Edite - Um sistema de acesso a uma base de dados em linguagem natural

Luísa Marques da Silva, Nuno Mamede e David Matos (INESC / IST)

luisa,Nuno.Mamede@inesc.pt,David.Matos@acm.org

O sistema Edite, surgiu para ser incorporado nos quiosques multimédia desenvolvidos pelo INESC, e colmatar as fraquezas inerentes ao acesso a base de dados através de um sistema de janelas. Assim, o sistema Edite é um sistema de acesso a uma base de dados de recursos turísticos em Linguagem Natural, funcionando para o Português, Espanhol, Francês e Inglês. Descreve-se aqui a sua arquitectura, e alguns problemas levantados aquando do seu desenvolvimento. Em particular focam-se as simplificações que foram feitas no que diz respeito ao reconhecimento da língua, tendo em vista os resultados práticos que se esperam de um sistema desta natureza.

Edite, uma Interface em Linguagem Natural

Portugal é um país cada vez mais consciente da importância da indústria do turismo no equilíbrio da sua economia: são cada vez mais e de melhor qualidade os suportes turísticos a oferecer. Num levantamento exaustivo dos recursos turísticos nacionais, surge o Inventário de Recursos Turísticos ou IRT, cuja base de dados é da responsabilidade do Instituto de Engenharia de Sistemas e Computadores (INESC).

De maneira a permitir que qualquer pessoa possa aceder a essa base de dados, o INESC desenvolveu quiosques multimédia que, pela sua aparência familiar, facilitam a comunicação Homem-Máquina. No

entanto, por muito atractivos que sejam os seus gráficos e vídeos, um quiosque multimédia tem as suas limitações:

- um utilizador tem de se adaptar à linguagem do sistema, por muito simples que seja;
- um utilizador que procure uma informação inexistente pode perder muito tempo a navegar, de janela em janela, até perceber que o sistema não o vai poder ajudar (e o sistema não tem maneira de o avisar).

Uma maneira de contornar estes problemas é a incorporação, nos quiosques multimédia, de um sistema que torna possível o acesso à base de dados em Linguagem Natural[2] (LN): o utilizador passa a poder exprimir-se na sua própria língua, e, se por acaso a informação que procura não existir, o sistema informa-o do facto. Para além desta, uma interface de acesso a uma base de dados em LN, tem ainda outras vantagens:

- as perguntas a que, por alguma razão, o sistema não consegue responder, podem ser registadas e a causa do insucesso localizada e reparada. Desde modo o sistema está sempre em crescimento, tornando-se cada vez mais robusto e competente;
- passam a ser possíveis perguntas que envolvam quantificação – *Indique-me 10 hotéis com sauna.* – qualificação – *Indique-me uma pensão barata.* – ou negação – *Onde é que há parques de campismo que não exijam cartão de campista?*;
- suportam-se expressões elípticas, permitindo ao utilizador exprimir-se de uma forma muito rápida – *A torre de Belém está aberta ao sábado? E ao domingo?*.

[2] Entende-se por Linguagem Natural aquela que evoluiu naturalmente através da comunicação entre as pessoas (por exemplo Português, Espanhol, Italiano). Por oposição, tem-se as linguagens inventadas ou códigos, como as linguagens de programação.

Assim, com o objectivo de aperfeiçoar o desempenho dos quiosques, o INESC decidiu incorporar um sistema de Interface em Linguagem Natural (ILN), que se baptizou com o nome de Edite. Este sistema não funciona apenas para a língua portuguesa, mas também responde a perguntas em Inglês, Francês e Espanhol.

