Iúri Chaer

Um estudo sobre a Teoria da Predição aplicada à análise semântica de Linguagens Naturais

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

São Paulo
2010

Iúri Chaer

Um estudo sobre a Teoria da Predição aplicada à análise semântica de Linguagens Naturais

Dissertação apresentada à Escola Politécnica da Universidade de São Paulo para obtenção do Título de Mestre em Engenharia Elétrica.

Área de concentração:
Sistemas Digitais

Orientador:
Prof. Dr. Ricardo Luis de Azevedo da Rocha

São Paulo
2010

Este exemplar foi revisado e alterado em relação à versão original, sob responsabilidade única do autor e com a anuência de seu orientador.

São Paulo, 3 de março de 2010

Assinatura do autor _____

Assinatura do orientador _____

FICHA CATALOGRÁFICA

Chaer, Iúri

Um estudo sobre a Teoria da Predição aplicada à análise semântica de Linguagens Naturais / I. Chaer. -- ed.rev. -- São Paulo, 2010.

72 p.

Dissertação (Mestrado) -- Escola Politécnica da Universidade de São Paulo. Departamento de Engenharia de Computação e Sistemas Digitais.

1. Aprendizado computacional 2. Inteligência artificial 3. Linguagem natural 4. Semântica formal I. Universidade de São Paulo. Escola Politécnica. Departamento de Engenharia de Computação e Sistemas Digitais. II. t.

À minha esposa, Aline

Agradecimentos

À minha esposa, Aline. O seu trabalho como revisora foi essencial e a sua insistência – delicada, mas constante – para que eu não procrastinasse é um dos principais motivos pelos quais este projeto foi levado a termo.

Ao meu amigo e orientador, Professor Doutor Ricardo Luis de Azevedo da Rocha, sempre presente e pronto para me apontar a direção certa.

À minha mãe, que sempre me apoiou e que ajudou diretamente com revisões e sugestões desde as fases iniciais deste trabalho. A sua perseverança é um modelo que nunca deixou de me motivar.

Ao meu pai que, sem perceber, me ensinou a amar o método científico e escolher o caminho que me trouxe até aqui. Sua biografia deveria figurar nas enciclopédias ilustrando o verbete "pragmatismo".

Aos meus amigos, que me apoiaram quase tanto quanto me interromperam. Às vezes, é necessário um intervalo.

Finalmente, ao Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq) pelos recursos fornecidos e à agência de notícias Reuters que, por meio do Instituto Nacional de Padrões e Tecnologia (NIST) dos EUA, forneceu o material para alguns dos testes mais importantes deste trabalho.

Resumo

Neste trabalho, estuda-se o aprendizado computacional como um problema de indução. A partir de uma proposta de arquitetura de um sistema de análise semântica de Linguagens Naturais, foram desenvolvidos e testados individualmente os dois módulos necessários para a sua construção: um pré-processador capaz de mapear o conteúdo de textos para uma representação onde a semântica de cada símbolo fique explícita e um módulo indutor capaz de gerar teorias para explicar sequências de eventos.

O componente responsável pela indução de teorias implementa uma versão restrita do Preditor de Solomonoff, capaz de tecer hipóteses pertencentes ao conjunto das Linguagens Regulares. O dispositivo apresenta complexidade computacional elevada e tempo de processamento, mesmo para entradas simples, bastante alto. Apesar disso, são apresentados resultados novos interessantes que mostram seu desempenho funcional.

O módulo pré-processador do sistema proposto consiste em uma implementação da Análise da Semântica Latente, um método que utiliza correlações estatísticas para obter uma representação capaz de aproximar relações semânticas similares às feitas por seres humanos. Ele foi utilizado para indexar os mais de 470 mil textos contidos no primeiro disco do *corpus* RCV1 da Reuters, produzindo, a partir de dezenas de variações de parâmetros, $71,5GB$ de dados que foram utilizados para diversas análises estatísticas. Foi construído também um sistema de recuperação de informações para análises qualitativas do método. Os resultados dos testes levam a crer que o uso desse módulo de pré-processamento leva a ganhos consideráveis no sistema proposto.

A integração dos dois componentes em um analisador semântico de Linguagens Naturais se mostra, neste momento, inviável devido ao tempo de processamento exigido pelo módulo indutor e permanece como uma tarefa para um trabalho futuro. No entanto, concluiu-se que a Teoria da Predição de Solomonoff é adequada para tratar o problema da análise semântica de Linguagens Naturais, contanto que sejam concebidas formas de mitigar o problema do seu tempo de computação.

Palavras-chave: Aprendizado computacional; Inteligência artificial; Linguagem natural; Semântica formal

Abstract

In this work, computer learning is studied as a problem of induction. Starting with the proposal of an architecture for a system of semantic analisys of Natural Languages, the two modules necessary for its construction were built and tested independently: a pre-processor, capable of mapping the contents of texts to a representation in which the semantics of each symbol is explicit, and an inductor module, capable of formulating theories to explain chains of events.

The component responsible for the induction of theories implements a restricted version of the Solomonoff Predictor, capable of producing hypotheses pertaining to the set of Regular Languages. Such device presents elevated computational complexity and very high processing time even for very simple inputs. Nonetheless, this work presents new and interesting results showing its functional performance.

The pre-processing module of the proposed system consists of an implementation of Latent Semantic Analisys, a method which draws from statistical correlation to build a representation capable of approximating semantical relations made by human beings. It was used to index the more than 470 thousand texts contained in the first disk of the Reuters RCV1 *corpus*, resulting, through dozens of parameter variations, 71.5*GB* of data that were used for various statistical analises. The test results are convincing that the use of that pre-processing module leads to considerable gains in the system proposed.

The integration of the two components built into a full-fledged semantical analyser of Natural Languages presents itself, at this moment, unachievable due to the processing time required by the inductor module, and remains as a task for future work. Still, Solomonoff's Theory of Prediction shows itself adequate for the treatment of semantical analysis of Natural Languages, provided new ways of palliating its processing time are devised.

Keywords: Computer learning; Artificial intelligence; Natural language; Formal semantics

Sumário

Lista de Figuras

Lista de Tabelas

Lista de Algoritmos

1 Introdução

Em 1989, Richard Saul Wurman, um premiado arquiteto americano, ganhou notoriedade ao escrever o livro *Information Anxiety*, no qual constatava que um único exemplar do jornal *The New York Times* continha mais informações do que uma pessoa de nível médio era capaz de acumular durante toda a vida 300 anos antes (WURMAN, 1989). Oito anos depois, em 1997, uma reportagem publicada pela *Reuters Magazine* entitulada "Information overload causes stress" baseou-se em estatísticas levantadas pela universidade Humboldt, nos EUA, para demonstrar que a quantidade de informação escrita disponível no mundo dobrava a cada 5 anos (INFORMATION. . . , 1997). Esse crescimento tem-se intensificado e seus efeitos tornaram-se evidentes a partir do final do século XX, com a popularização dos sistemas de telecomunicações. Um indicador interessante desse fenômeno está na decisão de Wurman de publicar um novo volume de seu livro, em 2001, em que afirma (WURMAN, 2001, p. 8):

> Because we have greater access to information, many of us have become more involved in researching and making our own decisions, rather than relying on experts. [...] The opportunity is that there is so much information, the catastrophe is that 99 percent of it isn't meaningful or understandable. [...] We need to rethink how we present information because the information appetites of people are much more refined. [...] Everyone needs a personal measure to distinguish useful information from raw data.[1]

De fato, nos dias atuais, o volume de informação parece ter alcançado um ponto muito distante da capacidade humana de gerenciamento. Diante de um problema de desempenho associado a escala, espera-se que um estudante de computação inicie sua investigação pelo método que se encontra por trás do sistema. Isso porque é muito comum haver maneiras de aumentar a eficiência de um sistema a partir da alteração de seus métodos. Entretanto, essa não tem sido a abordagem mais adotada atualmente. Predomina o estudo de sistemas de recuperação de informação e de tradução automática

[1] Vertido para o português: "Porque temos mais acesso a informações, muitos se tornaram mais envolvidos em pesquisar e tomar suas próprias decisões em lugar de confiar em especialistas. [...] A oportunidade é que há tanta informação, e a catástrofe é que 99 porcento dela não tem significado nem é compreensível. [...] Nós temos que repensar a maneira como apresentamos a informação porque os gostos por informação das pessoas estão muito mais refinados. [...] Todos precisam de uma medida pessoal para distinguir informações úteis de dados puros."

de textos e similares, o que, aparentemente, são paliativos dedicadas a lidar com os sintomas de um problema fundamental: a incompatibilidade entre as formas de armazenamento da informação e a sua utilização.

A palavra *logos* tem origem no Grego Antigo (CRAIG, 1998, p. 818):

> *Logos* became an important term in almost all philosophical schools. It emerged about 700 BC as the accepted term for discourse at any length, though seldom if ever naming a single word [...] Plato uses the term logos *in* almost all senses [...] moreover for intellect, thought or intelligence [...] contrasted with sense-experience which Plato associates with instability and untruth [...] Aristotle defines *logos* as a composite significante utterance, but actually uses it in a wide variety of senses [...] In his metaphysics *logos* indicates and sometimes equates with the substance, form or essence of things...[2]

Embora tenha adquirido variados significados ao longo dos tempos, de forma sintética é possível compreender *logos* como pensamento, fala, razão ou significado. Há séculos estudiosos apresentam teses controversas sobre se o pensamento precede a linguagem ou vice-versa. Filósofos como René Descartes e Wilfrid Stalker Sellars e educadores como Jean Piaget e Lev Semenovich Vytgotsky sustentam opiniões divergentes sobre o assunto (ASHER, 2005). Aqui, interessa fundamentalmente firmar um dos pressupostos básicos deste trabalho: ao considerar o processamento de linguagem por máquinas, é essencial ter pensamento e linguagem como fatores profundamente vinculados.

1.1 Objetivo

O objetivo desta pesquisa é investigar a viabilidade e as características de um sistema de análise semântica de Linguagens Naturais baseado na Teoria da Predição de Solomonoff[3]. Nesse ensejo, os módulos pertinentes a esse sistema foram desenvolvidos e testados de maneira que se pudesse verificar o seu comportamento.

No que respeita à operacionalização da proposta, há fatores relevantes a observar. Por um lado, é necessário um dispositivo de indução capaz de observar uma seção de

[2] Vertido para o português: "*Logos* se tornou um termo importante em quase todas as escolas filosóficas. Ele emergiu por volta de 700 AC sendo aceito com o significado de discurso em qualquer concepção, apesar de raramente ou nunca designar uma única palavra [...] Platão usa o termo *logos* em quase todos os sentidos [...], mas mais comumente para intelecto, pensamento ou inteligência [...], em contraste com a sensação-experimentação que Platão associa à instabilidade e inverdade [...] Aristóteles define *logos* como uma expressão de significado composto, mas na verdade usa o termo numa ampla variedade de sentidos [...] Na sua metafísica, logos indica, e algumas vezes se equipara, a substância, forma ou essência de coisas..."

[3] Neste trabalho, por coerência com a terminologia em inglês "Solomonoff Prediction", o termo "predição", em "Preditor de Solomonoff", é usado como equivalente a "previsão", apesar deste último ser mais utilizado na língua portuguesa. A comunicação ou não dos fatos previstos é desimportante para o modelo, de maneira que a diferenciação dos termos não é necessária.

um universo e tecer teorias úteis sobre o comportamento desse universo como um todo, ou seja, um mecanismo que, a partir de amostragens, processe generalizações, análises e sínteses que possam atender a diferentes tipos de demandas. Por outro lado, dada a complexidade das Linguagens Naturais, precisa-se de mecanismos capazes de transpor textos para formas mais tratáveis sem prejuízo para o seu conteúdo. Unindo essas duas partes, propõe-se um analisador semântico de Linguagem Natural em escopo e escala restritos, mas que, em seu domínio, será capaz de transpor informação para uma representação de conhecimento. Sabe-se de antemão que a complexidade computacional do dispositivo proposto por Solomonoff é excessivamente alta para a construção de uma aplicação completa neste momento, de maneira que o objetivo principal deste trabalho é investigar o comportamento e as propriedades dos módulos de um sistema de análise semântica de Linguagens Naturais como o proposto. O resultado contribui para o estabelecimento da viabilidade, das qualidades e dos defeitos do sistema como um todo.

Os objetivos específicos, fatores motivadores importantes para as pesquisas envolvidas, são a investigação de cognição humana, mecanismos de indução, modelos de linguística e formalismos computacionais adequados aos requisitos de adaptatividade e capacidade indutiva do problema. Nominalmente, pretende-se pesquisar e desenvolver um dispositivo de indução de hipóteses a partir da observação de um número limitado de eventos, pesquisar e desenvolver um sistema capaz de transpor textos em Linguagem Natural para uma representação mais adequada ao processamento computacional e realizar experimentos com os sistemas construídos. Com isso, viabiliza-se uma meta de encontrar maneiras de integrar as contribuições dessas diferentes linhas de pesquisa.

1.2 Metodologia

A abordagem convencional à análise semântica de Linguagens Naturais se restringe ao estabelecimento de relações entre termos, tanto na área da computação (e.g. (MCGUINNESS; HARMELEN et al., 2004)) quanto na linguística (tem-se como exemplo a Semântica Formal proposta por (MONTAGUE, 1973)). É um tratamento que pode ser feito de maneira relativamente simples, mas que tem sérias limitações (suas características são discutidas no capítulo 3). A proposta de Solomonoff para a formulação de teorias para sequências de eventos (SOLOMONOFF, 1964) contém, na sua construção, uma maneira diferente de observar o problema. O processo de condensação de observações em algoritmos perde muito em simplicidade, mas a descrição de conhecimento resultante pode resolver os problemas introduzidos no início deste capítulo, particularmente o de incompatibilidade entre descrição e uso do conhecimento.

A solução final do dispositivo preditor proposto por Solomonoff (SOLOMONOFF, 1989) é incomputável até para problemas simples. Mesmo aplicando artifícios para obter soluções subótimas, a complexidade computacional do sistema é exponencial. Tendo em mente que, em Linguagens Naturais, a relação entre significante e significado é arbitrária – conforme defendido pelo linguista Saussure (SAUSSURE, 1983 apud ASHER, 2005, Vol. X p. 758), um dos pais da linguística moderna –, as chances de obter resultados significativos em um tempo praticável não são promissoras. Para melhorar as chances de viabilizar a aplicação do dispositivo sobre cadeias de eventos tão complexas quanto trechos de texto em Linguagem Natural, propõe-se pré-processar os dados utilizando métodos que exponham relacionamentos semânticos entre termos.

A Análise da Semântica Latente é uma técnica estatística que utiliza álgebra linear para expor relacionamentos indiretos entre termos em textos em Linguagem Natural. Ele foi proposto em (DEERWESTER et al., 1990) como um método para processamento de documentos para recuperação de informações. A sua baixa complexidade computacional, associada à capacidade de aproximar associações feitas por seres humanos (LANDAUER; DUMAIS, 1997; PAPADIMITRIOU et al., 2000) e à representação conveniente dos seus resultados, torna o método um candidato excelente. Uma discussão mais aprofundada sobre os motivos dessa escolha é colocada no capítulo 3.

Dos motivos expostos, delineia-se a construção do analisador semântico para Linguagens Naturais proposto: um sistema composto por um módulo de pré-processamento, que torna os textos de entrada mais tratáveis aplicando a eles o método de Análise da Semântica Latente, e por um módulo indutor de teorias sobre o conteúdo que gerou a descrição sendo processada, um dispositivo baseado na Teoria da Predição de Solomonoff. Há indicações em outros trabalhos de como se implementaria a teoria na sua forma mais completa (ROCHA, 2000), mas, no interesse de reduzir a complexidade computacional do sistema como um todo e desenvolvê-lo em um prazo curto, optou-se por desenvolver uma versão limitada da proposta de Solomonoff, restrita ao conjunto das Linguagens Regulares.

1.2.1 Embasamento linguístico

A proposta de analisar a semântica inerente a uma secção de texto em Linguagem Natural considera a dificuldade que os seres humanos, criadores e usuários dessa classe de linguagem, encontram em definir até o significado de palavras isoladas formalmente. As tentativas de aproximação de conceitos por analogia, que se encontram em dicionários e em enciclopédias, muito raramente chegam a uma raíz. Casa de pombo é um tipo de casa, casa é uma forma de construção, construção é uma organização de

elementos – ao final, sempre resta algo por definir.

O filósofo inglês John Locke escreveu, no século XVII, seu *Ensaio Acerca do Entendimento Humano*, em que registra (LOCKE, 1973, p. 258):

> As palavras não tendo naturalmente significado, a ideia que cada uma significa deve ser apreendida e retida pelos que farão intercâmbios de pensamentos e manterão discursos com outros em qualquer língua. Mas isto é difícil de ser realizado onde:
> Primeiro, as ideias que significam algo são muito complexas e formadas por um grande número de ideias reunidas.
> Segundo, onde o significado das ideias não tem conexão evidente na natureza, não havendo, deste modo, modelo estabelecido em nenhuma parte da natureza para retificá-las e ajustá-las.
> Terceiro, onde o significado da palavra é referido a um modelo, não sendo o próprio modelo reconhecido com facilidade.
> Quarto, onde o significado da palavra e a essência real da coisa não são exatamente equivalentes.

A aparente ausência de axiomas semânticos no conhecimento humano é o primeiro problema que se enfrenta na análise do sentido das palavras e na interpretação das sentenças e dos enunciados de textos em Linguagem Natural. Um sistema capaz de realizar essa tarefa tem de estabelecer, em primeiro lugar, as relações entre as palavras e os seus sentidos – nos termos definidos por Ferdinand de Saussure (ASHER, 2005; FALK, 2003), as relações entre significado e significante.

O segundo obstáculo a ser superado neste projeto é a análise da estrutura dos textos em Linguagem Natural. Claramente, as frases “João chutou a pedra” e “a pedra chutou João” têm interpretações radicalmente diferentes e a organização de palavras “pedra João a chutou” sequer admite interpretação. Assim, mesmo que haja um sistema que permita associar a cada palavra um significado, interpretar sentenças segue como um desafio.

A teoria da sintaxe que se encontra nos livros de gramática tradicionais é pouco formal, falha ao explicar a compreensão de boa parte das construções gramaticais pelas pessoas e, assim, não serve como ferramenta a ser utilizada para a solução das questões propostas (CHOMSKY, 1986). Entretanto, alguns dos conceitos de linguagem defendidos por pesquisadores que atuam nessa área mostram-se extremamente relevantes para este projeto.

Professor de Teoria Linguística, Sintaxe, Semântica e Filosofia da Linguagem no Massachusetts Institute of Technology (MIT), Noam Chomsky tem proposto uma nova abordagem à gramática das Linguagens Naturais: a Gramática Gerativa (CHOMSKY, 1965). Sua intenção é descrever, de maneira completa, formal e universal, as regras de construção de sentenças válidas em Linguagens Naturais. Trata-se de um projeto em

andamento, com uma boa dose de controvérsia em tópicos teóricos fundamentais, de maneira que, considerando seu atual estado de desenvolvimento, não se julgou prudente, e de fato se evitou, alicerçar a implementação concreta desta investigação na Gramática Gerativa. A opção, no entanto, não exime do exame do trabalho desenvolvido pelo linguista – em especial porque, neste trabalho, assume-se a hipótese que levou Chomsky a idealizar a Gramática Gerativa.

A hipótese inicial de Chomsky é a existência de um sistema de regras compartilhado por todos os seres humanos, um sistema natural a todos os indivíduos da espécie, que possibilita sua comunicação. Seu principal argumento tem como base a velocidade de aprendizado e a competência que crianças, expostas aos ambientes mais diversos, têm para aprender e usar Linguagens Naturais – o chamado "Paradoxo da Pobreza de Estímulo" ou o "Problema de Platão". Se é verdade que o mecanismo gerador de todas as Línguas Naturais é o mesmo e que a compreensão de suas regras é inata na espécie humana, não é necessário expor um indivíduo a mais que uma pequena fração das possibilidades de um idioma para que ele seja capaz de compreendê-lo em sua totalidade. (HAUSER; CHOMSKY; FITCH, 2002).

Em outro de seus trabalhos, o estudioso afirma (CHOMSKY, 1986, p. 27, 33–35):

> There does exist what we have called an internalized language and that it is a problem of the natural sciences to discover it. [...] In the sciences, at least, disciplines are regarded as conveniences [...] and their boundaries shift or disappear as knowledge and understanding advance. In this respect, the study of language [...] is like chemistry, biology, solar physics, or the theory of human vision.[...] Linguistics becomes part of psychology...[4]

No decorrer deste projeto, a análise sintática da linguagem é encarada como um problema de indução das regras que produzem construções válidas dessa linguagem. Na escolha do método para a definição dessas regras, tem-se como fundamental "Problema de Platão", ou seja, considera-se a necessidade de estabelecer um conjunto de regras finito e relativamente simples frente à infinitude de construções permitidas pelas línguas naturais. Quanto à análise semântica de trechos completos de texto, foram escolhidos excertos com conceitos e ideias em construções básicas produzidas a partir da organização das palavras. Dessa maneira, *a priori*, a proposta metodológica desta investigação é dar à semântica tratamento análogo ao conferido à sintaxe.

[4] Vertido para o português: "De fato existe o que nós chamamos de linguagem internalizada e é um problema das ciências naturais descobri-la. [...] Nas ciências, ao menos, divisões disciplinares são vistas como conveniências [...] e as suas fronteiras movem-se ou desaparecem conforme o conhecimento e a compreensão avançam. Sob esse aspecto, o estudo da linguagem [...] é como o da química, biologia, física solar ou a teoria da visão humana. [...] Linguística se torna parte da psicologia..."

1.3 Estrutura da dissertação

O trabalho relatado nesta dissertação consta de uma pesquisa prévia sobre Aprendizado de Máquina e tratamento de Linguagens Naturais, do desenvolvimento de ferramentas para a avaliação das propostas feitas e, finalmente, de testes e da exploração dos resultados obtidos. A estrutura do texto reflete essa evolução.

