e o conjunto S com os documentos recuperados por ela.

De acordo com os conjuntos R e S da Figura 2.2, os cálculos dos valores de *recall* e de *precision* também podem ser expressos, respectivamente, pelas equações 2.5 e 2.6:

$$recall = \frac{|R \cap S|}{|R|} \tag{2.5}$$

$$precision = \frac{|R \cap S|}{|S|} \tag{2.6}$$

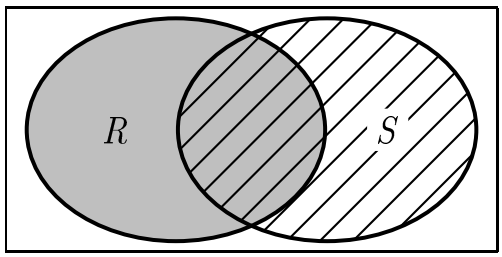


Figura 2.2: Conjunto de documentos relevantes e de documentos recuperados

Alguns problemas relacionados a essas medidas são:

- Para calcular os valores de *recall* e *precision*, os documentos precisam ser considerados relevantes ou não para o cálculo do *recall* e *precision*. Obviamente, pode haver documentos mais ou menos relevantes que outros. Além disso, o que é relevante para uma pessoa pode não ser para outra;
- Não é fácil saber quantos documentos relevantes existem no conjunto de dados e quais são eles, o que torna difícil o cálculo do valor de *recall*. Para isso, costuma-se procurar manualmente pelos documentos relevantes dentre todos os disponíveis. O conhecimento do conjunto de documentos que são relevantes não costuma estar disponível em grandes coleções de documentos (BAEZA-YATES; RIBEIRO-NETO, 1999). Nesses casos, não se pode determinar precisamente o valor de *recall*;
- Por fim, os valores de *recall* e *precision* se baseiam no fato de que o conjunto de documentos relevantes para uma *query* é sempre o mesmo, independente do usuário. Entretanto, usuários diferentes podem interpretar de formas diferentes se cada um dos documentos é ou não relevante. Uma forma de contornar esse problema é haver uma interatividade entre o usuário e o sistema (BAEZA-YATES; RIBEIRO-NETO, 1999).

Apesar dos problemas citados, a eficácia dos sistemas de recuperação de informação hoje em dia ainda é medida em termos desses valores. Não há um estudo estatístico adequado que mostre formas melhores de se medir a performance desses sistemas.

Segundo (Van RIJSBERGEN, 1979; YANG; LIU, 1999), outras medidas que podem ser utilizadas são: a medida F, a medida E e o *fallout*. A decisão de quais medidas utilizar em uma avaliação depende da aplicação e há sempre discussões sobre a confiabilidade de tais medidas (SU, 1998). Um exemplo é o artigo (GWIZDKA; CHIGNELL, 1999), onde se discute como avaliar máquinas de busca. Não é claro, por exemplo, o quanto pequenas diferenças nos valores de *precision* e *recall* têm efeito no sucesso na busca de um usuário.

Segundo (BAEZA-YATES; RIBEIRO-NETO, 1999), a medida F é uma média harmônica entre os valores de *recall* e *precision*, computada da seguinte forma:

$$F(j) = \frac{2}{\frac{1}{r(j)} + \frac{1}{P(j)}}, \tag{2.7}$$

onde $r(j)$ é o valor de *recall* do j-ésimo documento a ser classificado, e $P(j)$ é o valor de *precision* para esse j-ésimo documento. Logo, $F(j)$ é a média harmônica de $r(j)$ e $P(j)$ do j-ésimo documento da classificação. A função F assume valores no intervalo $[0, 1]$. Se F vale 0, nenhum documento relevante foi recuperado. Se vale 1, todos os documentos recuperados são relevantes. Logo, essa medida assume um valor alto somente quando tanto o valor de *recall* quanto o de *precision* são altos.

A medida E, proposta (Van RIJSBERGEN, 1979), também combina os valores de *recall* e *precision*, mas permite que seja especificado se há um interesse maior no valor de uma ou de outra. A medida E é definida da seguinte forma:

$$E(j) = 1 - \frac{1 + b^2}{\frac{b^2}{r(j)} + \frac{1}{P(j)}}, \tag{2.8}$$

onde $r(j)$ é o valor de *recall* do j-ésimo documento a ser classificado, e $P(j)$ é o valor de *precision* para esse j-ésimo documento. $E(j)$ é uma medida relativa a $r(j)$ e $P(j)$, e b é um parâmetro que reflete a importância relativa de *recall* e de *precision* no cálculo de $E(j)$. Para $b = 1$, $E(j)$ é o complemento da média harmônica F. Valores de b maiores que 1 indicam que se está mais interessado no valor de *precision* do que no de *recall*, enquanto valores de b menores que 1 indicam que se está mais interessado em *recall* do que em *precision*.

O capítulo 6 mostra características da metodologia proposta neste trabalho que influenciam o cálculo dos valores de *recall* e *precision* e, conseqüentemente, das medidas F e E.

"*Não viva no passado, não sonhe com o futuro,*
concentre a mente no momento presente."
Buda

3 *Uma proposta de resolução de anáforas*

"*Não tenha medo da perfeição.*
Você nunca vai atingi-la."
Salvador Dalí

Neste capítulo é apresentada a proposta de resolução de anáforas usada neste trabalho com a finalidade de recuperar informações relevantes em documentos digitais.

3.1 Introdução

Anáfora é um fenômeno lingüístico no qual uma entidade já introduzida no discurso é referenciada em outra frase através de alguma expressão lingüística, tal como no texto:

Texto 3.1 a) *Débora* quer *uma cachorrinha.*
b) **O pêlo** pode ser dourado.

A frase (3.1a) introduz duas entidades: *Débora* e *uma cachorrinha*[1]. Já a frase (3.1b) apresenta apenas uma entidade – **o pêlo**. No processo de interpretação, humano ou computacional, a utilização do artigo definido "o" é um indicativo de que a entidade já havia sido introduzida no discurso, i.e. apresenta um caráter anafórico. Resolver uma anáfora é, *a priori*, identificar a quem ou a que se refere esta anáfora. Mas no caso acima é mais do que isto: sem dúvida **o pêlo** existe no texto por causa da existência de *uma cachorrinha*, porém a interpretação de **o pêlo** deve identificar ainda de que forma ele está ligado com o animal (neste caso, **o pêlo** é uma parte da *cachorrinha*). Isto é uma Anáfora Nominal Definida (AND). Considere agora o seguinte texto:

Texto 3.2 a) *Júnior* adora *Valentina.*
b) **Ela** é sua sobrinha.

O pronome **ela** da segunda frase do Texto 3.2 poderia referenciar o termo *Valentina* ou o termo *Júnior*, ambos na primeira frase do discurso. Como o pronome não concorda em gênero com o substantivo *Júnior*, mas concorda em gênero e número com *Valentina*, é fácil identificar que **ela** faz uma referência a *Valentina.* A expressão lingüística que introduz a anáfora na frase é denominada **expressão anafórica**. A informação previamente introduzida, a que deve ser ligada à expressão anafórica, é denominada **antecedente** e o processo pelo qual é identificado o antecedente de uma expressão anafórica é denominado **resolução anafórica** ou **resolução de anáforas**. No exemplo, *Valentina* é o antecedente da expressão anafórica **ela**. Ainda no Texto 3.2, a concordância de gênero e número permite identificar que só existe um antecedente possível para a expressão anafórica (o sintagma nominal *Valentina*). Considere agora o texto:

Texto 3.3 *Fernandinha* adora sua *cachorrinha.*
Ela não late muito à noite.

[1] A convenção deste trabalho é que, nos exemplos, as partes em **negrito** assinalam as anáforas e as partes em *itálico* assinalam seus possíveis antecedentes.

No exemplo, considerando apenas o número e o gênero, existem dois antecedentes possíveis para a resolução da anáfora introduzida pelo pronome **ela**: o sintagma nominal *Fernandinha*, que ocupa a posição de sujeito da frase, e o sintagma nominal definido *cachorrinha*, que ocupa a posição de objeto da frase. Devido a essa ambigüidade, torna-se necessária a existência de outras informações que permitam escolher entre os dois antecedentes possíveis. Entre as possíveis fontes de informações, destacam-se:

1. O conhecimento de senso comum induzindo um interlocutor a dizer que os homens em condições normais não latem, enquanto os cachorros o fazem;

2. O conhecimento *inconsciente* do transmissor de que a emissão de um texto descrevendo situações sobre um mesmo objeto, i.e, mantendo o mesmo *centro de atenção*, facilita a interpretação por parte do receptor (KRUIJFF-KORBAYOVá; STEEDMAN, 2003; BEAVER, 2004). Os centros de atenção são repetidos, preferencialmente, sob a forma de anáforas, sendo que o objeto sobre o qual o texto é centrado tende a estar na posição do sujeito a cada frase.

As duas fontes de informação permitem resolver a anáfora referente ao pronome **ela** na segunda frase do Texto 3.3: quem late à noite é a *cachorrinha*. Há uma tendência de que a informação estrutural (GUNDEL; HEGARTY; BORTHEN, 2003) seja determinante na escolha dos antecedentes em contextos onde não se conhece muito sobre o domínio no qual o texto discursa, enquanto que em contextos onde existe um conhecimento razoável sobre o domínio, há uma tendência de se considerar o conhecimento de senso comum como determinante. Em ambos os contextos, uma informação não anula a outra na escolha do antecedente (BEAVER, 2004).

3.2 Resolução de Anáforas

A interpretação das anáforas nominais definidas ou de qualquer fenômeno anafórico pode ser generalizada como um processo que atribui valores aos itens da seguinte equação:

$$\mathcal{R}(\mathcal{A}, \mathcal{T}), \tag{3.1}$$

onde $\mathcal{A}$ denota a entidade introduzida pela interpretação fora de contexto de um pronome, de uma elipse ou de um sintagma nominal definido, $\mathcal{T}$ denota o seu antecedente e $\mathcal{R}$ é a relação existente entre $\mathcal{A}$ e $\mathcal{T}$. O processo de resolução da equação, que é propriamente o processo de resolução de anáforas, consiste em descobrir $\mathcal{T}$ e $\mathcal{R}$ dado $\mathcal{A}$.

Freitas propõe em (FREITAS, 2005) uma metodologia computacional que interpreta as anáforas nominais definidas cuja relação $\mathcal{R}$ é uma dentre: *parte de*, *membro de*, *subcategorizado por* e *co-referência* (detalhadas na seção 3.2.1). A obtenção das relações é feita por um conjunto de regras pragmáticas e o resultado é uma metodologia que permite, de forma integrada, resolver anáforas e elipses (FILHO; FREITAS, 2003; FREITAS; LOPES; MENEZES, 2004). A seção 3.2.2 define o *foco* de um discurso, que é utilizado na criação da Estrutura Nominal do Discurso (ver seção 3.2.3). O capítulo 4 mostra que a Estrutura Nominal do Discurso pode ser utilizada na recuperação de informações.

3.2.1 As regras pragmáticas

Baseado no conhecimento que as pessoas têm sobre a língua que falam, é possível estabelecer um conjunto pragmático de regras a serem utilizadas na determinação da relação entre a expressão anafórica e seus antecedentes. As informações sobre gênero, número e grau, coletivos e animacidade (SIDNER, 1979) podem ser utilizadas na determinação das seguintes relações:

- **co-referência**: indicando que tanto $\mathcal{A}$ quanto $\mathcal{T}$ denotam a mesma entidade: $\mathcal{A} = \mathcal{T}$. Por exemplo, considere o Texto 3.4:

  Texto 3.4 *Jociel* comprou *uma aliança*.
  Ele deu-a para Mariella.

  No texto Texto 3.4, a entidade do discurso introduzida pelo pronome **ele** co-referencia a entidade introduzida por *Jociel*, já que o pronome concorda em gênero e número com o substantivo próprio *Jociel* (o que não acontece em relação ao substantivo *aliança*);

- **membro de**: indicando que a entidade denotada por $\mathcal{A}$ é um membro do conjunto de entidades denotada por $\mathcal{T}$. Por exemplo, considere o Texto 3.5:

  Texto 3.5 Rafael tem duas *irmãs*.
  A irmã mais velha mora nos EUA.

  Como pode ser visto no Texto 3.5, *irmãs* é uma entidade *coletiva* composta de diversos indivíduos simples do mesmo tipo. O mesmo acontece nos Textos 3.6 e 3.7:

Texto 3.6 Helen avistou um *cardume*.
Os peixes eram coloridos.

Texto 3.7 Helen avistou um *cardume*.
Um peixe era imenso.
O peixe azul era o menor.

- **parte de**: indicando que a entidade denotada por $\mathcal{A}$ é parte (estrutural) da entidade denotada por $\mathcal{T}$, como ocorre no Texto 3.8, no qual **a cerveja** pode ser considerada parte de *um isopor*.

Texto 3.8 Diogo trouxe *um isopor*.
A cerveja estava quente.

- **subcategorizado por**: indicando que a entidade denotada por $\mathcal{A}$ é, de alguma forma, uma parte conceitual da entidade denotada por $\mathcal{T}$, como no Texto 3.9, no qual **aeromoça** é subcategorizado por *avião*.

Texto 3.9 *Thiago* viajou de avião.
Ele se apaixonou pela **aeromoça**.

- **acomodação**: essa pseudo relação surge quando todas as outras possibilidades de interpretação de um sintagma nominal definido terminaram e nenhuma das relações anteriores pôde ser estabelecida. Mesmo neste caso, algo deve ser feito para que a interpretação do discurso continue, pois um emissor não deseja transmitir discursos desconexos. Como conseqüência, a entidade introduzida pelo sintagma nominal definido, a qual não se configurou como sendo anafórica, deve então ser acomodada na representação semântica, comportando-se de maneira semelhante a um indefinido.

3.2.2 Foco do discurso

Foco é um termo utilizado para designar a entidade mais em evidência no discurso (ver (SIDNER, 1981; GROSZ, 1977; FREITAS; LOPES, 1994b; HAJIČOVá; SKOUMALOVá; SGALL, 1995)). Tipicamente, o foco é a entidade sobre a qual o transmissor centra sua atenção em determinado ponto do discurso, sendo que a utilização continuada de uma determinada entidade através do uso de anáforas é um forte indício de que esta entidade está em

foco (GROSZ; JOSHI; WEINSTEIN, 1995; SIDNER, 1981). É necessário ainda definir dois tipos de entidades salientes: um **foco explícito** e um **foco implícito**.

O foco explícito é resultante da utilização de anáforas pronominais, elipses e ANDs diretas (relação de co-referência), tal como no Texto 3.10. Na frase (b), o pronome pessoal **ele** referencia o substantivo *Daniel* da frase (a).

Texto 3.10 a) *Daniel* está fazendo Mestrado.
b) **Ele** está morando na Itália.

O foco implícito é resultante da utilização de conhecimento subjacente ao discurso, indicando uma entidade que continua a ser referenciada de forma indireta, tal como acontece no Texto 3.11. Ao contrário do que ocorre no exemplo anterior, existe a necessidade de encontrar também a relação $\mathcal{R}$. É preciso saber que há motoristas de ônibus (como existem motoristas de táxis, de caminhões *etc*); e que há portas de ônibus (como de táxis, de caminhões, de casas *etc*); mas que não há portas (no plural) de motorista.

Texto 3.11 a) *Um ônibus* acabou de chegar.
b) **O motorista** abriu ***as portas***.
c) **Os passageiros** desceram **pela porta de trás**.

Ainda no Texto 3.11, o sintagma nominal definido **os passageiros** na frase (4.1c) tem como antecedente implícito o *ônibus* introduzido na frase (4.1a) (a resolução deve descartar as entidades introduzidas na frase (4.1b)), com a ligação entre *ônibus* e **passageiros** não sendo uma relação de co-referência direta, mas sim uma relação em que os passageiros são parte do ônibus.

Usar os dois tipos de foco é como acompanhar as continuações e mudanças do centro de atenção em cada frase e/ou no conjunto, indicando como organizá-las em relação ao assunto atual do discurso. O foco explícito serve de medida para a coerência local, ou seja, entre duas frases consecutivas, e o foco implícito serve tanto para medir a coerência local quanto de um conjunto de frases que versam sobre um mesmo assunto.

3.2.3 Estrutura Nominal do Discurso

A resolução de anáforas é um processo em que, dados $\mathcal{A}$ e $\mathcal{R}$, deve-se determinar o antecedente $\mathcal{T}$ (ver seção 3.1). Os problemas que precisam ser resolvidos nesta determinação são:

1. Freqüentemente existe mais de um candidato para $\mathcal{T}$. O processo de interpretação pode optar então por duas alternativas: (a) escolher um candidato e prosseguir com a interpretação ou (b) considerar uma interpretação para cada um dos n candidatos e prosseguir com n interpretações. A primeira hipótese tem a vantagem da velocidade de resolução, porém pode acontecer que informações introduzidas posteriormente (pressuposto de não monotonicidade) invalidem a solução anterior, obrigando a um reprocessamento da informação já interpretada. Na segunda hipótese, todo o processamento das alternativas possíveis já foi feito. Assim, já não é necessário o reprocessamento, mas sim a busca por uma interpretação alternativa (já pronta). A desvantagem desta segunda hipótese é que a interpretação do discurso é o produto cruzado das interpretações possíveis para cada frase, o que torna o processamento oneroso. O ideal seria uma metodologia que utilizasse o melhor de cada uma destas hipóteses.

2. $\mathcal{T}$ pode estar em qualquer frase do discurso. A conseqüência para a interpretação é que, à medida que o discurso vai sendo interpretado, torna-se maior o número de possíveis antecedentes $\mathcal{T}$, por um dado A e é maior o esforço computacional do processo de resolução como um todo. A solução encontrada na literatura consiste em limitar o espaço de busca a um determinado número m de frases anteriores. Como encontrar o valor ideal para m? A escolha de um valor pequeno pode impossibilitar a escolha de um antecedente $\mathcal{T}$ que esteja em uma frase anterior à frase m. A escolha de um valor grande torna o processamento oneroso.

