INSTITUTO SUPERIOR TÉCNICO
Universidade Técnica de Lisboa

Já Te Digo — Uma interface em língua natural para uma base de dados de cinema

Ana Raquel de Oliveira Guimarães

Dissertação para obtenção do Grau de Mestre em
Engenharia Informática e de Computadores

Júri

Presidente:	Doutor Ernesto José Marques Morgado
Orientador:	Doutora Maria Luísa Torres Ribeiro Marques da Silva Coheur
Co-orientador:	Doutor Nuno João Neves Mamede
Vogal:	Doutor Porfírio Pena Filipe

Setembro 2007

Agradecimentos

Ao meu João, pela compreensão face a fins-de-semana e férias que não pudemos passar juntos por causa da tese. Por *private jokes* como "Nesta meia-hora que estás a ver televisão, podias ter acabado a tese" que, invariavelmente, me faziam rir mesmo quando a minha disposição estava algures pelos meus calcanhares.

À minha colega, pelo sentido de humor cáustico e sempre oportuno, pelos almoços, pelo café matinal na Cidália, pela companhia no turco, pela partilha do seu dicionário mental de sinónimos, pela disponibilidade para alinhar num suicídio colectivo nos momentos mais deprimentes, por uma empatia inigualável que, por vezes, inclui a telepatia, pela capacidade trocadilheira, pelo apoio em momentos difíceis, por 6 anos que teriam sido penosos sem a sua companhia. "Amigas não, colegas..." .

Aos meus pais, por terem disfarçado decentemente a sua ansiedade face à entrega da tese. Pronto, agora já está...! ◡̈

À Luisinha (nome pelo qual eu e a colega a tratamos secretamente), pela constante disponibilidade quer para partilhar conhecimento a nível académico, quer para partilhar conhecimento sobre a vida e o Mundo. Aquando da primeira "entrevista" fiquei com a sensação que seria muito bem orientada, num ambiente desprovido de qualquer formalidade ou constrangimento e recheado de entreajuda (o verdadeiro ambiente académico, digo eu). As minhas expectactivas já eram altas e mesmo assim foram superadas. Pela paciência face à minha expressão de "Oh não!" a cada correcção feita. Agradeço cada vírgula adicionada ◡̈ !

Ao Professor Nuno Mamede, pela motivação para esta área científica já desde o semestre em que frequentei a cadeira de Língua Natural. Pela capacidade de tentar sempre ajudar, mesmo que não concorde com o que se está a fazer: "Eu não estou a dizer mal, mas...". Não só fui bem orientada, como também fui bem co-orientada. Tenho hoje perfeita consciência que fui contemplada com o 1º prémio da lotaria dos orientadores científicos.

Ao Professor David Matos, pela quantidade de vezes em que cheguei àquele gabinete começando com "Tenho um problema..." (invariavelmente retorquido com "Interessante...") e saí com uma solução.

Ao grupo formado pela colega, João, Luís, Telmo e Rolo (numa fase inicial) que facilitou o "arranque" da tese.

A todos os L^2Fianos por me terem acolhido nesta casa, pelos bolos, chocolates e gelados que, embora tenham arruinado a dieta, foram extremamente importantes para ganhar forças e voltar ao trabalho "consoladinha".

Ao Marco Fernandes do Cinema PTGate por ter facultado os títulos em Português sem qualquer contrapartida, numa altura em que já tinha batido a diversas portas sem quaisquer resultados.

Ao George Michael, e, numa fase final em que o meu vocabulário estava entrar em falência técnica, ao Jorge Palma pela inspiração. (Sim, é possível gostar dos dois, pois cada um tem a sua função na minha vida.)

Ao Instituto Superior Técnico por não dar azo a saudades da vida de estudante. *Things can only get better now...*

A todos os que não agradeci convenientemente, ou não agradeci sequer, as minhas desculpas: a minha capacidade de verbalizar o meu pensamento está, nesta altura, a dar as últimas...

Lisboa, 17 de Novembro de 2007

Ana Raquel Guimarães

Aos meus pais e ao meu irmão.

It is better to know some of the
questions than all of the answers.

Resumo

As interfaces em língua natural para bases de dados já são desenvolvidas desde os anos 60 e têm como principais vantagens a expressividade, fácil utilização e capacidade de incorporar figuras de estilo tais como anáfora e elipse. Os sistemas desenvolvidos têm por base diversas abordagens, existindo assim os sistemas de emparelhamento, os sistemas baseados em sintaxe, os sistemas baseados em semântica e os sistemas que recorrem a uma linguagem de representação intermédia.