A primeira parte desta dissertação contém uma breve coletânea dos conceitos envolvidos no desenvolvimento do projeto – uma revisão bibliográfica resumida dos fundamentos teóricos para as escolhas feitas na investigação proposta. Nela, são explorados os assuntos chave abordados no trabalho, respectivamente: aprendizado por máquinas, sistemas de representação de conhecimento e o estudo das Linguagens Naturais. Os dois primeiros capítulos são mais voltados à questão dos sistemas computacionais necessários para a realização de um sistema capaz de analisar a semântica de textos em Linguagem Natural, enquanto o terceiro embasa a abordagem do ponto de vista linguístico e cognitivo.

A segunda parte deste trabalho trata dos experimentos feitos com os programas de computador desenvolvidos para verificar o comportamento dos algoritmos que se propõem usar no sistema de análise semântica de Linguagens Naturais. Primeiro, é descrita a implementação, os testes e os resultados obtidos de uma adaptação simplificada do Preditor de Solomonoff – o coração da proposta deste trabalho. Em seguida, são expostos uma implementação do sistema de Análise da Semântica Latente proposto em (DEERWESTER et al., 1990), o ambiente usado para verificar o seu desempenho e os resultados dos testes. Esse segundo sistema compõe o módulo de pré-processamento de textos, conforme descrito na seção 1.2.

A terceira parte da dissertação encerra o trabalho com algumas considerações finais. Ela apresenta as conclusões sobre o funcionamento do Preditor de Solomonoff e do Analisador da Semântica Latente desenvolvidos, considerações sobre a sua integração e expõe as vias mais promissoras que se vislumbrou para desenvolvimentos futuros desta pesquisa.

Parte I

Conceitos

2 Aprendizado por Máquina

A expressão “Aprendizado por Máquina” é utilizada para designar o funcionamento de sistemas computacionais capazes de alterar o seu comportamento em resposta a estímulos externos ou a experiências acumuladas no passado (ALPAYDIN, 2004). A apresentação desse conceito suscita uma questão pertinente: qual o motivo que justifica a criação de um sistema desse tipo em lugar de simplesmente solucionar o problema (NILSSON, 1996) (e.g. criar um programa que converte fala em escrita em vez de um programa que aprende a fazer a mesma conversão a partir de exemplos).

Há várias situações nas quais esse comportamento é desejável. O foco deste trabalho é um exemplo de aplicabilidade de técnicas de Aprendizado por Máquina, considerando que o problema da análise semântica de Linguagens Naturais é extremamente difícil de descrever – sequer tratando de seres humanos se encontram instruções formais que esclareçam como se realiza essa atividade. As Linguagens Naturais podem produzir uma quantidade infinita de sentenças e textos contendo também uma quantidade infinita de ideias e conceitos. As regras que regem sua composição, se existem, não são conhecidas.

A exposição sobre Representação de Conhecimento do capítulo 3 aborda e fornece parte da fundamentação para a definição dos métodos de aprendizado por máquina escolhidos para este trabalho, mas há motivos além da adequação de representação que devem ainda ser esclarecidos. O primeiro ponto relevante é a divisão das tarefas estabelecidas no capítulo 4: a obtenção dos significados das palavras deve ser tratada como uma tarefa separada e anterior à análise da semântica de sentenças e textos. Dado o foco do projeto no processamento de textos, a importância do módulo responsável pela análise de palavras está relacionada, principalmente, à adequação dos seus resultados ao módulo responsável por trechos compostos. Os requisitos deste restringem as alternativas disponíveis para aquele e a sua escolha deve ser priorizada.

Um atributo importante que diferencia os métodos de aprendizado por máquina é o grau de interferência por seres humanos que eles exigem. Na análise semântica de Linguagens Naturais, é desejável que este seja o menor possível. A análise semântica de

textos está no rol das atividades humanas consideradas subjetivas e é de se esperar que as características do operador influenciem os resultados do sistema. Isso não só dificultaria a sua validação, mas também comprometeria a sua utilidade como dispositivo de uso genérico, uma vez iniciado o treinamento. Quatro métodos de aprendizado por máquina que poderiam ser aplicados, tendo em mente esse requisito, são os Algoritmos Genéticos, as Redes Neurais, os métodos probabilísticos baseados no Teorema de Bayes e métodos estatísticos.

Algoritmos Genéticos são um método de aprendizado por máquina que emula a teoria biológica evolucionista (MITCHELL, 1996). São construídas diversas instâncias de um protótipo da solução para o problema que se quer resolver, todas diferentes em relação a certas características pré-determinadas. Colocadas para trabalhar no problema, selecionam-se as *n* mais bem-sucedidas, de acordo com uma função-objetivo. Estas são, então, combinadas de maneiras diferentes em uma nova geração de instâncias. Se o protótipo da solução, os parâmetros escolhidos para variação, a forma de recombinação e, principalmente, a função-objetivo forem adequados, o sistema converge para uma solução. No entanto, quando se trata de análise semântica de Linguagens Naturais esses requisitos são um obstáculo difícil de superar. Não se conhecem as características do problema bem o suficiente sequer para construir um protótipo da sua solução. O acompanhamento dos resultados por meio de uma função exigiria também que houvesse ao menos uma medida do que é correto como descrição formal do conhecimento contido em um trecho de texto – medida essa que também não está disponível. Por isso, a escolha pela aplicação de Algoritmos Genéticos como técnica independente não é adequada para o problema em estudo.

Redes Neurais, ou Redes Multicamada de Perceptrons (RUSSELL; NORVIG, 2003; ALPAYDIN, 2004), são sistemas inspirados no funcionamento do cérebro. Diversos elementos extremamente simples, normalmente capazes apenas de retransmitir ou bloquear sinais que chegam a eles, são conectados em camadas. As conexões são direcionadas e têm pesos que determinam quanto do sinal emitido por um elemento chega àqueles aos quais está conectado. Esses pesos variam de acordo com o resultado das comunicações anteriores entre os elementos envolvidos. O resultado é que, dada uma rede adequada, depois de um período de treinamento, que consiste na exposição a diferentes sinais de entrada, o sistema passa a reconhecer determinados padrões. No entanto, da maneira como são construídas atualmente, redes neurais não podem ser retreinadas. Esse é o primeiro obstáculo sério para a sua aplicação neste projeto já que, conforme foi esclarecido anteriormente, a análise semântica requer um processo de aprendizado contínuo. A linguagem e o universo de ideias são infinitos. É necessário que o sistema seja capaz de se modificar e de contextualizar novas informações permanentemente. Redes Neurais

também não compõem, assim, uma alternativa adequada para este trabalho.

Existem diversos métodos de Aprendizado por Máquina baseados na equação de probabilidades condicionais do Teorema de Bayes. A maioria dos eventos que se quer descrever e dos problemas que se quer resolver segue padrões e a maneira de restrição do universo de possibilidades fornecida pelo Teorema é extremamente atraente. Ele é usado em dispositivos simples de processamento de Linguagem Natural, como *N-Gramas* (JURAFSKY; MARTIN, 2008), mas também é base de métodos bem mais complexos como Modelos Ocultos de Markov (RUSSELL; NORVIG, 2003). Uma teoria particularmente relevante baseada nesse aparato probabilístico é a Teoria da Predição de Ray Solomonoff. Usando elementos da Teoria da Informação, Solomonoff propôs uma teoria e um método de aprendizado (SOLOMONOFF, 1964) extremamente adequados ao módulo de análise semântica deste projeto. Além de estar embasado em uma explicação convincente do processo de aprendizado, o método proposto por Solomonoff tem a convergência garantida para casos genéricos em tempo extremamente curto (seção 2.1) e é todo estruturado para uma aprendizagem contínua. Por ser um método genérico, entretanto, a sua complexidade é bastante elevada. Exceto por relatos do autor, que pesquisou esse dispositivo ativamente de 1964 a 2009, e por implementações tutoriais extremamente simples, não foram encontrados, para citação aqui, trabalhos que demonstrem a sua aplicação e os seus resultados. Se esse não é um sinal positivo para a sua adoção neste trabalho, também não se pode dizer que seja negativo. O problema da análise semântica de textos em Linguagem Natural é muito estudado, mas não-resolvido. Há pouca chance de verificar avanços significativos nessa área com os métodos aplicados tradicionalmente. Assim, faz-se a opção por essa teoria bem embasada, mas pouco usada, com a convicção de que qualquer resultado obtido será relevante.

Decidido o método para análise semântica, resta ainda a escolha pela maneira de executar o processamento das palavras. A Teoria da Predição de Solomonoff trabalha com a verificação de hipóteses para descrever sequências de eventos no intento de extrapolar a ocorrência de novos eventos. Dessa maneira, é importante priorizar a simplicidade das descrições dos eventos, de maneira que os atributos envolvidos nas relações de causa e efeito estejam sempre expostos para o mecanismo. Discorre-se brevemente, no capítulo 3, sobre representações vetoriais do conhecimento. Esse tipo de representação é extremamente adequado para a situação que se coloca e se torna particularmente interessante quando se observam os resultados de pesquisas realizadas nas últimas décadas acerca de técnicas que produzem representações desse tipo.

Entre os métodos estatísticos de Aprendizado por Máquina, destaca-se um que surgiu em 1990: a Análise da Semântica Latente (DEERWESTER et al., 1990) (mais

conhecido pela sigla LSA, do original em inglês *Latent Semantic Analysis*). Baseado na redução de um espaço vetorial contendo uma amostra de eventos, o método constrói uma base ortogonal de atributos implícitos extraídos automaticamente. O resultado da sua aplicação a um *corpus* não tem significado direto para um ser humano, mas, fixando um número adequado de dimensões para o espaço vetorial resultante, as associações que se pode obter do sistema são impressionantes (LANDAUER; DUMAIS, 1997). Há que considerar que o procedimento é puramente matemático e o número de dimensões adequado para o espaço vetorial resultante, por exemplo, tem de ser obtido empiricamente por meio de experimentos. Essa técnica será adotada neste trabalho para o processamento de palavras por causa da sua adequação ao estágio posterior e por causa dos resultados extremamente consistentes que se têm obtido. É importante notar que há propostas recentes de melhorias ao modelo original, com a aplicação de artifícios estatísticos mais complexos (HOFMANN, 1999; BLEI; NG; JORDAN, 2003). Essas propostas, apesar de trazerem melhorias ao desempenho do método, não alteram radicalmente o seu funcionamento. Assim, neste trabalho, opta-se por utilizar a proposta original de LSA, mais testada e de implementação mais simples. Caso o desempenho do método venha a ser um obstáculo, a sua troca poderá ser feita sem prejuízo.

2.1 A Teoria da Predição

Em 1964, Ray Solomonoff propôs uma maneira de sistematizar o processo indutivo (SOLOMONOFF, 1964). Essa maneira, a Teoria da Predição ou Teoria da Indução de Solomonoff, é baseada no teorema de Bayes, interpretando-o como um método de teste de hipóteses:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{2.1}$$

Ou seja, dada a ocorrência de um evento, a probabilidade de uma hipótese ser verdadeira é dada pela probabilidade que ela, sendo verdadeira, atribui ao evento, multiplicada pela probabilidade de que a hipótese esteja correta e dividida pela chance de ocorrência do evento. Lido dessa maneira, o teorema de Bayes parece mesmo um instrumento poderosíssimo de verificação de hipóteses. Não fosse a dificuldade de formular hipóteses e estabelecer a correção das probabilidades *a priori* $P(H)$, o problema da descoberta de explicações para eventos estaria essencialmente resolvido. Os passos fundamentais dados por Solomonoff foram, justamente, a formalização de um mecanismo para estabelecer $P(H)$ e a sinalização do universo de hipóteses a ser explorado. O matemático propõe também métodos para a busca dessas hipóteses,

mas esses são mais complexos (inclusive na conotação computacional) e menos bem definidos. Mesmo nos seres humanos, o processo indutivo aparenta ser essencialmente heurístico. Em um trabalho mais recente (SOLOMONOFF, 2003a), Solomonoff aponta numa direção que parece promissora para mitigar o problema da complexidade de geração e verificação de hipóteses: a divisão de problemas em diferentes domínios.

As próximas subseções aprofundarão um pouco mais os principais tópicos da Teoria da Predição de Solomonoff, no intuito de mostrar sua aplicabilidade na síntese de conhecimento a partir de dados e, portanto, na análise semântica de textos em Linguagem Natural.

2.1.1 Complexidade de Kolmogorov

A complexidade algorítmica, ou complexidade de Kolmogorov, é uma medida da quantidade de informação contida em uma cadeia de símbolos. É intuitivo perceber que há menos informação na cadeia 10^{100} do que na imensa maioria dos números de 100 dígitos que se pode imaginar. Uma proposta óbvia para a medida da complexidade é o comprimento da menor cadeia que representa aquela mesma informação. No entanto, esse método simples não é, a princípio, um bom indicativo da complexidade da informação contida na cadeia, uma vez que é possível criar infinitas linguagens para descrever os mesmos conceitos, cada uma usando cadeias de comprimento diferente. Como garantir que essa medida seja mesmo relativa à informação e não à linguagem de referência utilizada?

A solução para esse dilema vem com a Tese de Church (apud (LI; VITÁNYI, 1997)). Sua proposta: o conceito de computabilidade é independente do método utilizado para a computação. Conforme a tese, qualquer procedimento efetivamente calculável pode ser representado por um computador universal e todos os computadores universais são equivalentes por poderem simular o comportamento uns dos outros executando um número constante de passos. Com isso, a definição a partir do comprimento da cadeia passa a fazer sentido, de maneira que a complexidade $K(x)$ da cadeia x, de comprimento $l(x)$, descrita usando o formalismo f, capaz de computação universal, é dada simplesmente por:

$$K(x) = min\{l(p)|f(p) = x\} \tag{2.2}$$

Dessa equação depreende-se que a complexidade da cadeia x, quando descrita a partir de um formalismo f, é o comprimento da menor cadeia que, quando aplicada a f, gera x. Como foi dito, segue da Tese de Church que, usando um outro formalismo

M com capacidade de computação universal, é possível simular f com um número constante de passos e obter para a mesma cadeia a mesma complexidade exceto por uma constante. Essa constante pode ser arbitrariamente grande, mas do ponto de vista formal, isso não invalida o resultado. Mais que isso: intuitivamente faz muito sentido. Alguns sistemas levam a descrições mais naturais para certos problemas do que outros, ainda que seja possível converter diretamente entre eles. O problema não se torna mais complexo devido à descrição mais longa e sempre é possível explicar o outro sistema de descrição antes de utilizá-lo.

A equação (2.2) é, no entanto, incomputável. A única maneira de descobrir se um algoritmo produz uma dada cadeia x é executar seus passos. É impossível saber *a priori* se ele para para alguma entrada. O melhor que se pode obter realisticamente (i.e. em um tempo compatível com a expectativa humana) são aproximações restringindo-se o tempo de busca ou o espaço de algoritmos a serem explorados.

2.1.2 Probabilidade algorítmica

O Princípio da Navalha de Occam ou Lei da Parcimônia, atribuído a Guilherme de Occam, que viveu na Alta Idade Média, diz (ENCYCLOPÆDIA. . . , 2010):

> *"Pluralitas non est ponenda sine necessitate."*
> ou *"Pluralidades não devem ser assumidas sem necessidade."*

Isso significa, *grosso modo*, que a simplicidade é fator decisivo da plausibilidade de uma descrição. Explicações mais complexas devem ser preteridas em favor das mais simples. A partir do momento em que existe uma definição formal de complexidade, é viável verificar esse princípio que parece permear a ciência, o que de fato foi feito.

Considere um alfabeto binário $A = \{0,1\}$. Se cada símbolo desse alfabeto tiver a mesma probabilidade de ocorrência, é claro que a probabilidade de uma cadeia d de comprimento $|d|$ será:

$$\mu(d) = 2^{-|d|} \tag{2.3}$$

Ou seja, a probabilidade da cadeia *1* é 2^{-1}, de *11* é 2^{-2}, *et cetera*. O número decimal *8* pode ser representado, em binário, como *100*, ou *0100*, ou *00100*, ... Na verdade, há infinitas representações para esse número. A probabilidade algorítmica é a probabilidade, fixado um alfabeto, de geração da cadeia que representa aquele algoritmo por um dispositivo que fique escrevendo símbolos aleatoriamente. É a probabilidade de que um algoritmo seja gerado espontaneamente em um ambiente caótico. Note-se que

essa distribuição de chances, por assim dizer, dá maior peso às descrições mais curtas, sendo dominada pelo termo referente à complexidade de Kolmogorov. Por causa das suas características, Solomonoff propõe que essa semimedida μ seja usada para estimar a probabilidade *a priori* das hipóteses, de maneira que:

$$P(H) = 2^{-|H|} \tag{2.4}$$

É sabido, entretanto, que diferentes algoritmos podem levar aos mesmos resultados. Considerada a linguagem livre de prefixo D, formada por todas as descrições (algoritmos geradores) d_i de uma cadeia x, pode-se dizer que as probabilidades dadas pela equação (2.3) são uma semimedida[1] dessas descrições para x. Essa semimedida pode ser normalizada para a obtenção de uma relação de probabilidade[2]: a probabilidade de uma descrição, dada uma cadeia de saída, ou a probabilidade de um algoritmo ser o responsável pelo resultado observado. A restrição de D ser livre de prefixo existe para eliminar as descrições trivialmente redundantes (e.g. se lidas do dígito menos significativo para o mais significativo, todas as representações dadas, no exemplo acima, para o numeral *8* são trivialmente redundantes), que não adicionam nenhuma informação ao conjunto.

2.1.3 Predição

Solomonoff propôs aplicar o conceito de probabilidade algorítmica à predição de resultados (SOLOMONOFF, 1964). A questão básica é criar um dispositivo capaz de responder à seguinte pergunta: dada uma cadeia inicial x, qual a probabilidade de que o próximo símbolo seja a? Da teoria clássica de probabilística, sabe-se que a probabilidade de que se obtenha a cadeia xy, dada a ocorrência de x, obedece à equação:

$$P(xy|x) = \frac{P(xy)}{P(x)} \tag{2.5}$$

Para prever a probabilidade de desenvolvimento da sequencia x em xy, pode-se usar o comportamento observado até x, empregando as probabilidades obtidas pela normalização da semimedida descrita na equação (2.3). A função de probabilidades P'_M obtida pela aplicação dessas semimedidas acompanha a função P_M que gera a cadeia

[1] Uma semimedida é uma relação que associa um número real $\mu \in [0,1]$ a cada elemento de um conjunto, onde $\sum \mu \leq 1$.

[2] Uma probabilidade é apenas uma semimedida cuja soma resulta em um.

com um erro quadrático definido por (WILLIS, 1970; SOLOMONOFF, 1978):

$$\sum_{m=1}^{n} (P_M(x_{m+1} = \sigma | x_1 \ldots x_m) - P'_M(x_{m+1} = \sigma | x_1 \ldots x_m))^2 \leq 0.5 \cdot \ln D'_M \tag{2.6}$$

Onde:

m é o índice do símbolo da cadeia em análise;

$\mathbf{P_M}(\mathbf{x_{m+1}} = \sigma | \mathbf{x_1} \ldots \mathbf{x_m})$ é a probabilidade de que o gerador de dados produza o símbolo σ após a cadeia $x_1 \ldots x_m$; e

$\ln \mathbf{D'_M}$ é um fator constante dependente do dispositivo gerador da distribuição universal e é aproximadamente igual a $K \cdot \ln 2$, sendo K a complexidade de Kolmogorov do dispositivo.

Como esse erro quadrático é independente do comprimento da cadeia, nota-se, por indução, que ele decai mais rapidamente que o inverso do comprimento da cadeia. É uma taxa de convergência alta que vale mesmo para dispositivos preditores relativamente complexos (nos quais D'_M é maior), mas que favorece seus contrapartes mais simples.

Solomonoff estendeu o modelo para lidar com conjuntos de dados não ordenados (SOLOMONOFF, 1999). Mais tarde, em 2003, descreveu uma maneira de utilizá-lo para uma questão ainda mais genérica: o mapeamento de um conjunto de cadeias para outro, de uma maneira que pode ser vista como a localização da melhor resposta para uma pergunta (SOLOMONOFF, 2003b).

A aplicação inicialmente proposta da Teoria da Predição se presta a análises mais simples de textos em Linguagem Natural. Poderia ser usada, por exemplo, para criar um corretor capaz de analisar consistência de estilo de um texto. O modelo estendido para conjuntos de dados não ordenados já seria capaz de sintetizar conhecimento a partir de textos, criando uma descrição comprimida da informação neles contida, suficiente à extrapolação para outras informações. A terceira aplicação, se implementada, forneceria algo verdadeiramente interessante: um sistema especialista genérico apto a aprender de maneira similar aos humanos.

2.1.4 Busca por hipóteses

A subseção anterior descreve um método de validação e aperfeiçoamento de hipóteses para descrever uma série de eventos, mas não menciona a origem das hipóteses. Para que pelo menos uma aproximação do procedimento possa ser completada, é necessária

alguma maneira de encontrar hipóteses potencialmente válidas num espaço restrito de funções. Em (SOLOMONOFF, 2003a) é proposto que esse domínio seja o das funções parcialmente recursivas e em (SOLOMONOFF, 2005) são enumerados alguns candidatos para localizar as hipóteses:

a. Lsearch, um método de busca universal proposto por Levin e estendido com a contribuição de Solomonoff.

b. PPM (*Prediction by partial matching* ou Predição por correspondência parcial), técnica estatística usada para compressão de dados que, segundo Solomonoff, é "surpreendentemente rápido e preciso para induções aproximadas [...], mas na forma atual tem um custo de processamento muito alto." (SOLOMONOFF, 2005, p. 2 – vertido para o português)[3].

c. Descoberta de gramáticas livres de contexto estocásticas.

d. Programação genética.

Qualquer que seja o método escolhido, é certo que essa operação será responsável pela maior parte do tempo de processamento de um dispositivo baseado na Teoria da Predição.

[3]No original: "a very fast, surprisingly precise method for approximate induction [...] but at the moment there seems to be a very large speed penalty."