3. Finalmente, a limitação do número de frases e a consideração de que as entidades nelas introduzidas constituem apenas um conjunto de simples escolhas reduz a contribuição semântica destas mesmas entidades para a interpretação do discurso como um todo (FREITAS; LOPES, 1996). Cada frase (e suas entidades) traz uma contribuição semântica tanto para a sua própria interpretação (fora de contexto) quanto para a estruturação do conhecimento disperso em cada frase do discurso. Considerar a contribuição da frase para a estruturação do discurso permite ao processo de interpretação inserir restrições naturais[2] ao processo de escolha de $\mathcal{T}$ podendo então aumentar o número de frases consideradas. A Teoria da Centragem (GROSZ; JOSHI; WEINSTEIN, 1995) utiliza essa abordagem considerando a contribuição de cada entidade para o acompanhamento da movimentação do centro de atenção (foco) à medida que o discurso avança e como resultado deste acompanhamento são geradas restrições para a escolha de um antecedente. O que a Teoria

[2]Impostas pelo emissor e codificadas no discurso.

da Centragem não considera é que a informação sobre a movimentação do foco não só gera restrições imediatas no processo de resolução de anáforas (e.g. nas próximas duas frases) como também pode gerar restrições estruturais sobre a interpretação de qualquer entidade do discurso (LOPES; FREITAS, 1994; FREITAS; LOPES, 1994a).

Freitas apresenta em (FREITAS, 2005) uma metodologia de obtenção do antecedente $\mathcal{T}$ da fórmula $\mathcal{R}(\mathcal{T},\mathcal{A})$ através da criação da Estrutura Nominal do Discurso (END), que permite ao sistema de interpretação de anáforas restringir o número de antecedentes $\mathcal{T}$ possíveis para uma expressão anafórica $\mathcal{A}$, objetivando levar em consideração todos os itens observados acima. Esta estrutura permite: (1) restringir o espaço de busca por antecedentes sem limitar o número de frases e (2) criar um semiprocessamento de interpretações, i.e. intermediário entre uma interpretação completa e um reprocessamento, de forma a agilizar uma reinterpretação. Esta estrutura permite assim explicitar a movimentação dos focos durante todo o discurso.

A END é uma árvore na qual cada folha representa o conteúdo semântico de uma determinada frase do discurso e cada nó interno representa o conteúdo semântico resultante do acompanhamento das entidades mais em evidência (focos) de seus filhos. Uma propriedade importante desta árvore é que somente os nós mais à direita estão abertos para interpretação (POLANYI; BERG; AHN, 2003; POLANYI, 1988). Uma forma esquemática desta árvore pode ser vista na Figura 3.1, na qual F1, F2, F3 e F4 são frases, S1, S2 e S3 são nós internos e F4 é a frase a ser interpretada em relação aos nós visíveis S1, S3 e F3.

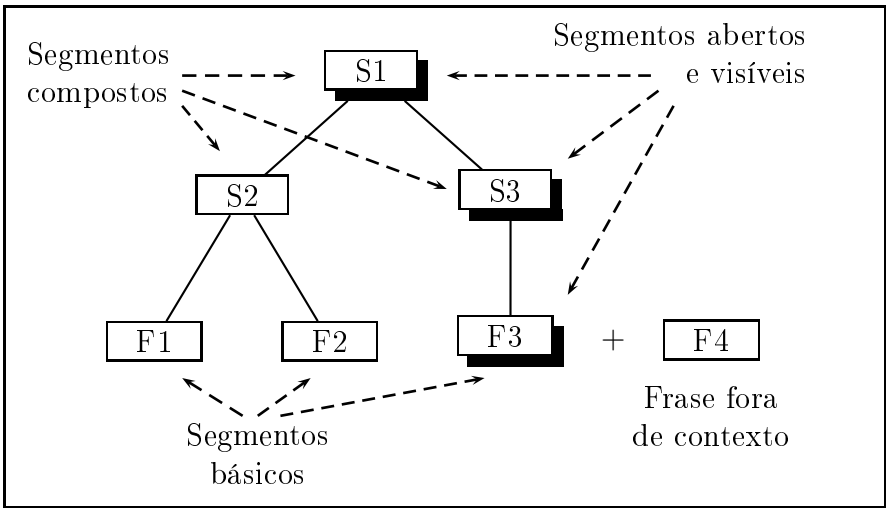


Figura 3.1: Estrutura Nominal do Discurso

Existe concordância nas áreas da Lingüística Computacional (SIDNER, 1979; SIBUN, 1992; GROSZ; SIDNER, 1986), Inteligência Artificial (HOBBS, 1993; HOBBS, 1985) e Filo-

sofia da Linguagem (POLANYI; BERG, 1996; MANN; THOMPSON, 1987) de que um agente cooperativo[3], produz um discurso de maneira planejada e organizada, reduzindo o esforço de interpretação por parte de seu interlocutor – o receptor. Esta forma organizada de transmissão é expressa sob a forma de uma estrutura que na maioria das vezes está implícita no discurso, a denominada *Estrutura do Discurso*.

A estrutura é fundamental para a compreensão do discurso pois organiza a informação transmitida, auxiliando sua interpretação por parte do receptor. O processo de estruturação do discurso está diretamente relacionado com a comunicação entre o transmissor e o receptor (FREITAS; LOPES, 1994a; ABBOTT, 1993), sendo que sua eficiência pode ser medida pela rapidez com que o receptor recupera as interpretações possíveis para um dado trecho do discurso (BLUTNER, 2000). Em termos computacionais, isso equivale a um menor tempo de processamento. Em termos lógicos, equivale a um número mais baixo de inferências sobre o menor número possível de modelos.

A END é estruturada como uma árvore motivada pela estrutura apresentada por Polanyi *et al.* em (POLANYI; BERG; AHN, 2003; POLANYI, 1988), que permite relações de subordinação e coordenação entre os segmentos constituintes. A estrutura pode apresentar restrições à interpretação, permitindo considerar, por exemplo, que somente os nós mais à direita da árvore estão abertos para a interpretação de novas frases (Figura 3.2).

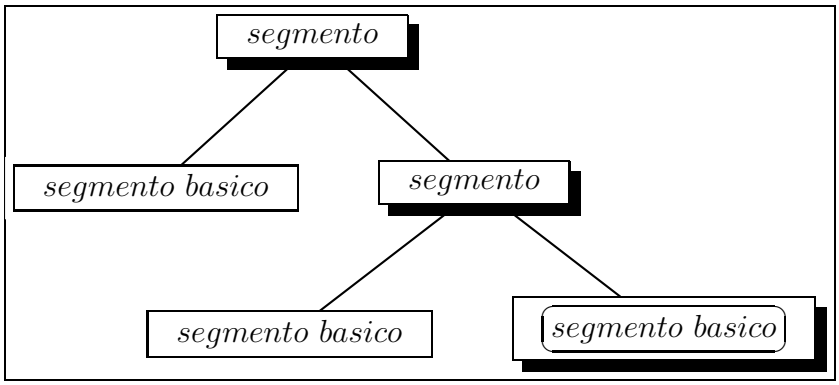


Figura 3.2: Árvore com os nós mais à direita abertos

Assim, o processo de interpretação de um novo segmento relativo aos segmentos previamente introduzidos se resume em um processo de encontrar um dos segmentos mais à direita na árvore (denominado *segmento visível*), o qual possa servir de referência para interpretação da nova frase.

A END é uma árvore específica para a resolução de anáforas. Devido a isso, a estrutura é fortemente baseada no acompanhamento do modo como as entidades introduzidas em

[3]Um agente que não tem a intenção de enganar transmitindo, deliberadamente, informações incorretas.

cada frase (as que estão mais em evidência ou em *foco*) evoluem durante o discurso (mantendo-se salientes ou não). Ela é criada analisando-se a movimentação dos focos implícitos e explícitos no decorrer do discurso.

Durante a construção da END, todo o material semântico existente na interpretação de uma frase – o chamado **segmento básico** – é agrupado em uma estrutura em árvore. Os nós internos são chamados **segmentos** e são compostos de material semântico herdado de seus nós filhos (a árvore é construída a partir das folhas). Apenas três atributos existentes em cada segmento da END são aproveitados neste trabalho: (1) o seu **foco explícito**, (2) o seu **foco implícito** e (3) as **relações de co-referência** que possam existir nele. Relações de co-referência indicam que termos distintos referenciam uma mesma entidade no discurso. O capítulo 4 mostra que essas relações dão origem a um **dicionário de sinônimos** referente ao discurso que possibilita, por exemplo, que o resultado da busca por **sabiá** no Texto 1.1 seja o mesma que o da busca pelo termo **passarinho**, pois os dois substantivos referenciam a mesma entidade no texto em questão.

A Estrutura Nominal do Discurso é criada com a finalidade específica de resolver anáforas. Entretanto, ela contém informações que podem ser úteis para um sistema de RI: a estrutura permite saber exatamente a quantidade de vezes que cada entidade é referenciada nos documentos. Essa informação adiciona ao sistema a capacidade de decidir com mais precisão se certa entidade é mais relevante que outra em um documento. A utilização da END introduz certas características à recuperação de informação que tendem a aumentar a quantidade de documentos relevantes que ela recupera, como a identificação das múltiplas referências a uma mesma entidade, que torna possível saber o número de vezes que a entidade é realmente referenciada, e a identificação dos termos sinônimos em um documento (os termos que se co-referenciam).

Este trabalho propõe uma metodologia computacional para recuperar informações relevantes a partir da resolução das anáforas de um documento, visando aumentar a qualidade dos resultados de uma *query*. Dessa forma, os documentos relevantes recuperados são classificados pela quantidade de informação que apresentam a respeito dos termos buscados, e não apenas pela localização e/ou quantidade de ocorrências de tais termos. O próximo capítulo explica a principal contribuição deste trabalho: como a Estrutura Nominal do Discurso é aplicada na recuperação de informação.

"O segredo é não correr atrás das borboletas...
é cuidar do jardim para que elas venham até você."
Mário Quintana

4 *Recuperação de informações na Estrutura Nominal do Discurso*

"Não existe nenhum caminho lógico
para a descoberta das leis elementares do universo
– o único caminho é o da intuição."
Albert Einstein

Este capítulo apresenta a forma como a Estrutura Nominal do Discurso é utilizada na recuperação de informações em documentos digitais. Além disso, detalha o cálculo de relevância de um documento interpretado pelo processamento de anáforas em relação a uma *query*, bem como o processo que é feito quando o usuário a define no sistema.

4.1 Introdução

Para que um método de busca em documentos digitais seja considerado bom, deve ser feita uma análise dos valores de *recall* e *precision* que seus resultados alcançam (ver capítulo 2). Essa é uma forma de avaliar a *qualidade* dos resultados do método: quando um usuário faz determinada busca em uma coleção de documentos, ele espera que lhe seja informado qual documento é o mais relevante em relação à sua consulta. O documento é relevante se seu *assunto* principal é o objeto de busca do usuário. O documento mais relevante é aquele que introduz mais informações sobre o que está sendo buscado. Entretanto, um documento escrito em linguagem natural exige grande esforço computacional para ser interpretado, já que sua relevância pode depender de quem o lê, dificultando a determinação dos assuntos principais dos documentos.

Os métodos de busca tradicionais se baseiam essencialmente na disposição das palavras nos documentos (ver capítulo 2). Eles consideram a quantidade de vezes e o local em que as palavras aparecem nos documentos, não apresentando soluções para se *interpretar* esses documentos. Para tentar contornar tal problema, este trabalho propõe utilizar a Estrutura Nominal do Discurso (END) criada pelo processamento de anáforas (ver capítulo 3) para tornar possível essa recuperação de informação em documentos digitais. Utilizando essa estrutura, é possível acompanhar como as entidades mais relevantes permanecem em evidência ao longo do documento. Com a resolução das anáforas, os termos contidos em um segmento visível da END gerada expõem entidades e ligações que podiam estar obscuras no documento original. O processamento de anáforas consegue capturar tais termos, possibilitando uma busca mais refinada dos documentos. Por exemplo, é possível notar que *carro* é referenciado em todas as frases do Texto 4.1 e, por isso, pode-se inferir que este é o assunto principal do texto.

Texto 4.1 Sérgio comprou um *carro* usado.
Mariana não gostou da **cor**.
O motor estava quebrado.

Este capítulo mostra como a END gerada pelo processamento de anáforas deve ser usada na busca em documentos digitais. Primeiro, é detalhado na seção 4.2 como ela deve ser adaptada para tal, quais informações presentes na estrutura são aproveitadas e como ela é analisada pela metodologia proposta neste trabalho. Com essa análise pode-se, então, calcular o grau de relevância de cada termo presente no documento. As formas de calcular essa relevância são mostradas na seção 4.4. Antes disso, na seção 4.3, é

apresentada a proposta de geração de um dicionário de sinônimos extraídos da END de cada documento. A seção 4.5 detalha o índice que armazena as informações extraídas da END dos documentos que são utilizadas pela metodologia. Na seção 4.6, uma busca simples (na qual apenas uma palavra é procurada no documento) a partir da END é totalmente detalhada. A seção 4.7 mostra como são feitas as buscas compostas (aquelas que possuem conjunções e/ou disjunções de palavras). Finalmente, na seção 4.8, é dada uma visão geral sobre as diversas fórmulas e abordagens apresentadas neste capítulo e sobre a metodologia proposta neste trabalho.

4.2 Estrutura Nominal do Discurso para Busca

A Estrutura Nominal do Discurso é resultado do acompanhamento das entidades existentes nas frases do discurso, em especial os focos (ver capítulo 3). Korbayová *et al.* definem em (KRUIJFF-KORBAYOVá; STEEDMAN, 2003) que ela reflete parte da estrutura mental do transmissor em relação aos indivíduos ou entidades existentes no discurso. Em outras palavras, a END, resultante da interpretação de um discurso ou documento, é uma hierarquização das suas entidades de forma que as mais salientes estejam mais em evidência na árvore final.

A END é uma árvore na qual cada folha representa o conteúdo semântico de uma determinada frase do discurso e cada nó interno representa o conteúdo semântico resultante do acompanhamento das entidades mais em evidência (focos) de seus filhos. Uma forma esquemática desta árvore pode ser vista na Figura 4.1.

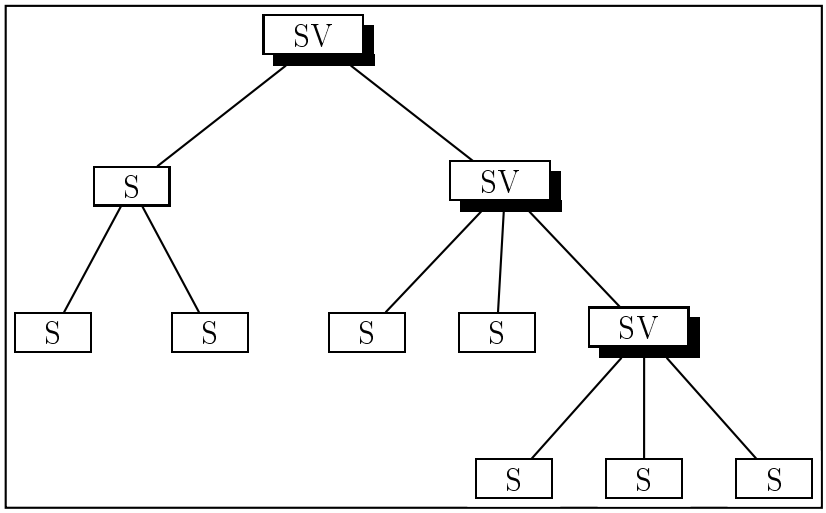


Figura 4.1: Esquema da Estrutura Nominal do Discurso

Durante a criação da estrutura, somente os nós mais à direita, denominados *segmentos*

visíveis (*SVs*), estão abertos para interpretação. Na Figura 4.1, cada *S* é um segmento que não é utilizado durante a busca pelo antecedente de uma resolução anafórica. Cada folha ou nó interno da estrutura possui um *foco explícito* e um *foco implícito*.

Originalmente, uma END tem o formato apresentado na Figura 4.1. Note que o último segmento visível (*SV*), que seria a representação da última frase interpretada, foi considerado como sendo um segmento (*S*). Para se realizar uma busca, a END deve ser transformada em uma *Estrutura Nominal do Discurso para Busca* (ENDB). É feita então uma transformação na árvore de forma que todos os segmentos visíveis passam a ser a raiz de uma árvore que contém seus subsegmentos. Cada *SV*, por sua vez, passa a ser elemento de uma lista. A Figura 4.2 ilustra a ENDB gerada a partir da END mostrada na Figura 4.1.

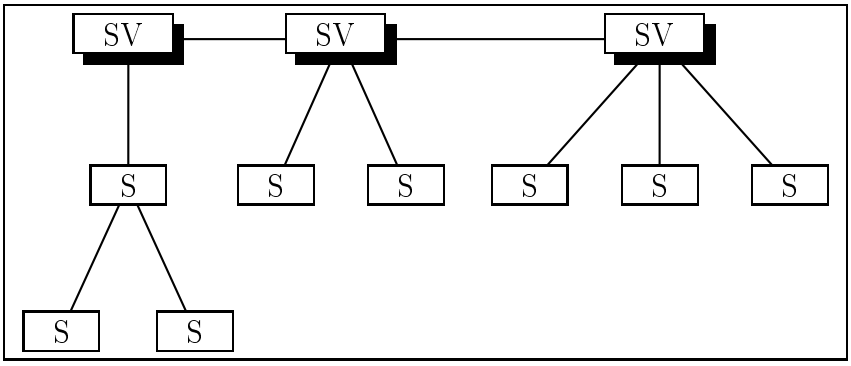


Figura 4.2: Esquema da Estrutura Nominal do Discurso para Busca

De forma geral, apenas transforma-se uma árvore em uma lista. Os segmentos da ENDB herdam apenas os focos implícitos e explícitos dos segmentos da END. Com isso, pode-se notar que nem todos os termos do documento podem ser localizados por esta metodologia de busca, pois a ENDB contém apenas os assuntos principais (focos) da END original, descartando as demais informações contidas nela. Quando um termo for encontrado em um segmento da END de um documento, é sinal de que o termo realmente possui relevância em relação ao discurso.

A END resultante da interpretação do Texto 4.2 é mostrada na Figura 4.3.

Texto 4.2 a) O Sabiá furou a gaiola.
b) (e) voou.
c) A menina adorava o bichinho.
d) Ela chorou.