Sendo a construção de uma ILN um processo complicado, decidiu-se como objectivo a curto prazo, desenvolver um sistema muito simples, que apenas respondesse a um tipo bem definido de perguntas, mas que fosse depois fácil de estender. Deste modo, tornou-se fundamental, no desenvolvimento deste projecto, separar o "trigo do joio", *i.e.*, o que realmente interessava para os propósitos delineados, de certos pormenores que, apesar de atribuírem um maior valor linguístico ao sistema, seriam redundantes no que respeito à sua aplicação real. Assim, sem nunca subvalorizar o rigor linguístico, muitas vezes este foi sacrificado em função da necessidade de atingir resultados práticos.

Domínio da Aplicação

A primeira etapa do desenvolvimento de uma ILN passa pela definição do subconjunto da linguagem de que o utilizador se poderá servir para alcançar os seus propósitos. Esse subconjunto deverá cobrir todo o domínio da aplicação. No caso do sistema Edite, como a base de dados em causa guarda informação sobre recursos turísticos, interessam todas as perguntas que alguém possa querer fazer sobre turismo.

Repare-se que a delimitação do subconjunto da linguagem a tratar facilita o desenvolvimento do sistema, pois apenas se torna necessário dominar uma parcela da língua.

Lembrando que o objectivo deste sistema é permitir que um utilizador possa extrair informações de uma base de dados (e não acrescentar), parece bastante natural que se entenda um acesso por parte do utilizador

como uma pergunta. Assim, interessa que o sistema compreenda as frases interrogativas ou imperativas que exprimam um pedido. É, aliás, esta a motivação para a eliminação dos caracteres de pontuação (com excepção da vírgula e do ponto e vírgula). Como apenas se admitem perguntas, a frase afirmativa *O Ritz tem piscina.* será tratada do mesmo modo que a interrogação *O Ritz tem piscina?.*

A gramática usada pelo sistema Edite compreende maioritariamente frases simples (por exemplo, *Indique-me os hotéis de 4 estrelas de Marinhais, Que campos de golfe têm mais de 18 buracos? O Tivoli tem sauna?*), aceitando, no entanto, algumas construções compostas. Assim, as orações coordenadas não são permitidas, apesar de se estudar a conjunção coordenada copulativa *e* e a conjunção coordenada disjuntiva *ou*, quando se encontram a ligar palavras ou sintagmas, *i.e.*, quando não estão a iniciar uma oração coordenada (por exemplo, a frase *Indique-me um hotel com bar e piscina* é permitida, mas não o é a frase *Indique-me um hotel com bar e indique-me um hotel com piscina*). Quanto às orações subordinadas, as relativas, interrogativas e infinitivas são suportadas pela gramática do Edite.

Depois de assente qual o subconjunto da linguagem que se quer disponível, há que decidir quais são as palavras que cobrem esse subconjunto, ou seja, há que definir o vocabulário do Edite. Repare-se que palavras como *fechadura* e *avó* podem não constar no vocabulário do sistema, mas nomes comuns como *hotel* ou *piscina* têm de lá estar. Também alguns verbos como *ter, indicar* e *alugar* ocupam certamente um lugar no vocabulário do Edite, bem como nomes próprios que identifiquem algum recurso: *Jerónimos, Tejo, Lisbo*a, etc.

ARQUITECTURA DO SISTEMA EDITE

A arquitectura do sistema Edite é uma arquitectura "clássica", que se apresenta na figura que se segue. Cada etapa do sistema é descrita ao longo deste capítulo.