O JáTeDigo é uma interface em língua natural, em Português, para uma base de dados aplicada ao domínio de cinema. Os dados sobre cinema recolhidos provêm de diferentes fontes, tendo sido principalmente obtidos através do IMDB — *Internet Movie Database*. Na base de dados de cinema figuram 1 502 517 nomes de pessoas e 672 048 títulos de filmes resultantes do processamento de diversos ficheiros de texto de elevada dimensão. Adicionou-se ainda informação relativa aos Óscares da Academia, bem como cerca de 5000 títulos em Português.

A arquitectura da aplicação baseia-se em quatro etapas principais: reconhecimento de entidades mencionadas, desambiguação, processamento de língua natural e, finalmente, acesso à base de dados. Na primeira fase são reconhecidos os títulos de filmes e nomes de pessoas que estão presentes na questão. Seguidamente, no caso de haver em base de dados mais que uma entidade com o mesmo nome (diversos filmes com o mesmo título, diversas pessoas com o mesmo nome), é solicitado ao utilizador a sua desambiguação. Concluídas as duas anteriores fases, a questão é submetida a uma cadeia de processamento de língua natural. Se a questão formulada emparelhar com um dos padrões sintácticos definidos nessa cadeia, é escolhido o *script* adequado para obter a resposta à questão através do acesso à base de dados.

Para determinar a eficácia da aplicação foi concebida uma interface *Web* e foram realizados vários testes com diversos utilizadores. Num dos testes efectuados, em que figuravam exemplos de questões, a aplicação foi capaz de responder a 66% das questões com uma taxa de sucesso de 87,9%. Num outro teste, em que a interface era somente composta por uma caixa de texto, a aplicação respondeu a 40% das questões, sendo que 90% das respostas estavam correctas. Verificou-se assim que a existência de exemplos de questões na interface com o utilizador tem impacto no seu desempenho, sendo que a percentagem de questões respondidas é bastante superior para o caso de figurarem esses exemplos.

Abstract

The natural language interfaces for databases (NLIDB) are being developed since 1960. Their main advantages are: expressivity, high usability and ability to process anaphora and ellipsis. Regarding the architecture, there are different types of NLIDB: pattern-matching systems, syntax-based systems, semantic-based systems and systems based on intermediate representation languages.

The JáTeDigo system is a natural language interface developed in Portuguese for a cinema database. The cinema data was obtained through several sources, mainly through IMDB — Internet Movie Database. There are 1 502 517 person names and 672 048 film titles in the database that were inserted by processing several big-sized plain text files. It was also added data regarding the Academy Awards (Oscars) and also about 5000 titles in Portuguese.

The architecture consists of four main steps: named entity recognition, desambiguation, natural language processing and, finally, database querying. In the first step, film titles and person names are identified within the input question. Following, if there is an entity repeated in the database (more than one film with the same title or more than one person with the same name), the user is asked to desambiguate. When the two previous steps are finished, the question is submited to a natural language processing chain. If the input question matches one of the syntactic patterns defined in the chain, the script that will handle the database access is given as output.

To evaluate the system's performance several tests with users were made. In one of the tests where there were examples of questions handled by the system, 66% of the questions were answered with a success rate of 87,9%. In another test, where the interface was only a text box, 40% of the questions were answered with a 90% success rate. It was concluded that giving examples of questions handled by the system had influence in its performance given that the percentage of questions answered was greater when those questions where available in the interface.

Palavras Chave
Keywords

Palavras Chave

Língua Natural

Interpretação

Cinema

Ambiguidade

Interface

Informação

Keywords

Natural Language

Interpretation

Cinema

Ambiguity

Interface

Information

Índice

Lista de Figuras

Lista de Tabelas

1 Introdução

Os sistemas de pergunta-resposta (*question-answering* em Inglês, abreviado para *QA*) em língua natural têm sido alvo de estudo por parte da comunidade científica desde a década de 60. O objectivo é bastante simples: dada uma questão em língua natural formulada pelo utilizador, providenciar a resposta ao utilizador pelo mesmo meio. Esta abordagem acaba por "humanizar" a interacção pessoa-máquina uma vez que permite que o utilizador comunique com um computador da mesma forma que comunicaria com um humano.