3 Representação do conhecimento

O volume de livros e artigos dedicados à busca de formas para a representação do conhecimento na disciplina da Inteligência Artificial (MYLOPOULOS, 1981; BRACHMAN; FIKES; LEVESQUE, 1983; BRACHMAN; LEVESQUE, 1985; IWANSKA; SHAPIRO, 2000; LENAT et al., 1990) é um indicador significativo da importância e da complexidade do tema. Coleta e processamento de informações sobre qualquer assunto exigem uma representação dos elementos pertinentes a ele.

Representações são modelos de algo que se deseja analisar. No caso, o que as diferencia, tornando importante seu estudo e escolha de uma determinada alternativa para uma certa finalidade, é a facilidade de seu processamento. Representações diferentes privilegiam atividades diferentes, de maneira que a adequação da forma de representação às intenções do usuário é muitas vezes fator que determina uma solução elegante ou uma falha retumbante (DAVIS; SHROBE; SZOLOVITS, 1993).

Boa parte dos estudos atuais sobre interpretação e descrição do conhecimento na área da computação fundamenta-se no inter-relacionamento de diferentes termos. Define-se "gato" como "mamífero", "mamífero" como "vertebrado", "vertebrado" como "possuidor de coluna vertebral", *et cetera*. Essa abordagem, chamada de Rede Semântica (MYLOPOULOS, 1981), tem mérito, pois o conhecimento é extremamente dependente dos relacionamentos entre conceitos. Por causa da relativa facilidade de implementação e por sua organização ser perfeitamente adequada à distribuição das informações em núcleos independentes, esse tipo de representação tem ganhado popularidade na última década e está sendo usada em propostas como a Web Semântica (MCGUINNESS; HARMELEN et al., 2004; HENDLER; BERNERS-LEE, 2010). No entanto, mesmo que sistemas desse tipo possam levar a resultados bastante positivos na análise da semântica de um texto sobre, por exemplo, biologia, não se pode dizer que tenham mais conhecimento sobre o assunto do que uma máquina de escrever tem sobre gramática. A presença de elementos fundamentais não é suficiente para caracterizar o conhecimento. As inferências que se podem extrair de uma representação desse tipo estão limitadas à Lógica Proposicional, o que restringe extrapolações corriqueiras entre os seres humanos.

Na área de Recuperação de Informações, tem obtido bastante sucesso nas últimas décadas uma forma de representação que apresenta similaridades com as Redes Semânticas. Trata-se das representações em espaços vetoriais (MAGNINI; CAVAGLIA, 2000; DEERWESTER et al., 1990; HOFMANN, 1999; BLEI; NG; JORDAN, 2003). São extremamente simples: cria-se um espaço vetorial de dimensões análogas a alguns atributos e a representação de conceitos é feita por meio de vetores nesse espaço. Tradicionalmente, em textos sobre o assunto, essa técnica não compõe o rol de métodos para representação do conhecimento. Entretanto, ela tem tido uma participação significativa e crescente no campo da inferência devido, principalmente, aos resultados obtidos com sua aplicação, além da possibilidade de automatização completa da construção de bases de conhecimento desse tipo, cuja objetividade dos resultados depende apenas das escolhas feitas na distribuição do *corpus* usado para gerá-las, o que, é de frisar, é uma grande vantagem em relação às redes semânticas, frequentemente dependentes da interferência direta de especialistas.

Outra abordagem, comum em provadores de teorema automáticos, é o uso de asserções em forma lógica (Lógica de Primeira Ordem, Lógica Fuzzy ou outra) (FITTING, 1996). Esse método de representação do conhecimento ganha muito em poder em relação às redes semânticas, devido à possibilidade que oferece de melhor especificação do contexto das asserções. Por exemplo, é possível criar uma descrição para um avião e depois fazer a asserção de que todo avião sem asas cai, sem qualquer implicação para aviões com asas ou outros elementos sem asa. Numa rede semântica, uma tentativa desse tipo resultaria em generalizações indesejáveis ou na criação de novas e exageradas especializações de elementos (em vez de criar somente uma entidade “Avião”, teríamos algo como “AviãoComAsas” e “AviãoSemAsas”). Uma aplicação da abordagem com asserções lógicas que merece atenção especial foi feita pelo linguista Richard Montague, que usou Lógica Intencional para representar a semântica de sentenças em Linguagem Natural (ASHER, 2005; MONTAGUE, 1973), trabalho de grande impacto sobre a compreensão de análise semântica de línguas naturais.

Uma terceira abordagem, menos comum nos dias de hoje, é a representação do conhecimento na forma de algoritmos. Linguagens funcionais como LISP (família de linguagens de programação concebida por John McCarthy em 1958 que evoluiu e se desenvolveu para aplicações em Inteligência Artificial (RUSSELL; NORVIG, 2003)), que fazem pouca distinção entre dados e procedimentos, já foram muito usadas para esse tipo de tarefa em sistemas especialistas e similares (MYLOPOULOS, 1981). A facilidade de inferência a partir de descrições algorítmicas e o potencial para concisão das asserções nessa representação certamente excedem as duas outras técnicas expostas nesta seção. Por outro lado, a dificuldade de alimentar e corrigir esse tipo de base de

conhecimento é um obstáculo para sua manutenção manual. No entanto, a proposta de um sistema indutivo automatizado não só oferece uma maneira para superar essa dificuldade como também estabelece, na representação algorítmica, uma teoria do aprendizado convincente e um método de inferência eficiente (SOLOMONOFF, 1964) (mais detalhes na seção 2.1).

Este projeto se propõe a usar algoritmos como representação interna do conhecimento contido em organizações de palavras por causa das suas vantagens e por ser uma forma de representação adequada a uma técnica de Aprendizado por Máquinas extremamente atraente. Sempre assumindo, como Saussure (cujas teorias são brevemente apresentadas no capítulo 4), que o conteúdo semântico das palavras é arbitrário, no entanto, admite-se que obter uma representação algorítmica do conhecimento contido em cada uma delas provavelmente exigiria recorrer à interferência humana e à criação, mesmo que disfarçada, de uma rede semântica ou outra estrutura taxonômica similar. A proposta, então, é a utilização de uma representação vetorial para as palavras, obtida a partir de um método de extração da semântica latente – Análise da Semântica Latente (DEERWESTER et al., 1990). Essa técnica será mais bem descrita na seção 3.1

3.1 Análise da Semântica Latente (LSA)

A inferência automática de associações semânticas entre palavras e seus contextos é um problema bastante intrincado da Inteligência Artificial. Um dos métodos mais simples e bem-sucedidos de indexação de textos para recuperação de informação (YATES; NETO, 1999) foi proposto por Salton em (SALTON; BUCKLEY, 1988). Conhecida por *tf-idf* ("frequência do termo-inverso da frequência no documento" ou, no original, "*term frequency-inverse document frequency*"), essa técnica consiste em fazer uma contagem global e texto-a-texto de todos os termos em um *corpus*. Estabelece-se então um espaço vetorial onde cada dimensão corresponde a um termo e cada texto é representado como um somatório dos seus componentes. O módulo de cada termo é calculado pela equação:

$$|t| = \frac{tf \cdot \log \frac{N}{n}}{\sqrt{\Sigma W_{t_i}^2}} \tag{3.1}$$

Onde:

t é o vetor do termo;

tf é a frequência de ocorrência do termo no documento;

N é o número total de documentos no corpus sendo examinado;

n é o número de documentos no corpus que contêm o termo; e

$\sqrt{\sum \mathbf{W_{t_i}^2}}$ é um fator de normalização que consiste na raiz da soma dos quadrados dos pesos de todos os termos pertencentes a esse documento.

Seguindo a abordagem vetorial para a representação de textos em *corpora*, Deerwester, junto a um grupo de pesquisadores, propôs em 1990 (DEERWESTER et al., 1990) o método de Análise da Semântica Latente (LSA, *Latent Semantic Analisys* no original). A ideia por trás do método é bastante simples: reduzir o número de dimensões do espaço vetorial produzido em modelos como o *tf-idf* retendo, tanto quanto possível, as relações entre os vetores originais. Existe um número mínimo de dimensões necessárias para representar, sem perda, o espaço vetorial determinado por um *corpus*. A redução dimensional além desse limite, dependendo de como for feita, pode reduzir o ruído da amostra (relações contextuais indevidas ou irrelevantes) e trazer à tona relações fortes indiretas. Resultados com essas características são mostrados em diversos trabalhos, em especial (DEERWESTER et al., 1990; LANDAUER; DUMAIS, 1997).

Para reduzir o número de dimensões em um espaço vetorial com a menor perda possível de informação, o LSA se baseia no método de Decomposição em Valores Singulares (SVD, no original *Singular Value Decomposition*), proposto em 1965 por Golub e Kahan (GOLUB; KAHAN, 1965). Esse método decompõe uma matriz A em três outras:

$$A = U \cdot \Sigma \cdot V^T \tag{3.2}$$

Nessa decomposição:

U é uma matriz ortogonal unitária cujas colunas são os autovetores de $A \cdot A^T$;

V é outra matriz ortogonal unitária, mas suas colunas são os autovetores de $A^T \cdot A$; e

Σ é uma matriz diagonal contendo os valores singulares de A – esses valores são as raízes quadradas não-negativas dos autovalores de $A \cdot A^T$ e $A^T \cdot A$.

Escolhendo a decomposição na qual os valores de Σ estão em ordem decrescente, o resultado obtido mantendo somente as n primeiras linhas e colunas de Σ (e, assim, as n primeiras colunas de U e n primeiras linhas de V) é, para um grande número de aplicações, a melhor aproximação em n dimensões que se pode obter para A. É aquela que minimiza o somatório do quadrado dos erros: $|A - A_n|^2 = \sum_{i,j}(A_{i,j} - C_{i,j})^2$

(DEERWESTER et al., 1990; BERRY; DUMAIS; O'BRIEN, 1995; PAPADIMITRIOU et al., 2000). Essa aproximação A_n será dada então pela seguinte equação:

$$A_n = U_n \cdot \Sigma_n \cdot {V_n}^T \tag{3.3}$$

Uma maneira comum de computar a proximidade entre vetores em aplicações de recuperação de informação, utilizada no artigo seminal sobre LSA, é o cosseno do ângulo entre eles (DEERWESTER et al., 1990; LANDAUER; DUMAIS, 1997; YATES; NETO, 1999; WIDDOWS, 2008). Note, no entanto, que a matriz A_n obtida não pode ser utilizada para isso. Depois da redução do número de dimensões, não é mais possível diferenciar documentos ou termos. Para obter os vetores correspondentes a cada documento, é necessário transpôr os documentos para o novo espaço vetorial. Por causa dos seus papéis na reconstrução de A, U representa a matriz de termos e V, a matriz de documentos (BERRY; DUMAIS; O'BRIEN, 1995). O objetivo no modelo é calcular a distância entre documentos. Manipulando a equação (3.2) para isolar a matriz V, obtém-se:

$$V = A^T \cdot U \cdot \Sigma^{-1} \tag{3.4}$$

É a partir da equação (3.4) que se chega à seguinte fórmula, apresentada em (BERRY; DUMAIS; O'BRIEN, 1995):

$$d_k = d^T \cdot U_k \cdot \Sigma_k^{-1} \tag{3.5}$$

Na equação (3.5), d_k é a representação do vetor de documento d no espaço reduzido obtido truncando o resultado da aplicação do SVD. A mesma equação pode ser usada para textos de busca, já que estes são simplesmente textos (presumivelmente) curtos expressando uma consulta de usuário. A similaridade entre um documento d e uma consulta q é dada então simplesmente pelo cosseno do ângulo entre os vetores que os representam. Esse cálculo pode ser feito a partir do produto escalar entre os vetores, de maneira que:

$$sim(d,k) = \cos(\widehat{dk}) = \frac{d \cdot k}{|d||k|} \tag{3.6}$$

Em (LANDAUER; DUMAIS, 1997) são apresentados resultados de alguns experimentos diretamente ligados ao aprendizado de línguas. Utilizou-se o método LSA para indexar a *Enciclopédia Acadêmica Americana Grolier*, que consistia na época

em 4,6 milhões de palavras organizadas em 30473 artigos. O *corpus* foi mapeado em um espaço de 300 dimensões e o resultado foi utilizado para resolver 80 questões de sinonímia do TOEFL ("Teste de Inglês como Língua Estrangeira", no original *"Test of English as a Foreign Language"*). Essas questões consistem em uma palavra modelo e quatro alternativas. O indivíduo sujeito ao teste deve escolher aquela cujo significado mais se aproxima do modelo. O resultado obtido usando LSA foi um índice de acertos de 64,4%, contra uma média de 64,5% obtida por seres humanos.

Outro estudo relevante é apresentado em (PAPADIMITRIOU et al., 2000). Nesse trabalho, os pesquisadores buscam os motivos para o desempenho do LSA na associação de termos semanticamente relacionados. Partindo de algumas premissas restritivas mas bastante razoáveis em situações normais de comunicação entre seres humanos, provam-se alguns teoremas que demonstram a eficácia do LSA. Apesar de essas premissas serem, indubitavelmente, fruto da intuição dos autores, o estudo dá uma boa percepção das situações nas quais se pode obter bons resultados aplicando LSA e, por conseguinte, daquelas nas quais o método pode se sair mal. A premissa que, intuitivamente, é mais importante é de que os documentos do *corpus* são separáveis, ou seja, que cada texto é pertinente a somente um tópico. A redução dimensional realizada com SVD faz com que relacionamentos contextuais fracos sejam ainda mais enfraquecidos e que relacionamentos fortes, mesmo que indiretos, sejam reforçados. Se esses relacionamentos contextuais não tiverem significado semântico bem definido, ou seja, se cada texto falar sobre muitos, ou, no limite, sobre nenhum tópico em particular, os relacionamentos entre palavras derivados pelo LSA não terão qualquer valor semântico.

É natural que seja difícil entender e, mais ainda, aprender com um um texto que fala sobre muitos assuntos. Seres humanos usam artifícios como a pontuação, paragrafação, divisão em seções e capítulos para estruturar e separar tópicos em textos mais longos. A abordagem de trabalhos anteriores, de ignorar divisões internas e considerar cada texto do *corpus* como um átomo, certamente não é a mais precisa, mas parece funcionar muito bem para textos curtos e orientados, tais como os artigos acadêmicos em (DEERWESTER et al., 1990), artigos de enciclopédia de (LANDAUER; DUMAIS, 1997) e definições da WordNet em (ESULI; SEBASTIANI, 2005).

Do ponto de vista prático, para este trabalho, a característica mais importante do LSA é a sua capacidade de mapear palavras para um espaço semântico onde os seus relacionamentos de significado são explicitados na sua codificação. É um método que requer ajustes empíricos – o número de dimensões ótimo para o espaço de saída precisa ser determinado por meio de testes – e cuja qualidade dos resultados só pode ser garantida para *corpora* de textos coesos e com escopo restrito. No entanto, a conveniência da representação dos textos e a sua proximidade das associações feitas

por seres humanos não deixam dúvida sobre o seu valor.

4 O estudo das Linguagens Naturais

A comunicação humana é motivo de estudos desde a Antiguidade. Na sua "Carta para Heródoto", o filósofo Epicuro da Grécia Antiga apresenta uma teoria sobre o surgimento das Línguas Naturais (Epicuro, apud FALK, 2003, p. 68):

> names ... were not at first deliberately given to things, but men's natures according to their different nationalities had their own peculiar feelings and received their peculiar impressions, and so each in their own way emitted air formed into shape by each of these feelings and impressions, according to the differences made in the different nations by the places of their abode as well.[1]

Essa abordagem ao estudo da linguagem como uma ciência natural parece só ter sido retomada na Idade Contemporânea. Uma frase famosa de Santo Isidoro de Sevilha, que viveu nos séculos IV e V, define gramática como "a arte da expressão correta, a primeira das Artes Liberais e o fundamento delas todas" e, de fato, do início da Idade Média até o século XIX, o conceito de gramática estava essencialmente ligado a compêndios de regras ditando a correção de construções (ASHER, 2005). A busca pela compreensão dos fenômenos linguísticos, em distinção à mera observação e categorização, só foi retomada no século XIX com Ferdinand de Saussure.

O salto no tempo não foi casual. Em verdade, durante séculos, a busca de conhecimento foi considerada proibida. A esse respeito, o historiador italiano Carlo Ginzburg conta uma história interessante. Segundo ele, a Epístola aos Romanos 11.20, um dos textos bíblicos escritos pelo apóstolo Paulo, exortava os romanos convertidos ao cristianismo a não desprezar os hebreus. O texto, em grego, foi traduzido de forma muito literal para o latim, o que resultou num mal entendido de graves consequências (GINZBURG; CAROTTI, 1990, p. 95–98).

> ... 'sapere' foi entendido não como um verbo de significado moral ('sê sábio'), mas como um verbo de significado intelectual ('conhecer');

[1] Vertido para o português: "nomes ... não foram a princípio dados às coisas deliberadamente, mas tendo naturezas distintas de acordo com as suas diferentes nacionalidades, os homens tiveram sensações e impressões peculiares, de maneira que cada um à sua maneira emitiu no ar formas geradas por essas sensações e impressões, adequadas às suas diferenças de nação e local onde viviam".

> a expressão adverbial ‘altum’, por outro lado, foi entendida como um substantivo que designa ‘aquilo que está no alto’. [...] Assim, a condenação da soberba moral pronunciada por São Paulo tornou-se uma censura contra a curiosidade intelectual. [...] Encontramo-nos, portanto, frente a um lapso não individual, mas coletivo ou quase coletivo. O deslize das palavras [...] foi certamente favorecido por fatores de ordem linguística e textual [...], a mente humana é comparável a um computador que opera na base de uma lógica de tipo sim-não, tudo-nada. Mesmo que a física moderna já seja suficientemente imune ao antropomorfismo para não se vincular a esse tipo de lógica, os seres humanos continuam a se comportar e a pensar da maneira mencionada. Para eles, a realidade enquanto refletida pela linguagem e, consequentemente, pelo pensamento, não é um continuum, mas um âmbito regulado por categorias descontínuas, substancialmente antitéticas.

Passado o período de obscurantismo e acentuando ainda mais a importância de considerar o falante e o ouvinte para a compreensão da linguagem como forma de comunicação e transmissão de conhecimento, merece destaque a lição do escritor, sociólogo, crítico literário, semiólogo e filósofo francês Roland Barthes, datada de 1978, a respeito do papel e da importância de Saussure para o estudo da linguística (BARTHES, 2003, p. 90–92):

> Talvez alguém ainda se lembre (pois está bem fora de moda): Saussure formulou com clareza a oposição langue-parole: dialética clara e sutil entre o sujeito falante e a massa falante. A partir de então, Saussure, se não foi atacado, foi pelo menos ‘esvaziado’ por diferentes vagas de pesquisa: Chomsky (competência-desempenho), Derrida, Lacan (la-langue). Acredito, pessoalmente, que, nessa oposição, alguma coisa é inabalável: necessidade de dois lugares, dois espaços em relação dialética: 1) uma reserva, onde são guardadas as leis da linguagem de uma comunidade (espécie de tabernáculo); 2) um momento de atualização, escolha do sujeito, recolhas na reserva [...] regras ‘mundanas’ (lógica, conveniência, dialética sob a escuta do outro, jogo de imagens, etc.).

4.1 O Estruturalismo na Linguística

Ferdinand de Saussure, acadêmico suíço, viveu entre o final do século XVIII e o começo do século XIX e é considerado, de forma quase unânime, o pai do estudo da linguística como ela é conhecida atualmente (FALK, 2003). Muitas de suas proposições não foram superadas pela comunidade científica e algumas delas são particularmente importantes para este trabalho.

Saussure defendia que a relação entre significante e significado é arbitrária, como

se pode verificar em um de seus escritos (SAUSSURE, 1983 apud ASHER, 2005, Vol. X p. 758):

> There is no internal connexion, for example, between the idea 'sister' and the French sequence of sounds s-ö-r which acts as its signifiant. The same idea might as well be represented by any other sequence of sounds. This is demonstrated by differences between languages, and even by the existence of different languages. The signification 'ox' has as its signal b-ö-f on one side of the border [between French and German-speaking regions] but o-k-s (Ochs) on the other side.[2]

O fato de existirem palavras compostas e onomatopeias não é ignorado pelo autor, mas a proposição de que o relacionamento entre ideias e suas representações nas Linguagens Naturais é essencialmente arbitrário tem implicações importantes. Denota, por exemplo, que a única maneira de conhecer o significado de todas as palavras de uma língua é enumerar a totalidade das duplas significante-significado pertencentes a ela. Significa, também, que o aprendizado de todas essas duplas em uma língua não tem qualquer utilidade para o aprendizado de outras línguas.

O entendimento de Saussure acerca do estudo da linguagem abriu espaço para pesquisas sobre a estrutura das línguas em si, em lugar de estudos comparativos entre línguas ou do estabelecimento de leis gramaticais para a escrita correta. Em particular, a Teoria Gerativista proposta por Chomsky estabeleceu ideias importantes para o desenrolar deste projeto.

4.2 A Teoria Gerativista

Noam Chomsky estabeleceu os fundamentos da teoria da sintaxe conhecida como Gramática Gerativa, ainda em gestação (CHOMSKY, 1965). A intenção é responder três questões consideradas fundamentais:

a. O que constitui o conhecimento da linguagem?

b. Como o conhecimento da linguagem é adquirido?

c. Como o conhecimento da linguagem é usado?

O autor argumenta que essas perguntas não são sequer abordadas pela Gramática Tradicional, preocupada essencialmente em listar os usos considerados formalmente

[2] Vertido para o português: "... não há conexão interna, por exemplo, entre a ideia 'irmã' e a sequência de sinais 's-ö-r' usada no francês como seu significante. A mesma ideia pode ser representada sem perda por qualquer outra sequência de sons. Isso é demonstrado pelas diferenças entre as línguas, e até pela existência de diferentes línguas. O significado de 'boi' tem como sinal 'b-ö-f' de um lado da fronteira [entre as regiões onde se fala francês e alemão], mas 'o-k-s' (Ochs) do outro lado".

corretos da linguagem. O conhecimento da linguagem é aceito como pré-existente e completo e os compêndios gramaticais visam a restringir o uso daquilo que é sabido às formas consideradas corretas (CHOMSKY, 1986).