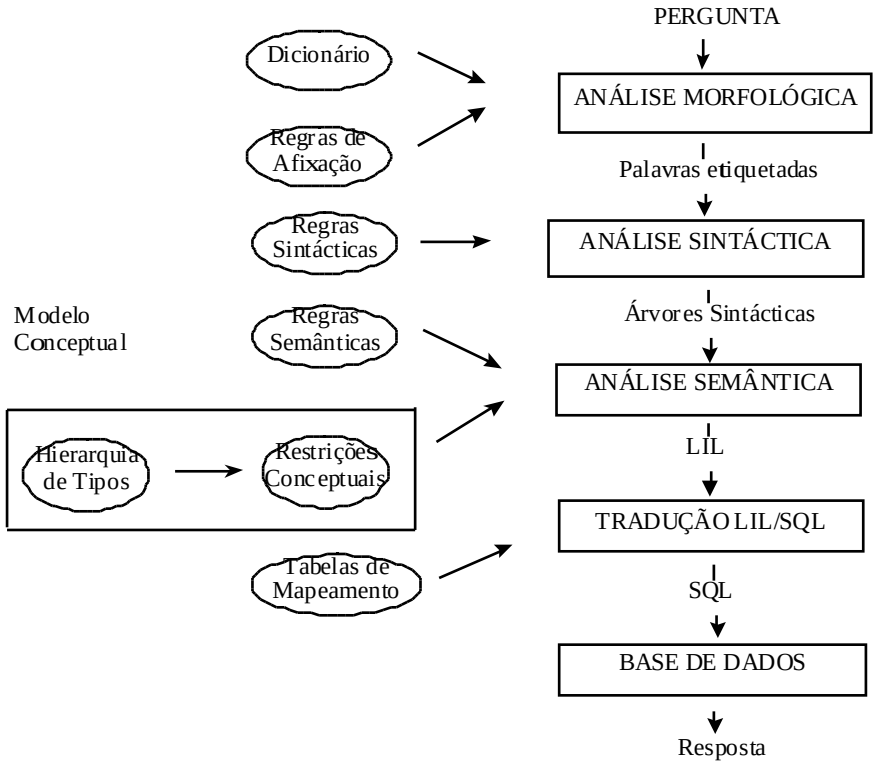

Figura 0.1– Arquitectura do sistema Edite

ANÁLISE MORFOLÓGICA

Apesar do domínio de linguagem do Edite restringir bastante o espaço linguístico, continua a ser demasiado vasto para um dicionário e, deste modo, o sistema utiliza o analisador morfológico Jspell. O Jspell é uma ferramenta desenvolvida na Universidade do Minho ([Almeida & Pinto-94], [Almeida & Pinto–95]), e baseada no Ispell (um corrector ortográfico do UNIX); apoia-se num dicionário de Português (mais de 35000 palavras) e num ficheiro de afixos que contém um conjunto de regras para a formação de palavras a partir de palavras da mesma família. Essas regras também se dizem regras de afixação, pois

uma palavra pode ser formada a partir de outra por acrescento de letras à cabeça desta (regras de prefixação) ou no fim (regras de sufixação). O facto do Jspell já ter um dicionário de Português foi a principal razão para a sua escolha, apesar de não se usar, na totalidade, o dicionário do Jspell, pois o domínio do discurso do Edite é pequeno. Deste modo, a procura torna-se muito mais rápida e ainda se evita que, em caso de erro por parte do utilizador, o número de sugestões de correcção seja muito grande.

Na Figura 0.2 pode-se observar o resultado simplificado da análise morfológica à frase *O Ritz tem piscina?*.

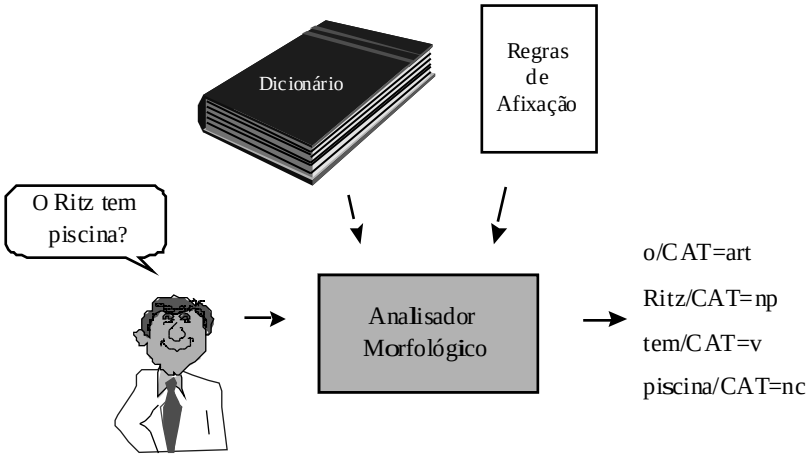

Figura 0.2 – Análise Morfológica

Descreve-se de seguida, e em traços largos, o funcionamento do Jspell, que se reflecte nas seguintes fases:

- Identificação das palavras da frase: esta etapa passa pelas definições que se encontram no ficheiro de afixos (que caracteres são letras, podendo deste modo fazer parte das palavras, que caracteres são separadores, etc.), bem como por certas opções (aceitam-se, ou não, palavras concatenadas, etc.). Assim, de acordo com o ficheiro de afixos, pode-se considerar, por exemplo, *1000$* como uma palavra (caso se tenha em conta que o carácter $ pode fazer parte de uma palavra), ou não, e, neste último caso, a palavra que o Jspell

identifica é *1000*. Também como exemplo, de acordo com outra opção do Jspell, este pode identificar, ou não, em *suitenupcial* as palavras *suite* e *nupcial*.

- Validação da palavra: quando acaba o processo de identificação das palavras, o Jspell é usado para descobrir se essas palavras existem. Começa por aplicar as regras de sufixação/prefixação "ao contrário". Para que se perceba o que quer dizer "aplicar as regras ao contrário", tome-se um exemplo: considere-se a palavra *amavelmente.* O Jspell vai procurar todas as regras de sufixação que acrescentam os sufixos que se podem obter da palavra (*amavelmente, mavelmente, avelmente, velmente,* etc.…). No ficheiro de afixos, encontram-se várias regras que o fazem, mas destacam-se duas. A primeira, a regra m, gera um advérbio a partir de um adjectivo, retirando à palavra a terminação *ável* e acrescentando-lhe o sufixo *avelmente* (- *ável* + *avelmente*). Com esta regra aplicada, "ao contrário" à palavra *amavelmente* (- *avelmente* + *ável*), gera-se a palavra *amável.* A outra regra, a regra n, transforma um verbo no infinitivo num adjectivo, substituindo o sufixo *ir* pelo sufixo *ente* (- *ir* + *ente*). Do mesmo modo, gera-se a palavra *amavelmir* (- *ente* + *ir*). De seguida, procuram-se as formas obtidas no dicionário: a palavra *amável* existe e a palavra *amavelmir* não. Falta ainda observar se a regra m se encontra como uma regra que pode ser aplicada à palavra *amável.* Em caso afirmativo, o Jspell devolve a classificação da palavra *amavelmente* (*vd.* Figura 0.3).