As interfaces em língua natural para bases de dados (designadas doravante por ILNBD) constituem um domínio particular dos sistemas pergunta-resposta. A distinção reside no formato dos dados — nas ILNBD, estes estão, como o próprio nome indica, dispostos numa base de dados. Nesse sentido, a língua natural constitui uma "linguagem" alternativa de interacção com base de dados, sendo esta, indiscutivelmente, a forma mais *user-friendly* de interagir com uma base de dados.

Neste documento, é descrito o desenvolvimento de uma interface em língua natural para uma base de dados de cinema[1].

1.1 Motivação

Na década de 80, as interfaces em língua natural eram bastante populares, tendo, na altura, sido desenvolvidos bastantes sistemas deste género. No entanto, a sua popularidade decaiu, não havendo grandes desenvolvimentos nesta área nos últimos anos. O aparecimento de motores de pesquisa na Internet (como o muito popular *Google*) são, de alguma forma, responsáveis pelo abandono nesta área. No entanto, embora a pesquisa por palavras-chave (*keywords*) acabe por satisfazer algumas procuras rápida e facilmente, por vezes os utilizadores perdem-se na consulta a diversas páginas, só obtendo a informação pretendida ao fim de alguma tentativa-erro.

Considere-se, por exemplo, a questão: "Que actor entra nos filmes Magnolia e Top Gun?". Esta questão pretende saber o(s) actor(es) que entra(m) em ambos os filmes. Se o utilizador utilizar o conhecido "Google" para responder a esta questão, consegue fazê-lo logo à primeira tentiva com as palavras-chave "actor", "Magnolia" e "Top Gun". Tal acontece por diversos motivos:

[1] Disponível em `http://www.l2f.inesc-id.pt/~arog` em Setembro de 2007

- Tom Cruise é dos actores mais conhecidos, havendo inúmeras páginas que o referem juntamente com ambos os títulos;
- Estão a ser utilizados os títulos originais dos filmes;
- A palavra "actor" tem o mesmo significado em Português e Inglês.

Continuando no mesmo género de questão, veja-se "Que actriz entra em As Horas e Magnolia?". Para responder a esta questão, recorrer-se-ia às palavras-chave "actriz, As Horas e Magnolia", contudo, desta vez, na primeira página de respostas, não se conseguiria encontrar a resposta.

De salientar que este tipo de pesquisa não devolve somente a resposta à questão, devolve sim, por ordem de relevância, documentos onde as palavras-chave foram encontradas — cabe ao utilizador encontrar, nesses documentos, a resposta à sua questão. Pelo contrário, nos sistemas de QA (em que se incluem as ILNBD), o utilizador apenas tem que formular a questão de acordo com a informação que pretende obter e aguardar a resposta exacta à sua questão (e não textos onde terá que procurar pela resposta). Note-se que não está em causa a viabilidade da pesquisa por palavras-chave enquanto método de pesquisa, mas sim a sua usabilidade, especialmente para os menos acostumados com a utilização de computadores e *Internet*.

Para além das vantagens de uma ILNBD face a uma pesquisa por palavras-chave, as ILNBD trazem igualmente alguns benefícios aquando da consulta a bases de dados. Segue-se uma análise de algumas interfaces para bases de dados retiradas de (Androutsopoulos et al., 1995).

O método mais popular de consulta a base de dados tem sido o SQL, linguagem formal mais utilizada. Tome-se, como exemplo, que as tabelas 1.1 e 1.2 estão guardadas numa base de dados relacional.

tabela_funcionarios

funcionario	departamento	telefone
Alberto Silva	Marketing	22334455
Mário Gomes	Vendas	11223344

Tabela 1.1: Informação sobre os funcionários.

tabela_departamentos

departamento	chefe	cidade
Vendas	Luís Sousa	Viseu
Contabilidade	António Cunha	Guarda

Tabela 1.2: Informação sobre os departamentos.

Para gerar uma lista que mostre o chefe de cada funcionário usar-se-ia a seguinte instrução SQL:

```
SELECT tabela_funcionarios.funcionario, tabela_departamentos.chefe
FROM tabela_funcionarios, tabela_departamentos WHERE
tabela_funcionarios.departamento=tabela_departamentos.departamento
```

A interrogação acima requer à base de dados todos os pares existentes que consistam num valor para *funcionario* e para *chefe*, em que o valor para *funcionario* provenha de uma entrada em *tabela_funcionarios*, o valor para *chefe* provenha de uma entrada em *tabela_departamentos* e as duas entradas tenham o mesmo valor para *departamento*.