A partir da estrutura vista na Figura 4.3, é criada a ENDB do documento em questão (Figura 4.4). Como se pode observar, a relação de co-referência entre os termos *sabiá* e

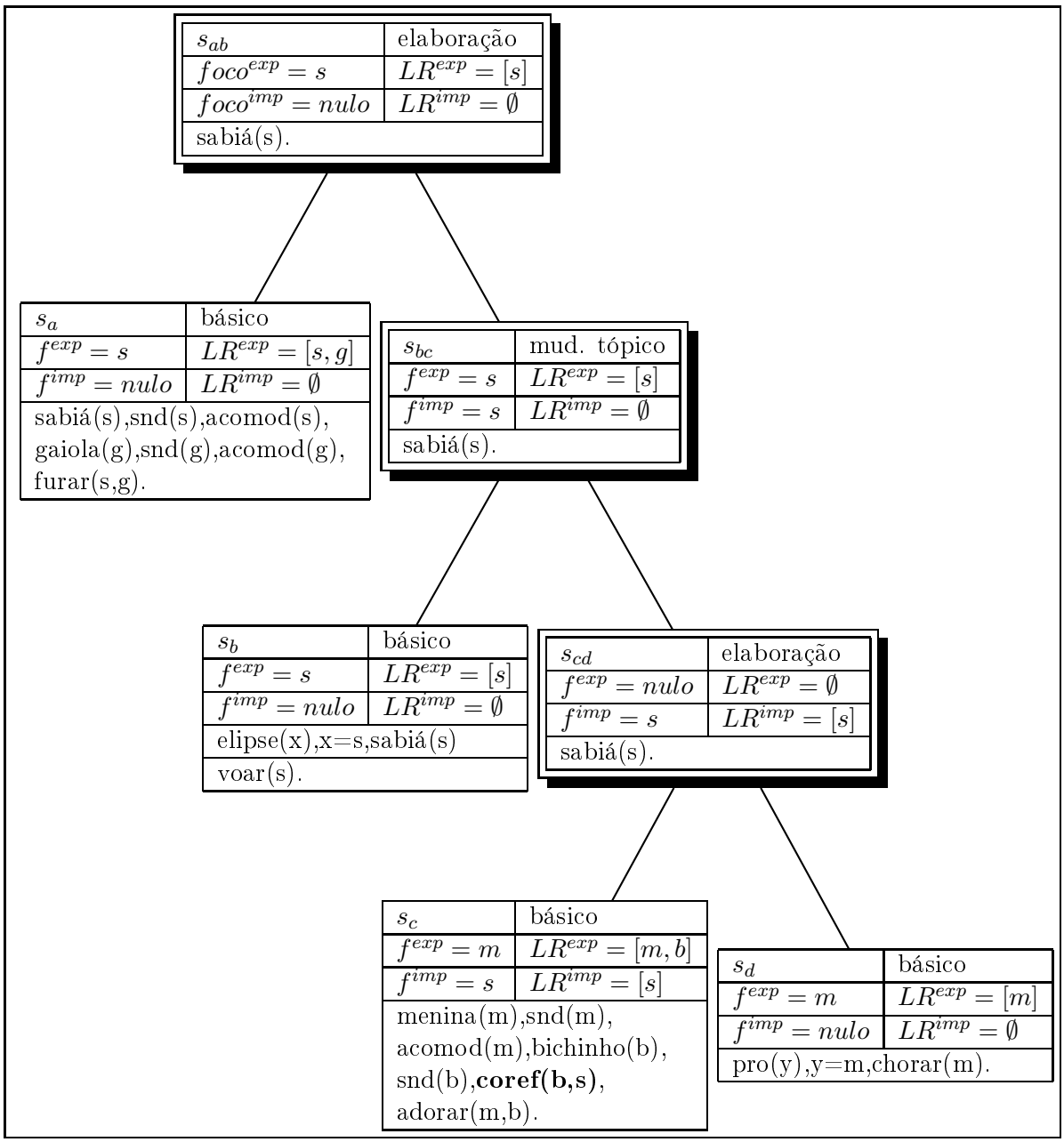


Figura 4.3: END do Texto 4.2

bichinho faz com que tais termos sejam sinônimos na ENDB resultante. Portanto, todas as ocorrências de *bichinho* são substituídas por *sabiá*.

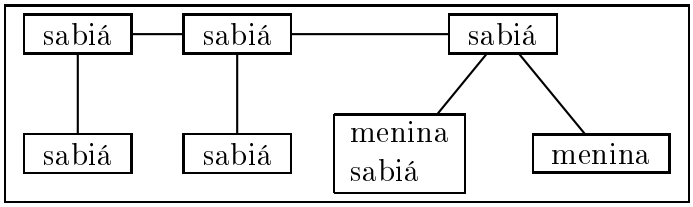


Figura 4.4: ENDB do Texto 4.2

4.3 Dicionário de sinônimos

Durante o processo de resolução de anáforas, mostrado no capítulo 3, uma anáfora nominal definida pode ser resolvida através de 4 regras pragmáticas: *co-referência*, *membro de*, *parte de* e *subcategorizado por*. A regra *co-referência* pode ser aplicada quando a expressão anafórica tiver sido introduzida no discurso por meio de um pronome, de uma elipse ou de um sintagma nominal definido que concorde em número e gênero com o seu antecedente. Nesse último caso, apenas o antecedente aparece como foco na END, já que a expressão anafórica apenas referencia a entidade introduzida por ele no discurso. Por exemplo, na END extraída do texto a seguir:

Texto 4.3 O *sabiá* fez um buraquinho na gaiola.
A menina gostava muito do **bichinho**.

o termo **bichinho** não aparece na estrutura como foco do discurso, apesar de a relação de co-referência estar presente na END. No lugar dele está o termo *sabiá* pois, segundo o processamento de anáforas, ambos referenciam a mesma entidade. Uma busca pelo termo **bichinho** na END não retornaria, portanto, o texto em questão. Entretanto, já que o próprio texto deixa claro que, nele, **bichinho** e *sabiá* são co-referências, os dois termos deveriam ter a mesma relevância em relação ao discurso.

Este trabalho sugere a criação de um dicionário de sinônimos (DS) para cada documento submetido ao processamento de anáforas, com base nas relações de *co-referência* armazenadas na END. A geração do dicionário pode ser feita ao mesmo tempo que a transformação da END em ENDB. O dicionário permite que, antes de a busca por um termo na ENDB ser realizada, seja verificado se esse termo foi substituído na estrutura por algum sinônimo. Em outras palavras, ao se buscar pelo termo **bichinho** no Texto 4.3, a resposta é a mesma que a de uma busca pelo termo *sabiá*, o que significa que a mesma quantidade de informação é introduzida no texto sobre esses dois termos.

O mesmo acontece com os termos *Iracema* e **virgem** no livro *Iracema*, de José de Alencar, como pode-se notar no trecho "Quando ele transmontou o vale e ia penetrar na mata, surgiu um vulto de *Iracema*. **A virgem** seguira o estrangeiro como a brisa sutil que resvala sem murmurejar por entre a ramagem.". A identificação desses dois sinônimos é essencial para a interpretação do livro, pois nele o termo **virgem** referencia 112 vezes a entidade introduzida no discurso pelo termo *Iracema*, que por sua vez ocorre 217 vezes

no discurso.

O dicionário de sinônimos é usado apenas em buscas no documento do qual ele foi extraído. Isso fica claro notando-se que no Texto 4.4:

Texto 4.4 *Márcio* se casará com Kelly.
O noivo vai levá-la para Campinas.

os termos *Márcio* e *noivo* são tratados como sinônimos. Neste caso, o resultado da busca por *noivo* é o mesmo da busca por *Márcio*. Não é verdade, entretanto, que em todos os documentos existentes os dois termos se co-referenciam.

Os termos *animalzinho* e **bichinho**, entretanto, são sinônimos na maioria dos casos. Uma solução para essa questão é utilizar, além do dicionário de sinônimos local ao documento, um dicionário de sinônimos global (DSG), que representaria *co-referências* do senso comum. Esse dicionário global não faz parte da metodologia proposta neste trabalho mas, caso ele esteja disponível, não há problemas em utilizá-lo.

O restante deste trabalho considera que a busca por um termo qualquer na ENDB passa, implicitamente, pela busca por seus sinônimos no DS.

4.4 Medida da quantidade de informação introduzida por uma entidade

A geração da ENDB e do DS permite que seja calculada a relevância de cada termo presente na estrutura em relação ao documento original. Em (FREITAS, 2005), Freitas sugere que as duas características a seguir devem ser consideradas no cálculo dessa relevância:

1. Nota-se que a Figura 4.2 representa uma seqüência de subárvores ordenadas pelos segmentos visíveis. Cada SV representa um assunto que ficou visível após a interpretação do documento. Mais ainda: a lista de SVs constitui a forma pela qual os assuntos foram sendo conduzidos pelo transmissor. Assim, um assunto no início da lista possui um peso maior do que um assunto no final da lista.

2. Olhando para cada SV individualmente, observa-se que este possui uma subárvore agregada representando a forma na qual o assunto do SV (i.e. focos) foi desenvolvido no decorrer de um documento. Este desenvolvimento é também estruturado na

forma de árvore e considera que os nós pais são mais relevantes que nós filhos. Portanto nós mais próximos da raiz devem assinalar assuntos mais relevantes para um dado SV.

De acordo com a proposta de cálculo da relevância proposta em (FREITAS, 2005), localizando um conjunto de documentos relevantes $\mathcal{CD} = d_1, d_2, ..., d_i, ..., d_m$ para um dado termo t, deve-se então fazer sua classificação. O mecanismo de classificação calcula para cada documento d_i um determinado valor de relevância $VR(t, d_i)$. Valores maiores para $VR(t, d_i)$ significam documentos mais relevantes. O cálculo de $VR(t, d_i)$ leva em consideração as posições onde o termo t foi localizado na ENDB. A Figura 4.5 destaca essa relação entre a profundidade em que um termo foi encontrado e a influência que o mesmo deve ter no cálculo de $VR(t, d_i)$. Quanto mais próximo da raiz tiver sido localizado um termo, maior deve ser $VR(t, d_i)$ (i.e. menor profundidade). Quanto mais próximo da primeira subárvore (i.e. mais à esquerda) estiver a subárvore que contenha um termo localizado, maior deve ser $VR(t, d_i)$. Levando em conta estes critérios são atribuídos:

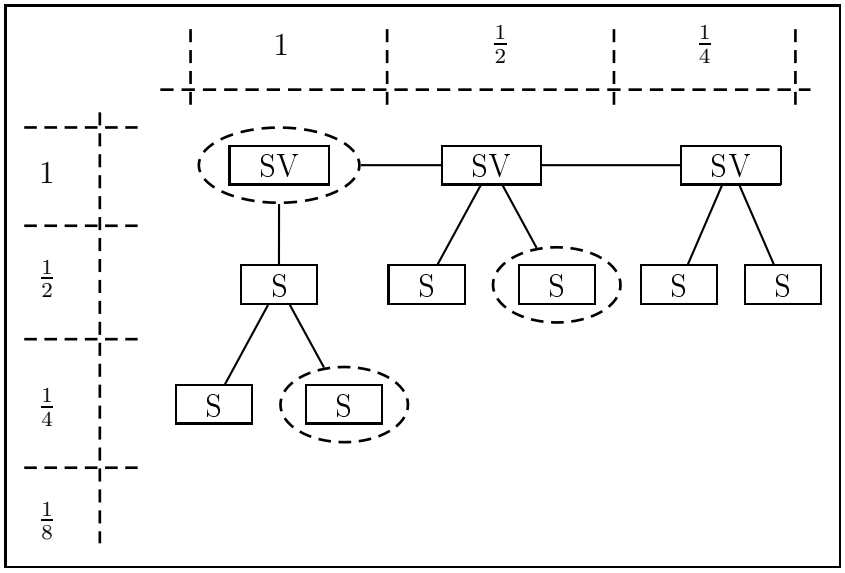


Figura 4.5: Pesos para o cálculo do valor de relevância

- Para cada nível de deslocamento horizontal, a fração $\frac{1}{2^v}$ onde $v = 0, 1, 2, ...(mv - 1)$ e mv é o número de segmentos visíveis existentes na ENDB de di_i (considerando 0 para a subárvore mais à esquerda e incrementando o valor para a direita);
- Para cada nível de profundidade, a fração $\frac{1}{2^p}$ onde $p = 0, 1, 2, ...(mp - 1)$ e mp é a máxima profundidade da árvore em um determinado nível de deslocamento horizontal.

Utilizando esses valores, o cálculo do valor de relevância de um certo termo t em um documento d_i é feito pela equação 4.1:

$$VR(t, d_i) = \prod_{v=0}^{(mv-1)} \left\{ \frac{1}{2^v} \sum_{p=0}^{(mp-1)} \left[\frac{1}{2^p} f\left(t, d_i, v, p\right) \right] \right\}, \tag{4.1}$$

onde:

$$f(t, d_i, v, p) = \begin{cases} 0 & \text{caso } t \text{ não seja encontrado nas coordenadas } (v, p) \text{ de } d_i, \\ 1 & \text{caso contrário.} \end{cases} \tag{4.2}$$

Essa é a forma de se calcular a relevância de uma entidade em um dado documento, segundo a proposta de (FREITAS, 2005). Na Figura 4.5, considerando que o termo foi localizado apenas nos segmentos circulados, o valor de relevância seria dado por:

$$VR(t, d_i) = \left[\frac{1}{2^0} \times \left(\frac{1}{2^0} + \frac{1}{2^2} \right) \right] \times \left(\frac{1}{2^1} \times \frac{1}{2^1} \right) = \left(1 + \frac{1}{4} \right) \times \frac{1}{4} = \frac{5}{16}.$$

Dois problemas podem ser identificados nesse cálculo. O primeiro, mais facilmente perceptível, é que a fórmula 4.1 *não* pode ter um produtório. Para isso, basta notar que se o termo pesquisado no exemplo acima não aparecesse na segunda subárvore da ENDB, ele teria um valor de relevância $VR(t, d_i) = \frac{5}{4}$ (ou seja, maior do que o valor $VR(t, d_i) = \frac{5}{16}$ encontrado anteriormente). Isso não faz sentido, pois se a entidade voltou a ser foco do documento, é porque ela tem mais chances de ser o seu assunto principal. A forma mais intuitiva de solucionar essa incoerência é transformar o produtório da fórmula 4.1 em um somatório, gerando a fórmula 4.3.

$$VR(t, d_i) = \sum_{v=0}^{(mv-1)} \left\{ \frac{1}{2^v} \sum_{p=0}^{(mp-1)} \left[\frac{1}{2^p} f\left(t, d_i, v, p\right) \right] \right\} \tag{4.3}$$

Utilizando a fórmula 4.3, a incoerência acima não aconteceria mais. Entretanto, nota-se ainda um segundo problema: os valores das razões associados aos segmentos decrescem rapidamente à medida que se caminha horizontalmente ou verticalmente na estrutura. Por exemplo, se uma entidade se encontra nas raízes de todas as subárvores (a partir da sexta) da ENDB, os fatores presentes no cálculo de sua relevância se assemelham aos termos da *Progressão Geométrica* (PG) abaixo:

$$PG = \left\{ \frac{1}{2^5}, \frac{1}{2^6}, \frac{1}{2^7}, \ldots \right\} = \left\{ \frac{1}{32}, \frac{1}{64}, \frac{1}{128}, \ldots \right\},$$

cujo último elemento tende a se aproximar de 0 (zero). A soma dos termos dessa PG é dada por:

$$\frac{a_n \times q - a_1}{q-1} \simeq \frac{0 \times \frac{1}{2} - \frac{1}{2^5}}{\frac{1}{2} - 1} = \frac{-\frac{1}{2^5}}{-\frac{1}{2}} = \frac{1}{2^4} = 0.0625,$$

onde a_n é o último termo da PG (que se aproxima de 0), $q = \frac{1}{2}$ é a razão e a_1 é o primeiro termo (nesse caso, $a_1 = \frac{1}{2^5}$, já que o termo está presente apenas a partir da sexta subárvore) da PG. O valor encontrado (0.0625) é menor que o valor 1 (obtido caso o termo fosse encontrado apenas no primeiro segmento visível), o que significa que, mesmo que uma entidade apareça na raiz de *todas* as subárvores a partir da sexta, ela nunca terá uma relevância maior do que se ela estiver presente *apenas* na raiz da primeira. Entretanto, nem sempre o assunto do documento aparece nas 5 primeiras subárvores do documento. Se o termo contido no primeiro segmento é realmente o assunto principal do texto, então ele continua presente no restante da ENDB. Esse problema pode ser contornado de duas formas simples:

1. Alterando as fórmulas das frações $\frac{1}{2^p}$ e $\frac{1}{2^v}$ para, respectivamente, $\frac{1}{p+1}$ e $\frac{1}{v+1}$, gerando novos pesos para cada segmento no cálculo da relevância. Esses novos pesos são mostrados na Figura 4.6.

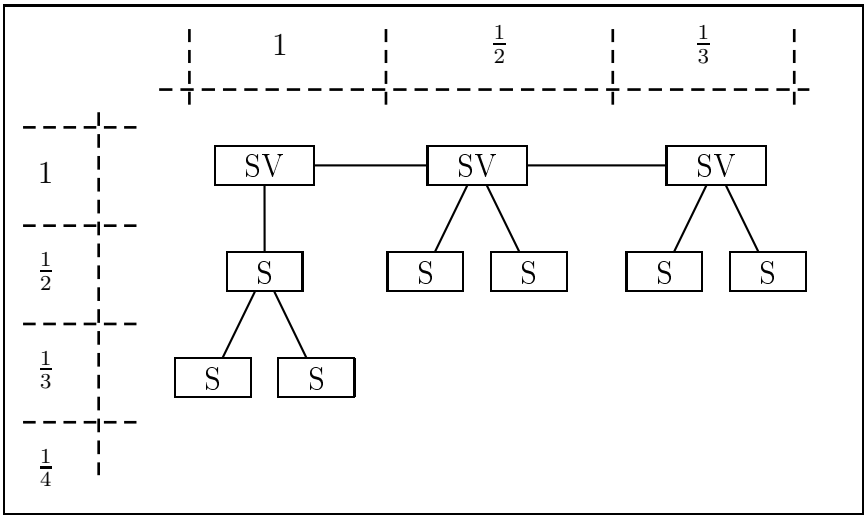


Figura 4.6: Novos pesos para o cálculo do valor de relevância

Dessa forma, o valor da relevância de uma entidade decresce menos à medida que ela se encontra mais à direita ou mais abaixo da estrutura, como se pode ver na Figura 4.7. O gráfico da figura mostra a fração $\frac{1}{x+1}$ e a fração $\frac{1}{2^x}$ (ambas em função de x).