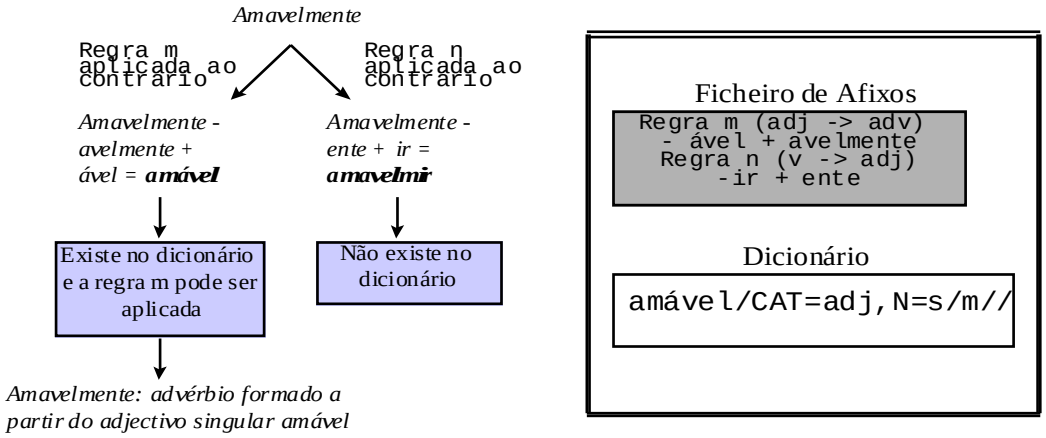

Figura 0.3 – Validação da palavra *amavelmente*

No fim da análise morfológica, se esta for bem sucedida, cada palavra da frase de entrada terá a(s) sua(s) categoria(s) morfológica(s) associada(s). No entanto, a análise morfológica nem sempre é bem sucedida, podendo falhar por duas razões:

- a palavra tem um erro ortográfico, mas a sua forma corrigida é conhecida pelo sistema;
- a palavra é desconhecida.

Em ambos os casos haverá uma tentativa de tratar o potencial erro, mas enquanto que no primeiro caso é possível que se descubra a palavra correcta, confrontando-se o utilizador com o seu erro, no caso da palavra ser desconhecida, como a palavra simplesmente não faz parte do vocabulário do Edite, a análise morfológica é mal sucedida. Por exemplo, em *Algave* o Edite será capaz de reconhecer *Algarve*, mas a palavra *hjfjgstfwet* já provocaria um insucesso morfológico, pois o analisador não seria capaz de obter uma sugestão para uma palavra tão "disforme".

Um conceito importante que surge na análise morfológica e que vem resolver inúmeros problemas é o conceito de termo composto, que não tem qualquer significado sob o ponto de vista linguístico. Um termo

composto, no contexto do Edite, é qualquer sequência de palavras que interesse agrupar (por exemplo, *Vila-Real-de-Santo-António*, *Linda-a-*Velha, mais-*de*, *mais-do-que*; etc.). A importância dos termos compostos deve-se ao facto de permitirem o tratamento de um grupo de palavras como uma única, aligeirando-se assim, não apenas o processo sintáctico (por exemplo, a sequência *suite-nupcial* será tratada como um nome comum), como o semântico (por exemplo, a semântica de *dez-mil-e-quinhentos-escudos* será 10500$).

No caso de surgirem dois termos compostos que começam na mesma palavra, selecciona-se o maior. Por exemplo, em *Vila Real de Santo António* toma-se o termo composto *Vila-Real-de-Santo-António*, e não *Vila-Real*.

ANÁLISE SINTÁCTICA

Depois de se terem decidido quais as palavras que constarão no dicionário, chega-se à fase em que se decide como é que essas palavras se podem combinar para formar frases, com base na sua categoria morfológica. Ou seja, define-se a estrutura sintáctica das frases que o utilizador poderá usar. Como já se disse, interessa que o Edite "compreenda" frases interrogativas ou imperativas que exprimam um pedido, expressas ou numa frase simples, ou numa frase composta através de orações subordinadas relativas, interrogativas ou infinitivas.

Entra-se, então, na análise sintáctica, etapa em que, com base no algoritmo de Earley ([Moll et al.–88],[Fonseca–93],[Reichwein–93]) e numa gramática (onde se guardam as regras sintácticas), se geram uma ou mais árvores sintácticas, contendo a estrutura sintáctica da frase em estudo. Isto se a frase fornecida pelo utilizador for sintacticamente correcta, pois caso a gramática não suporte a frase, nenhuma árvore é criada.

Na Figura 0.4 pode-se observar uma árvore (simplificada) gerada durante a análise sintáctica da frase *O Ritz tem piscina?*.