Ao responder as questões acima enumeradas, a Teoria Gerativista propõe que o conhecimento da linguagem é justamente o conhecimento da Gramática Gerativa – o conjunto de regras que gera todas as construções válidas de uma língua – e que esse conhecimento é adquirido por meio de um aparato inato a todos os seres humanos, um conjunto de regras que foi chamado de Gramática Universal.

A terceira questão, sobre como é usado o conhecimento da linguagem, fica inicialmente por conta do Modelo-T (figura 4.1) proposto por Chomsky.

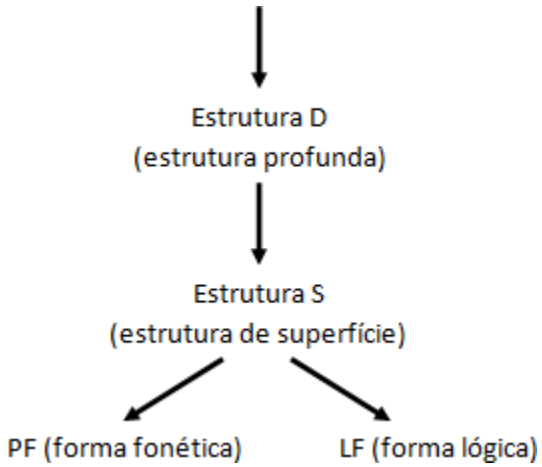

Figura 4.1: Modelo-T da Linguagem (CHOMSKY, 1986).

Conforme o Modelo-T, o processo de uso da linguagem começa na Estrutura-D, organização abstrata das ideias que se desejam comunicar. Por meio de regras transformacionais, essas estruturas podem ser mapeadas em Estruturas-S, mais próximas daquelas utilizadas pelos seres humanos no processo de comunicação. A partir daí, essas estruturas podem ser convertidas na forma fonética e na forma lógica do que se quer comunicar.

Na década de 1990, Chomsky deu início ao Programa Minimalista, em que vários desses conceitos, em especial o Modelo-T, foram abandonados e vários outros foram e ainda estão sendo revistos. Independentemente de os modelos concretos gerados pelo Programa Gerativista serem capazes de responder as perguntas que lhe deram o primeiro impulso, sua motivação e as hipóteses colocadas para explicar os problemas em estudo estão entre as linhas mais promissoras para a explicação do fenômeno da linguagem humana.

Considerando o estado atual de desenvolvimento contínuo da teoria, não será viável aplicar, neste trabalho, os modelos de gramática já produzidos pelo Programa Gerativista.

No entanto, a hipótese da Gramática Universal é uma solução bastante convincente para o "Problema de Platão" e está em linha com teorias sobre o aprendizado desenvolvidas em outras áreas do conhecimento (seção 2.1). A teoria da modularidade do cérebro humano e do nativismo de certos comportamentos do homem – uma das maiores barreiras para a aceitação da Gramática Universal – tem sido aceita por pesquisadores (FODOR, 1983).

As teorias e argumentos apresentados nesta seção fundamentam a escolha, neste trabalho, de assumir que existe uma Gramática Universal codificada na mente humana e que a compreensão de línguas naturais exige, pelo menos, a emulação do dispositivo que a processa.

4.3 A linguagem no cérebro humano

Há, como em todos os setores do conhecimento, críticos da concepção cerebralista da mente. Para eles, o cérebro é um instanciador das regras da linguagem, um mecanismo responsável simplesmente pelo processamento físico de algo que não é criado ali. Esse, entretanto, não tem sido o entendimento da maioria dos estudiosos (NERO, 2002). Del Nero, médico psiquiatra, mestre em filosofia e doutor em engenharia eletrônica pela Universidade de São Paulo, afirma (NERO, 2002):

> A linguagem é um dos grandes artífices da mente e da cultura. Inicialmente departamento concreto exclusivo do cérebro humano, torna-se virtual pela exposição ao meio. Se carrega consigo a capacidade potencial de reconhecer a natureza proposicional de uma sequência de símbolos, posteriormente deverá equipar-se para manipular regras superficiais da gramática, significados e discursos.

O conhecimento que se tem sobre o processamento da linguagem no cérebro humano foi aprendido, como é comum na pesquisa neurológica do Homem, a partir de testes cognitivos realizados em indivíduos que sofreram lesões cerebrais. Há diversos estudos detalhados de casos desse tipo (APHASIA, 2008; CARAMAZZA; ZURIF, 1976).

Um, em especial, foi marcante no século XIX por ter evidenciado, de forma pioneira, a ligação entre uma lesão cerebral específica e uma limitação da racionalidade. O caso Phineas Gage, ocorrido em 1848, foi fartamente registrado. Tratava-se de um trabalhador da construção civil de 25 anos de idade que, à época, assentava trilhos para uma empresa ferroviária nos Estados Unidos. Um rapaz forte, saudável, preciso em seus movimentos, além de sociável e simpático. Isso até que a explosão de uma rocha o atingisse. Uma barra de ferro de um metro de comprimento e três centímetros de

diâmetro, pesando seis quilos, atravessou seu crânio e ali permaneceu. Dois meses depois, Gage deixou o hospital ainda com a barra de ferro na cabeça, mas fisicamente recuperado e aparentemente sem problemas de fala e compreensão. No entanto, tornou-se irreverente e passou a insultar as pessoas. A história, rica em detalhes, é contada por António Damásio, neurologista português radicado nos Estados Unidos. Ele explica (DAMÁSIO, 1999, p. 30–31):

> Enquanto outros casos de lesões neurológicas ocorridas na mesma época revelaram que o cérebro era o alicerce da linguagem, da percepção e das funções motoras [...], a história de Gage sugeriu este fato espantoso: em certo sentido, existiam sistemas no cérebro humano mais dedicados ao raciocínio do que a quaisquer outros [...] As alterações na personalidade de Gage não foram sutis. Ele já não conseguia fazer escolhas acertadas...

Esta seção propõe-se a descrever o essencial sobre o assunto para esclarecer a fundamentação do trabalho proposto.

De acordo com o que se verificou até o momento em que este texto é redigido (uma vez que as pesquisas nessa área têm evoluído rapidamente), o reconhecimento da linguagem pelo cérebro humano se dá essencialmente em duas regiões do cérebro: a Área de Broca, localizada no lóbulo frontal do córtex, e a Área de Wernicke, que fica no cerebelo, em torno do córtex auditivo (APHASIA, 2008). Essas duas áreas se comunicam por um feixe neuronal denominado Fascículo Arcuato. Lesões em cada uma dessas regiões resultam em três tipos de afasia: a Afasia de Broca, a Afasia de Wernicke e a Afasia de Condução, respectivamente (figura 4.2).

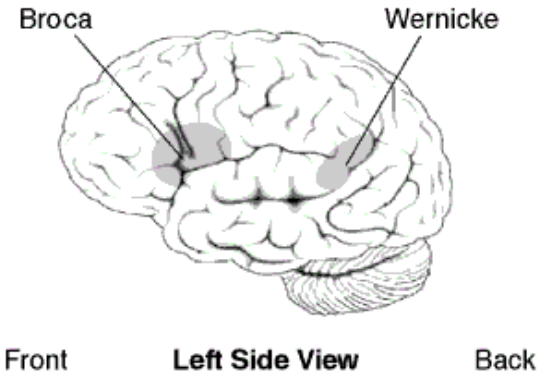

Figura 4.2: Regiões de Wernicke e de Broca (APHASIA, 2008).

A Afasia de Broca e a Afasia de Condução são bastante similares, afetando principalmente a capacidade de organizar e compreender a estrutura de orações. Pessoas que sofrem desses problemas normalmente se comunicam usando sentenças muito simples e concisas, desprovidas de artigos e de inflexões ou conjugações verbais (CARAMAZZA;

ZURIF, 1976). Testes mais minuciosos indicam que aparentemente esses indivíduos se baseiam nos possíveis conteúdos semânticos de cada sentença para compreender o que lhes está sendo dito e também que têm grande dificuldade em utilizar-se de informações sintáticas para fazer escolhas quando defrontados com ambiguidades.

A Afasia de Wernicke é um distúrbio mais incapacitante: suas vítimas normalmente têm grande dificuldade em se comunicar. Constroem discursos estruturalmente complexos, mas desprovidos de significado. Entre esses indivíduos, é comum a utilização de orações com sujeito indefinido em momentos onde este não está implícito, a troca de substantivos e o uso de anáfora de maneira incompreensível. Aos estudiosos do problema, parece bastante claro que a capacidade de construção sintática e de compreensão do papel semântico de cada palavra não é afetada, mas que o entendimento dos significados, este sim, é prejudicado, muitas vezes impossibilitando a comunicação. Pacientes recuperados da lesão e de seus efeitos relatam que, apesar de falarem, não compreendiam nada do que outros ou eles próprios diziam.

Há uma característica curiosa da Afasia de Wernicke que merece nota: ela não afeta a capacidade de cantar músicas conhecidas antes da ocorrência da lesão. Entretanto, de maneira geral, refuta-se que isso esteja associado à compreensão (HEBERT et al., 2003).

As diferenças entre os distúrbios acima descritos levam estudiosos e autores à conclusão de que o processamento da linguagem no cérebro ocorre a partir de duas abordagens complementares. Especula-se, até pela tendência de organização topológica do cérebro, que a compreensão semântica das palavras e de seus papéis se dá na Área de Wernicke e a estruturação sintática, na Área de Broca. O Fascículo Arcuato seria o canal de ligação entre as duas áreas (HEBERT et al., 2003).

Os entendimentos acima descritos corroboram com a hipótese de que a mente humana é modular e de que possui componentes inatos, em concordância com as proposições de Chomsky e Fodor dispostas na seção 4.2. A proposta, neste projeto, é construir um dispositivo estruturado para realizar a análise dos significados das palavras em um estágio separado da análise das estruturas das sentenças, de maneira análoga a um indivíduo com Afasia de Broca. Se privado de seu interpretador sintático, esse dispositivo reconheceria apenas o sentido geral de agrupamentos de palavras. Privado do módulo que contém as associações semânticas das palavras, ficaria completamente incapacitado.

Parte II

Experimentos

5 Implementação do Preditor de Solomonoff restrito a Linguagens Regulares

Uma das contribuições mais relevantes deste trabalho foi a implementação de uma versão fiel, ainda que restrita, do Preditor de Solomonoff. Não foi localizado qualquer outro exemplar de concretização desse dispositivo proposto 46 anos atrás em (SOLOMONOFF, 1964), apesar do número de trabalhos publicados que o referenciam (LI; VITÁNYI, 1989; LI; VITÁNYI, 1997; HUTTER, 2005; RISSANEN, 1983; MINSKY, 1967; CHAITIN, 1997) – existe uma tentativa informal, chamada *Solomonoff-lite Evaluator* e exposta somente em uma página na Internet (HAY, 2007). No entanto, uma rápida inspeção no código-fonte dessa implementação mostra que os seus métodos de geração de teorias não são consistentes com a proposta de Solomonoff.

Nesta seção, é descrita a implementação feita para este trabalho do Preditor de Solomonoff, restrita à indução de teorias com o poder computacional de Gramáticas Regulares. Posteriormente, são mostrados alguns resultados de testes dessa implementação e medidas de tempo de processamento para cadeias de entrada de diferentes comprimentos: um indicador empírico da complexidade computacional do programa.

É importante manter em mente que o Preditor de Solomonoff, conforme proposto em (SOLOMONOFF, 1964), é completo, mas incomputável (SOLOMONOFF, 1975). Ele opera, originalmente, sobre linguagens recursivamente enumeráveis. A redução do modelo para computar hipóteses somente no universo das Linguagens Regulares o torna computável, mesmo que com complexidade computacional assintótica $\Omega(e^n)$. Solomonoff demonstra (SOLOMONOFF, 1986) que são duas características mutuamente excludentes: ser completo e computável. O modelo implementado neste trabalho é, portanto, necessariamente incompleto. Isso não significa que, para um grande conjunto de problemas comuns na prática, ele não possa produzir resultados úteis. O seu interesse como preditor universal, no entanto, é intencionalmente comprometido em prol da sua computabilidade.

5.1 Descrição do programa

Conforme foi exposto na seção 2.1, a Teoria da Predição de Solomonoff dá uma forma de calcular a probabilidade de cada algoritmo capaz de explicar um conjunto de eventos e, dessa forma, delineia um dispositivo preditor. Na sua forma mais simples, esse dispositivo vasculha o espaço de algoritmos cegamente, verificando, para cada elemento, a sua probabilidade *a priori* e o espaço de saídas válidas que ele produz. A probabilidade *a priori* de cada hipótese é dada pela equação (2.4), reproduzida a seguir:

$$P(H) = 2^{-|H|} \tag{2.4}$$

A probabilidade de uma hipótese dado um evento é dada pela Teoria de Bayes, representada pela equação (2.1) e reproduzida abaixo:

$$P(H|E) = \frac{P(E|H) \cdot P(H)}{P(E)} \tag{2.1}$$

O dispositivo derivado da Teoria da Predição de Solomonoff conjuga essas duas equações ao espaço de saídas produzidas por cada hipótese (o universo previsto por ela), obtendo $P(H|E)$: a probabilidade da hipótese dados os eventos observados.

O programa que foi desenvolvido para este trabalho implementa a forma mais simples do dispositivo preditor exposto por Solomonoff. O espaço de algoritmos vasculhado é restrito àquele dos códigos processáveis por uma máquina com poder computacional equivalente ao das gramáticas regulares, reduzindo a qualidade das teorias obtidas, mas simplificando imensamente o processamento.

A escolha do espaço das Linguagens Regulares como restrição foi baseada na simplicidade dessa classe de linguagens formais em relação aos níveis superiores da Hierarquia de Chomsky. Há uma série de qualidades decorrentes dessa simplicidade. As mais importantes são a existência de algoritmos de baixo custo para a obtenção do Autômato de Estados Finitos Determinístico (AFD) ótimo que produz uma Linguagem Regular, a unicidade desse dispositivo ótimo e a complexidade computacional linear do seu processamento para a aceitação de uma cadeia (HOPCROFT; MOTWANI; ULLMAN, 2000). Todas essas características foram exploradas para reduzir a complexidade computacional e de implementação do programa. A estrutura geral do dispositivo implementado pode ser vista no algoritmo 5.1.

Algoritmo 5.1: Preditor de Solomonoff – Estrutura geral

```
1 for h in hipoteses
```

```
2   afd = cria_automato_finito_deterministico(h)
3   P(E|h) = afd.probabilidade(E)
4   for evento in E
5     if P(evento|h) = 0
6       remove_hipotese(h)
7       P(h|E) = 0
8       break
9     else
10      P(h|E) += P(evento|h)*P(h)/P(evento)
```

Mostra-se em algoritmo 5.2 uma descrição detalhada do núcleo do preditor implementado. Ele toma a forma de uma classe, *Preditor*, que tem como atributos o alfabeto de símbolos terminais com o qual se deseja trabalhar, o número de ocorrências de cada evento observado, as probabilidades de todas as teorias já formuladas pelo dispositivo e o comprimento da mais longa entre elas. Toda a complexidade dessa classe fica no método *observa*, que é responsável por alterar o conjunto interno de teorias do *Preditor* conforme novos eventos são apresentados.

Cada chamada ao método *observa* de uma instância da classe *Preditor* altera o estado interno do objeto. As probabilidades das teorias compatíveis com os eventos observados são recalculadas para considerar o novo evento. Caso seja possível obter uma teoria mais longa do que as já exploradas que seja melhor do que as já conhecidas, o método *observa* continua a sua exploração do espaço das expressões regulares. No momento em que o comprimento das teorias só vai reduzir a sua probabilidade, ou em que se fosse exceder o limite imposto na chamada do método, a busca por hipóteses é interrompida.

Algoritmo 5.2: Preditor de Solomonoff - Método *observa*

```
1  Preditor::observa(novo_evento, comprimento_limite_hipotese)
2    for teoria in this.probabilidades_teorias
3      # Criar o autômato de estados finitos determinístico (AFD) mínimo que
4      # reconhece o código sendo processado
5      afd = automato_finito_deterministico_minimo(this.alfabeto, teoria)
6      saidas_validas = 0
7      total_saidas = 0
8
9      # Exercitar o AFD - verificar quantas cadeias ele reconhece até o
10     # comprimento da maior cadeia de entrada (total_saidas) e, dentre elas,
11     # quantas são equivalentes a cada uma das cadeias de entrada (saidas_validas)
12     afd.obtem_contagem_cadeias(novo_evento, saidas_validas, total_saidas)
13
14     if saidas_validas == 0
15       probabilidades_teorias.remove(teoria)
16     else
17       probabilidade_teoria = (1/tamanho(alfabeto))^tamanho(teoria)
18       this.probabilidades_teorias[teoria] =
             ((this.probabilidades_teorias[teoria] *
             this.contagem_eventos_conhecidos) + (probabilidade_teoria *
```

```
            saidas_validas / total_saidas)) / (this.contagem_eventos_conhecidos+1)
19
20        if this.probabilidades_teorias[teoria] > probabilidade_melhor_teoria
21          probabilidade_melhor_teoria = this.probabilidades_teorias[teoria]
22          melhor_teoria = teoria
23
24    if this.ocorrencias_eventos[novo_evento] > 0
25      this.ocorrencias_eventos[novo_evento]++
26      comprimento_maximo = this.comprimento_maior_teoria
27    else
28      this.ocorrencias_eventos[novo_evento] = 1
29      if this.contagem_eventos_conhecidos == 0
30        comprimento_maximo = tamanho(novo_evento)
31      else
32        comprimento_maximo =
              this.obtem_comprimento_maximo(this.comprimento_maior_teoria+1,
              this.comprimento_maior_teoria + tamanho(novo_evento) + 1,
              probabilidade_melhor_teoria)
33      comprimento_maximo = min(comprimento_maximo, comprimento_limite_hipotese)
34
35    this.contagem_eventos_conhecidos++;
36    gerador = gerador_expressoes_regulares(this.alfabeto,
          this.comprimento_maior_teoria, comprimento_maximo)
37    while gerador.proxima_valida(teoria, chaves(this.probabilidades_teorias)) &&
          tamanho(teoria)<=comprimento_maximo
38      probabilidade_teoria = (1/tamanho(this.alfabeto))^tamanho(teoria)
39      afd = automato_finito_deterministico_minimo(this.alfabeto, teoria)
40
41      saidas_validas = []
42      total_saidas = 0
43      afd.obtem_contagem_cadeias(chaves(this.ocorrencias_eventos), saidas_validas,
            total_saidas)
44      probabilidade_hipotese_dados_eventos = 0.0
45      for evento in this.ocorrencias_eventos
46        if saidas_validas[evento] == 0
47          probabilidade_hipotese_dados_eventos = 0.0
48          break
49        else
50          probabilidade_hipotese_dados_eventos += (probabilidade_teoria *
                saidas_validas[evento] * this.ocorrencias_eventos[evento]) /
                (total_saidas * this.contagem_eventos_conhecidos)
51      if probabilidade_hipotese_dados_eventos > 0.0
52        this.probabilidades_teorias[teoria] = probabilidade_hipotese_dados_eventos
53
54        if tamanho(teoria) > this.comprimento_maior_teoria
55          this.comprimento_maior_teoria = tamanho(teoria)
56
57        # se encontramos uma teoria melhor que todas as anteriores, é possível que
58        # possamos estabelecer um novo limite para o comprimento das hipóteses
59        if probabilidade_hipotese_dados_eventos > probabilidade_melhor_teoria
60          melhor_teoria = teoria
61          probabilidade_melhor_teoria = probabilidade_hipotese_dados_eventos
62          comprimento_maximo = this.obtem_comprimento_maximo(tamanho(teoria),
                comprimento_maximo, probabilidade_melhor_teoria)
```

O processamento do método *observa* começa na linha 2, onde há um laço responsá-

vel por recalcular as probabilidades de todas as teorias já computadas considerando o novo evento observado. É criado o Autômato de Estados Finitos Determinístico mínimo equivalente à expressão regular que representa a teoria na linha 5. Esse autômato é exercitado na linha 12, informando o número de saídas total computadas pelo autômato e quantas delas têm o evento observado como prefixo. Se a teoria não é capaz de explicar o novo evento (nenhuma das saídas o tem como prefixo), ela é descartada. Senão, a probabilidade da teoria é recalculada para levar em consideração o novo evento usando equação (2.1), na linha 18. Entre as linhas 24 e 33 é estabelecido um limite superior para o comprimento das hipóteses que serão geradas para teste, de uma maneira que será descrita posteriormente. Esse limite é usado na exploração do espaço de hipóteses ainda não visitado. No laço da linha 37, são geradas novas hipóteses aplicando o fecho de Kleene aos símbolos da linguagem. A seguir, calcula-se a probabilidade da hipótese dados todos os eventos conhecidos de maneira análoga ao que é feito no laço da linha 2 e, caso a hipótese seja capaz de descrever todos os eventos observados, ela é adicionada ao rol de teorias válidas juntamente com a probabilidade $P(H|E)$ obtida (linha 52). Caso a nova hipótese seja a melhor já encontrada, o limite de comprimento das hipóteses a serem visitadas é atualizado.

O limite inicial estabelecido para o comprimento das hipóteses a serem investigadas é calculado entre as linhas 24 e 33 do algoritmo 5.2. Caso o evento já tenha sido observado, por causa da maneira como o algoritmo foi construído, todas as hipóteses com chance de ser a melhor possível já terão sido visitadas. Dessa maneira, limita-se o espaço a ser investigado ao comprimento da maior hipótese já testada, efetivamente interrompendo a exploração de uma área maior do espaço de hipóteses. Se o evento é novo, há duas possibilidades: ou o novo evento é o primeiro a ser observado ou não. Se o evento for o primeiro observado pelo preditor, o maior comprimento de descrição útil consiste no próprio evento: é a hipótese trivial que invariavelmente maximizará o termo $P(E|H)$. Qualquer hipótese mais longa terá probabilidade *a posteriori* menor. Caso o evento não seja o primeiro, é invocado o método *obtem_comprimento_maximo* (algoritmo 5.3), estabelecendo como limite inferior o comprimento da maior teoria já explorada acrescido de um (afinal, deseja-se apenas hipóteses que ainda não foram exploradas) e como limite superior o comprimento da cadeia $melhor\_hipotese\_computada|novo\_evento$, que cumpre o papel de maximizar $P(E|H)$. O algoritmo 5.3 tem uma lógica interna bastante simples: para cada comprimento entre o limite inferior e o limite superior fornecidos, ele verifica se é possível que haja uma hipótese cuja probabilidade *a posteriori* supere a da melhor teoria já encontrada. Caso seja encontrado um limite para a melhoria das hipóteses inferior ao limite superior informado na chamada do método, ele é retornado. Senão, retorna-se o próprio parâmetro de limite superior.