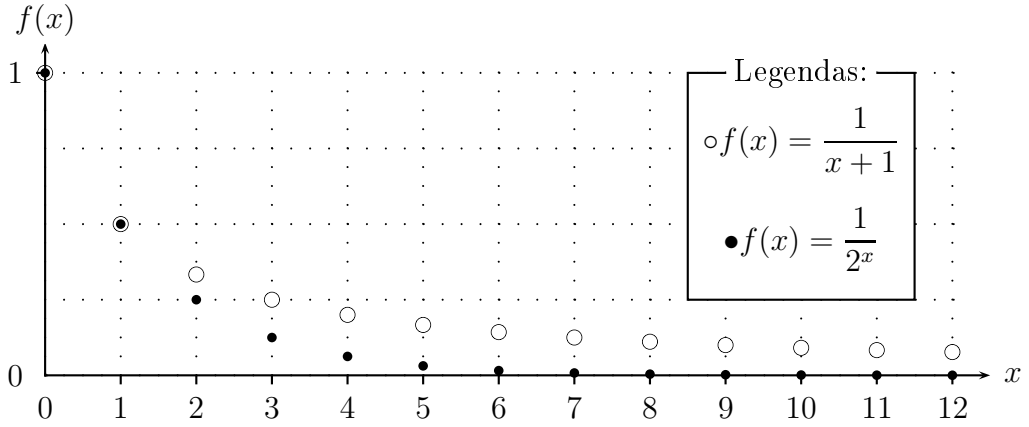


Figura 4.7: Decréscimo de $\frac{1}{x+1}$ e $\frac{1}{2^x}$ com o aumento de x

Dessa forma, a fórmula 4.3 é então substituída por 4.4:

$$VR(t,d_i) = \sum_{v=0}^{(mv-1)} \left\{ \frac{1}{v+1} \sum_{p=0}^{(mp-1)} \left[\frac{1}{p+1} f\,(t,d_i,v,p) \right] \right\}; \tag{4.4}$$

2. Desconsiderando a idéia de que entidades que aparecem no início da estrutura são mais relevantes. Tal idéia parte do princípio de que o assunto principal do texto é citado logo no início dele e, por estar no foco da atenção do leitor, não precisa ser citado explicitamente de novo. Entretanto, como este trabalho utiliza a ENDB gerada a partir da *interpretação* do texto, a entidade citada no início do texto continua aparecendo ao longo da estrutura caso seja realmente o seu assunto. Se isso não ocorrer, a entidade não é o assunto principal do texto, apesar de ter aparecido em seu início. Assim, a fração $\frac{1}{v+1}$ é retirada da formula 4.4, gerando a fórmula 4.5:

$$VR(t,d_i) = \sum_{v=0}^{(mv-1)} \sum_{p=0}^{(mp-1)} \left[\frac{1}{p+1} f\,(t,d_i,v,p) \right] \tag{4.5}$$

   Cada nó interno da estrutura é resultante da interpretação de duas frases contidas nas folhas. Quanto mais próximo da raiz estiver um nó, mais vezes ele foi submetido à interpretação das frases. Por isso, a fração $\frac{1}{2^p}$ não precisa ser retirada da fórmula.

Das observações acima, surgem três propostas para o cálculo da relevância de uma entidade em um documento, listadas a seguir:

a) **1ª Proposta**: Utiliza-se a fórmula 4.3, na qual o produtório de 4.1 foi transformado em um somatório. Este é o cálculo menos aconselhado a ser usado pois, como

foi explicado anteriormente, os valores das razões associadas aos segmentos decrescem muito rápido à medida que se caminha horizontalmente ou verticalmente na estrutura;

b) **2ª Proposta**: Utiliza-se a fórmula 4.4, na qual as frações $\frac{1}{2^v}$ e $\frac{1}{2^p}$ foram substituídas por $\frac{1}{v+1}$ e $\frac{1}{p+1}$. Este cálculo diminui o problema existente na proposta anterior, mas ainda parece pior do que a *3ª Proposta*, explicada a seguir;

c) **3ª Proposta**: Utiliza-se a fórmula 4.5, na qual a fração $\frac{1}{v+1}$ foi eliminada. Este cálculo aproveita melhor o processo de interpretação do documento, dando mais relevância ao que o autor realmente escreveu.

Cada um dos três cálculos gera ainda um *valor de relevância relativo*, calculando-se a razão entre o valor de relevância encontrado e o valor de relevância que um termo teria se aparecesse em todos os segmentos da ENDB. O novo valor significa o quanto o termo é relevante em relação ao documento como um todo. Enquanto o valor de relevância original (também chamado de *valor de relevância absoluto*) mede a quantidade de informação introduzida pelo termo, o *valor de relevância relativo* mede quanto o termo é importante no documento. Um *valor relativo* próximo de 1 significa que a entidade introduzida pelo termo foi referenciada em todo o decorrer do documento. Utilizando *3ª Proposta*, o cálculo da relevância absoluta do termo *sabiá* na ENDB do Texto 4.2, mostrada na Figura 4.4, é feito da seguinte forma:

$$VR_{ABS}(sabia, 4.2) = \left(\frac{1}{1} + \frac{1}{2}\right) + \left(\frac{1}{1} + \frac{1}{2}\right) + \left(\frac{1}{1} + \frac{1}{2}\right) = \frac{9}{2} = 4.5$$

O cálculo da relevância do termo *menina* na ENDB do mesmo texto é:

$$VR_{ABS}(menina, 4.2) = \frac{1}{2} + \frac{1}{2} = 1$$

O cálculo do valor de relevância máximo do texto em questão é:

$$VR_{MAX}(4.2) = \left(\frac{1}{1} + \frac{1}{2}\right) + \left(\frac{1}{1} + \frac{1}{2}\right) + \left[\frac{1}{1} + \left(\frac{1}{2} + \frac{1}{2}\right)\right] = \frac{10}{2} = 5$$

A relevância relativa do termo *sabiá*, portanto, é dada por:

$$VR_{REL}(sabia, 4.2) = \frac{4.5}{5} = 90\%$$

O cálculo da relevância relativa do termo *menina* é:

$$VR_{REL}(menina, 4.2) = \frac{1}{5} = 20\%$$

4.5 Índice de documentos interpretados

O processamento das anáforas de um documento gera uma END e a partir dela são criados a ENDB e o DS. Para que os documentos disponíveis não precisem ser interpretados novamente a cada busca, é proposta também a criação de um *Índice de Documentos Interpretados* (IDI). Esse índice contém, para cada documento d_i disponível na coleção:

- Uma tabela com os termos presentes na ENDB de d_i e seus respectivos *valores de relevância absolutos* (calculados de acordo com alguma das três fórmulas apresentadas na seção 4.4). Essa tabela é chamada de *tabela de valores de relevância*;
- O *valor de relevância absoluto* que um termo teria caso estivesse presente em todos os segmentos da ENDB de d_i (VR_{MAX}). Esse valor é usado para o cálculo do *valor de relevância relativo* de um termo em d_i, não sendo necessário armazenar o *valor de relevância relativo* de todos os termos presentes em d_i;
- O dicionário de sinônimos de d_i.

A Figura 4.8 mostra como o Texto 4.2 pode ser armazenado no índice de documentos interpretados, utilizando a *3ª Proposta* de cálculo de relevância apresentada na seção 4.4.

Documento	*Valores de relevância*	VR_{MAX}	*Dicionário de sinônimos*
Texto 4.2	*Termo* | VR_{ABS} sabiá | 4.5 menina | 1	5	*Termo* | *Co-ref* bichinho | 2 menina | -1 sabiá | -1

Figura 4.8: Um do documentos armazenados no IDI

Armazenar o índice é suficiente para que as buscas possam ser realizadas.

O dicionário de sinônimos mostrado na Figura 4.8 apresenta todos os termos contidos no documento original. Caso um termo t referencie uma entidade introduzida no discurso por outro termo s (seu sinônimo), o DS armazena com o termo t o índice de s no dicionário. No exemplo, o termo *bichinho* co-referencia o termo *sabiá*, que ocupa o índice 2 do DS.

Por isso, o número 2 aparece ao lado de *bichinho*. O valor (-1) ao lado dos termos *sabiá* e *menina* indicam que esses termos não fazem referência a entidades introduzidas por outros termos. O capítulo 5 apresenta mais detalhes sobre esse armazenamento.

4.6 Busca de termos simples

Após o índice de documentos interpretados ser criado, o sistema pode ser disponibilizado ao usuário. O principal objetivo do sistema é proporcionar a busca em documentos a partir da resolução das anáforas nos mesmos. Para definir uma busca, o usuário escolhe as palavras-chave que possam caracterizá-la. Caso a *query* contenha apenas um termo t (busca de termos simples), ela é feita da seguinte forma:

- O primeiro passo a ser feito é percorrer o IDI em busca dos documentos nos quais o termo t ocorre. Para cada documento d_i presente no índice, busca-se por t no dicionário de sinônimos do documento. Se o dicionário indicar que t co-referencia outro termo s, o restante do processo de busca deve ser feito com esse termo s, ao invés de t;

- Em seguida, busca-se por t (ou por seu sinônimo) na *tabela de valores de relevância* de d_i. Se o usuário deseja obter o maior número possível de informações sobre o termo, opta-se pelo *valor de relevância absoluto* de t. Caso ele queira encontrar documentos cujos assuntos principais sejam o termo buscado, opta-se pelo *valor de relevância relativo* do termo. Se o termo não se encontra na tabela, o documento não é relevante. Caso contrário, d_i e o valor de relevância de t são incluídos em uma lista de documentos relevantes;

- A coleção com os documentos relevantes encontrada é ordenada decrescentemente pelo valor de relevância do termo no documento;

- Por fim, a coleção de documentos já ordenada é apresentada ao usuário como resultado de sua *query*. O primeiro documento da coleção resultante é encarado como o mais relevante para o usuário, o documento seguinte é o segundo mais relevante, e assim por diante. O último documento apresentado é o menos relevante em relação a t.

O algoritmo 4.1, mostrado a seguir, descreve a busca pelo termo t no índice de documentos interpretados. O algoritmo apresenta um laço principal que percorre cada entrada

do índice de documentos interpretados. Cada entrada contém o identificador de um documento, a tabela de valores de relevância dos termos presentes em sua END, seu valor de relevância máximo e seu dicionário de sinônimos. Em cada iteração, o algoritmo procura por algum sinônimo de t no documento e, em seguida, calcula o valor de relevância absoluto do termo nesse documento. Caso VR_{ABS} de t seja diferente de zero, o documento e o VR_{ABS} são inseridos na lista de documentos relevantes, que no final é ordenada decrescentemente em relação aos valores de relevância de t em cada documento.

Algoritmo 4.1: Busca por um termo no IDI

```
1  {Entrada:
2   - t = termo buscado
3   - IDI = índice com os documentos interpretados
4   - tipo_busca = indica se o valor de relevância buscado é o valor
5                  relativo ou o absoluto}
6  ------------------------------------------------------------------
7  {Variáveis:
8   - DOC = um elemento do IDI (cada elemento do índice contém o
9           identificador de um documento, a tabela com os valores de
10          relevância dos termos presentes no documento, seu valor de
11          relevância máximo e seu dicionário de sinônimos)
12  - d = identificador de um documento
13  - TVR = tabela de valores de relevância dos termos de um documento
14          (cada elemento da tabela possui um termo e seu valor de
15          relevância em relação ao documento)
16  - VR_MAX = valor de relevância máximo de um documento
17  - DS = dicionário de sinônimos de um documento (cada elemento de
18         DS possui o nome de um termo e o índice de seu sinônimo
19         no próprio dicionário)
20  - LDR = lista de documentos relevantes, inicialmente vazia
21  - index = índice de um termo em um documento
22  - s = sinônimo de um termo em um documento
23  - VR = valor de relevância relativo de um termo em um documento
24 ------------------------------------------------------------------
25 {Saída: a lista ordenada com os documentos relevantes em relação
26        ao termo t}
27 ------------------------------------------------------------------
28 LDR ← [ ]
29 Para todo DOC ∈ IDI faça
```

```
30      {As quatro funções a seguir acessam cada item de DOC:
31       identificador de um documento; tabela com valores de relevância
32       dos termos presentes no documento; valor de relevância máximo do
33       documento; e dicionário de sinônimos do documento}
34      d ← identificador(DOC)
35      TVR ← tabela_relevancia(DOC)
36      VR_MAX ← vr_max(DOC)
37      DS ← dicionario(DOC)
38      {O primeiro passo é verificar se t possui um sinônimo no
39       documento atual. Como DS é um array ordenado, a pesquisa
40       é binária}
41      index ← pesquisa_binária(t, DS)
42      {O retorno da pesquisa binária é a posição do sinônimo de t
43       no próprio DS. Caso o índice retornado seja (−1), o termo
44       não co−referencia outro no documento}
45      Se index ≠ −1 então
46          s ← termo(DS, index)
47          {A função acima retorna o termo na posição index do DS}
48          {Em seguida, t assume o valor de seu sinônimo}
49          t ← s
50      Fim então
51      {O passo seguinte é descobrir a relevância de t (ou de seu
52       sinônimo) no documento}
53      {A pesquisa binária, neste caso, retorna o valor de relevância
54       associado ao termo buscado na tabela de relevância}
55      VR ←pesquisa_binária(t, TVR)
56      {Em seguida, deve−se adicionar à lista de documentos relevantes
57       o documento associado ao seu valor de relevância}
58      Se tipo_busca = ''RELATIVO'' e VR > 0 então
59          LDR ← LDR + [(d, VR)]
60      Fim então
61      Senão
62          LDR ← LDR + [(d, VR/VR_MAX)]
63      Fim senão
64  Fim para todo
65
66  {Por fim, a lista de documentos relevantes deve ser ordenada
67   decrescentemente em relação ao valor de relevância de t nos
```

```
68   documentos}
69 ordenação(LDR)
70 retorne LDR
```

4.7 Busca de termos compostos

A necessidade de informação do usuário costuma ser representada através de uma *expressão de busca*, apesar de que, para o usuário, é difícil saber quais palavras estarão presentes nos documentos que satisfaçam sua necessidade. Em geral, para fazer uma pesquisa em documentos digitais, o usuário digita um conjunto de palavras e espera encontrar, dentre todos os documentos disponíveis, aquele(s) que introduza(m) o maior número possível de informações sobre o conjunto de palavras buscado.

Uma *busca de termos compostos* difere-se de uma *busca de termos simples* por permitir que vários termos sejam pesquisados ao mesmo tempo em uma única busca. A busca pode ser representada como uma conjunção de termos, uma disjunção de termos, ou mesmo uma combinação entre conjunções e disjunções de termos. Caso eles sejam apresentados sem o usuário explicitar se trata-se de uma conjunção ou de um disjunção de termos, considera-se que foi feita uma conjunção.

Uma conjunção pode ser explicitada, por exemplo, das seguintes formas: (1) sabiá *and* gaiola *and* menina; (2) sabiá *e* gaiola *e* menina; (3) sabiá *&* gaiola *&* menina; ou (4) sabiá gaiola menina.

Uma disjunção pode ser explicitada, por exemplo, das seguintes formas: (1) sabiá *or* gaiola *or* menina; (2) sabiá *ou* gaiola *ou* menina; ou (3) sabiá | gaiola | menina.

Cada um desses dois tipos de busca é descrito nas seções seguintes.

4.7.1 Conjunção de termos

Em uma conjunção de termos espera-se, acima de tudo, que todos os termos buscados estejam presentes na coleção de documentos retornada ao usuário. A primeira ação a ser tomada, portanto, é encontrar essa coleção de documentos. Em seguida, é necessário que se faça uma ordenação dessa coleção, como foi feita na *busca de termos simples*.

Em uma conjunção, o documento que não apresenta determinado termo (i.e. aquele cuja *tabela de valores de relevância* no IDI não apresenta algum dos termos buscados) não

é relevante. Seguindo esse raciocínio, se existem dois documentos que apresentam todos os termos buscados, qual é o mais relevante? Para responder essa pergunta, considere os valores de relevância obtidos da *tabela de valores de relevância* de algum documento no IDI, mostrados na tabela 4.1.

	Termo 1	*Termo 2*
Documento 1	10	31
Documento 2	5073	1

Tabela 4.1: Exemplo de valores de relevância obtidos do IDI

Pela tabela 4.1, nota-se que o segundo documento apresenta pouquíssima informação sobre o segundo termo da busca. Esse documento pode ser considerado como o menos relevante para o usuário, que espera receber o maior número possível de informações sobre todos os termos buscados. Em outras palavras, apesar de o *Documento 2* citar mais vezes o primeiro termo do que o *Documento 1*, ele praticamente nem cita o *Termo 2*. Como o usuário deseja obter informação sobre *ambos* os termos, o *Documento 2* poderia até ficar fora da coleção final. Por isso faz sentido considerá-lo como menos relevante.

De forma geral, para se ordenar a coleção de documentos que possuem todos os termos citados, ordena-se primeiramente a lista de *valores de relevância* de cada um dos documentos da coleção. Em seguida, ordena-se decrescentemente os documentos pelo primeiro elemento da lista de relevância de cada um deles. Se o primeiro elemento da lista de mais de um documento forem iguais, elas são ordenadas decrescentemente pelo segundo elemento de suas listas de relevâncias, e assim sucessivamente. No fim da ordenação, a lista ordenada é dada como resultado ao usuário, assim como na *busca de termos simples*.

Supondo agora que uma determinada busca pelos termos a, b, c e d tenha resultado na coleção com os documentos 1, 2, 3 e 4, e que os valores de relevância dos termos buscados em cada documento são os mostrados na tabela 4.2, é mostrado a seguir um exemplo de cada iteração da ordenação desses documentos.

	Termo 1	*Termo 2*	*Termo 3*	*Termo 4*
Documento 1	10	31	53	78
Documento 2	5073	1	8	17
Documento 3	50	1	10	17
Documento 4	50	10	31	78

Tabela 4.2: Valores de relevância a serem ordenados em uma conjunção

Estas são as listas de relevâncias iniciais de cada um dos documentos que contêm os termos a, b, c e d:

```
Documento 1: [10, 31, 53, 78]
Documento 2: [5073, 1, 8, 17]
Documento 3: [50, 1, 10, 17]
Documento 4: [50, 10, 31, 78]
```

O primeiro passo é ordenar a lista de valores de relevância de cada documento:

```
Documento 1: [10, 31, 53, 78]
Documento 2: [1, 8, 17, 5073]
Documento 3: [1, 10, 17, 50]
Documento 4: [10, 31, 50, 78]
```

Em seguida, ordena-se decrescentemente os documentos pelo primeiro elemento da lista de cada um:

```
Documento 1: [10, 31, 53, 78]
Documento 4: [10, 31, 50, 78]
Documento 2: [1, 8, 17, 5073]
Documento 3: [1, 10, 17, 50]
```

Da mesma forma, como houve empate nos valores de alguns documentos, ordena-se decrescentemente pelo segundo elemento os documentos que tiverem empatado na iteração anterior:

```
Documento 1: [10, 31, 53, 78]
Documento 4: [10, 31, 50, 78]
Documento 3: [1, 10, 17, 50]
Documento 2: [1, 8, 17, 5073]
```

Analogamente, ordena-se agora pelo terceiro elemento:

```
Documento 1: [10, 31, 53, 78]
Documento 4: [10, 31, 50, 78]
Documento 3: [1, 10, 17, 50]
Documento 2: [1, 8, 17, 5073]
```

Como não houve empate no terceiro elemento, não há mais o que ordenar. Portanto, os documentos são retornados ao usuário na seguinte ordem de relevância (do mais relevante para o menos relevante):

```
Documento 1, Documento 4, Documento 3, Documento 2
```

Outra forma de se ordenar a coleção de documentos é *multiplicar* todos os valores de relevância relativos aos termos buscados em cada documento. Cada documento terá, então, um valor único de relevância. Dessa forma, os documentos são ordenados decrescentemente por esse valor de relevância, como na *busca de termos simples*, e o resultado é fornecido ao usuário. Esse raciocínio também funciona para documentos que não apresentem algum dos termos procurados, pois nesse caso, o valor de relevância referente a esse termo é nulo, e o produto dos valores também sera nulo, fazendo com que o documento não seja considerado relevante. Ao contrário da abordagem anterior, esta considera que o segundo documento da tabela 4.1 é mais relevante que o primeiro, pois o valor $(5073 \times 1 = 5073)$ é maior que $(10 \times 31 = 310)$.