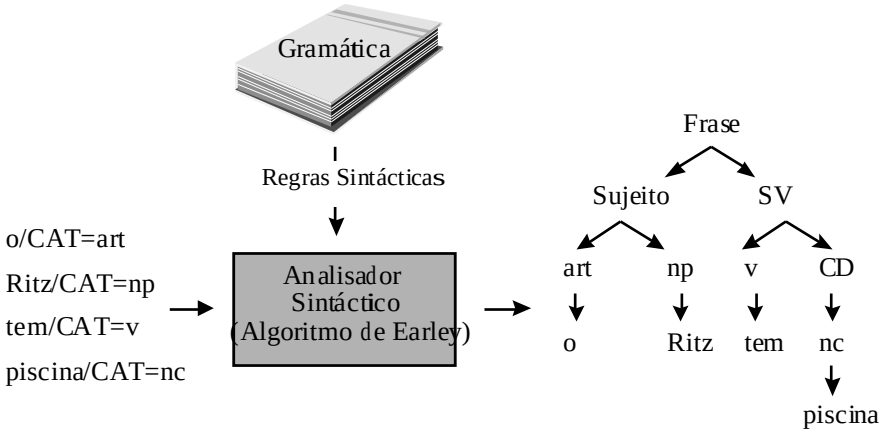

Figura 0.4 – Análise Sintáctica

Como um sistema robusto deve ser capaz de compreender frases mal formadas sempre que possível [Allen–95], a gramática do Edite não é uma gramática exacta para o Português. Em particular, algumas frases telegráficas[3] são sintacticamente aceites.

ANÁLISE SEMÂNTICA

Decidiu-se que este sistema de processamento de Linguagem Natural usaria uma linguagem de representação intermédia para expressar o significado das frases em estudo; intermédia, pois situa-se entre a frase escrita em LN e a linguagem com que é feito o acesso à base de dados.

Segundo [Allen–95], as linguagens de representação, para serem competentes, têm de ter as seguintes propriedades:

- a representação tem de ser precisa e não pode ser ambígua;
- a representação deve captar a estrutura intuitiva das frases em linguagem natural.

[3] Designam-se por telegráficas as frases demasiado sintécticas. Geralmente são falsas frases, *i.e.* sem verbo. Como exemplo, tome-se *hotéis com piscina*.

Chama-se análise semântica à etapa do sistema em que se transforma a estrutura em árvore obtida durante a análise sintáctica, numa forma lógica que representa o significado, independente de contexto, da frase. À linguagem usada chamou-se LIL (Linguagem Intermédia Lógica).

O facto da representação lógica ser independente de contexto leva a que a análise semântica, só por si, não reprove frases como *Indique-me um hotel com meias.* Na realidade, a essa frase será atribuída uma representação em LIL, e a sua análise semântica apenas falhará porque durante esta fase são feitas consultas ao chamado Modelo Conceptual (secção 0).

A maioria das palavras do dicionário têm uma semântica associada (por exemplo, a semântica de *apartamento-com-dois-quartos* é T3, a semântica de *instrutor* é a mesma de *professor*, ou seja, é professor). No entanto, há algumas palavras, ou mesmo sequências de palavras, que são semânticamente desprezáveis, pois não contribuem com informação relevante. Estes elementos são eliminados (por exemplo, os elementos a negrito das frases seguintes são desprezados: *Indique-**me** a direcção do hotel Estoril-Sol*, ***Eu queria saber** onde fica o Museu de Arte Antiga*, *Onde **é que se pode** comer um bom bacalhau com natas?*).

Usando de novo o exemplo anterior, uma aproximação da expressão LIL, resultante da análise semântica da frase *O Ritz tem piscina?,* pode ser vista na Figura 0.5.