Solomonoff propôs em (SOLOMONOFF, 1989) que o preditor fosse implementado para ser incremental e aceitar interferência externa quanto ao seu tempo de execução, de maneira que fosse possível obter a melhor resposta encontrada até um determinado instante – algo muito interessante, dado que é impossível prever o tempo necessário para obter a melhor resposta dentre todas. Essa proposta foi incorporada no algoritmo 5.2 por meio do argumento *comprimento_limite_hipotese* passado ao método *observa*. A interrupção ao processamento do preditor não pode, dessa maneira, ser feita arbitrariamente, mas ainda assim há uma via direta para interferência no tempo de execução do sistema. No caso comum, a maioria do tempo de execução do programa fica por conta da geração e verificação de novas hipóteses contra todos os eventos já observados. Interrompendo a criação de novas hipóteses, o tempo necessário para a inclusão de novos eventos tende a crescer muito mais lentamente.

Algoritmo 5.3: Preditor de Solomonoff - Método *obtem_comprimento_maximo*

```
1 Preditor::obtem_comprimento_maximo(limite_inferior, limite_superior,
      probabilidade_melhor_teoria)
2   for teoria in this.probabilidades_teorias
3     if limite_inferior >= limite_superior
4       break
5     probabilidade_maxima = (1/tamanho(this.alfabeto))^limite_inferior
6     if probabilidade_maxima < probabilidade_melhor_teoria
7       return limite_inferior-1
8   return limite_superior
```

O comportamento do algoritmo 5.3 explica por que, ao final do laço da linha 37 do algoritmo 5.2, o método *obtem_comprimento_maximo* é invocado para atualizar o limite de comprimento das hipóteses caso tenha sido encontrada uma teoria que supere todas as anteriores. É possível que a probabilidade da nova hipótese supere o limite do que se pode encontrar no espaço que se planeja explorar. Isso seria extremamente custoso, justificando todas as tentativas de redução do espaço de busca.

Há três pontos críticos no algoritmo 5.2: a geração de teorias, na linha 37; a criação de um dispositivo capaz de executar a teoria, na linha 5; e o exercício de cada teoria, na linha 12. Esses três pontos críticos são o principal motivo para a complexidade computacional e a dificuldade de implementação de um preditor como o proposto por Solomonoff. São também os motivadores principais para a escolha de Linguagens Regulares para a implementação apresentada.

A geração de teorias pode ser feita, a princípio, como um simples fecho de Kleene sobre a linguagem de descrição dessas teorias. A implementação é aparentemente simples, mas o resultado é infinito. É necessário, para um sistema factível, restringir o universo de teorias a ser vasculhado. Na implementação apresentada, essa restrição foi

feita restringindo o comprimento máximo das teorias, conforme a descrição acima do funcionamento do algoritmo.

O segundo e terceiro pontos críticos do algoritmo 5.2 citados acima – a criação e o exercício de um dispositivo capaz de executar cada teoria que se tem – dificultam a execução do sistema por motivos similares. É necessário assegurar, antes de mais nada, que o dispositivo que será usado terminará o seu processamento em um período de tempo finito. Não é praticável iniciar um processamento que pode recair no Problema da Parada da Máquina de Turing (LI; VITÁNYI, 1997). Há artifícios que poderiam ser usados para lidar com essa questão, como o revezamento do processamento de todas as teorias que se tem (SOLOMONOFF, 1989), mas ainda assim o custo de processamento tende ao impraticável. O Autômato de Estados Finitos determinístico não sofre desses problemas. Ele é capaz de reconhecer qualquer cadeia que pertença à linguagem que define em tempo linearmente proporcional ao comprimento dessa cadeia de entrada (HOPCROFT; MOTWANI; ULLMAN, 2000). Além disso, a obtenção desse dispositivo, apesar de computacionalmente custosa (a complexidade computacional de converter um AF não-determinístico, que é a tradução natural de uma expressão regular, em um AF determinístico, é de ordem exponencial (HOPCROFT, 1971)), é facilmente implementada.

5.2 Resultados

A implementação feita neste trabalho do Preditor de Solomonoff restrito a Linguagens Regulares foi validada em uma série de testes bastante simples, mas que devem comprovar a sua eficácia. É possível verificar a divisibilidade de números inteiros usando Autômatos de Estados Finitos. O conjunto dos números inteiros divisíveis por dois, por exemplo, forma uma Linguagem Regular. O que se fez para testar o programa criado foi alimentá-lo com sequências de números, escritos na sua forma binária, divisíveis por um certo inteiro. A intenção é verificar a velocidade de convergência para a expressão regular correta, tanto em relação ao número de eventos quanto ao tempo real de execução.

5.2.1 Tempo de processamento

O tempo de processamento do programa depende, principalmente, do número de verificações que precisam ser feitas para validar as hipóteses contra os eventos observados. A complexidade computacional assintótica da transformação de cada expressão regular no Autômato Finito Determinístico (AFD) mínimo equivalente é $O(e^n)$, mas, na prática,

o seu comportamento assintótico não é o fator que mais contribui para o tempo de processamento. São geradas $|\Sigma|^n$ cadeias para cada comprimento n de hipótese e todas elas precisam ser transformadas em AFD mínimo para serem testadas todas as suas saídas possíveis. Assim, o que se observa é que o grande responsável pelo tempo de processamento, no caso comum em que há poucos eventos frente ao número de hipóteses possíveis, é a quantidade de hipóteses que precisam ser verificadas.

Para verificar a evolução do tempo de processamento, o programa foi executado com cadeias que forçassem a geração de hipóteses até o seu próprio comprimento. A intenção desses testes foi somente obter dados empíricos para embasar a escolha de testes exequíveis com os recursos disponíveis, de maneira que foram executados somente uma vez cada, com entradas escolhidas manualmente. O gráfico de tempo de execução de acordo com o comprimento da entrada (e, assim, do maior comprimento de hipótese explorado) é mostrado na figura 5.1.

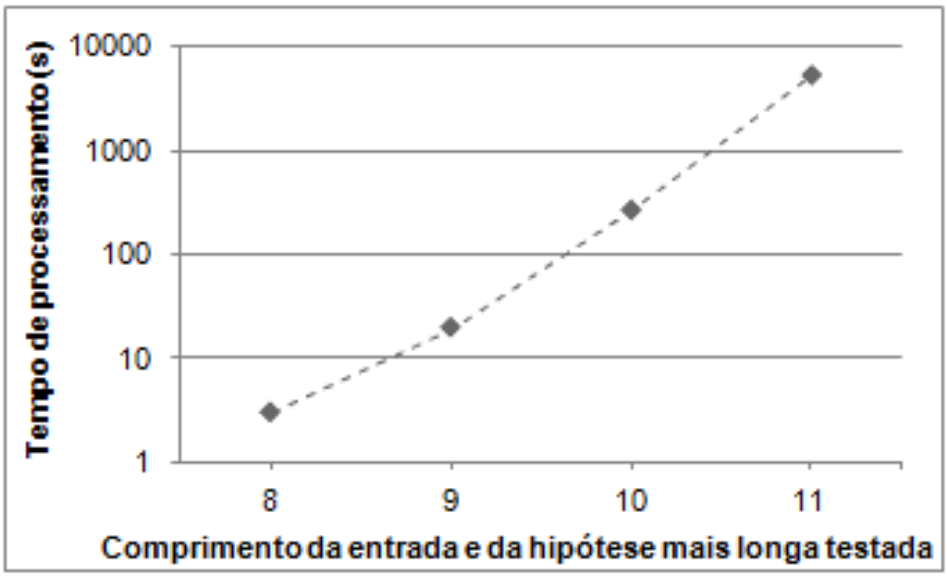

Figura 5.1: Gráfico do tempo de processamento do preditor contra o comprimento da maior hipótese gerada.

Foram omitidos dos resultados apresentados na figura 5.1 processamentos de comprimentos de cadeia que exigiram menos de 1 segundo para facilitar a visualização em escala logarítmica. A escala logarítmica, por sua vez, foi escolhida para ressaltar a similaridade entre a curva de progressão do tempo de processamento obtida empiricamente e o comportamento esperado para um algoritmo com complexidade computacional assintótica $O(e^n)$. A conclusão prática desses testes é que, dados os recursos disponíveis para este trabalho, não é adequado planejar testes funcionais que requeiram a geração de hipóteses com mais de 11 símbolos.

5.2.2 Testes com uma linguagem sintética

A complexidade computacional do Preditor de Solomonoff limita bastante o universo de testes praticáveis. Com isso em mente, decidiu-se verificar, inicialmente, o comportamento do protótipo construído desse dispositivo com uma linguagem sintética, sem interesse semântico. A proposta é verificar quantos exemplos da linguagem $(01)*$ são necessários para que o dispositivo chegue a essa expressão regular.

De início, foi apresentada ao preditor a cadeia "01", à qual o preditor respondeu somente com a hipótese "01". Então foi apresentada ao preditor a cadeia "0101", da qual resultou a resposta análoga "0101". Alimentando o dispositivo com as duas cadeias, a descrição da linguagem proposta já figurava como o primeiro entre os resultados, conforme mostra a tabela 5.1.

Hipótese	**Probabilidade**
$(01)*$	0.0078125
$\|(01)*$	0.00390625
$01\|0101$	0.00390625
$0101\|01$	0.00390625
$01(\|01)$	0.00390625
$01(01\|)$	0.00390625
$0(\|10)1$	0.00390625
$0(10\|)1$	0.00390625
$(\|01)01$	0.00390625
$(01\|)01$	0.00390625

Tabela 5.1: Dez principais hipóteses propostas pelo preditor para a sequência de cadeias "01 0101".

O resultado apresentado na tabela 5.1 exigiu um tempo de processamento de $0,43s$. É certamente um exemplo bastante simplificado que não pode ser estendido para todas as situações. No entanto, o desempenho do preditor para localizar uma hipótese capaz de descrever os eventos observados não pode ser ignorado. Mesmo que o gerador original dos eventos expostos ao preditor fosse substancialmente mais complexo, é inegável que a teoria considerada mais provável pelo dispositivo é coerente e elegante na sua simplicidade.

5.2.3 Teorias geradas para o teste de divisibilidade por dois

Verificado o funcionamento inicial do preditor implementado, decidiu-se testar uma linguagem definida por um problema simples da matemática: a divisibilidade, dentro do conjunto dos Números Naturais, por dois. Mantendo o alfabeto utilizado nas seções anteriores deste capítulo, os números serão representados na forma binária.

Conforme exposto na seção 2.1, é esperado que o Preditor de Solomonoff obtenha o algoritmo gerador de uma sequência de eventos observando um número relativamente pequeno deles. O teste proposto consiste, portanto, em alimentar o preditor com uma sequência de números binários divisíveis por dois. A expressão regular mais curta que define essa Linguagem Regular é "$(1|0)*0$" – qualquer cadeia composta de zeros e uns que termine com um zero.

Utilizou-se como entrada para o preditor os números no intervalo $[0,126]$ e estabeleceu-se o limite de comprimento para a busca de hipóteses em 7 – o comprimento da expressão regular que reconhece os eventos. Cada valor foi repetido com todos os comprimentos de cadeia até 7 símbolos (comprimento do número 126 em binário) e cada repetição foi completada com um prefixo de zeros, não-significativos. Assim foi incluída na amostra a informação de que zeros à esquerda não são significativos. As cadeias foram alimentadas ao programa em grupos de tamanho crescente em ordem lexicográfica. Como resultado, foi enviado ao programa o conjunto de eventos $\{0\}$, depois $\{0,00\}$, $\{0,00,10\}$ e assim por diante. Um gráfico da evolução da posição da hipótese geradora dos dados "$(1|0)*0$" é mostrado na figura 5.2.

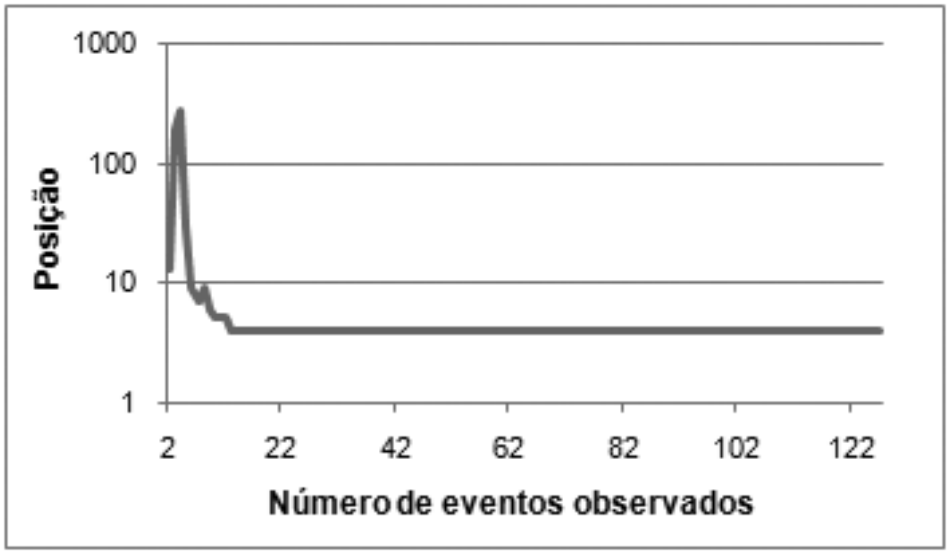

Figura 5.2: Gráfico da evolução da hipótese "$(1|0)*0$" do preditor com o aumento do número de eventos observados.

Observa-se um fato interessante do teste descrito: a hipótese que de fato gerou a sequência de eventos não chegou à primeira posição dentre as alternativas propostas pelo preditor, mesmo depois que este foi exposto a 126 eventos. É necessário ver a lista dos principais candidatos, mostrada na tabela 5.2, para compreender esse fenômeno.

A diferença entre a linguagem aceita pela expressão regular "$(1*0)*$" e aquela aceita pela expressão "$(1|0)*0$" (e a sua equivalente "$(0|1)*0$") é a cadeia vazia. É notável que uma hipótese tão próxima da correta domine as propostas do preditor a

Hipótese	Probabilidade
$(1*0)*$	0.000143537
$(1\|0)*$	0.000111827
$(0\|1)*$	0.000111827
$(1\|0)*0$	$7.56032e-05$
$(0\|1)*0$	$7.56032e-05$
$\|(1*0)*$	$7.17684e-05$
$\|(1\|0)*$	$5.59133e-05$
$\|(0\|1)*$	$5.59133e-05$
$(\|1\|0)*0$	$3.78016e-05$
$(\|0\|1)*0$	$3.78016e-05$

Tabela 5.2: Dez principais hipóteses propostas pelo preditor para o problema da divisibilidade de números binários por dois, dadas 126 amostras.

partir do ponto em que foi alimentado com 12 eventos. No entanto, não é desejável que as hipóteses incorretas mais curtas dominem a lista por tanto tempo. É o efeito colateral negativo de escolher uma linguagem tão abrangente como a dos inteiros divisíveis por dois – metade dos números binários existentes é divisível por dois, de maneira que uma expressão regular que reconhece absolutamente todos os números binários, "$(1|0)*$", tem uma probabilidade de somente 50% de reconhecer uma cadeia que não é divisível por dois. O comprimento da hipótese, nesse caso, faz com que teorias mais genéricas sejam preferidas às corretas mesmo após a observação de um número relativamente grande de eventos. Tentou-se prosseguir a execução com até 254 amostras sem alteração da posição das hipóteses esperadas.

O resultado obtido é inesperado por evidenciar uma classe de problemas em que o desempenho do preditor fica aquém do esperado. No entanto, o preditor propôs hipóteses muito coerentes para os eventos observados. A hipótese preferida pelo preditor a partir da observação de 12 eventos é particularmente interessante por ser extremamente simples e descrever com grande precisão a Linguagem Regular proposta. Mesmo em um teste que expõe um caso de convergência lenta é difícil ignorar as qualidades do dispositivo.

6 Implementação de um Sistema de Buscas Baseado em Análise da Semântica Latente

Conforme descrito na seção 3.1, o LSA é um método para mapear um *corpus* em um espaço vetorial com número reduzido pré-definido de dimensões. Para poder criar essa representação, é necessário, em primeiro lugar, processar os documentos do *corpus*, armazenando dados sobre as palavras que ocorrem em cada texto e consolidando esses dados em uma matriz. A essa matriz deve então ser aplicado o método SVD, cujo resultado, truncado, fornece a aproximação desejada do espaço vetorial determinado pelo *corpus*. O resultado deve ser dado por uma série de matrizes que podem ser usadas no processamento de consultas.

Por causa do desacoplamento entre o processamento inicial do *corpus* e o tratamento de consultas, o desenvolvimento desse tipo de sistema é feito tradicionalmente (YATES; NETO, 1999) em dois módulos separados. O primeiro módulo, chamado *indexador*, é responsável pelo processamento do *corpus* em si e pela geração de um índice para buscas. O segundo módulo, o *processador de consultas*, recebe cadeias de termos e retorna uma listagem ordenada dos documentos do *corpus*, sem necessidade de acesso direto a este, somente ao índice gerado pelo *indexador*. Esses dois módulos, representados na figura 6.1, serão descritos individualmente nas seções seguintes.

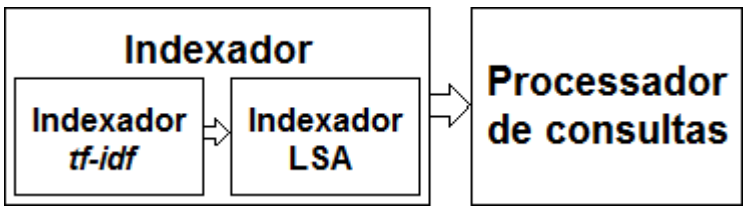

Figura 6.1: Sistema de buscas aplicando LSA (CHOMSKY, 1986).

Uma vez implementados o indexador e o processador de consultas, foi executada uma série de testes para verificar o desempenho da indexação usando LSA. Para isso, foi usado o *corpus* de textos jornalísticos em inglês da Reuters, o *Reuters Corpus Volume 1* (INFORMATION. . . , 1997), que contém textos publicados pela agência de notícias

entre agosto de 1996 e março de 1997. Descrições da metodologia e dos resultados dos testes executados encerram esta seção.

Será mantida uma cópia dos resultados consolidados dos testes – alguns milhares de gráficos – em um servidor acessível pelo endereço http://iuri.chaer.org/pesquisa/. Também serão mantidos um buscador e um comparador de termos (na medida do possível, devido ao consumo de recursos computacionais pela ferramenta) para que outros possam observar os resultados da indexação utilizando LSA. Os textos originais do *corpus* da Reuters, no entanto, não podem ser disponibilizados por motivo de direito autoral.

6.1 Descrição do indexador

O primeiro passo da indexação consiste simplesmente na geração da matriz esparsa relacionando termos e documentos, resultado normal de uma indexação com o método *tf-idf* (SALTON; BUCKLEY, 1988). A essa matriz pode ser aplicado o método de redução dimensional do LSA para produzir índices com qualquer número de dimensões. Por isso, foi criado um pré-indexador, responsável somente pela separação de termos, extração dos seus radicais e cálculo do *tf-idf* relacionando termos a documentos. Os dados processados são armazenados em arquivos estruturados substancialmente menores que o *corpus* original, reduzindo o tempo necessário para a geração de índices com diferentes números de dimensões.

A segunda fase do indexador consiste propriamente na aplicação do método LSA. O programa responsável por essa parte basicamente realiza a decomposição em valores singulares da matriz obtida pela primeira fase, trunca a matriz de valores singulares mantendo o número de dimensões desejadas e gera as matrizes de termos e documentos necessárias para que se possa processar consultas. Os resultados são armazenados em arquivos compactos, de maneira a facilitar a carga pelo processador de consultas.

6.1.1 Indexador *tf-idf*

Para reduzir o tempo consumido em operações de leitura do disco, o indexador *tf-idf* foi construído para acessar diretamente os arquivos comprimidos do *corpus* da Reuters, exatamente da maneira como eles são distribuídos. A indexação é feita em duas passagens. Na primeira, é feita uma contagem dos termos, tanto global quanto local, em cada documento. A partir dessas contagens, é possível alocar o espaço necessário para a matriz esparsa gerada pelo método *tf-idf*, assim como calcular os valores atribuídos a cada termo por cada documento. Como o *corpus* de teste contém um número relevante

de erros de ortografia, para o *corpus* de mais de 470 mil documentos foram removidos os termos com menos de 6 ocorrências – um heurística bastante simplista, mas que verificou-se, empiricamente, ter resultados muito bons em relação à complexidade da sua implementação. Como resultado do processamento desse módulo são gerados três arquivos:

Léxico : Índice de termos que ocorrem no *corpus*.

Índice de documentos : Índice que relaciona todos os documentos que compõem o *corpus*.

Índice invertido : Matriz esparsa cujas linhas são relativas aos termos e as colunas, aos documentos. Os valores nessa matriz são calculados pela equação (3.1), repetida a seguir.

$$|t| = \frac{tf \cdot \log \frac{N}{n}}{\sqrt{\sum W_{t_i}^2}} \tag{3.1}$$

A maior parte do tempo consumido pelo indexador *tf-idf* é gasta em operações de leitura e escrita no disco. Conforme já foi dito, o *corpus* é lido em formato comprimido para reduzir o impacto da leitura. Para a escrita, em lugar de comprimir, utilizou-se simplesmente o formato binário mais compacto possível. Todas as leituras e escritas são feitas usando o mecanismo de mapeamento de arquivos em memória do sistema operacional.