Ambas as abordagens são corretas. Isso ocorre porque a relevância de uma busca depende do que o usuário realmente deseja como resultado. Entretanto, este trabalho sugere a utilização do primeiro tipo de ordenação, que aproveita melhor a estrutura gerada pela interpretação do documento.

4.7.2 Disjunção de termos

Uma disjunção de termos em uma busca menos comum de ser feita do que uma conjunção. Quando um usuário procura por *sabiá OU gaiola*, que tipo de relevância ele está buscando realmente? Sabe-se que, ao contrário da conjunção, os documentos resultantes não precisam conter todos os termos da busca, mas como ordenar tais os documentos?

Em uma disjunção de termos espera-se, acima de tudo, que *pelo menos um* dos termos buscados esteja presente na coleção de documentos retornada ao usuário. A primeira coisa a se fazer é, portanto, encontrar essa coleção de documentos. Para entender como é feita a ordenação dessa coleção de documentos, considere os valores de relevância obtidos do índice IDI mostrados na tabela 4.3.

	Termo 1	*Termo 2*
Documento 1	230	431
Documento 2	507	0

Tabela 4.3: Exemplo de valores de relevância obtidos do índice

Pela tabela, nota-se que o segundo documento apresenta uma quantidade maior de informação referente ao primeiro termo da busca. Esse documento pode ser considerado

como o mais relevante para o usuário, que espera receber o maior número possível de informações sobre *algum* dos termos buscados. Em outras palavras, apesar de o *Documento 2* falar bem menos sobre o segundo termo do que o *Documento 1*, a disjunção indica que o usuário está na dúvida de qual termo buscar. Por exemplo, uma disjunção possível é *dálmatas ou vira-latas*. Nela, pode-se concluir que o usuário deseja encontrar documentos que falem o máximo possível sobre *dálmatas*, ou que falem o máximo possível sobre *vira-latas*. Por isso faz sentido considerar o *Documento 2* da tabela 4.3 como mais relevante.

Portanto, para se ordenar a coleção de documentos que possuem algum dos termos citados, primeiro ordena-se decrescentemente a lista de *valores de relevância* de cada um dos documentos da coleção. Em seguida, ordena-se decrescentemente os documentos pelo primeiro elemento da lista de relevância de cada um deles. Se o primeiro elemento da lista de mais de um documento forem iguais, elas são ordenadas decrescentemente pelo segundo elemento de suas listas de relevâncias, e assim sucessivamente. No fim da ordenação, a lista ordenada é dada como resultado ao usuário, assim como na conjunção de termos.

Supondo agora que uma determinada busca pelos termos a, b, c ou d tenha resultado na coleção com os documentos 1, 2, 3 e 4, e que os valores de relevância dos termos buscados em cada documento são os mostrados na tabela 4.4, é mostrado a seguir um exemplo de cada iteração da ordenação desses documentos.

	Termo 1	*Termo 2*	*Termo 3*	*Termo 4*
Documento 1	230	431	53	78
Documento 2	507	0	8	17
Documento 3	50	1	10	17
Documento 4	50	10	31	78

Tabela 4.4: Valores de relevância a serem ordenados em uma disjunção

Estas são as listas de relevâncias iniciais de cada um dos documentos que contêm algum dos termos a, b, c ou d:

```
Documento 1: [230, 431, 53, 78]
Documento 2: [507, 0, 8, 17]
Documento 3: [50, 1, 10, 17]
Documento 4: [50, 10, 31, 78]
```

O primeiro passo é ordenar decrescentemente a lista de valores de relevância de cada documento:

```
Documento 1: [431, 230, 78, 53]
Documento 2: [507, 17, 8, 0]
Documento 3: [50, 17, 10, 1]
Documento 4: [78, 50, 31, 10]
```

Em seguida, ordena-se decrescentemente os documentos pelo primeiro elemento da lista de cada um:

```
Documento 2: [507, 17, 8, 0]
Documento 1: [431, 230, 78, 53]
Documento 4: [78, 50, 31, 10]
Documento 3: [50, 17, 10, 1]
```

Como já não houve empate no primeiro elemento, não há mais o que ordenar. Portanto, os documentos são retornados ao usuário na seguinte ordem de relevância (do mais relevante para o menos relevante):

```
Documento 2, Documento 1, Documento 4, Documento 3
```

Assim como na conjunção, outra forma de se ordenar a coleção de documentos é *somar* todos os valores de relevância relativos aos termos buscados em cada documento. Cada documento terá, então, um valor único de relevância. Dessa forma, os documentos são ordenados decrescentemente por esse valor de relevância, como na *busca de termos simples* e na *conjunção*, e o resultado é fornecido ao usuário. Ao contrário da abordagem anterior, essa considera que o primeiro documento da tabela 4.3 é mais relevante que o segundo, pois o valor $(230 + 431 = 661)$ é maior que $(507 + 0 = 507)$.

Novamente, ambas as abordagens são corretas, dependendo do que o usuário realmente deseja como resultado. Este trabalho sugere a utilização da primeira ordenação, que aproveita melhor a estrutura gerada pelo processo de interpretação do documento.

4.8 Considerações finais

Como pode ser observado, alguns procedimentos propostos neste trabalho admitem mais de uma forma de serem executados: (1) A seção 4.4 apresenta 3 formas de cálculo da relevância de um termo em um documento (equações 4.3, 4.4 e 4.5). Todas podem

ser utilizadas, mas a sugerida nesta dissertação é a 4.5 pois, devido ao processo de interpretação que é feito no texto, não é preciso considerar uma palavra como mais relevante que outra por ela aparecer antes de outra no discurso. (2) Além disso, são propostas duas formas de cálculo de relevância de uma *query* em relação a um documento (*valor de relevância relativo*, que mede quanto o termo é relevante em relação ao texto, e *valor de relevância absoluto*, que calcula a quantidade de informação que é introduzida no texto pelo termo em questão). Nesse caso, ambas podem ser disponibilizadas ao usuário. (3) Por último, são sugeridas também diferentes formas de se ordenar a lista de documentos retornada ao usuário numa conjunção (seção 4.7.1) ou numa disjunção (seção 4.7.2) de termos. Este trabalho considera que a primeira forma apresentada em cada uma delas é a mais indicada.

Vale ressaltar ainda que o processamento dos algoritmos propostos nesta metodologia se divide em duas categorias: o processamento *offline*, que inclui as tarefas que são feitas apenas uma vez, antes que as buscas sejam disponibilizadas aos usuários (geração da ENDB, dos dicionários de sinônimos e do IDI); e o processamento *online*, que compreende as funcionalidades que podem ser feitas diversas vezes a partir de quando o sistema é disponibilizado aos usuários (além das buscas, existe a possibilidade de se incluir novos documentos no índice de documentos interpretados). O capítulo 5 apresenta mais detalhes sobre os algoritmos referentes a cada etapa proposta neste trabalho.

"*O pensamento parece uma coisa à toa*
mas como é que a gente voa
quando começa a pensar."
Lupcínio Rodrigues

5 O Protótipo

"*Os fins justificam os meios.*"
Maquiavel

Este capítulo apresenta detalhes do protótipo construído neste trabalho, a partir da metodologia proposta, e analisa a complexidade de cada algoritmo. É possível, entretanto, que sejam feitas modificações em cada implementação desta metodologia.

5.1 O sistema

Foi implementado um protótipo da metodologia de recuperação de informação proposta neste trabalho. Vale lembrar que um sistema que baseado nesta metodologia funciona apenas em conjunto de um mecanismo de resolução das anáforas de um discurso, que deve retornar a END do documento analisado. O sistema descrito aqui utiliza o processamento de anáforas proposto em (FREITAS, 2005).

Este capítulo detalha as estruturas de dados (BENTLEY; SEDGEWICK, 1997; AHO et al., 1983; BENTLEY, 1982; CORMEN; LEISERSON CHARLES; RIVEST, 1990) utilizadas na implementação deste protótipo, e propõe outras estruturas que podem aperfeiçoá-lo. Além disso, são analisadas as complexidades (KNUTH, 1978; SEDGEWICK; FLAJOLET, 1996; AHO; HOPCROFT, 1974; KNUTH, 1998) de tempo de processamento dos algoritmos propostos. O objetivo da análise é encontrar apenas um limite superior para o tempo de execução do sistema.

As seções seguintes apresentam as principais características do sistema de busca implementado. A seção 5.2 apresenta o processamento *offline*, que inclui os pré-processamentos que precisam ser feitos apenas uma vez, antes de o sistema ser disponibilizado aos usuários. São eles: resolução das anáforas de um documento, transformação da END em ENDB, criação do dicionário de sinônimos e geração do índice com os documentos interpretados. Em seguida, a seção 5.3 analisa os algoritmos que são executados cada vez que uma *query* é feita por um usuário ou que um novo documento é incluído no IDI (processamento *online*).

5.2 Processamento *offline*

Antes que o sistema seja disponibilizado para o usuário, é necessário que todos os documentos disponíveis sejam interpretados e que o resultado dessa interpretação seja armazenado. Cada documento passa por um processo de:

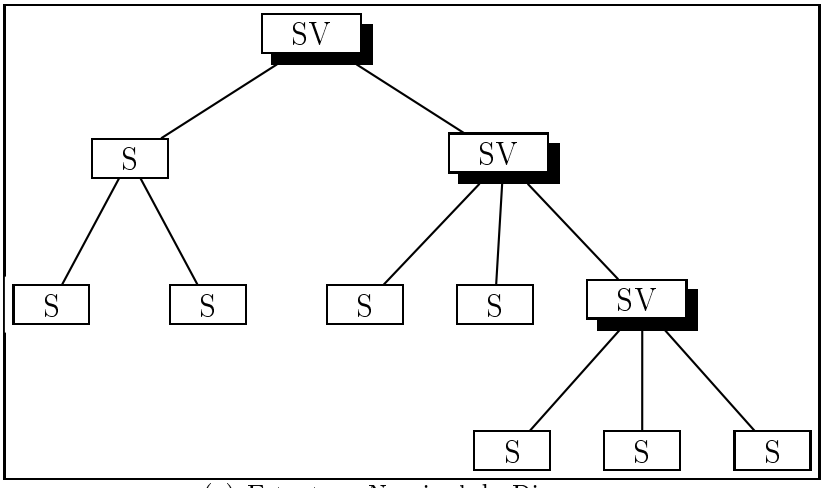
1. resolução de suas anáforas e criação de sua Estrutura Nominal do Discurso;
2. transformação da END em uma Estrutura Nominal do Discurso para Busca; e
3. criação de um dicionário com as palavras do discurso que se co-referenciam (dicionário de sinônimos).

Feito isso, deve ser criado o índice de documentos interpretados, contendo as informações de cada documento que precisam ser armazenadas para que as *queries* possam ser feitas.

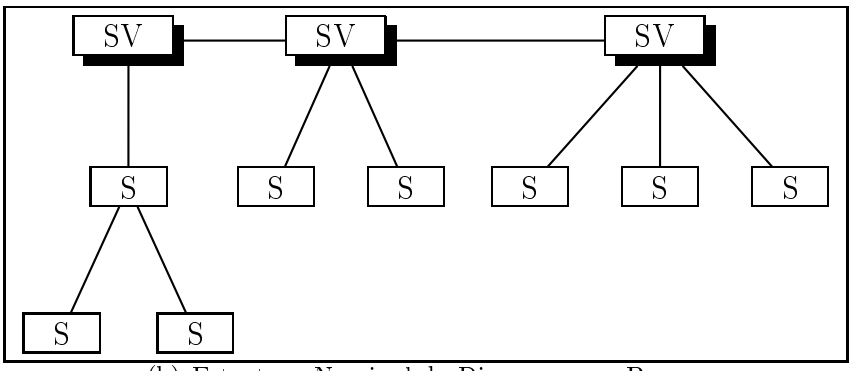
Como este trabalho utiliza o algoritmo proposto em (FREITAS, 2005) para resolver as anáforas dos documentos, sua complexidade não é discutida aqui. A complexidade dos demais algoritmos *offline* são discutidas nas seções seguintes. Alguns deles necessitam como entrada a END e uma lista $\mathcal{L}$ contendo os termos presentes na Estrutura Nominal do Discurso em ordem crescente.

5.2.1 Geração da ENDB

Após a END de um certo documento ser criada, o sistema gera sua ENDB. De forma geral, para que a Estrutura Nominal do Discurso seja transformada na ENDB, só é necessário transformar uma árvore (END) em uma lista (ENDB), como pode ser visto na Figura 5.1.



(a) Estrutura Nominal do Discurso



(b) Estrutura Nominal do Discurso para Busca

Figura 5.1: Transformação da END em ENDB

Cada segmento mais à direita da END (SV) passa a ser a raiz da árvore de seus subsegmentos, tornando-se elemento de uma lista (ENDB). Os segmentos contidos na ENDB herdam apenas os focos implícitos e explícitos dos segmentos da END. Portanto, é preciso percorrer cada segmento da Estrutura Nominal do Discurso (apenas uma vez), descobrindo seus focos. Além disso, se o segmento for um SV, ele é incluído no final de uma lista.

A complexidade da transformação da END em ENDB depende apenas da quantidade de segmentos presentes na estrutura original. Para calcular o número de segmentos da END de um documento D_i, considere que:

1. O documento D_i possui um número f_i de frases;
2. A interpretação de cada frase do discurso gera um segmento. Ao todo, esses segmentos ocupam f_i nós da estrutura;
3. A interpretação de cada duas frases consecutivas do discurso também gera um novo segmento, adicionando outros $(f_i - 1)$ novos nós internos à estrutura;
4. Eventualmente, alguns segmentos resultantes da interpretação das frases do discurso podem ser agrupados em um único segmento, diminuindo a quantidade de segmentos da estrutura.

Considerando que não houve agrupamentos entre segmentos, existem $(f_i + (f_i - 1) = 2 \times f_i - 1)$ segmentos na estrutura nominal de um discurso que contém f_i frases. Como esse é o número máximo de segmentos que a END pode conter, a complexidade de geração da ENDB de D_i é dada por:

$$T_{CRIAR-ENDB-D_I}(f_i) \in O(f_i), \tag{5.1}$$

já que cada segmento é analisado exatamente uma vez e a inclusão no final de uma lista (no caso de o segmento ser um SV) possui complexidade $\Theta(1)$. De 5.1, tem-se que a complexidade de gerar a ENDB de todos os n documentos disponíveis é:

$$T_{CRIAR-ENDB}(n) \in \sum_{i=1}^{n} O(f_i). \tag{5.2}$$

5.2.2 Dicionário de sinônimos

Para criar o dicionário com os sinônimos presentes em um documento, é necessário percorrer toda a END identificando as relações de co-referência presentes no discurso. Uma relação de co-referência é armazenada na estrutura na forma $coref(b, a)$, indicando que o termo b referencia a entidade introduzida no discurso pelo termo a. O DS deve indicar todos os termos que, assim como b, fazem referência à entidade introduzida por a.

Para um documento D_i contendo f_i frases, o número máximo de focos presentes em D_i é $(2 \times f_i)$, pois cada frase introduz no máximo dois novos focos ao discurso (foco implícito e explícito). Dessa forma, o dicionário pode ser representado como um *array* com $(2 \times f_i)$ elementos. Cada elemento possui uma estrutura contendo um termo da ENDB e o índice (relativo ao próprio *array*) do termo que ele co-referencia. Esse valor pode ser iniciado com (-1), indicando que o termo não co-referencia outro. O dicionário de cada documento D_i, portanto, é criado da seguinte forma:

- Cada segmento da ENDB de D_i é analisado, em busca das relações de co-referência existentes na estrutura. Quando é encontrada uma relação indicando que um termo b co-referencia uma entidade introduzida *a priori* no discurso por um termo a, deve ser descoberto o índice de a no *array*. Se o *array* estiver ordenado, é possível realizar uma busca binária, cuja complexidade é dada por (KNUTH, 1998):

$$T_{BUSCAR-ANTECEDENTE}(f_i) \in O(lg\ f_i). \tag{5.3}$$

- Em seguida, o termo b é buscado no *array*. O campo de b relativo ao termo que ele co-referencia passa a valer o índice de a no *array*, encontrado no passo anterior. Após encontrar o termo b, essa alteração pode ser feita em $\Theta(1)$. A complexidade deste passo, portanto, é:

$$T_{BUSCAR-EXPRESSAO-ANAFORICA}(f_i) \in O(lg\ f_i) + \Theta(1) \subset O(lg\ f_i). \tag{5.4}$$

O algoritmo 5.1 descreve os passos acima.

Algoritmo 5.1: Criação do dicionário de sinônimos

```
1 {Entrada:
2  - END = Estrutura Nominal do Discurso;
3  - L = Lista ordenada com os termos presentes na END.
4 ------------------------------------------------------
```

```
{Variáveis:
 - s = um segmento da END;
 - t = um termo presente na END;
 - CR = uma relação de co-referência presente na END;
 - b = um termo que referencia outro (expressão anafórica);
 - a = um termo que é referenciado por outro (antecedente);
 - DIC = lista que associa cada termo a seu possível sinônimo
         (inicialmente vazia);
 - i = index de um termo no dicionário;
 - j = index de um termo no dicionário.
-----------------------------------------------------------------
{Saída: a lista com todos os termos presentes na END do documento,
        ordenada pelo nome desses termos.}
-----------------------------------------------------------------
DIC ← [ ]
{A seguir, cada termo presente na END é associado, inicialmente, ao
 valor (−1) e adicionado no dicionário.}
Para todo t ∈ L faça
    DIC ← DIC + [(t, −1)]
Fim para todo
Para todo s ∈ END faça
    {Para cada coref(b, a) no documento, o índice de a é associado ao
     termo b no dicionário.}
    Para todo CR ∈ s faça
        CR ← coref(s)
        {As duas funções a seguir acessam os termos do discurso que
     se co-referenciam.}
        A ← antecedente(CR)
        B ← expressao_anafórica(CR)
        i ← pesquisa_binária(A, DIC)
        j ← pesquisa_binária(B, DIC)
        {A função a seguir associa o valor i ao termo na posição j
         do dicionário.}
        associar(DIC, j, i)
    Fim para todo
Fim para todo
retorne DIC
```

No Texto 5.2, só há dois termos que se co-referenciam: *sabiá* e **bichinho**. No Texto 5.1 a seguir, existem mais termos que se co-referenciam:

Texto 5.1 a) O cachorro se soltou da coleira.
b) Letícia tentou fugir do animal.
c) Mas o cão mordeu a mulher.