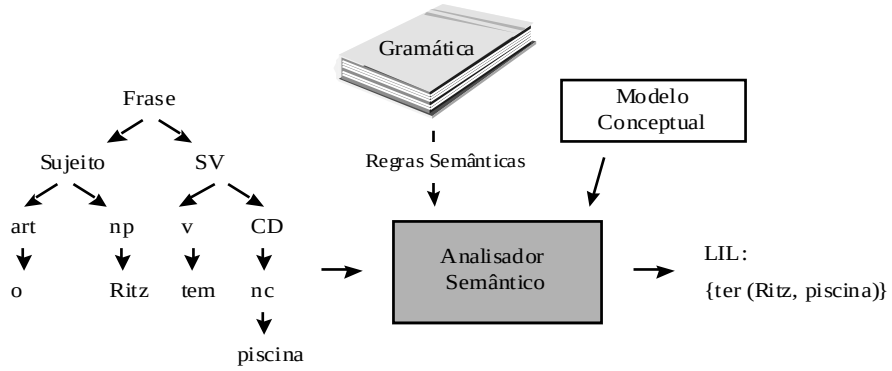

Figura 0.5 – Análise Semântica

MODELO CONCEPTUAL

O Modelo Conceptual [Reis & Mamede–96b] é consultado durante a análise semântica. No caso do pedido do *hotel com meias,* é o Modelo Conceptual que indicará que um hotel não pode ter meias, e provocará o insucesso da análise semântica da frase. Já o exemplo do *hotel com piscina* terá a aprovação do Modelo Conceptual, pois tem sentido dizer que um hotel tem (ou não) piscina.

O ACESSO À BASE DE DADOS

Se a análise semântica tiver sido bem sucedida, resta transformar a expressão em LIL que se obteve, numa fórmula em SQL (Structured Query Language), linguagem em que é feita a consulta à base de dados. Este processo de tradução é descrito em [Reis & Mamede–96a].

CONCLUSÃO

Um sistema como o Edite nunca estará concluído, não sendo só a complexidade da LN a contribuir para este facto: a evolução constante da língua, com novas palavras e novas expressões sempre a surgirem, obrigará a uma manutenção contínua do sistema.

O sistema Edite não é apenas um projecto teórico e já são muitos os tipos de perguntas a que este sistema responde, avizinhando-se a data em que

passará a ser livremente utilizado – não imediatamente pelo grande público, mas por utilizadores da *Internet*. Deste modo, vão-se descobrir as palavras e expressões mais usadas, os erros mais frequentes e a informação mais requisitada. Dessa experiência – onde o Edite revelará as suas fraquezas – espera-se obter muita informação, pois as perguntas não respondidas serão armazenadas e, numa actualização do sistema, a informação que permite responder a essa perguntas incorporada no sistema. Espera-se, assim, que o sistema se torne cada vez mais competente e útil.

BIBLIOGRAFIA

James **Allen**

1995 *Natural Language Understanding*
2ª edição
The Benjamin/Cummings Publishing Company, INC.

José João Dias de **Almeida** e Ulisses **Pinto**

1994 *Jspell – um Módulo para Análise Léxica Genérica de Linguagem Natural*

1995 *Manual do Utilizador do Jspell*

Ana Paiva **Fonseca**

1993 Comunicação em Linguagem Natural para um Tutor Inteligente
Tese de mestrado
Instituto Superior Técnico – Universidade Técnica de Lisboa

Robert N. **Moll**, Michael A. **Arbib**, A. J. **Kfoury**

1988 *An Introduction to Formal Language Theory*
Spring-Verlag

Georg **Reichwein**

1993 Sintaxe e Semântica de Linguagens 2". Apontamentos da cadeira de Sintaxe e Semântica da Linguagem 2, do 4º ano do curso de Matemática Aplicada e Computação, Secção de Ciência da Computação, Departamento de Matemática – Instituto Superior Técnico

Paulo **Reis**, Nuno **Mamede**

em prep. LIL-SQL – Processamento de Interrogações LIL por tradução para SQL". Relatório Técnico. Grupo de Sistemas e Serviços Telemáticos, INESC

1996 Modelo Conceptual. A Hierarquia de Tipos
Relatório Técnico. Grupo de Sistemas e Serviços Telemáticos, INESC

D. M. **Ritchie**, K. **Thompson**

1974 The Unix Time-Sharing System" in *Communications of the ACM* 17 (7). Julho