6.1.2 Indexador LSA

O indexador LSA necessita somente do índice invertido produzido pelo indexador *tf-idf* para completar o seu trabalho. A parte mais custosa do seu processamento é a extração dos valores singulares dessa matriz. Utilizou-se da biblioteca SVDLIBC para essa tarefa, por ser bem escrita e bastante utilizada para aplicações de análise da semântica latente (GLIOZZO; STRAPPARAVA, 2005; TURNEY, 2005). Ela reimplementa o método LAS2 da biblioteca SVDPACKC e boa parte da análise do algoritmo no trabalho de apresentação da biblioteca SVDPACKC (BERRY, 1992) vale igualmente para a SVDLIBC.

Resolvendo a equação (3.2), a biblioteca SVDLIBC fornece, dada a matriz contendo o índice invertido, os valores singulares que são a base da análise da semântica latente.

$$A = U \cdot \Sigma \cdot V^T \tag{3.2}$$

Uma vez conhecidas as matrizes U e Σ, elas são truncadas (processo descrito na seção 3.1) e calculam-se, para armazenamento nos arquivos de consulta, o produto $\cdot U \cdot \Sigma^{-1}$, necessário para a transposição de novos documentos para o espaço vetorial de busca, e a matriz de documentos do *corpus* já transpostos, d_k, seguindo a equação (3.5).

$$d_k = d^T \cdot U_k \cdot \Sigma_k^{-1} \tag{3.5}$$

As duas matrizes resultantes são armazenadas em dois arquivos. Esses arquivos formam o núcleo do sistema de buscas baseado em LSA desenvolvido neste trabalho e, junto ao léxico e ao índice de documentos produzidos pelo indexador *tf-idf*, compõem a interface entre o módulo indexador e o processador de consultas.

6.2 Descrição do processador de consultas

O módulo processador de consultas foi desenvolvido como uma aplicação cliente-servidor. O cliente é trivial, consistindo somente em uma interface visual para o envio de consultas ao servidor e a exibição das respostas. Ao servidor, portanto, é legada toda a complexidade do processamento da consulta, desde a preparação da cadeia de termos que a compõem até a organização dos resultados.

Cada cadeia de termos enviada como consulta é tratada pelo servidor como um novo documento a ser comparado com os demais do *corpus*. O programa, inicialmente, carrega em memória os dados dos arquivos que contêm o léxico L (índice de termos), a matriz reduzida de documentos D_{LSA} e a matriz M_{LSA} de mapeamento de novos documentos para o espaço vetorial no qual será feita a busca por análise da semântica latente. Ao receber uma consulta, ele separa os termos da cadeia de consulta e aplica o Algoritmo de Porter para obter o radical de cada um deles. A partir do léxico L criado pelo indexador, é criado um vetor esparso q que representa a consulta, da mesma maneira que um documento seria representado no modelo *tf-idf*. Multiplicando esse vetor q pela matriz de mapeamento M_{LSA}, obtém-se o vetor q_{LSA}, que representa, dentro do novo espaço vetorial, o que seria um documento composto somente pela consulta. O programa calcula então o cosseno do ângulo entre o vetor da consulta q_{LSA} e os vetores de cada um dos documentos do *corpus*. Quanto maior o cosseno desse ângulo, maior a similaridade entre a consulta e o documento dentro do modelo de LSA. Os documentos são ordenados por similaridade e o resultado é retornado ao usuário.

A implementação material do processador de consultas foi feita em duas linguagens. O módulo cliente foi feito em PHP para simplificar o desenvolvimento de uma interface

visual rica acessível remotamente. O módulo servidor foi desenvolvido em C++, aproveitando as bibliotecas já utilizadas no indexador para realizar as tarefas comuns. A comunicação é toda feita em baixo nível usando sockets. O desenvolvimento do servidor foi todo feito com otimização em mente. As leituras explícitas do disco são feitas uma única vez: quando o programa é iniciado. Mesmo assim, a complexidade computacional assintótica a $O(n \cdot \log n)$ de cada consulta (dominada pela ordenação feita ao final do cálculo da similaridade entre cada documento e a cadeia de consulta) deixa, no ambiente usado para testes, o tempo de resposta em torno dos sete décimos de segundo para um *corpus* de 473.876 documentos. Os resultados obtidos em testes serão explorados na seção seguinte.

6.3 Resultados

Todos os testes foram realizados em um único computador, com as seguintes características:

- **Processador:** Intel E6420, dotado de 4MB de memória *cache* nível 1 e com frequência de *clock* de 2,13GHz.
- **Memória de acesso aleatório:** 2GB de memória DDR2 com frequência de operação de 800MHz.
- **Armazenamento:** Dois discos rígidos de 8 mil rotações por minuto, com 8MB de *cache* interno e espaço de 500GB em RAID1 por *hardware*.
- **Sistema operacional:** Linux, com núcleo versão 2.6.24, configurado para utilizar todo o espaço de uma partição de 8GB para paginação de memória virtual.

Para os testes realizados, foi utilizado o primeiro disco de textos jornalísticos em inglês do *Reuters Corpus Volume 1*. Ele é composto de 224 arquivos comprimidos em formato *ZIP*, totalizando 577MB de dados comprimidos. Descomprimidos, são 473.876 arquivos somando 2,3GB de texto em codificação ISO-8859-1. Cada arquivo descomprimido contém um único texto jornalístico incluindo código de formatação HTML, encapsulado, junto a alguns dados, em XML.

6.3.1 Indexação e medidas objetivas de desempenho

Cada texto do *Reuters Corpus Volume 1* é classificado quanto aos seus assuntos e quanto aos setores da economia relacionados ao seu conteúdo. Originalmente, são 128

classificações de assunto e uma árvore de 872 setores da economia. Essas classificações, no entanto, contêm redundâncias e códigos sem significado identificados no próprio mapeamento fornecido pela Reuters como *dummy codes*. Para realizar os testes, a árvore dos setores de economia foi planificada e as classificações redundantes foram fundidas e corrigidas manualmente, resultando em 121 classificações de assunto e 367 setores da economia. Excertos das possibilidades nessas duas classificações podem ser vistas nas tabelas 6.1 e 6.2.

accounts/earnings
advertising
advertising/promotion
annual results
arts, culture, entertainment
asset transfers
balance of payments
biographies, personalities, people
bond markets
bonds/debt issues
brands
business news
capacity utilization
capacity/facilities
comment/forecasts
...

Tabela 6.1: Excerto do rol de classificações em assuntos do *corpus* de teste.

O primeiro passo nos testes foi a indexação do primeiro disco do *Reuters Corpus Volume 1*. Para verificar o efeito do número de dimensões mantidas no espaço vetorial representado pelos índices LSA, foram gerados índices para os 60 espaços vetoriais com 10 a 600 dimensões, de 10 em 10; para os 14 índices de 620 a 880 dimensões, de 20 em 20; e, aumentando a resolução das amostras em um setor que se mostrou interessante, foram feitos os 4 índices de 804 a 816 dimensões, 4 entre 817 e 820 e mais 3, de 20 em 20 dimensões, entre 840 e 880 dimensões. O processo de geração desses índices envolve somente uma execução do módulo de indexação *tf-idf*, mas 81 execuções do módulo LSA. O índice *tf-idf* resultante tem 425MB de dados e a sua geração exige aproximadamente 10 minutos. Os tamanhos e tempos de processamento para os índices LSA variam com o número de dimensões de uma maneira aparentemente linear para o volume de dados processado, conforme pode ser visto nas figuras 6.2 e 6.3[1]. De fato, em (BERRY, 1992) demonstra-se que o custo computacional do algoritmo utilizado pela biblioteca SVDLIBC para extração dos valores singulares de uma matriz cresce

[1]Os dados para os processos de indexação acima de 600 dimensões foram omitidos por forçarem o sistema a utilizar memória virtual alocada em disco, distorcendo os dados com um aumento no tempo de processamento que não é devido ao algoritmo.

linearmente com o aumento do número de dimensões. A complexidade computacional dos produtos entre matrizes efetuados após a aplicação do SVD, da mesma maneira, cresce linearmente com o aumento do número de dimensões, de maneira que a complexidade computacional assintótica do processo como um todo é $O(n)$, em acordo com a progressão observada para os tempos de execução.

abrasive products
accountancy and auditing
adhesives
advertising agencies
aero-engines
aerospace
agricultural equipment hire
agricultural machinery
agriculture
agriculture and horticulture
agriculture, forestry and fishing
air transport
aircraft components, not electrical
aircraft hiring and leasing
aircraft maintenance
...

Tabela 6.2: Excerto do rol de classificações em setores da economia do *corpus* de teste.

Com os índices LSA em mãos, foram feitos testes para verificar a influência do número de dimensões na qualidade dos resultados. As classificações em assuntos e setores da economia feitas pelos jornalistas da Reuters foram tratados como metas para o sistema. Supõe-se que os jornalistas fizeram boas associações semânticas, de maneira que o sistema será bom na medida em que for capaz de reproduzir os resultados das escolhas manuais.

Em primeiro lugar, foram feitas medidas dos cossenos dos ângulos nos espaços LSA obtidos para todas as classificações usadas pela Reuters para caracterizar o *corpus*. Essas medidas foram centralizadas na média e normalizadas para ficar entre -1 e 1, de modo que se pudesse avaliar o tipo de distribuição estatística gerada pelo método. Chamou-se a essa medida normalizada do cosseno *distância* – apesar de não ser imediatamente intuitivo, em coordenadas polares o uso do termo é justificável. Também com esse intuito, foram calculadas a assimetria e curtose de todas as curvas. São medidas padrão muito utilizadas para avaliar distribuições estatísticas (JOANES; GILL, 1998).

Esperava-se que fosse possível localizar alguma distribuição clássica, de comportamento já bem conhecido, que caracterizasse os dados obtidos. No entanto, isso não foi possível. Para a classificação em assuntos, a assimetria das distribuições variou entre

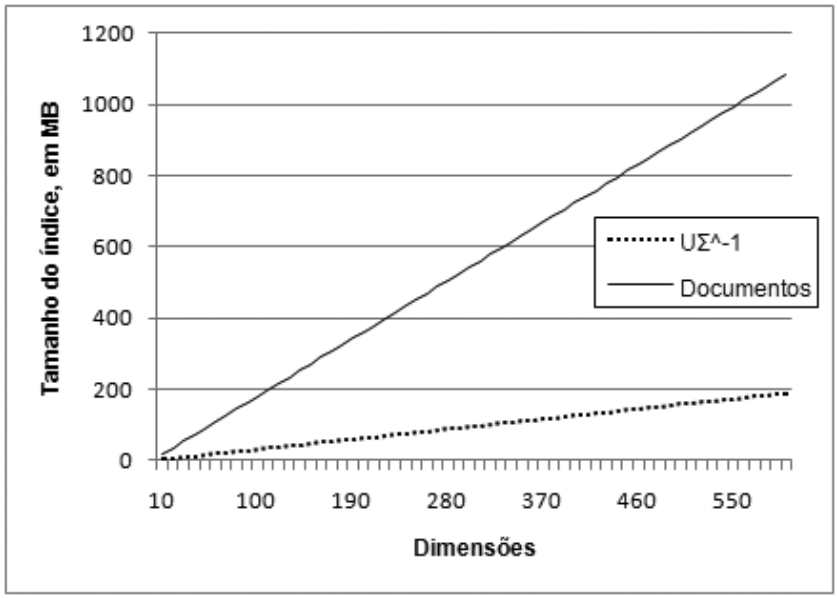

Figura 6.2: Gráfico de progressão do tamanho dos índices com o número de dimensões.

$0,335$ para 10 dimensões e até $2,039$ para 600 dimensões. A curtose também cresceu com o número de dimensões, variando entre $-0,450$ e $12,059$. Alguns dos histogramas obtidos são mostrados na figura 6.4. A classificação em setores da economia se comportou de maneira similar, com a assimetria variando entre $0,172$ e $2,423$ e a curtose entre $-0,526$ e $16,356$. Histogramas análogos aos das classificações sobre assuntos são mostrados na figura 6.5.

Apesar de as distribuições obtidas não se encaixarem em nenhum modelo clássico, a sua análise dá algumas informações interessantes sobre o método em relação às classificações utilizadas pela Reuters. O primeiro fato interessante é que as distribuições das classificações em assuntos e em setores da economia são muito semelhantes e mostram comportamento muito parecido conforme se aumenta o número de dimensões dos índices, tanto em relação ao aspecto dos gráficos quanto às medidas de assimetria e curtose. Outro dado interessante é a maneira como o LSA tende a medir a distância entre os textos e as classificações utilizadas conforme se altera o número de dimensões dos índices. Em espaços com número mais baixo de dimensões, as distâncias são distribuídas de maneira mais uniforme, enquanto que em espaços com número mais alto de dimensões há um acúmulo bem marcado na média (é importante notar que os histogramas apresentados nas figuras 6.4 e 6.5 estão em escala logarítmica, de maneira que a disparidade é bem maior do que aparenta à primeira vista).

Uma possível explicação para as distribuições presentes nos histogramas apresentados está na analogia entre as dimensões dos espaços construídos usando LSA e classificações semânticas. Um espaço com menos dimensões dá menos possibilidades

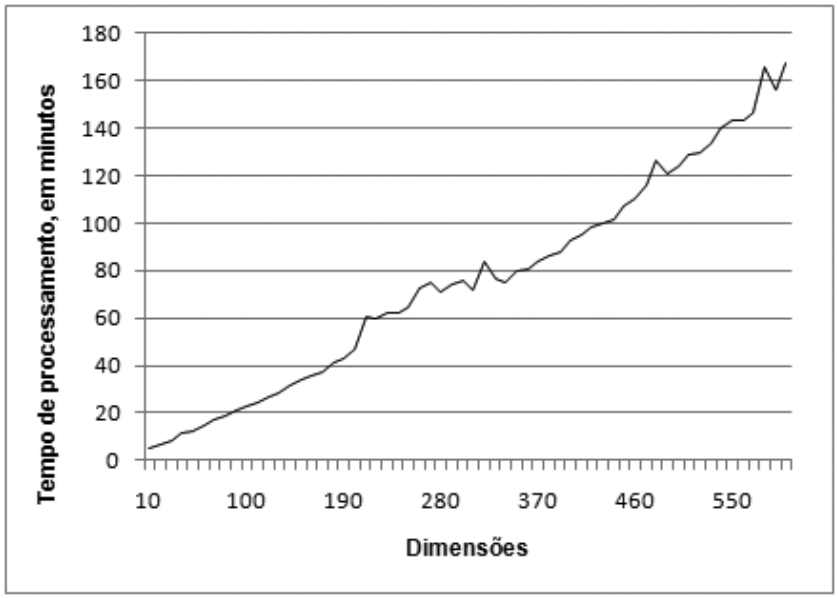

Figura 6.3: Gráfico de progressão do tempo de indexação LSA com o número de dimensões.

de identificações semânticas para os textos. Um índice com 10 dimensões daria a possibilidade de classificar o *corpus* em escalas de intensidade de 10 conceitos diferentes. Uma analogia possível seria tentar caracterizar todos os termos contidos em uma edição de jornal em intensidades diferentes de um conjunto de palavras como *{vermelho, política, criança, carro, presidente, violência, televisão, economia, guerra, clima}*. Observando o problema dessa maneira, seria razoável que um número pequeno de dimensões resultasse em uma distribuição equilibrada de distâncias, afinal, nenhuma das dimensões descreve realmente bem a maioria dos termos. Resultados que corroboram com essa conjectura são apresentados em (LANDAUER; DUMAIS, 1997).

A busca por uma distribuição clássica capaz de descrever os dados produzidos pelo método teve como motivação, principalmente, o próximo passo dos testes. Duas medidas extremamente comuns para a avaliação de sistemas de Recuperação de Informação são a precisão e a revocação[2] dos resultados retornados (YATES; NETO, 1999).

A precisão, em sistemas de Recuperação de Informação, é dada pela equação (6.1) e dá a proporção de documentos selecionados pelo sistema que de fato são relevantes. A precisão é uma medida de falsos positivos resultantes do método utilizado para a seleção de documentos.

$$precisão = \frac{|\{documentos\ relevantes\} \cap \{documentos\ selecionados\ automaticamente\}|}{|\{documentos\ selecionados\ automaticamente\}|} \tag{6.1}$$

[2]O termo em português *revocação* é equivalente ao original em inglês *recall*, este mais facilmente localizável na literatura acadêmica.

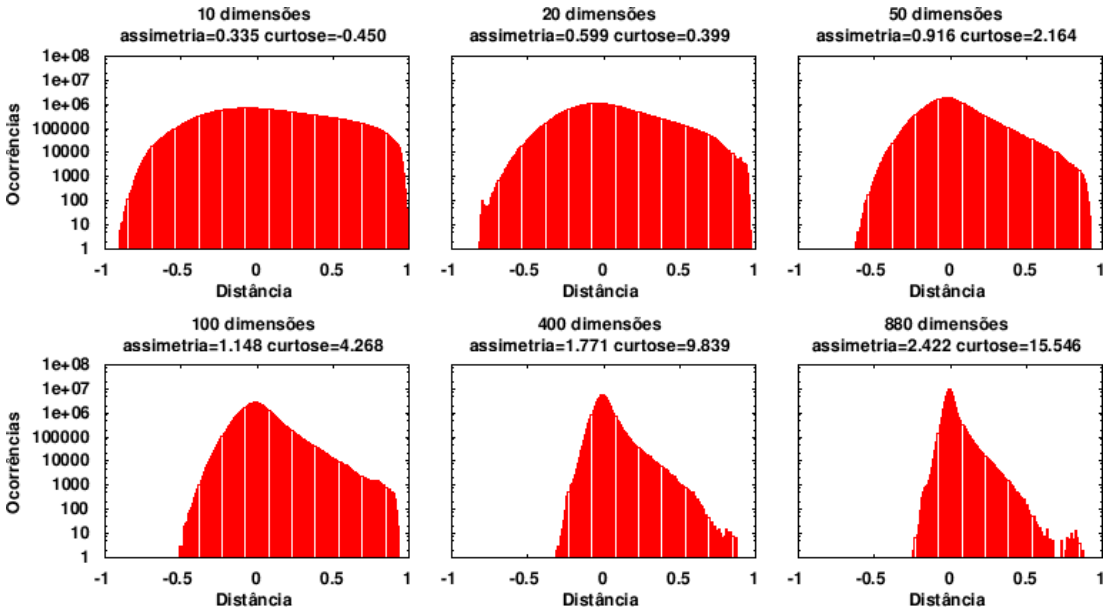

Figura 6.4: Distribuições normalizadas dos cossenos dos ângulos entre textos e classificações de assunto.

A revocação é uma medida diferente da precisão por ser em relação a todo o *corpus* e não somente aos documentos recuperados pelo sistema. Ela é dada pela equação (6.2) e pode ser vista como uma medida da proporção de falsos negativos obtidos pelo sistema.

$$revocacão = \frac{|\{documentos\ relevantes\} \cap \{documentos\ selecionados\ automaticamente\}|}{|\{documentos\ relevantes\}|} \tag{6.2}$$

Por não ter sido possível encontrar uma distribuição clássica capaz de descrever o comportamento do método LSA, os limiares para classificação automática pelo sistema de buscas foram testados arbitrariamente. Verificou-se o desempenho do sistema utilizando-se de 1 a 10 vezes o desvio padrão da distribuição para todos os índices. Pode-se observar, na figura 6.6, a progressão das médias da precisão e da revocação para todos os índices LSA em relação aos limiares de escolha. Para ter um parâmetro único de decisão, foi colocada também nos gráficos a média geométrica entre essas duas grandezas. Diferentemente da média aritmética, a média geométrica valoriza o equilíbrio entre as grandezas, e não somente o seu módulo. Tratando-se de duas medidas complementares, julgou-se que esse equilíbrio é fundamental. Para a classificação em assuntos, o maior valor obtido da média geométrica é $0,233$ com limiar de 2σ. Para setores da economia, o topo foi atingido em 3σ, com média geométrica de $0,143$ (bastante próximo do valor obtido com 2σ, $0,141$).

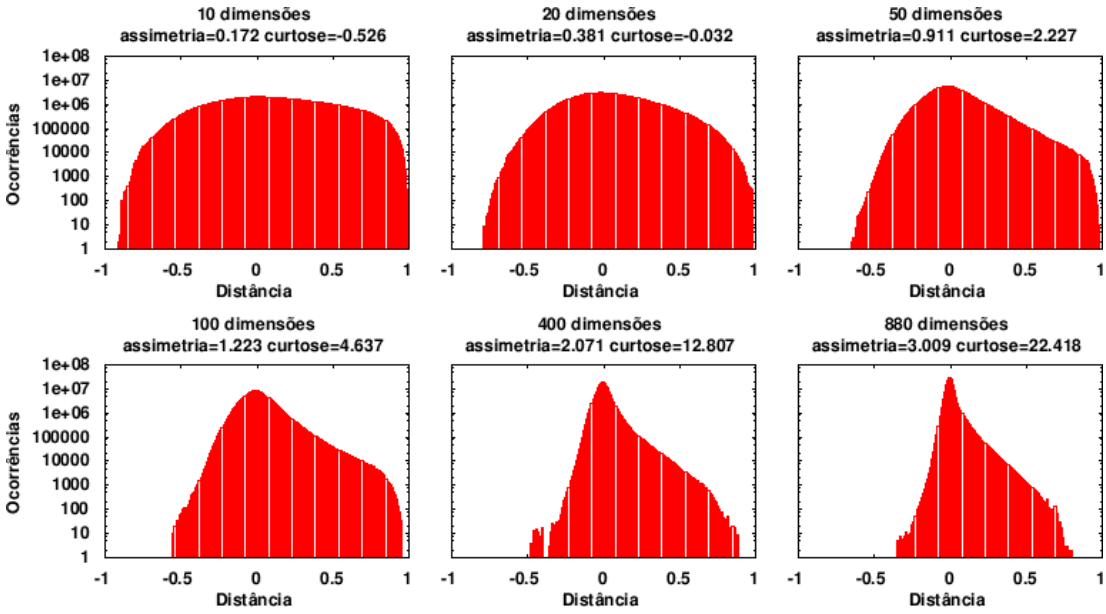

Figura 6.5: Distribuições normalizadas dos cossenos dos ângulos entre textos e classificações de setores da economia.