No Texto 5.1, os termos *cachorro*, *animal* e *cão* se co-referenciam. O mesmo acontece com os termos *Letícia* e *mulher*. O dicionário de sinônimos desse texto é ilustrado na Figura 5.2.

Termo	*Co-ref*
animal	1
cachorro	-1
cão	1
letícia	-1
mulher	4

Figura 5.2: Dicionário de Sinônimos

O número máximo de relações de co-referências presentes em uma ENDé $2 \times (f_i - 1)$, pois cada foco a partir da segunda frase pode referenciar uma entidade introduzida no discurso por outro termo. Com isso, pode-se concluir a partir de 5.3 e 5.4 que a complexidade de criação do dicionário de sinônimos de cada D_i é dada por:

$$T_{CRIAR-DS-D_I}(f_i) \in [2 \cdot (f_i - 1)] \cdot (O(lg\ f_i) + O(lg\ f_i)) \subset O(f_i \cdot lg\ f_i), \tag{5.5}$$

e que a complexidade da criação do dicionário de sinônimos de todos os n documentos disponíveis é dada por:

$$T_{CRIAR-DS}(n) \in \sum_{i=1}^{n} O(f_i \cdot lg\ f_i). \tag{5.6}$$

5.2.3 Índice de documentos interpretados

Após a criação da ENDB de cada documento disponível, cria-se o IDI (Índice de Documentos Interpretados – ver seção 4.6), que associa cada documento aos 3 itens listados a seguir:

- O valor de relevância que um termo possuiria caso estivesse presente em todos os segmentos da ENDB de um documento. Esse valor, chamado de *valor de relevância*

máximo (VR_{MAX}) é usado no cálculo do *valor de relevância relativo* de um termo. No exemplo da ENDB da Figura 5.3, há 3 segmentos no primeiro nível da lista (cada um contribui em 1 unidade nesse cálculo) e 4 no segundo (cada um deles contribui com o valor $\frac{1}{2}$). Portanto, o valor máximo que um termo alcançaria se aparecesse em todos os segmentos da ENDB da Figura 5.3 é:

$$VR_{max} = 3 \times 1 + 4 \times \tfrac{1}{2} = 5;$$

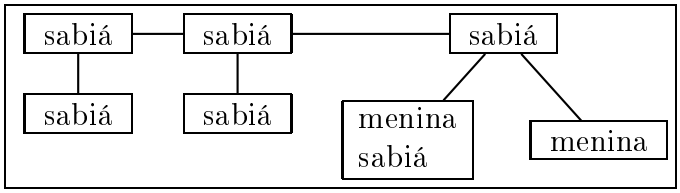


Figura 5.3: Exemplo de uma ENDB

Para calcular VR_{MAX}, é necessário percorrer cada segmento da estrutura, já que cada segmento possui um peso no cálculo da relevância de acordo com sua localização na END (ver capítulo 4). Como o número de segmentos é, no máximo, $(2 \times f_i - 1)$, então a complexidade para calcular o valor de relevância máximo e associá-lo ao índice é:

$$T_{CALCULAR-VR_{MAX}}(f_i) \in O(f_i) + \Theta(1) \subset O(f_i); \tag{5.7}$$

- Uma tabela denominada *Tabela de Valores de Relevância* ($\mathcal{TVR}$), associando cada termo presente na END ao seu valor de relevância absoluto (VR_{ABS}), que é calculado de acordo com a fórmula 4.5, mostrada no capítulo 4. Se o termo não aparece na ENDB do documento, ele não é armazenado na lista. Portanto, não é considerado pelo sistema de busca como um dos assuntos centrais do documento em questão. As duas primeiras colunas da tabela 5.1 ilustram a $\mathcal{TVR}$ criada pelo IDI com os termos da ENDB da Figura 5.3. A última coluna da tabela mostra o valor de relevância relativo (VR_{REL}) de cada termo.

Termo	VR_{ABS}	VR_{REL}
sabiá	4,5	0,9
menina	1	0,2

Tabela 5.1: Tabela de Valores de Relevância

Considerando que a lista $\mathcal{L}$ contendo os termos presentes na ENDB está ordenada e possui no máximo $(2 \times f_i)$ elementos, a criação da lista $\mathcal{TVR}$ que associa cada termo a seu valor de relevância pode ser feita de duas formas:

1. Para cada termo presente na ENDB de um documento D_i, percorre-se essa estrutura calculando o valor de relevância do termo. Nesse caso, a complexidade de criação de $\mathcal{TVR}$ seria:

$$T_{CRIAR-TVR}(f_i) \in O(f_i \cdot f_i) \in O(f_i^2), \tag{5.8}$$

   para cada documento D_i, já que a ENDB possui no máximo $(2 \times f_i - 1)$ segmentos e existem no máximo $(2 \times f_i)$ termos na Estrutura Nominal do Discurso;

2. O cálculo da relevância de todos os termos de um documento D_i pode ser feito percorrendo a ENDB apenas uma vez. Como cada segmento possui um peso no cálculo da relevância de acordo com sua localização na estrutura, basta percorrê-la da seguinte forma:

   – Um segmento possui no máximo dois focos. Cada segmento, portanto, gera no máximo duas buscas pelos termos que ele contém, o que pode ser feito em $O(lg\ f_i)$ para cada segmento;

   – O peso do segmento em questão é somado ao valor de relevância dos termos contidos nele, em $\Theta(1)$.

   Neste caso, a complexidade de criação da lista de valores de relevância seria:

$$T_{CRIAR-TVR}(f_i) \in O(f_i \cdot lg\ f_i), \tag{5.9}$$

   para cada documento D_i, que é melhor que o valor encontrado em 5.8.

- O dicionário com os sinônimos encontrados no documento. Associar o dicionário, já criado, a um documento D_i possui complexidade:

$$T_{ASSOCIAR-DS} \in \Theta(1); \tag{5.10}$$

Portanto, de 5.10, 5.7 e 5.9, tem-se que a complexidade de inserção de cada documento no IDI é dada por:

$$T_{INSERIR-D_I-IDI}(f_i) \in \Theta(1) + O(f_i) + O(f_i \cdot lg\ f_i) \subset O(f_i \cdot lg\ f_i), \tag{5.11}$$

e a complexidade de geração de todo o IDI, para um número n de documentos, é:

$$T_{CRIAR-IDI}(n) \in \sum_{i=1}^{n} O(f_i \cdot lg(f_i)), \tag{5.12}$$

sendo f_i o número de frases do documento D_i. Vale ressaltar que o IDI não precisa ser ordenado pois, quando é feita uma *query*, todos os documentos sempre são pesquisados no índice.

Existem duas alternativas simples para tornar o processamento *offline* mais rápido, enumeradas a seguir:

1. Criar o dicionário de sinônimos de cada documento ao mesmo tempo que sua ENDB é gerada (ver capítulo 4). Dessa forma, a estrutura é percorrida apenas uma vez, ao invés de duas;

2. Não criar a ENDB de cada documento. Através da estrutura, são calculados os valores de relevância dos termos e o valor de relevância máximo do documento, que são armazenados no IDI. Entretanto, esses valores podem ser calculados a partir da própria END. A estrutura não chega a ser armazenada pelo sistema. Ela é sugerida neste trabalho apenas para tornar mais claro o processo de recuperação de informações baseado na resolução de anáforas, pois permite identificar mais facilmente as informações da END que são utilizadas na busca por informações.

   O único processamento *offline* que precisa ser feito, portanto, é a criação do índice de documentos interpretados. Percorrendo a END de cada documento apenas uma vez, é possível criar o dicionário e calcular os valores de relevância enquanto o IDI é criado. Dessa forma, para cada documento D_i disponível, devem ser realizados os seguinte passos:

   - Interpretar D_i, ou seja, resolver suas anáforas e criar sua END;
   - Para cada segmento da END, verificar a presença de relações de co-referências (caso existam, inseri-las no dicionário de sinônimos) e adicionar o peso relativo à sua posição na estrutura (deslocamento horizontal e/ou vertical) ao valor de relevância dos termos presentes neste segmento;
   - Associar o dicionário criado ao documento D_i no IDI (cuja complexidade é $\Theta(1)$).

   Das complexidades 5.6, 5.10, 5.7 e 5.9, tem-se que a complexidade total do processamento *offline*, neste caso, é dada por:

$$T_{CRIAR-IDI}(n) \in \sum_{i=1}^{n}(O(f_i \cdot lg\ f_i) + \Theta(1) + O(f_i) + O(f_i \cdot lg\ f_i)) \subset \sum_{i=1}^{n} O(f_i \cdot lg\ f_i) \tag{5.13}$$

Apesar de essas alternativas diminuírem o número de cálculos a serem feitos, nenhuma delas altera a complexidade de criação do IDI. Mesmo que se invista em estruturas de dados mais eficientes para criar o DS, a complexidade de criação do índice de documentos interpretados ainda não diminuiria, devido ao processo de cálculo dos valores de relevância dos termos de D_i, que é $O(f_i \cdot lg\ f_i)$.

Armazenar o índice de documentos interpretados é suficiente para que as buscas sejam feitas. O espaço máximo necessário para armazená-lo é diretamente proporcional ao número de frases de cada documento. Para cada documento interpretado, é necessário armazenar no índice os itens seguintes:

- O valor de relevância máximo do documento. A complexidade de espaço para armazená-lo é dada por:
$$S_{VR_{MAX}} \in \Theta(1); \tag{5.14}$$
- A *Tabela de Valores de Relevância* ($\mathcal{TVR}$) dos termos presentes no documento. A cada termo é associado o seu valor de relevância absoluto no documento. Considerando que existem no máximo $(2 \times f_i)$ elementos distintos em uma END, a complexidade de armazenamento da $\mathcal{TVR}$ é dada por:
$$S_{TVR}(f_i) \in O(f_i); \tag{5.15}$$
- O dicionário com os sinônimos encontrados no documento. Considerando que existem no máximo $(2 \times f_i)$ elementos distintos em uma END, a complexidade de armazenamento do dicionário também é dada por:
$$S_{DS}(f_i) \in O(f_i). \tag{5.16}$$

De 5.14, 5.15 e 5.16, e considerando que o identificador de um documento é armazenado em $\Theta(1)$, então a complexidade de armazenamento dos n documentos disponíveis é dada por:
$$S_{IDI}(f_i) \in \Theta(1) + O(f_i) + O(f_i) + \Theta(1) \subset O(f_i). \tag{5.17}$$

5.3 Processamento *online*

A partir do momento em que o sistema é disponibilizado, o usuário pode requisitar novas *queries* a qualquer momento. Da mesma forma, o administrador do sistema pode

incluir novos documentos no IDI assim que eles forem interpretados. A complexidade de uma busca e a da inclusão de novos documentos no índice são analisadas nas seções seguintes.

5.3.1 Busca de termos simples

Quando uma *query* é feita pelo usuário, o sistema verifica se ela é uma *busca de termos simples* ou uma *busca de termo compostos*. No primeiro caso, percorre o IDI verificando em quais documentos o termo buscado t é um dos assuntos centrais. Ou seja, cria uma lista com cada documento cuja *lista de valores de relevância* contém t. Em seguida, ordena crescentemente essa lista de documentos pelo valor de relevância[1] do termo buscado em cada um deles. A lista resultante é retornada ao usuário como resultado de sua *query*, indicando que o primeiro documento da lista é o mais relevante para o usuário, o segundo documento é o segundo mais relevante *etc.*

Antes de buscar por t em um documento D_i, o sistema precisa verificar se ele referencia uma entidade introduzida no discurso por outro termo. Para isso, busca no DS se t foi substituído na em D_i por algum sinônimo. Como um documento D_i com f_i frases possui no máximo $(2 \times f_i)$ termos e o dicionário está ordenado, pode ser feita uma busca binária por t no DS de D_i com a seguinte complexidade (KNUTH, 1998):

$$T_{BUSCAR-DS}(f_i) \in O(lg\ f_i). \tag{5.18}$$

A busca por t no dicionário retorna (-1) caso ele não referencie uma entidade introduzida no discurso por outro termo. Caso contrário, retorna a posição que seu sinônimo s aparece no DS. Dado o índice de s no DS, a complexidade de encontrar esse termo s é dada por:

$$T_{BUSCAR-SINONIMO} \in \Theta(1). \tag{5.19}$$

Em seguida, busca-se por t (ou por seu sinônimo) na lista de valores de relevância do documento D_i, presente no IDI. Novamente, a busca pelo termo tem uma complexidade dada por:

$$T_{BUSCAR-TVR}(f_i) \in O(lg\ f_i). \tag{5.20}$$

De 5.18, 5.19 e 5.20, tem-se que a busca pelo valor de relevância de t em todos os

[1] Caso o usuário opte por saber quais documentos acrescentam mais informação sobre o(s) termo(s) buscado(s), o $VR_{absoluto}$ é usado. Caso contrário, o sistema de busca analisa o $VR_{relativo}$ de cada termo.

documentos tem a complexidade mostrada a seguir:

$$T_{BUSCAR-TERMO}(n) \in \sum_{i=1}^{n}(O(lg\ f_i) + \Theta(1) + O(lg\ f_i)) \subset \sum_{i=1}^{n} O(lg\ f_i). \tag{5.21}$$

Após calcular os valores de relevância de t, os documentos devem ser ordenados em relação a esses valores. Segundo (CORMEN; LEISERSON CHARLES; RIVEST, 1990; KNUTH, 1998), é possível ordenar uma lista com m elementos (os documentos relevantes, que contêm t) em:

$$T_{CLASSIFICAR-DOCUMENTOS}(m) \in O(n \cdot lg\ m), \tag{5.22}$$

sendo que m, o número de documentos relevantes encontrados, nunca é maior do que n, o número de documentos disponíveis.

O tempo de busca por um termo pode diminuir se for implementado um índice contendo todos os termos encontrados nos documentos. A cada termo seria associada uma lista informando o valor de relevância do termo nos documentos que ele está presente. O índice seria uma estrutura que permitisse o acesso rápido (ver (KNUTH, 1998; BAEZA-YATES; RIBEIRO-NETO, 1999; BENTLEY; SEDGEWICK, 1997; AHO et al., 1983; Van RIJSBERGEN, 1979)) a um determinado termo. Já que a busca é a principal funcionalidade disponibilizada aos usuários (e a que é feita mais vezes), vale a pena investir em estruturas eficientes, para o caso de haver um grande número de documentos disponíveis.

5.3.2 Busca de termos compostos

Quando uma *query* contendo k termos $t_i \in \{t_1, t_2, \cdots, t_k\}$ é feita, deve-se descobrir a relevância de cada um deles em relação aos documentos disponíveis. A partir da complexidade 5.21, relativa à busca pelo valor de relevância de um único termo, tem-se que a complexidade de se encontrar a relevância de todos os k termos buscados nos n documentos disponíveis é dada por:

$$T_{BUSCAR-TERMOS}(k, n) \in \sum_{j=1}^{k}\sum_{i=1}^{n} O(lg\ f_i) = k \cdot \sum_{i=1}^{n} O(lg\ f_i). \tag{5.23}$$

O capítulo 4 apresenta duas formas de se ordenar a lista com os documentos que contêm os termos buscados (tanto para uma conjunção como para uma disjunção de termos) e a complexidade de cada uma delas é mostrada a seguir:

- Na primeira forma de ordenação proposta, ordena-se primeiramente a lista de valo-

res de relevância de cada um dos documentos da coleção (na conjunção, ordena-se crescentemente; na disjunção, em ordem decrescente). A ordenação de k elementos (que são o valor de relevância de cada um dos k termos buscados em relação a um documento D_i) pode ser feita em $O(k \cdot lg\ k)$. Com isso, ordenar a lista relativa a cada um dos m documentos relevantes possui a seguinte complexidade:

$$T_{ORDENAR-LR}(k, m) \in \sum_{i=1}^{m} O(k \cdot lg\ k) \subset O(m \cdot k \cdot lg\ k). \tag{5.24}$$

Em seguida, os documentos são ordenados pelo primeiro elemento de sua lista de relevância. Se houver elementos iguais a serem ordenados, os documentos que contêm tais valores são ordenados pelo segundo elemento de suas listas de relevâncias, e assim sucessivamente. No pior caso, todos os documentos são ordenados em todas as k iterações, o que possui uma complexidade dada por:

$$T_{K-ITERACOES-ORD}(k, m) \in \sum_{j=1}^{k} O(n \cdot lg\ m) \subset O(k \cdot m \cdot lg\ m). \tag{5.25}$$

Novamente, este é um pior caso que dificilmente acontece. Para que todos os documentos sejam ordenados nas k iterações, é necessário que o valor de relevância dos termos buscados sejam iguais em todos os documentos da coleção, o que dificilmente pode acontecer. Em geral, a segunda iteração já contém uma lista pequena (ou vazia) de documentos a serem ordenados e, dessa forma, a complexidade seria dada por:

$$T_{K-ITERACOES-ORD}(m) \in O(m \cdot lg\ m). \tag{5.26}$$

Em geral, o número de termos buscados é menor do que número de documentos disponíveis. Nesse caso, a complexidade total de classificar os documentos relevantes de uma busca de termos compostos (no pior caso, quando todos os documentos são ordenados nas k iterações) é dada por:

$$T_{ORD-COMPOSTA}(k, m) \in O(k \cdot m \cdot lg\ m) + O(m \cdot k \cdot lg\ k) \subset O(k \cdot m \cdot lg\ m). \tag{5.27}$$

- A segunda forma de ordenação proposta consiste em calcular um único valor de relevância para os termos buscados (na conjunção, multiplica-se os valores de relevância encontrados; na disjunção, soma-se tais valores) e ordenar os m documentos encontrados em relação a esses novos valores, tal como na busca simples. A complexidade

dessa ordenação é dada por:

$$T_{ORD-COMPOSTA}(m) \in O(m \cdot lg\ m). \tag{5.28}$$

Essa segunda forma de ordenação produz resultados mais rapidamente, mas a primeira aproveita melhor as interpretações feitas nos documentos.