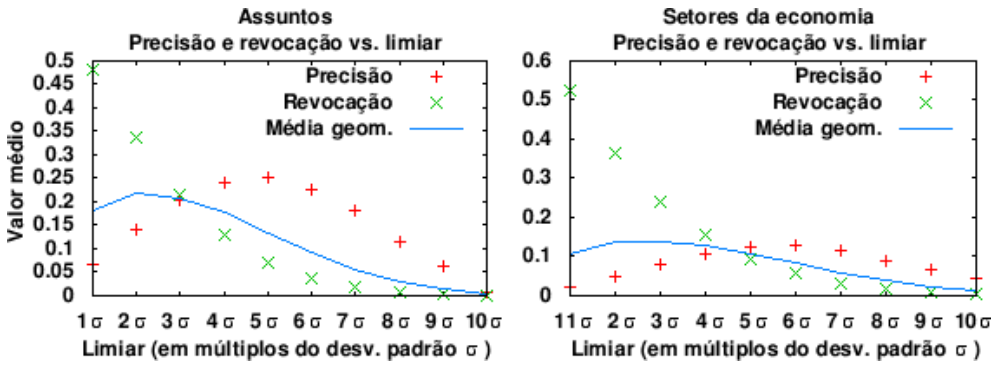

Figura 6.6: Gráficos das médias de precisão e revocação sobre o limiar de decisão do sistema (em múltiplos do desvio padrão σ).

Da mesma maneira que foi feito para a precisão e revocação em relação ao limiar de escolha, foram extraídos dados para a geração de gráficos das médias de progressão da precisão e revocação variando o número de dimensões dos índices da figura 6.7.

O gráfico da figura 6.6 é particularmente interessante. A queda abrupta no desempenho que se observa exatamente no índice de 820 dimensões poderia, a princípio, aparentar ser um resultado espúrio. Na verdade, ele mostra um comportamento curioso com que trabalhos anteriores sobre LSA já se depararam: não raro, pequenas alterações no número de dimensões de um índice LSA causam grandes mudanças na capacidade do sistema de fazer associações semânticas relevantes. Um gráfico apresentado em (LANDAUER; DUMAIS, 1997), reproduzido neste trabalho na figura 6.8, ilustra um caso representativo.

Em busca de dados mais precisos sobre a variação do desempenho, selecionaram-se

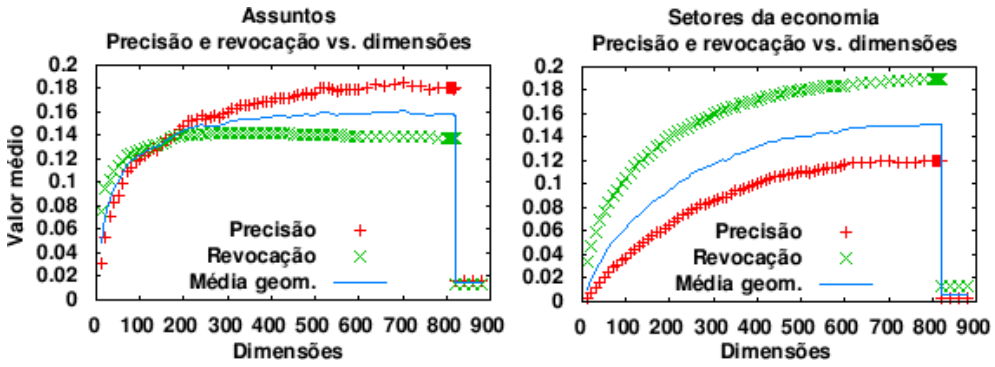

Figura 6.7: Gráficos das médias de precisão e revocação sobre o número de dimensões dos índices.

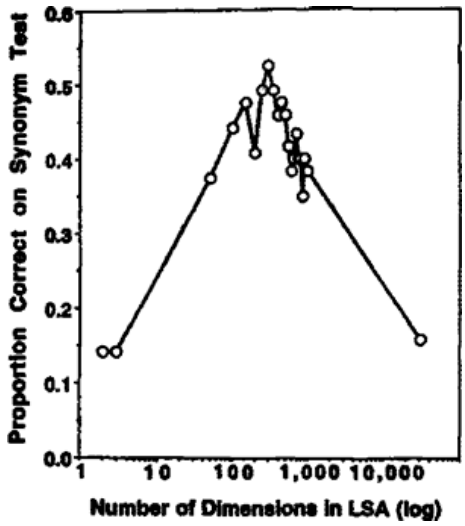

Figura 6.8: Gráfico do desempenho do sistema de LSA desenvolvido em (LANDAUER; DUMAIS, 1997) (extraído de (LANDAUER; DUMAIS, 1997)).

os histogramas dos melhores resultados de média geométrica entre precisão e revocação para cada número de dimensões. Eles são apresentados nas figuras 6.9 e 6.10.

O primeiro fato que chama atenção nos histogramas das figuras 6.9 e 6.10 é o número de dimensões no qual se atinge o desempenho máximo. A análise da semântica latente é baseada em um método de álgebra linear que gera um espaço vetorial no qual, idealmente, cada dimensão representa algum conceito semântico contido no *corpus* e explicitado pela co-ocorrência dos termos ao longo dos documentos. Nesse contexto, na medida em que os conceitos exprimidos pelas classificações usadas nos testes puderem ser ditos ortogonais, será razoável que haja uma correlação entre esse número de classificações e o número de dimensões do espaço que melhor descreve essas classificações. Conforme foi dito no início desta seção, foram usadas 121 classificações de assunto e 367 de setores da economia. O melhor resultado nos testes com classificações de assunto foi obtido com o índice de 80 dimensões e o melhor para setores de economia, com 490 dimensões. São resultados coerentes com as previsões feitas a partir da hipó-

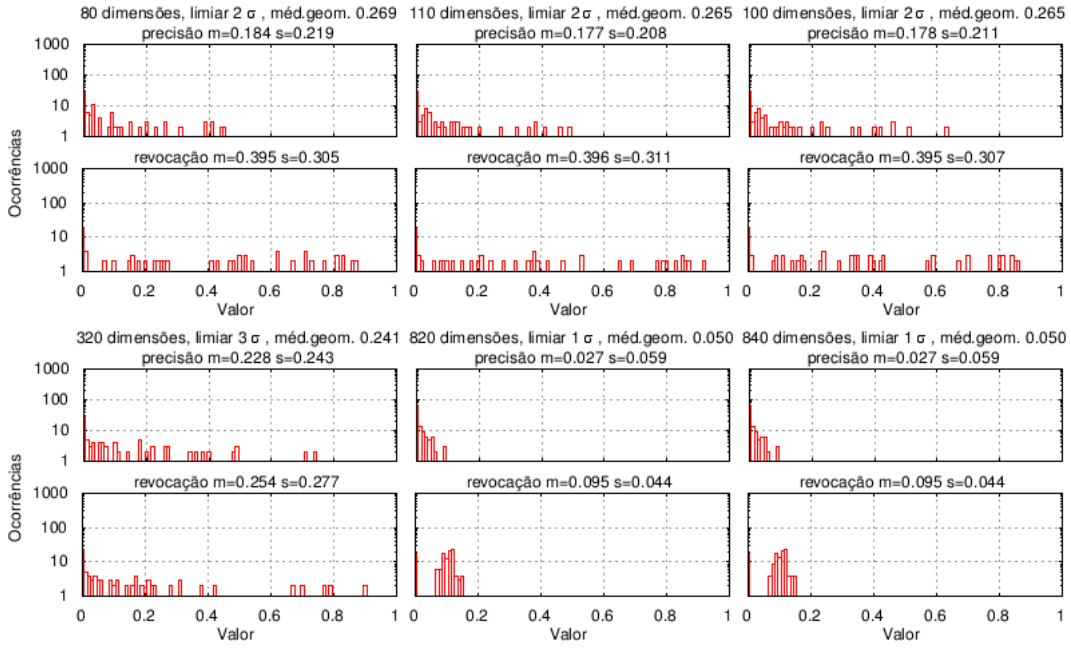

Figura 6.9: Histogramas de alguns dos melhores resultados, em ordem decrescente, variando o limiar de escolha para índices LSA. Classificação por assunto.

tese de que há correlação entre o espaço vetorial produzido usando indexação LSA e a organização semântica subjacente do *corpus*. Conversamente, partindo do pressuposto de que a hipótese está correta, o resultado indica que há uma superposição de $51,2\%$ nas classificações de assunto definidas pela Reuters, mas que, seguindo a mesma linha, seriam necessárias mais 123 classificações de setores da economia para descrever todas as possibilidades cobertas pelos documentos.

O segundo fato de interesse que se observa nos histogramas das figuras 6.9 e 6.10 é a progressão não-monotônica dos resultados, algo que fica mais óbvio nas classificações por setor da economia que nas classificações por assunto. Nos testes com classificações por assunto, os melhores resultados se dão em torno de 80 dimensões, caindo progressivamente conforme se distancia desse valor, mas com saltos como a melhoria de desempenho quando se diminui o número de dimensões de 110 para 100. Nas classificações por setores da economia, os saltos são ainda maiores, sendo o índice de 440 dimensões o melhor encontrado com menos de 490 dimensões e o de 510 dimensões, o terceiro melhor dentre os com mais de 500 dimensões.

O terceiro e último resultado de interesse que foi observado durante estes experimentos é a influência do limiar de decisão automática do sistema no seu desempenho. A progressão desse valor parece, a princípio, ter um comportamento bastante regular, como mostra a tabela 6.3. Entre 30 e 300 dimensões, o melhor limiar utilizado foi o dobro do desvio padrão da distribuição; de 310 a 819 dimensões, foi 3σ; e para os extremos, o limiar mais baixo, de 1σ.

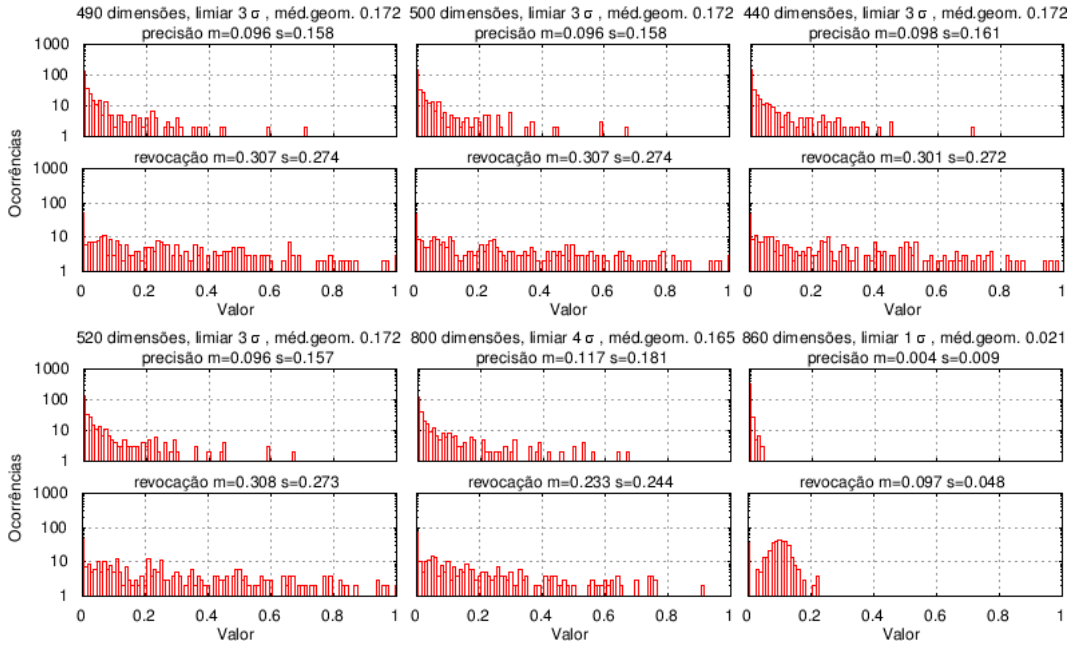

Figura 6.10: Histogramas de alguns dos melhores resultados, em ordem decrescente, variando o limiar de escolha para índices LSA. Classificação por setor da economia.

Limiar	Intervalo
1σ	$[10,20] \cup [820,880]$
2σ	$[30,300]$
3σ	$[310,819]$

Tabela 6.3: Melhores limiares observados para classificações por assunto para cada intervalo de número de dimensões em índices LSA.

Nos testes com classificações por setor da economia, o melhor limiar para escolha automática segue o padrão mostrado na tabela 6.4. O resultado é bastante interessante por explicitar que a queda de desempenho mostrada na figura 6.7 está correlacionada a uma queda um pouco menos abrupta no valor do melhor limiar para escolha. Para os índices entre 760 e 818 dimensões, o melhor limiar testado foi o de 4σ. Para o índice de 819 dimensões, imediatamente anterior à queda brusca de desempenho, esse valor cai para 3σ. Nos testes com índices subsequentes, com 820 a 880 dimensões, o limiar de escolha ótimo cai para 1σ, estabilizado nesse patamar mais baixo.

Limiar	Intervalo
1σ	$[10,40] \cup [820,880]$
2σ	$[50,230]$
3σ	$[240,740] \cup 819$
4σ	$[760,818]$

Tabela 6.4: Melhores limiares observados para classificações por setor da economia para cada intervalo de número de dimensões em índices LSA.

Os resultados apresentados corroboram com a teoria de que a queda de desempenho

observado na figura 6.7 é causada pelas alterações na distribuição das distâncias atribuídas pelos índices LSA às classificações fornecidas pela Reuters conforme é alterado o número de dimensões do espaço vetorial. O motivo para essa queda ser tão brusca continua obscuro, mas parece provável que testes com mudanças mais suaves no limiar para escolha automática mostrassem que essa queda não é tão brusca, mas sim que a escolha da medida para obtenção do limiar não foi acertada.

6.3.2 Resultados de buscas manuais

Um resultado bastante interessante, representativo da capacidade de associação de termos e documentos utilizando LSA, é obtido na busca pelo nome "Bush" no índice de documentos gerado. Conforme foi dito no início deste capítulo, o *corpus* indexado para os testes deste trabalho é composto de textos jornalísticos publicados em inglês pela agência de notícias Reuters entre agosto de 1996 e março de 1997. À época, Bill Clinton era o presidente dos Estados Unidos da América. Tinha vencido George H. W. Bush na eleição de 1992 e Bob Dole em 1996. A próxima eleição, quando George W. Bush se tornou presidente em seu lugar, só ocorreria em 2000. Mesmo assim, quando se faz uma busca por "Bush" (que em inglês significa *arbusto*), o primeiro resultado é um texto sobre a escolha do moderador do debate de candidatos à presidência entre Bill Clinton e Bob Dole. Nesse texto de 1447 caracteres, o termo "Bush" não ocorre nem uma única vez. Não é possível verificar se o motivo para essa associação é o então ex-presidente George H. W. Bush ou o então governador do Texas, futuro presidente dos EUA George W. Bush.

Resultados curiosos seguindo a linha de assuntos relacionados à política nos EUA em 1996 e 1997 são a verificação da proximidade entre "Bush" e "Republican", $0,38$; "Bush" e "Democrat", $0,05$; e, fora do contexto político, entre "Madonna" e "Evita" (filme no qual a artista estrelou em 1996), o cosseno é $0,90$. No entanto, as associações feitas pelo algoritmo não são sempre tão anedóticas. É necessário levar em consideração o fato de que o período coberto pelos textos do *corpus* é bastante curto e que textos jornalísticos tendem a ser muito concisos e concentrados no contexto do momento. Por algum motivo, o cosseno entre "Clinton" e "Democrat" é muito inferior àquele entre o sobrenome do então presidente e o nome do partido opositor, "Republican". Talvez isso seja resultado da quantidade de menções do partido do seu opositor no ano de eleição, mas também pode ser algo muito mais complexo.

Parte III

Considerações finais

7 Contribuições

Para este trabalho foi implementada uma versão do Preditor de Solomonoff restrita à produção de hipóteses pertencentes ao conjunto das Linguagens Regulares e um sistema de indexação e busca baseado na Análise da Semântica Latente. Esses dois sistemas foram testados em uma série de situações, gerando novos dados e interpretações. A questão da análise semântica de Linguagens Naturais por máquinas permanece em aberto, mas os resultados deste trabalho contribuem para a sua compreensão tanto pela classe dos algoritmos estatísticos quanto pelo conjunto das abordagens analíticas.

Nas seções que seguem são enumerados os resultados obtidos no correr deste trabalho e é apresentada uma breve discussão sobre aquilo que está relatado na parte II desta dissertação.

7.1 Preditor de Solomonoff

Conforme já mencionado na introdução do capítulo 5, não há antecedente na literatura para a implementação do Preditor de Solomonoff feita neste trabalho. Isso não é uma qualidade e traz consigo a dificuldade de não se poder fazer comparações de resultados. Os que se obteve na subseção 5.2.3 deste trabalho, durante o teste de divisibilidade de números inteiros, não têm precedente publicado. Em uma análise inicial, o comportamento observado no dispositivo poderia até ser interpretado como um defeito sério. No entanto, sob um foco pragmático, pode-se chegar a outras conclusões.

A hipótese preferencial proposta pelo preditor nos testes da subseção 5.2.3 é extremamente precisa. Ela prevê corretamente os resultados do gerador utilizado em todas as infinitas cadeias que este produz, mas inclui uma incorreta. Foram necessários mais de dois séculos para que gerações de cientistas corrigissem a equação newtoniana que descreve a energia cinética de um corpo com um termo que, em situações rotineiras, fica em torno de $10^{-8}J$. Um erro como o cometido pelo dispositivo preditor neste trabalho, infinitésimo, provavelmente nunca seria notado por pessoas que só pudessem observar os eventos produzidos pelo gerador original.

Independentemente da posição assumida sobre a taxa de convergência do protótipo simplificado do Preditor de Solomonoff que foi feito para este trabalho, as respostas fornecidas por ele apelam ao senso do pesquisador. Claramente são conjecturas razoáveis e a sua evolução segue o caminho lógico que se esperaria de um ser inteligente.

Solomonoff escreveu em diversos trabalhos que a incomputabilidade do seu preditor (incomputabilidade que é contornada com a redução feita neste trabalho para o universo das Linguagens Regulares) é uma característica desejável do sistema, e não um defeito[1] (SOLOMONOFF, 2003a). A argumentação é de que um modelo completo de aprendizado precisa, necessariamente, ser incomputável. O fato é que, em um ambiente real, a incomputabilidade – e mesmo a complexidade computacional exponencial em relação à entrada – limita extremamente a aplicabilidade de um sistema. A proposta de Solomonoff para mitigar esse problema – a interrupção do processamento incremental do preditor e a aceitação de respostas subótimas sempre que a demora for excessiva – não basta sequer para problemas relativamente simples nas máquinas de hoje.

Grande parte dos métodos de Inteligência Artificial que são usados hoje em dia se apoia em um artifício muito natural para seres humanos: medidas de utilidade. Respostas mais úteis são priorizadas em relação às menos úteis, mesmo quando as menos úteis são mais prováveis. Um bom médico não diagnostica um paciente como tendo uma doença incurável antes de verificar todas as doenças curáveis menos comuns. A resposta de que uma doença é incurável é marginalmente útil, enquanto que qualquer diagnóstico que possa levar a tratamento é interessante.

Neste ponto, é apenas uma conjectura, mas é possível que o Preditor de Solomonoff pudesse ter o seu desempenho bastante melhorado se, a cada evento, fosse relacionada uma informação sobre a sua utilidade. Em situações reais, não basta ter a melhor resposta que se conseguiu em termos de equilíbrio entre complexidade e capacidade de explicar o universo.

7.2 Representação semântica usando LSA

Em trabalhos anteriores sobre Análise da Semântica Latente (DEERWESTER et al., 1990; LANDAUER; DUMAIS, 1997; BERRY; DUMAIS; O'BRIEN, 1995; PAPADIMITRIOU et al., 2000; ESULI; SEBASTIANI, 2005) assim como neste, pode-se observar que a representação vetorial dada pelo método LSA a palavras em Linguagem Natural tem relação com a maneira como seres humanos interpretam relações semânti-

[1] As palavras exatas de Solomonoff são: "While these features are all very beautiful, there seemed at first to be a quite serious problem – that universal probability was incomputable. Surprisingly enough, this turned out to be not a *Bug* but a *Feature*!" (SOLOMONOFF, 2003a, p. 3)

cas. A aplicação desse método a sistemas de recuperação de informação não está no escopo deste trabalho, mas o seu desempenho ao responder a pedidos formulados por seres humanos é um parâmetro para a verificação do seu funcionamento. No entanto, o *corpus* utilizado nos testes deste trabalho é substancialmente maior e mais diverso que os utilizados nesses trabalhos clássicos sobre LSA e, nesta situação, vieram à tona algumas características interessantes do método.

Um fato interessante que foi observado nos testes é que, diferentemente do que é normalmente visto na literatura para outros métodos, nos testes feitos neste trabalho, a precisão e a revocação melhoraram juntas em um grande número de vezes. Essa é uma forte indicação de que o aumento de desempenho observado com mudanças no limiar de escolha e do número de dimensões do espaço vetorial a representar o *corpus* não é casual. O sistema mostrou uma progressão orientada no sentido de se alinhar com as expectativas das pessoas que fizeram a classificação dos textos.

O desempenho apresentado nos testes de classificação usando LSA neste trabalho foi bem inferior aos que podem ser encontrados em trabalhos como (LANDAUER; DUMAIS, 1997) e (SCHONE; JURAFSKY, 2000). Neles, fala-se de precisão e revocação acima dos 80%, enquanto que neste trabalho em momento algum se obteve mais que 23% de precisão ou 40% de revocação. Essa diferença, no entanto, se deve em grande parte à escolha do *corpus*. As provas de língua inglesa para estrangeiros TOEFL usadas em (LANDAUER; DUMAIS, 1997) e o *corpus* CELEX, organizado e anotado por linguistas, usado em (SCHONE; JURAFSKY, 2000) são muito inferiores em escala ao *corpus* RCV1 da Reuters usado neste trabalho. O conjunto de dados usado neste trabalho foi construído a partir do trabalho de centenas de jornalistas cujo compromisso primário era dar ao leitor a informação que o interessava. Se por um lado o foco restrito de cada texto e a consistência terminológica comum aos jornalistas são vantagens para testes como os que foram feitos neste trabalho, por outro, a inevitável falta de consistência no uso das classificações no *corpus* certamente afetou negativamente os resultados obtidos.