Por fim, após retornar o resultado da busca feita pelo usuário, o sistema fica aguardando uma nova *query* ou o cadastro de um novo documento.

5.3.3 Inclusão de novos documentos

Caso um novo documento seja disponibilizado, não é necessário refazer todo o índice (ao contrário de alguns métodos apresentados no capítulo 2). Apenas é adicionada ao IDI uma entrada associando o novo documento ao seu dicionário de sinônimos, sua tabela de valores de relevância e seu valor de relevância máximo. Os dados contidos no IDI são suficientes para que uma *query* seja realizada, portanto só ele precisa ser armazenado.

Antes de ser incluído no IDI, o novo documento disponibilizado deve passar pelos seguintes processos:

1. resolução de suas anáforas e criação de sua Estrutura Nominal do Discurso;
2. transformação da END em uma Estrutura Nominal do Discurso para busca; e
3. criação de um dicionário com as palavras do discurso que se co-referenciam (dicionário de sinônimos).

A transformação da END em ENDB e a criação do DS, já analisadas, possuem as complexidades 5.1 e 5.5, respectivamente. A complexidade de inserir o documento no IDI (incluir um elemento no final de uma lista) é $\Theta(1)$. Portanto, o tempo de inserir um novo documento no índice de documentos interpretados é:

$$T_{INCLUIR-NOVO-D_I}(f_i) \in O(f_i) + O(f_i \cdot lg\ f_i) + \Theta(1) \subset O(f_i \cdot lg\ f_i), \tag{5.29}$$

sendo f_i o número de frases de D_i.

5.4 Considerações Finais

Os resultados que *queries* feitas a partir do protótipo são promissores. O capítulo 6 analisa os resultados encontrados pela metodologia proposta neste trabalho.

Vale notar que esse protótipo não foi implementado para ser um *site* de buscas na *internet*. Ele foi pensado para atender às necessidades de corporações que possuem bancos com grande número de documentos digitais, que precisem ser consultados constantemente. O sistema, portanto, seria usado internamente por seus funcionários.

Não há uma restrição que impeça o uso desta metodologia na busca de documentos digitais disponíveis na *Internet*. Entretanto, vale notar que grande parte dos *sites* não possuem uma estrutura que permite uma interpretação de seu texto. Grande parte deles é constituído apenas de códigos no formato *HTML* contando *links* para outros *sites*, contendo poucos textos passíveis de serem interpretados. Nesses casos, a interpretação se torna desnecessária. Para que a ferramenta seja utilizada como um *site* que busque textos na *Internet*, deve haver um esforço adicional pra se descobrir, dentre os endereços virtuais existentes, quais apresentam textos que possam ser interpretados.

"*A minha musa inspiradora é o meu prazo de entrega.*"
Luis Fernando Veríssimo

6 Avaliação da metodologia

"*Viver é desenhar sem borracha.*"
Millôr Fernandes

Neste capítulo é avaliada a qualidade dos resultados obtidos pela metodologia proposta neste trabalho.

6.1 Qualidade dos resultados

Antes de finalizar a implementação de um sistema de recuperação de informação, deve ser feita uma avaliação dos resultados obtidos por ele. As formas mais comuns de se medir a performance de um sistema de recuperação de *dados* são tempo e espaço. Quanto menor o tempo de resposta e menor o espaço de armazenamento utilizado, melhor o sistema é considerado. Em (BAEZA-YATES; RIBEIRO-NETO, 1999), é feita uma análise detalhada de como se deve equilibrar a complexidade de tempo e a complexidade de espaço.

Em um sistema de recuperação de *informação*, cada um documentos localizados possui um *ranking* indicando sua relevância em relação à *query*. É importante que a qualidade da coleção retornada ao usuário também seja analisada. Avaliar a qualidade dos resultados de uma metodologia para recuperar informação em documentos digitais, como a proposta neste trabalho, não é tarefa simples. Textos escritos em linguagens naturais podem ser interpretados de formas diferentes (dependendo de quem os lê) e o uso de palavras-chave introduz uma diferença de semântica entre o que o usuário deseja e o conjunto de documentos retornados, já que esse tipo de texto nem sempre é bem estruturados e pode ser semanticamente ambíguo (BAEZA-YATES; RIBEIRO-NETO, 1999). A avaliação da metodologia proposta neste trabalho é feita destacando-se seu comportamento diante de determinadas *queries*, na tentativa de prever seus resultados.

Segundo (Van RIJSBERGEN, 1979), até hoje não foi criado um método que sempre recupere todos os documentos relevantes em relação a uma *query* e que nunca recupere algum documento irrelevante. Isso se agrava pelo fato de que o que é relevante para um usuário pode não ser para outro. A qualidade dos métodos, portanto, é medida pela quantidade de documentos relevantes e irrelevantes que eles recuperam. Quanto maior o número de documentos relevantes localizados e menor o número de documentos irrelevantes nesse conjunto, melhores são os valores de *recall* e *precision* alcançados pelo método.

As seções 6.1.1, 6.1.2, 6.1.3 e 6.1.4 mostram as principais características que diferenciam esta metodologia dos modelos tradicionais de RI, e a seção 6.1.5 mostra como essas características influenciam nos valores de *recall* e *precision* obtidos por esta proposta.

6.1.1 Múltiplas referências a uma mesma entidade

Utilizar o processamento de anáforas na recuperação de informação permite identificar todas as referências a uma determinada entidade no documento. Retomando o Texto 1.1,

mostrado a seguir:

Texto 1.1 "Sabiá lá na gaiola fez um buraquinho
Voou, voou, voou, voou
E a menina que gostava tanto do bichinho
Chorou, chorou, chorou, chorou"
(Sabiá Lá na Gaiola, Mário Vieira/Hervê Cordovil)

A resolução das anáforas do Texto 1.1 permite identificar que a entidade referenciada pelo termo *sabiá* na primeira frase do discurso continua sendo referenciada nas duas frases seguintes. Com isso, a resolução de anáforas acrescenta uma informação sobre tal entidade que os métodos de busca tradicionais não conseguem captar. Como a entidade continua em evidência ao longo do discurso, mesmo sem o termo *sabiá* ter sido repetido, ela apresenta mais relevância em relação ao texto do que se as referências a ela não tivessem ocorrido.

Cada método de busca calcula um certo valor de relevância de uma entidade em relação a um texto. Esse cálculo, porém, é feito de forma distinta em cada método. Não faz sentido apenas comparar o valor de relevância de um método com o de outro. O que se deve comparar é a classificação que cada um deles dá aos textos relevantes, em relação à *query* feita pelo usuário. Para comparar a classificação da metodologia proposta neste trabalho com a dos demais métodos, é preciso submeter mais de um texto a uma mesma busca. Então, retomando também o Texto 1.2, mostrado a seguir:

Texto 1.2 "Minha terra tem palmeiras, onde canta o Sabiá;
As aves, que aqui gorjeiam, não gorjeiam como lá.
Nosso céu tem mais estrelas, nossas várzeas têm mais flores,
Nossos bosques têm mais vida, nossa vida mais amores."
(Canção do Exílio, Gonçalves Dias)

Tanto o Texto 1.1 quanto o *1.2* possuem quatro frases e o termo *sabiá* é citado apenas na primeira frase de cada um deles. Os métodos tradicionais não têm informações suficientes para decidir qual dos textos é mais relevante para a busca pelo termo *sabiá*, por não identificarem se a entidade a que o termo se refere é referenciada novamente no decorrer do discurso.

Cada método usa um critério de desempate para classificar cada texto como mais ou menos relevante. A metodologia proposta neste trabalho certamente classifica o Texto 1.1

como mais relevante que o *1.2*, devido às múltiplas referências à entidade introduzida no primeiro texto pelo termo *sabiá*, mesmo sem a repetição desse termo. A maioria dos métodos tradicionais também utilizam a quantidade de vezes que o termo buscado aparece no discurso. Entretanto, eles não identificam referências a esses termo.

Uma forma de refereciar uma entidade é através de uma elipse[1], como no trecho "*Carlota* estava sentada sob a tolda, com a cabeça encostada ao ombro de sua mãe e com os olhos engolfados no horizonte, que ocultava o lugar onde tínhamos passado a primeira e última hora de felicidade. Quando (**ela**) me viu, (**ela**) fez um movimento como se quisesse lançar-se para mim; mas (**ela**) conteve-se, (**ela**) sorriu-se para sua mãe, e, cruzando as mãos no peito, (**ela**) ergueu os olhos ao céu, como para agradecer a Deus, ou para dirigir-lhe uma prece.", retirado do livro *Cinco minutos* de José de Alencar. A entidade introduzida no texto pelo termo *Carlota* é referenciada cinco vezes através de elipses. No trecho "*Minha mãe* era boa criatura. Quando lhe morreu o marido, Pedro de Albuquerque Santiago, (**ela**) contava trinta e um anos de idade, e (**ela**) podia voltar para Itaguaí. (**ela**) Não quis; (**ela**) preferiu ficar perto da igreja em que meu pai fora sepultado. (**ela**) Vendeu a fazendola e os escravos, (**ela**) comprou alguns que pôs ao ganho ou alugou, uma dúzia de prédios, certo número de apólices, e (**ela**) deixou-se estar na casa de Mata-cavalos, onde (**ela**) vivera os dois últimos anos de casada. (**ela**) Era filha de uma senhora mineira, descendente de outra paulista, a família Fernandes.", retirado do livro *Dom Casmurro*, de Machado de Assis, no qual a entidade introduzida no discurso pelo termo *Minha mãe* é referenciada através de nove elipses no restante do trecho.

Outra forma de anáfora é através de um sintagma nominal definido, como no trecho[2] "*João Romão* foi, dos treze aos vinte e cinco anos, empregado de um vendeiro que enriqueceu entre as quatro paredes de uma suja e obscura taverna nos refolhos do bairro do Botafogo; (...) Proprietário e estabelecido por sua conta, **o rapaz** atirou-se à labutação ainda com mais ardor, possuindo-se de tal delírio de enriquecer, que afrontava resignado as mais duras privações.", retirado do livro *O Cortiço*, de Aluízio Azevedo, no qual o termo **o rapaz** co-referencia o termo *João Romão*.

Por fim, um termo também pode ser repetido em um texto através de uma anáfora pronominal, como no trecho "*Meu pai* respondia a todos que eu seria o que Deus quisesse; e alçava-me ao ar, como se intentasse mostrar-me à cidade e ao mundo; perguntava a todos se eu me parecia com **ele**, se era inteligente, bonito...", retirado do livro *Memórias Póstuma de Brás Cubas* de Machado de Assis, no qual o termo *Meu pai* referencia a

[1] As elipses são mostradas no texto através dos pronomes entre parênteses.

[2] As reticências indicam uma passagem do texto que não é reproduzida neste trecho.

mesma entidade que o pronome **ele**.

Todas as três formas de anáforas mostradas acima são identificadas por esta metodologia. O cálculo da relevância dos termos *Carlota*, *Minha mãe*, *João Romão* e *Meu pai* consideram todas as vezes que eles são referenciados no documento, mesmo sem que o próprio termo reapareça no discurso.

6.1.2 Disposição dos termos no discurso

Em geral, os métodos tradicionais levam em consideração a posição em que as palavras aparecem no discurso, por terem que decidir quais delas são mais ou menos relevantes. Eles consideram que uma palavra deve aparecer no início do discurso para que tenha relevância em relação a ele. No Texto 1.1 isso é verdade: o termo *sabiá* referencia uma entidade que é relevante, e é citado já na primeira linha do mesmo. Mas nem sempre isso acontece.

Um autor pode começar um livro pela narração do ambiente onde a história se passará (no Texto 1.2, o termo *sabiá* se encontra na primeira frase do discurso e não é o foco do mesmo). Os personagens principais do livro podem demorar a aparecer, mas passarão a estar em evidência a partir de então. Um dos assuntos principais do livro *Fortaleza Digital*, de Dan Brown, é um supercomputador que supostamente é capaz de quebrar qualquer tipo de criptografia. O computador só passa a ser citado no livro do capítulo 4 em diante, mas continua em evidência a partir de então. Victoria Jones é a personagem principal do livro *Aventura em Bagdá*, de Agatha Christie, mas não é citada no primeiro capítulo do livro.

Considerar uma palavra como mais relevante que outra por estar presente no início do texto é um critério de desempate entre o valor de relevância que deve ser dado a elas. A metodologia de RI baseada na resolução de anáforas não precisa desse critério, pois identifica o quão relevante uma entidade é para o documento pela quantidade de vezes que ela é referenciada (através de pronomes, elipses, sinônimos *etc*).

6.1.3 Termos sinônimos

O Texto 1.1 apresenta ainda outro aspecto que torna a metodologia proposta neste trabalho vantajosa: a relação de co-referência entre os termos *sabiá* e *bichinho*. O fato de ambos referenciarem a mesma entidade no texto faz com que esta metodologia considere os dois termos como sinônimos. Isso implica que a busca por *sabiá* tem uma relevância

(para o Texto 1.1) igual à da busca por *bichinho*.

No decorrer de um discurso, uma entidade pode ser referenciada por diversos termos diferentes. O usuário pode pensar em qualquer um desses termos quando faz uma busca por ela. Um termo não é mais relevante que o outro. Ambos indicam uma única entidade, e a busca por um deles corresponde a uma busca por esta entidade. Essa característica também não é abordada pelos métodos tradicionais.

Um exemplo de termos sinônimos é encontrado no trecho "*Cecília* aparecera no alto da esplanada e lhe acenara; sua mãozinha alva e delicada agitando-se no ar parecia dizer-lhe que esperasse; *Peri* julgou mesmo ver no rostinho gentil de sua **senhora** apesar da distancia, brilhar um raio de felicidade. Quando com os olhos fitos naquela graciosa visão ele esforçava-se por adivinhar a causa de tão súbita alegria, *a índia* soltou um segundo grito selvagem, um grito terrível. Tinha pela direção do olhar do prisioneiro visto Cecília sobre a esplanada; tinha percebido o gesto da **menina**, e compreendera vagamente a razão por que Peri recusara a liberdade e o seu amor. Precipitou-se sobre o arco que estava atirado ao chão; mas apesar da rapidez desse movimento, quando ela estendia a mão, já Peri tinha posto o pé sobre a arma. **A selvagem**, com os olhos ardentes, os lábios entreabertos, trêmula de ciúme e de vingança, levantou sobre o peito do **índio** a faca de pedra com que lhe cortara os laços há pouco; mas a arma caiu-lhe da mão, e vacilando apoiou-se no seio que ameaçara.", retirado do livro *O Guarani* de José de Alencar, no qual os termos **menina** e **senhora** referenciam a entidade introduzida no discurso pelo termo *Cecília*. O mesmo acontece com os termos *a índia* e **A selvagem**, e com *Peri* e **índio**. O livro também contém o trecho "*D. Antônio*, ainda pálido e trêmulo do perigo que correra Cecília, volvia os olhos daquela terra que se lhe afigurava uma campa, para o selvagem que surgira, como um gênio benfazejo das florestas do Brasil. O **fidalgo** não sabia o que mais admirar, se a força e heroísmo com que ele salvara sua filha, se o milagre de agilidade com que se livrara a si próprio da morte.", no qual os termos **fidalgo** e *D. Antônio* se co-referenciam.

A resolução de anáforas permite, portanto, calcular com mais precisão a relevância dos termos do discurso, identificando os termos que possuem o mesmo conteúdo semântico no documento.

6.1.4 Sujeitos, verbos e predicados

Outra característica desta metodologia que deve ser destacada é que ela considera apenas o sujeito e o predicado de cada frase quando uma busca é feita, pois sua inten-

ção é determinar os assuntos principais do discurso. Uma busca pelo termo *chorou* não retornaria o Texto 1.1, apesar de a palavra *chorou* ser citada nele.

Essa característica tem um lado negativo e outro positivo. O negativo é que o usuário pode querer identificar um texto por determinadas ações (representadas por verbos, conjugados ou não) que acontecem nele. As ações que ligam o sujeito ao predicado das frases são guardadas na END, mas não são herdadas pela ENDB, por não identificarem o assunto do texto. Neste aspecto, a ausência de verbos na estrutura para busca é uma característica positiva da metodologia. O assunto (foco) de um texto é a entidade referenciada mais vezes nele. Analisando os verbos, o foco poderia ser considerado menos relevante do que é, em virtude de determinadas ações predominantes no discurso que poderiam ofuscá-lo. A presença dos verbos na estrutura para busca, portanto, atrapalharia o processo de identificação dos assuntos principais dos textos.

6.1.5 *Recall* e *precision*

Atualmente, as duas medidas mais utilizadas para avaliar métodos de RI em documentos digitais são *precision* e *recall* (ver capítulo 2). *Recall* é a proporção de documentos relevantes de uma coleção que foram recuperados (i.e. o número de acertos em relação ao número de casos existentes) e *precision* é a proporção dos documentos recuperados em uma busca que são relevantes (ou seja, o número de acertos em relação ao número de casos tratados), calculados pelas equações 6.1 e 6.2:

$$recall = \frac{n^o \textit{ de documentos relevantes recuperados}}{n^o \textit{ total de documentos relevantes do conjunto de dados}} \tag{6.1}$$

$$precision = \frac{n^o \textit{ de documentos relevantes recuperados}}{n^o \textit{ de documentos relevantes e irrelevantes recuperados}} \tag{6.2}$$

O valor do número total de documentos relevantes do conjunto de dados, usado no cálculo de *recall*, não varia de acordo com o método utilizado. Os valores que podem ser melhorados em cada método são o número de documentos relevantes recuperados (usado nos cálculos de *recall* e *precision*) e o número de documentos irrelevantes recuperados (que, somado ao anterior, é usado no cálculo do valor de *precision*). As características mencionadas nas seções 6.1.1, 6.1.2, 6.1.3 e 6.1.4 influenciam o cálculo dos valores de *recall* e *precision* da metodologia proposta neste trabalho da seguinte forma:

- A identificação das múltiplas referências a uma mesma entidade torna possível que um número maior de documentos relevantes seja encontrado. Com mais documentos

relevantes, tanto o valores de *recall* como o de *precision* aumentam;

- Desconsiderando o fato de que palavras no início do documento são *necessariamente* relevantes, a metodologia descarta alguns documentos irrelevantes que podem ser considerados relevantes em outros métodos. Com a diminuição do número de documentos irrelevantes recuperados, o valor de *precision* também aumenta;

- A procura por sinônimos de um determinado termo buscado é outro fator que permite aumentar o número de documentos relevantes recuperados, aumentando os valores de *recall* e *precision*;

- Por último, desconsiderando os verbos dos documentos, aumentam as chances de se identificar corretamente os seus focos. Dessa forma, cresce o número de documentos relevantes recuperados, já que as ações existentes no discurso não atrapalham a identificação de seus assuntos. Com isso, aumentam os valores de *recall* e *precision*.