É importante relembrar que o objetivo deste trabalho, no que toca à LSA, não é tecer novas teorias ou avançar o conhecimento sobre as características do método. O motivo para os testes de funcionamento da Análise da Semântica Latente feitos neste trabalho é verificar a validade da sua aplicação como módulo de pré-processamento de textos em Linguagem Natural. Nesse contexto, o que se observou é que as características do método são adequadas e, mais que isso, que os seus parâmetros de desempenho e tempo de processamento são extremamente atraentes para a aplicação planejada.

8 Conclusão

Neste trabalho, foi implementado um protótipo do Preditor de Solomonoff (SOLOMONOFF, 1964) restrito à produção de hipóteses pertencentes ao conjunto das Linguagens Regulares. Foram feitos alguns testes com esse programa e foi determinado que a sua complexidade computacional e o custo de cada uma das tarefas que o programa executa resultam em um tempo de processamento excessivamente alto para a maioria das aplicações práticas. Foram observadas algumas características interessantes sobre o seu funcionamento, mas, principalmente, foi verificada a sua eficiência, em termos de número de passos computacionais, para fornecer boas hipóteses sobre o gerador de uma sequência de eventos.

Em um segundo momento, foi implementado e testado um sistema de indexação e busca usando o método de Análise da Semântica Latente proposto em (DEERWESTER et al., 1990). Esse sistema foi exercitado ao longo de diversos dias de processamento para construir 81 índices de diferentes números de dimensões, totalizando 71,5GB de dados. Os índices criados foram então utilizados para medir os cossenos entre os vetores das classificações designadas pela Reuters e os documentos presentes no primeiro disco do *corpus* RCV1. A partir dessas medidas, foram consolidados gráficos para a análise de diversas características do método LSA. A conclusão final a partir dos testes é que o sistema tem potencial para funcionar da maneira proposta no início deste trabalho, como um módulo de pré-processamento de textos em Linguagem Natural dentro de um sistema de análise semântica.

Os resultados apresentados no capítulo 6 fazem crer que a representação resultante de uma indexação usando LSA pode ser muito vantajosa para um mecanismo que observe textos em Linguagem Natural como cadeias de eventos. No espaço vetorial resultante da aplicação do método, pelo menos parte das relações semânticas entre as palavras fica exposta na sua morfologia. O custo computacional envolvido na indexação de um *corpus* de grande porte é relativamente baixo e, mesmo que o conteúdo desse *corpus* não aborde diretamente todos os assuntos, se o número de dimensões do espaço vetorial criado não for baixo demais para que se possa recuperar os termos originais (não os documentos, já que isso inutilizaria a aplicação do método), não há como ter

perda de conteúdo. Tudo aponta para prováveis ganhos no processamento de textos em Linguagem Natural utilizando o Preditor de Solomonoff.

A complexidade computacional do Preditor de Solomonoff, apesar de todas as qualidades do modelo, ainda é um obstáculo de difícil transposição. Não foi possível, neste trabalho, fazer testes com o preditor implementado para casos realmente interessantes. Mesmo o processamento de uma única palavra de comprimento médio da língua portuguesa – por exemplo, a própria palavra "médio" – é excessivamente demorado. Os testes apresentados no capítulo 5 foram executados usando um alfabeto de dois símbolos e uma cadeia com 11 símbolos fez com que o programa demorasse pouco menos de uma hora e meia para completar o processamento. Considerando que o dispositivo testa, num caso em que não haja regularidades óbvias, $|\Sigma|^{comprimento\_entrada}$ hipóteses (sendo $|\Sigma|$ o número de símbolos do alfabeto e $comprimento\_entrada$ o comprimento da entrada do programa), o tempo médio consumido por hipótese fica em torno de $2,5s$. Ignorando acentuação e maiúsculas, o alfabeto da língua portuguesa atualmente tem 26 símbolos. Dessa maneira, numa predição bastante simplificada, o processamento da palavra "médio" exigiria em torno de 275 dias.

Apesar dos excelentes resultados intermediários, conforme já era previsto, a complexidade computacional do Preditor de Solomonoff inviabilizou a integração dos módulos. Serão necessários outros artifícios, como a heurística proposta na seção 7.1, ou avanços significativos na velocidade de processamento dos sistemas computacionais para que um dispositivo preditor como o implementado possa ser usado na escala necessária para o processamento de Linguagem Natural.

Os resultados deste trabalho serão mantidos acessíveis na Internet no endereço http://iuri.chaer.org/pesquisa/. O código-fonte dos programas utilizados também será disponibilizado no mesmo endereço para livre consulta e desenvolvimentos futuros após a aceitação deste trabalho para publicação. Espera-se, assim, potencializar as contribuições desta pesquisa.

9 Trabalhos futuros

Ao final deste relato, parece haver ainda mais para ser feito do que no começo. A continuação mais direta para este trabalho é encontrar maneiras de contornar o obstáculo representado pela alta complexidade computacional envolvida na execução do Preditor de Solomonoff, de maneira a possibilitar o processamento de Linguagens Naturais. Os resultados independentes com os dois módulos construídos para o sistema proposto são muito encorajadores. A sua integração, sem dúvida, exigirá um grande salto, mas tudo faz crer que o resultado será promissor.

A proposta da seção 7.1, de utilizar uma medida de utilidade para orientar as respostas subótimas do Preditor de Solomonoff, é um possível caminho para poder aplicar o preditor a problemas reais de maior porte. É uma proposta bastante complicada de generalizar por envolver um conceito tão fluído como "utilidade" e provavelmente afetaria negativamente a independência de contexto do modelo de Solomonoff. Ainda assim, é possível que um pouco de equilíbrio entre generalidade e complexidade resultasse em um método mais facilmente aplicável.

Quanto ao uso da Análise da Semântica Latente com o objetivo de reduzir o tempo de convergência do processamento de textos em Linguagem Natural pelo Preditor de Solomonoff, há um problema de incompatibilidade de representação que não foi discutido, mas que será importante uma vez encontrados métodos de mitigar o problema da complexidade computacional do preditor: o resultado da aplicação de LSA é um espaço vetorial contínuo. A representação usual desse tipo de dado nas máquinas de processamento discreto que se usam atualmente é muito custosa em espaço, o que tem um impacto direto no tempo de processamento pelo preditor. Seria valiosa uma avaliação do impacto da redução da precisão da representação dos valores contínuos resultantes do LSA no seu desempenho.

Finalmente, em uma linha que aparenta ser quase tão difícil de implementar quanto promissora, seria extremamente interessante que fosse construído um dispositivo implementando a versão completa do Preditor de Solomonoff, capaz de processar hipóteses de Linguagens de Nível 0. Em (ROCHA, 2000), é proposta a aplicação do Formalismo Adaptativo para esse tipo de tarefa. Em (CHAER; ROCHA, 2009), é proposto

um método de busca semântica dentro desse formalismo, algo diretamente aplicável a um solucionador de problemas capaz de classificar as suas entradas em diferentes grupos de problemas, conforme proposto em (SOLOMONOFF, 1986). Além disso, provavelmente será necessário adicionar o tempo de execução dos programas à sua Complexidade de Kolmogorov, seguindo a proposta de Schmidhuber realizada no seu Solucionador Ótimo de Problemas Ordenados (*Optimal Ordered Problem Solver* no original) (SCHMIDHUBER, 2004).

Parte IV

Referências

Referências

ALPAYDIN, E. *Introduction To Machine Learning*. Cambridge, MA, USA: MIT Press, 2004.

APHASIA. Bethesda, Maryland, USA: National Instutute of Health (NIH), 2008. Disponível em: <http://www.nidcd.nih.gov/health/voice/aphasia.asp>. Acesso em: Maio de 2008.

ASHER, R. E. (Ed.). *The Encyclopedia of Language and Linguistics*. 2. ed. Amsterdã, Holanda: Elsevier, 2005.

BARTHES, R. *O Neutro*. São Paulo, SP, Brasil: Martins Fontes, 2003.

BERRY, M. W. Large-scale sparse singular value computations. *International Journal of Supercomputer Applications*, MIT Press, Cambridge, MA, USA, v. 6, n. 1, p. 13–49, 1992.

BERRY, M. W.; DUMAIS, S. T.; O'BRIEN, G. W. Using linear algebra for intelligent information retrieval. *SIAM review*, Society for Industrial and Applied Mathematics, Philadelphia, PA, USA, v. 37, n. 4, p. 573–595, 1995.

BLEI, D. M.; NG, A. Y.; JORDAN, M. I. Latent dirichlet allocation. *The Journal of Machine Learning Research*, MIT Press, Cambridge, MA, USA, v. 3, p. 993–1022, 2003.

BRACHMAN, R.; FIKES, R.; LEVESQUE, H. Krypton: A Functional Approach to Knowledge Representation. *Computer*, IEEE Computer Society, Washington, DC, USA, v. 16, n. 10, p. 67–73, 1983.

BRACHMAN, R. J.; LEVESQUE, H. J. *Readings in Knowledge Representation*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 1985.

CARAMAZZA, A.; ZURIF, E. B. Dissociation of algorithmic and heuristic processes in language comprehension: Evidence from aphasia. *Brain and Language*, Elsevier, Amsterdã, Holanda, v. 3, n. 4, p. 572–582, 1976.

CHAER, I.; ROCHA, R. L. A. Um estudo sobre a Ação Elementar de Consulta no Formalismo Adaptativo. In: *Memórias do WTA 2009 – Terceiro Workshop de Tecnologia Adaptativa*. São Paulo, SP, Brasil: Escola Politécnica da USP, 2009. p. 129–134. ISBN 978-85-86686-51-1.

CHAITIN, G. J. Algorithmic information theory. Cambridge University Press, Cambridge, MA, USA, 1997.

CHOMSKY, N. *Aspects of the Theory of Syntax*. Cambridge, MA, USA: MIT Press, 1965.

CHOMSKY, N. *Knowledge of Language: Its Nature, Origin, and Use*. Westport, CT, USA: Praeger/Greenwood, 1986.

CRAIG, E. (Ed.). *Routledge Encyclopedia of Philosophy*. London, UK: Routledge, 1998.

DAMÁSIO, A. O. *O Erro de Descartes: emoção, razão eo cérebro humano*. Trad. Dora Vicente e Georgina Segurado. São Paulo, SP, Brasil: Companhia das Letras, 1999.

DAVIS, R.; SHROBE, H.; SZOLOVITS, P. What Is a Knowledge Representation? *AI Magazine*, AAAI, Menlo Park, CA, USA, v. 14, n. 1, p. 17–33, 1993.

DEERWESTER, S. et al. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, New York, NY, USA, v. 41, n. 6, p. 391–407, 1990.

ENCYCLOPÆDIA Britannica Online. Chicago, IL, USA: Encyclopædia Britannica, Inc., 2010. Disponível em: <http://www.britannica.com/EBchecked/topic%-/424706/Ockhams-razor>. Acesso em: Janeiro de 2010.

ESULI, A.; SEBASTIANI, F. Determining the semantic orientation of terms through gloss classification. In: ACM. *Proceedings of the 14th ACM international conference on Information and knowledge management*. New York, NY, USA: ACM Press, 2005. p. 624.

FALK, J. S. *Saussure and American linguistics. Cambridge companion to Saussure, ed. by Carol Sanders*. Cambrige, UK: Cambridge University Press, 2003.

FITTING, M. *First-Order Logic and Automated Theorem Proving*. Berlin/Heidelberg, Alemanha: Springer, 1996.

FODOR, J. A. *The modularity of mind*. Cambridge, MA, USA: MIT Press, 1983.

GINZBURG, C.; CAROTTI, F. *Mitos, emblemas, sinais: morfologia e história*. São Paulo, SP, Brasil: Companhia das Letras, 1990.

GLIOZZO, A.; STRAPPARAVA, C. Domain kernels for text categorization. In: ACL. *Proceedings of the 9th Conference on Computational Natural Language Learning (CoNLL)*. New Brunswick, NJ, USA: ACL, 2005. p. 56–63.

GOLUB, G.; KAHAN, W. Calculating the Singular Values and Pseudo-Inverse of a Matrix. *Journal of the Society for Industrial and Applied Mathematics: Series B, Numerical Analysis*, Society for Industrial and Applied Mathematics, v. 2, n. 2, p. 205–224, 1965.

HAUSER, M. D.; CHOMSKY, N.; FITCH, W. The Faculty of Language: What Is It, Who Has It, and How Did It Evolve? *Science*, AAAS, Washington, DC, USA, v. 298, n. 5598, p. 1569–1579, 2002.

HAY, N. Solomonoff-lite evaluator. 2007. Disponível em: <http://www.singinst-.org/blog/2007/07/09/solomonoff-lite-evaluator/>. Acesso em: Outubro de 2009.

HEBERT, S. et al. Revisiting the dissociation between singing and speaking in expressive aphasia. *Brain*, Oxford University Press, Oxford, UK, v. 126, n. 8, p. 1838, 2003.

HENDLER, J.; BERNERS-LEE, T. From the Semantic Web to social machines: A research challenge for AI on the World Wide Web. *Artificial Intelligence*, Elsevier, Amsterdã, Holanda, n. 174, p. 156–161, 2010.

HOFMANN, T. Probabilistic latent semantic indexing. In: *Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval*. New York, NY, USA: ACM Press, 1999. p. 50–57.

HOPCROFT, J. E. An n log n algorithm for minimizing the states in a finite automaton. Stanford University, Stanford, CA, USA, 1971.

HOPCROFT, J. E.; MOTWANI, R.; ULLMAN, J. D. *Introduction to automata theory, languages, and computation*. 2. ed. Boston, MA, USA: Addison-Wesley, 2000.

HUTTER, M. *Universal artificial intelligence: Sequential decisions based on algorithmic probability*. Berlin/Heidelberg, Alemanha: Springer, 2005.

INFORMATION overload causes stress. *Reuters Magazine*, Reuters, London, UK, 1997. Disponível em: <http://library.humboldt.edu/~ccm/fingertips/ioverloadstats.html>. Acesso em: Janeiro de 2009.

IWANSKA, L.; SHAPIRO, S. C. *Natural Language Processing and Knowledge Representation: Language for Knowledge and Knowledge for Language*. Cambridge, MA, USA: MIT Press, 2000.

JOANES, D. N.; GILL, C. A. Comparing measures of sample skewness and kurtosis. *The Statistician*, Blackwell Publishers, Hoboken, NJ, USA, v. 47, n. 1, p. 183–189, 1998.

JURAFSKY, D.; MARTIN, J. H. *Speech And Language Processing*. Englewood Cliffs, NJ, USA: Prentice Hall, 2008.

LANDAUER, T. K.; DUMAIS, S. T. A Solution to Plato's Problem: The Latent Semantic Analysis Theory of Acquisition, Induction, and Representation of Knowledge. *Psychological Review*, APA – American Psychological Association, New York, NY, USA, v. 104, p. 211–240, 1997.

LENAT, D. B. et al. Cyc: toward programs with common sense. *Communications of the ACM*, ACM Press, New York, NY, USA, v. 33, n. 8, p. 30–49, 1990.

LI, M.; VITÁNYI, P. M. B. Inductive reasoning and kolmogorov complexity. *Proceedings of the Fourth Annual Conference on Structure in Complexity Theory*, IEEE Computer Society, Washington, DC, USA, p. 165–185, 1989.

LI, M.; VITÁNYI, P. M. B. *An Introduction to Kolmogorov Complexity and Its Applications*. Berlin/Heidelberg, Alemanha: Springer, 1997.

LOCKE, J. *Ensaio acerca do entendimento humano. (Col. Os Pensadores)*. São Paulo, SP, Brasil: Ed. Abril, 1973.

MAGNINI, B.; CAVAGLIA, G. Integrating subject field codes into wordnet. In: *Proceedings of LREC-2000, Second International Conference on Language Resources and Evaluation*. Atenas, Grécia: LREC-2000, 2000. p. 1413–1418.

MCGUINNESS, D. L.; HARMELEN, F. van et al. OWL Web Ontology Language Overview. *W3C Recommendation*, W3C, Cambridge, MA, USA, v. 10, p. 2004–03, 2004.

MINSKY, M. L. *Computation: finite and infinite machines*. Englewood Cliffs, NJ, USA: Prentice Hall, 1967.

MITCHELL, M. *An Introduction to Genetic Algorithms*. Cambridge, MA, USA: Bradford Books, 1996.

MONTAGUE, R. The proper treatment of quantification in ordinary English. *Proceedings of the 1970 Stanford Workshop on Grammar and Semantics*, D. Reidel, Dordrecht, Holanda, v. 49, p. 221–242, 1973.

MYLOPOULOS, J. An overview of Knowledge Representation. In: *Proceedings of the workshop on Data abstraction, databases and conceptual modelling*. New York, NY, USA: ACM, 1981. v. 11, n. 2, p. 5–12.

NERO, H. S. D. *O sítio da mente pensamento, emoção e vontade no cérebro humano*. São Paulo, SP, Brasil: Collegium Cognitio, 2002.

NILSSON, N. J. Introduction to Machine Learning. Unpublished Draft – Department of Computer Science, Stanford University, Redwood City, CA, USA, 1996. Disponível em: <http://robotics.stanford.edu/people/nilsson/mlbook.html>. Acesso em: Janeiro de 2010.

PAPADIMITRIOU, C. H. et al. Latent semantic indexing: A probabilistic analysis. *Journal of Computer and System Sciences*, Elsevier, Amsterdã, Holanda, v. 61, n. 2, p. 217–235, 2000.

RISSANEN, J. A universal prior for integers and estimation by minimum description length. *Annals of Statistics*, Institute of Mathematical Statistics, Beachwood, OH, USA, v. 11, n. 2, p. 416–431, 1983.

ROCHA, R. L. *Um método de escolha automática de soluções usando tecnologia adaptativa*. Tese (Doutorado) — Escola Politécnica da USP, São Paulo, SP, Brasil, 2000.

RUSSELL, S. J.; NORVIG, P. *Artificial Intelligence – A Modern Approach*. 2. ed. Englewood Cliffs, NJ, USA: Prentice Hall, 2003.

SALTON, G.; BUCKLEY, C. Term-weighting approaches in automatic text retrieval. *Information Processing and Management: an International Journal*, Pergamon Press, Inc., Tarrytown, NY, USA, v. 24, n. 5, p. 513–523, 1988.

SAUSSURE, F. *Course in General Linguistics*. Trad. Roy Harris. La Salle, IL, USA: Open Court, 1983.

SCHMIDHUBER, J. Optimal Ordered Problem Solver. *Machine Learning*, Kluwer Academic Publishers, Amsterdã, Holanda, n. 54, p. 211–254, 2004.

SCHONE, P.; JURAFSKY, D. Knowledge-free induction of morphology using latent semantic analysis. In: ASSOCIATION FOR COMPUTATIONAL LINGUISTICS. *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning*. Morristown, NJ, USA, 2000. v. 7, p. 67–72.

SOLOMONOFF, R. J. A formal theory of inductive inference. parts I and II. *Information and Control*, Academic Press Inc., Boston, MA, USA, v. 7, n. 2, p. 224–254, 1964. Disponível em: <http://world.std.com/~rjs/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. Inductive inference theory - A unified approach to problems in pattern recognition and artificial intelligence. *Proceedings of the Fourth International Joint Conference on Artificial Intelligence*, Tbilisi, Georgia, U.S.S.R., p. 274–280, 1975. Disponível em: <http://world.std.com/˜rjs/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. Complexity-based induction systems: Comparisons and convergence theorems. *Information Theory, IEEE Transactions on*, IEEE, New York, NY, USA, v. 24, n. 4, p. 422–432, 1978. Disponível em: <http://world.std.com/˜rjs-/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. The application of algorithmic probability to problems in artificial intelligence. *Proceedings of the Second Annual Conference on Uncertainty in Artificial Intelligence*, Elsevier, Amsterdã, Holanda, p. 473–491, 1986. Disponível em: <http://world.std.com/˜rjs/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. A system for incremental learning based on algorithmic probability. *Proceedings of the Sixth Israeli Conference on Artificial Intelligence, Computer Vision and Pattern Recognition*, Tel Aviv, Israel, p. 515–527, 1989. Disponível em: <http://world.std.com/˜rjs/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. Two Kinds of Probabilistic Induction. *The Computer Journal*, British Computer Society, London, UK, v. 42, n. 4, p. 256–259, 1999. Disponível em: <http://world.std.com/˜rjs/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. The Universal Distribution and Machine Learning. *The Computer Journal*, British Computer Society, London, UK, v. 46, n. 6, p. 598–601, 2003. Disponível em: <http://world.std.com/˜rjs/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. Three Kinds of Probabilistic Induction: Universal Distributions and Convergence Theorems. *The Computer Journal*, British Computer Society, London, UK, 2003. Disponível em: <http://world.std.com/˜rjs/pubs.html>. Acesso em: Janeiro de 2009.

SOLOMONOFF, R. J. *Lecture 1: Algorithmic Probability*. 2005. Disponível em: <http://world.std.com/˜rjs/pubs.html>. Acesso em: Janeiro de 2009.

TURNEY, P. D. Measuring semantic similarity by latent relational analysis. *International Joint Conference on Artificial Intelligence*, Lawrence Erlbaum Associates Ltd., Philadelphia, PA, USA, v. 19, p. 1136–1141, 2005.

WIDDOWS, D. Semantic vector products: Some initial investigations. In: *Proceedings of the Second Quantum Interaction Symposium (QI-2008)*. UK: College Publications, 2008.

WILLIS, D. G. Computational Complexity and Probability Constructions. *Journal of the ACM (JACM)*, ACM Press, New York, NY, USA, v. 17, n. 2, p. 241–259, 1970.

WURMAN, R. S. *Information Anxiety*. New York, NY, USA: Doubleday, 1989.

WURMAN, R. S. *Information anxiety 2*. Indianapolis, IN, USA: Hayden/Que, 2001.

YATES, R. B.; NETO, B. R. *Modern Information Retrieval*. Boston, MA, USA: Addison-Wesley, 1999.