Portanto, três características desta metodologia contribuem para o aumento de seus valores de *recall*, por ajudarem a recuperar mais documentos relevantes: (1) a identificação das múltiplas referências a uma mesma entidade; (2) a procura por sinônimos de um determinado termo buscado; e (3) não analisar os verbos dos documentos durante o cálculo de relevância dos termos. Já o valor de *precision* aumenta devido a todas essas três características, e também porque a metodologia não considera que palavras no início do documento são *necessariamente* relevantes. Vale lembrar que quanto maiores os valores de *recall* e *precision* de um método, melhor é a sua qualidade.

"*O único lugar onde o sucesso vem antes do trabalho é no dicionário.*"
Albert Einstein

7 Conclusões

"*No fim tudo dá certo, e se não deu certo*
é porque ainda não chegou ao fim."
Fernando Sabino

Neste capítulo são apresentadas as conclusões deste trabalho e algumas propostas de trabalhos futuros.

7.1 Conclusões e Trabalhos Futuros

Linguagens naturais são tão interessantes quanto subjetivas. A interpretação de um texto pode produzir resultados diferentes, dependendo de quem o lê. O resultado deste trabalho é adicionar uma forma clara e simples de interpretação de documentos digitais aos mecanismos de recuperação de informação. Esta proposta aumenta a qualidade dos resultados das *queries* em relação às propostas tradicionais, unindo o estudo lingüístico à recuperação de informação. O mecanismo de busca aqui proposto classifica os documentos pela quantidade de informação que eles apresentam a respeito dos termos buscados, e não apenas pela disposição dessas palavras nos documentos. A tentativa de interpretar o texto através da resolução de suas anáforas permite dar mais enfoque ao que o autor escreveu.

A Estrutura Nominal do Discurso (FREITAS, 2005) foi criada com a finalidade específica de resolver anáforas. Entretanto, ela contém informações que podem ser úteis para um sistema de RI: a estrutura permite saber a quantidade de vezes que cada entidade é referenciada nos documentos. Essa informação adiciona ao sistema a capacidade de decidir se certa entidade é mais relevante que outra em um mesmo documento. Ainda não se tem conhecimento de um sistema capaz de recuperar sempre todos os documentos relevantes a uma *query*, e que nunca recupere documentos que sejam irrelevantes (Van RIJSBERGEN, 1979). O que diferencia cada método é a quantidade de documentos relevantes e irrelevantes que cada um retorna ao usuário. O ideal é maximizar o número de documentos relevantes localizados e minimizar o número de documentos irrelevantes nesse conjunto. A utilização da resolução de anáforas introduz características à recuperação de informação que a aumentam a quantidade de documentos relevantes recuperados. São elas: (1) a identificação das múltiplas referências a uma mesma entidade, que torna possível saber o número de vezes que a entidade é realmente referenciada; (2) a procura por sinônimos do termo buscado, que são definidos no próprio documento; e (3) a busca apenas pelos focos de cada frase, que impede que demais termos não relevantes possam atrapalhar a identificação do assunto principal do discurso. Essa última característica também diminui a quantidade de documentos irrelevantes recuperados, assim como o fato de desconsiderar que palavras no início do discurso são necessariamente relevantes.

Existem outras formas de classificar a relevância de documentos digitais. O *site* de buscas Google™, por exemplo, classifica as páginas da *internet* como mais ou menos importantes de acordo com o número de *links* que apontam para ela. *Links* de páginas mais importantes valem mais do que *links* de páginas menos importantes. Não é feito qualquer tipo de interpretação dos textos contidos nas páginas. Apesar de democrático,

esse tipo de classificação pode ser manipulado, atribuindo *links* não contextualizados como objetivo de certa página, modificando a ordenação de resultados na pesquisa pelo Google™ e induzindo a resultados pouco relevantes ou tendenciosos.

O tempo de processamento dos algoritmos da metodologia de busca não é alto:

- A busca por um termo é feita em $\sum_{i=1}^{n} O(lg\ f_i)$, para n de documentos com f_i frases cada, seguida da ordenação dos documentos relevantes em $O(n \cdot lg\ n)$. Essa complexidade pode ser menor se for criado um índice contendo todos os termos existentes, associando-os diretamente aos documentos nos quais eles estão presentes. O processamento *offline* aumentaria, mas caso haja um grande número de documentos disponíveis, um tempo de busca alto faria com que os usuários não utilizassem o sistema;

- O tempo processamento de uma busca composta cresce linearmente com o aumento do número de termos da *query*;

- A manutenção do índice com os documentos interpretados também tem baixo custo computacional, já que os elementos do índice não precisam ser ordenados;

- Por fim, o tempo de geração da Estrutura Nominal do Discurso não foi discutido neste trabalho pois o algoritmo que cria a estrutura não faz parte da metodologia proposta. Qualquer forma de criá-la pode ser utilizada. O algoritmo utilizado no protótipo (proposto em (FREITAS, 2005)) é linear, já que cada frase do discurso é interpretada apenas uma vez, e a quantidade de segmentos visitados durante a interpretação (em busca do antecedente da expressão anafórica) é limitado por uma constante.

A forma de utilizar a resolução de anáforas na recuperação de informação introduz ainda uma última questão: Por que utilizar a Estrutura Nominal do Discurso, e não o próprio texto após o processo de interpretação? Gerar um novo documento substituindo as expressões anafóricas por seus antecedentes já é um fator que melhora a qualidade dos resultados das buscas. Entretanto, a estrutura permite acompanhar como os focos se mantêm em evidência ao longo do discurso. Seus segmentos são resultados da interpretação de duas frases consecutivas no discurso. Além disso, a estrutura identifica os termos que se co-referenciam, permitindo a busca por termos sinônimos. Considerando o texto "O aluno comprou um *carro*. O **motor** veio quebrado.", se o termo **motor** for substituído pelo termo *carro*, uma informação útil à busca pode ser perdida (que pode ser recuperada

pela estrutura). A não substituição do termo também prejudica a qualidade do resultado, já que uma referência à entidade introduzida no discurso pelo termo *carro* não está sendo considerada.

Por fim, vale ressaltar que a metodologia proposta neste trabalho permite modificações em cada implementação. Foram sugeridas mais de uma fórmula para cálculo da relevância dos termos (equações 4.3, 4.4 e 4.5), mais de um tipo de valor de relevância (relativo e absoluto) e mais de uma forma de ordenar a lista de documentos relevantes em uma busca compostas (tanto em uma conjunção de termos, como em uma disjunção). Além disso, algumas estruturas de dados foram sugeridas neste trabalho, mas cada implementação pode utilizar estruturas diferentes (visando beneficiar as funcionalidades mais importantes em cada sistema).

Um trabalho futuro imediato a ser feito é: ao invés de retornar ao usuário, como resultados de uma *query*, apenas os documentos ordenados (do mais relevante ao menos relevante), mostrar trechos de cada um desses documentos (a fim de tornar mais fácil a tarefa do usuário de escolher um documento dentre os retornados). Isso é encontrado na maioria dos *sites* de busca da *Internet*, como Google™ e Yahoo!®.

Outra proposta de trabalho futuro é usar esta proposta para agrupar documentos semelhantes (prática conhecida como *clustering*).

Um trabalho futuro importante, que pode trazer enorme ganho à qualidade dos resultados, é refazer o processo de geração da END, visando apenas a recuperação de informação. A estrutura é gerada originalmente com o propósito único de encontrar os focos de um texto (ver capítulo 3). Durante sua criação, algumas informações relevantes ao processo de recuperação de informação podem estar sendo perdidas. Por exemplo: Em alguma situações, a junção de dois segmentos simples gera um segmento composto de foco nulo. Isso é feito durante a geração da END pois o foco nulo demonstra apenas que não houve mudança no assunto central do texto, não sendo necessário repeti-lo na estrutura. Entretanto, no caso da recuperação de informação, quanto mais vezes o foco aparecer na estrutura, mais relevante em relação ao documento ele será.

"*Se eu soubesse antes o que sei agora,*
erraria tudo exatamente igual."
Humberto Gessinger

Referências

ABBOTT, B. A pragmatic account of the definiteness effect in existencial sentences. *Journal of Pragmatics*, v. 19, p. 39–55, 1993.

AHO, A. V.; HOPCROFT, J. E. *The Design and Analysis of Computer Algorithms*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1974. ISBN 0201000296.

AHO, A. V. et al. *Data Structures and Algorithms*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1983. ISBN 0201000237.

BAEZA-YATES, R.; RIBEIRO-NETO, B. *Modern information retrieval*. Wokingham, UK: Addison-Wesley, 1999.

BEAVER, D. I. The optimization of discourse anaphora. *LingPhilo*, v. 27, p. 3–56, 2004.

BENTLEY, J. L. *Writing efficient programs*. Upper Saddle River, NJ, USA: Prentice-Hall, Inc., 1982. ISBN 0-13-970251-2; 0-13-970244-X (pbk.).

BENTLEY, J. L.; SEDGEWICK, R. Fast algorithms for sorting and searching strings. In: *SODA '97: Proceedings of the eighth annual ACM-SIAM symposium on Discrete algorithms*. Philadelphia, PA, USA: Society for Industrial and Applied Mathematics, 1997. p. 360–369. ISBN 0-89871-390-0.

BLUTNER, R. Some aspects of optimality in natural language interpretation. *Journal of Semantics*, v. 17, p. 189–217, 2000.

CARDOSO, O. N. P. Recuperação de informação. *INFOCOMP - Journal of Computer Science*, v. 2, n. 1, p. 33–38, nov 2000.

CARTER, D. *Interpreting Anaphors in Natural Language Texts*. [S.l.]: Ellis Horwood Books, 1987.

CLEVERDON, C. The cranfield tests on index language devices. In: JONES, K. S.; WILLETT, P. (Ed.). *Readings in Information Retrieval*. San Francisco, US: Morgan Kauffman, 1997. p. 47–59.

CORMEN, T. H.; LEISERSON CHARLES, E.; RIVEST, R. L. *Introduction to Algorithms*. [S.l.]: MIT Press, 1990.

FERNEDA, E. *Recuperação de Informação: Análise sobre a contribuição da Ciência da Computação para a Ciência da Informação*. Tese (Doutorado) — Escola de Comunicação e Artes da Universidade de São Paulo, 2003.

FILHO, A. M. C.; FREITAS, S. A. A. d. Interpretação do futuro do pretérito em narrativas. In: *Anais do 1º workshop em Tecnologia da Informação e da Linguagem Humana, TIL'2003*. São Carlos - SP, Brasil: [s.n.], 2003. Disponível em: <nilc.icmc.sc.usp.br/til2003>.

FREITAS, S. *Interpretação Automatizada de Textos: Processamento de Anáforas*. Tese (Doutorado) — Universidade Federal do Espírito Santo, 2005.

FREITAS, S. A. A. de. A utilização da drt em um sistema de representação do discurso. In: *IX Simpósio Brasileiro de Inteligência Artificial*. Rio de Janeiro - RJ - Brasil: [s.n.], 1992.

FREITAS, S. A. A. de. *Resolução de Anáforas e Estrutura do Discurso*. [S.l.], 1995.

FREITAS, S. A. A. de; LOPES, J. G. P. Discourse segmentation: Extending the centering theory. In: *XI Simpósio Brasileiro de Inteligência Artificial*. UFCE - Fortaleza - CE: [s.n.], 1994.

FREITAS, S. A. A. de; LOPES, J. G. P. Improving centering to support discourse segmentation. In: SANDT, R. V. der; BOSH, P. (Ed.). *Focus and Natural Language Processing*. Kassel, Germany, June 12-15, 1994: [s.n.], 1994. IBM Working Papers of the Institute for Logic and Linguistics, Heidelberg.

FREITAS, S. A. A. de; LOPES, J. G. P. Solving the reference to mixable entities. In: *Proceedings of the Indirect Anaphora Workshop*. University of Lancaster, Lancaster, UK: [s.n.], 1996.

FREITAS, S. A. A. de; LOPES, J. G. P.; MENEZES, C. da S. Abducing definte descriptions relations. In: *Anais do XXIV Congresso da Sociedade Brasileira de Computação*. Salvador - BA, Brasil: [s.n.], 2004.

GROSSMAN, D. A.; FRIEDER, O. *Information Retrieval: Algorithms and Heuristics*. Dordrecht, The Netherlands: Springer, 1998. ISBN 1-4020-3004-5.

GROSZ, B.; SIDNER, C. L. Attention, intentions and the structure of the discourse. *cl*, v. 12, n. 3, p. 175–204, 1986.

GROSZ, B. J. *The Representation and Use of Focus in a System for Understanding Dialogs*. SRI International, Menlo Park, California, 1977.

GROSZ, B. J.; JOSHI, A. K.; WEINSTEIN, S. Centering: A framework for modelling the local coherence of discourse. *cl*, v. 21, n. 2, p. 203–225, 1995.

GUNDEL, J. K.; HEGARTY, M.; BORTHEN, K. Cognitive status, information structure, and pronominal reference to clausally introduced entities. *jolli*, v. 12, p. 281–299, 2003.

GWIZDKA, J.; CHIGNELL, M. *Towards Information Retrieval Measures for Evaluation of Web Search Engines*. 1999. Disponível em: <citeseer.ist.psu.edu/593227.html>.

HAJIČOVá, E.; SKOUMALOVá, H.; SGALL, P. An automatic procedure for topic-focus identification. *cl*, v. 21, n. 1, p. 81–94, 1995.

HOBBS, J. R. *On the Coherence and Structure of Discourse*. CLSI - Stanford University, 1985. Relatório Técnico nº CSLI-85-37.

HOBBS, J. R. Intention, information, and structure in discourse: A first draft. In: *Proceedings of the NATO Advanced Research Workshop: Burning Issues in Discourse*. Maratea, Italy: [s.n.], 1993. p. 41–66.

JIZBA, R. *Measuring Search Effectiveness*. 2000. Disponível em: <http://www.hsl.creighton.edu/hsl/Searching/Recall-Precision.html>.

KNUTH, D. E. *The Art of Computer Programming, 2nd Ed. (Addison-Wesley Series in Computer Science and Information*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1978. ISBN 0201038099.

KNUTH, D. E. *The art of computer programming, volume 3: (2nd ed.) sorting and searching*. Redwood City, CA, USA: Addison Wesley Longman Publishing Co., Inc., 1998. ISBN 0-201-89685-0.

KRUIJFF-KORBAYOVá, I.; STEEDMAN, M. Discouse and information structure. *jolli*, n. 12, p. 249–259, 2003.

LOPES, I. L. Estratégia de busca na recuperação da informação: revisão da literatura. *Ci. Inf. [online]*, v. 31, n. 2, p. 60–71, 2002. ISSN 0100-1965.

LOPES, J. G. P.; FREITAS, S. A. A. de. Improving centering to support discourse segmentation. In: BOSCH, P.; SANDT, R. van der (Ed.). *Focus in Natural Language Processing*. Heidelberg, Germany: IBM, 1994, (Working Papers of the Institute for Logic and Linguistics, v. 3). p. 533–542.

MANN, W. C.; THOMPSON, S. A. *Rhetorical Structure Theory: A Theory of Text Organization*. ISI Reprint Series, 1987. Relatório Técnico nº ISI/RS-87-190.

POLANYI, L. A formal model of the structure of discourse. *Journal of Pragmatics*, n. 12, p. 601–638, 1988.

POLANYI, L.; BERG, M. van den. Discourse structure and discourse interpretation. In: DEKKER, P.; STOKHOF, M. (Ed.). *Tenth Amsterdam Colloquium*. Department of Philosophy, University of Amsterdam: [s.n.], 1996. p. 113–131. Disponível em: <citeseer.nj.nec.com/polanyi96discourse.html>.

POLANYI, L.; BERG, M. van den; AHN, D. Discourse structure and sentential information structure. *jolli*, v. 12, p. 337–350, 2003.

SALTON, G. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. [S.l.]: Addison-Wesley, 1989. 530 p. ISBN 0-201-12227-8.

SALTON, G.; MCGILL, M. J. *Introduction to Modern Information Retrieval*. New York, NY, USA: McGraw-Hill, Inc., 1986. ISBN 0070544840.

SEDGEWICK, R.; FLAJOLET, P. *An introduction to the analysis of algorithms*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc., 1996. ISBN 0-201-40009-X.

SIBUN, P. *Locally organized text generation*. Tese (Doutorado), Amherst, MA, USA, 1992.

SIDNER, C. L. *Towards a Computational Theory of Definite Anaphora Comprehension in English Discourse*. Cambridge, MA, USA, 1979.

SIDNER, C. L. Focusing for interpretation of pronouns. *Comput. Linguist.*, MIT Press, Cambridge, MA, USA, v. 7, n. 4, p. 217–231, 1981. ISSN 0891-2017.

SU, L. T. Value of search results as a whole as the best single measure of information retrieval performance. *Inf. Process. Manage.*, v. 34, n. 5, p. 557–579, 1998.

TEIXEIRA, C. M. S.; SCHIEL, U. A internet e seu impacto nos processos de recuperação da informação. *Ci. Inf. [online]*, v. 26, n. 1, 1997. ISSN 0100-1965.

Van RIJSBERGEN, C. J. *Information Retrieval*. [S.l.]: Butterworth-Heinemann, 1979. 208 p. ISBN 0408709294.

YANG, Y.; LIU, X. A re-examination of text categorization methods. In: HEARST, M. A.; GEY, F.; TONG, R. (Ed.). *Proceedings of SIGIR-99, 22nd ACM International Conference on Research and Development in Information Retrieval*. Berkeley, US: ACM Press, New York, US, 1999. p. 42–49.

Livros Grátis

(http://www.livrosgratis.com.br)

Milhares de Livros para Download:

Baixar livros de Literatura
Baixar livros de Literatura de Cordel
Baixar livros de Literatura Infantil
Baixar livros de Matemática
Baixar livros de Medicina
Baixar livros de Medicina Veterinária
Baixar livros de Meio Ambiente
Baixar livros de Meteorologia
Baixar Monografias e TCC
Baixar livros Multidisciplinar
Baixar livros de Música
Baixar livros de Psicologia
Baixar livros de Química
Baixar livros de Saúde Coletiva
Baixar livros de Serviço Social
Baixar livros de Sociologia
Baixar livros de Teologia
Baixar livros de Trabalho
Baixar livros de Turismo