Por outro lado, nas interfaces baseadas em formulários, são usados formulários pré-definidos que contêm campos a serem preenchidos. O utilizador preenche a informação nos campos respectivos e o sistema completa os campos restantes com base na informação contida na base de dados. Se um utilizador quiser saber quem é o chefe do Mário Gomes terá que preencher o formulário como indicado na tabela 1.3.

Informação sobre funcionários	
Funcionário:	Mário Gomes
Departamento:	
Telefone:	
Chefe:	

Tabela 1.3: Formulário preenchido pelo utilizador.

O sistema responde preenchendo os campos em falta como é visível na tabela 1.4.

Informação sobre funcionários	
Funcionário:	Mário Gomes
Departamento:	Vendas
Telefone:	11223344
Chefe:	Luís Sousa

Tabela 1.4: Formulário preenchido pelo sistema.

Se houver mais que uma resposta possível (mais que uma pessoa com o mesmo nome), o sistema gera uma lista de formulários preenchidos, em que cada formulário é uma possível resposta.

A interrogação por exemplo é mais um método de consulta a base de dados. O utilizador combina um determinado número de formulários em que cada um reflecte a estrutura da tabela na base de dados. A interrogação acima seria representada como está visível nas tabelas 1.5 e 1.6.

A informação na tabela 1.5 diz ao sistema para seleccionar as entradas na tabela de funcionários cujo nome seja Mário Gomes. As duas ocorrências da variável *X* dizem que cada entrada seleccionada a partir da tabela de funcionários deve ser agrupada com as entradas provenientes da tabela de departamentos que têm o mesmo valor para a coluna *departamento*. A entrada *P.Y* diz que o *chefe* correspondente à interrogação deve ser devolvido ao utilizador.

form_tabela_funcionarios

funcionario	departamento	telefone
Mário Gomes	*X*	

Tabela 1.5: Formulário para funcionários.

form_tabela_departamentos

departamento	chefe	cidade
X	*P.Y*	

Tabela 1.6: Formulário para departamentos.

Relativamente às interfaces gráficas, o utilizador começa por especificar as tabelas das base de dados a serem usadas (no exemplo da figura 1.1, *tabela_funcionarios* e *tabela_departamentos*). Seguidamente, o utilizador preenche os diversos atributos e liga os atributos entre tabelas com o uso do rato. Para saber quem é o chefe de Mário Gomes, o utilizador realiza a interrogação representada na figura 1.1.

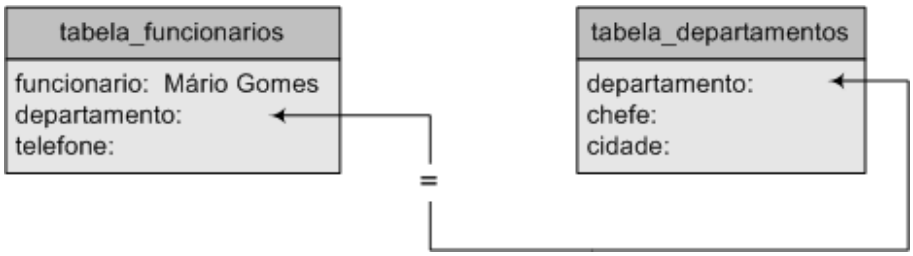

Figura 1.1: Esquema de uma *query* gráfica.

Em conclusão, as ILNBD destacam-se grandemente das outras interfaces pela sua fácil utilização. É difícil imaginar algo mais linear do que utilizar a própria língua para comunicar com uma base de dados: quando se quer saber uma dada informação, apenas se faz uma pergunta, tal como se faria se se estivesse a comunicar com uma pessoa. Com a possibilidade de usar a sua própria língua, o utilizador não necessita de aprender uma nova linguagem para poder interagir com a base de dados. As linguagens formais para manipulação de base de dados não são intuitivas para todos, muito menos para pessoas com pouca formação ao nível da Informática. As interfaces gráficas e as interfaces baseadas em formulários são mais acessíveis a utilizadores ocasionais, mas, mesmo assim, obrigam o utilizador a aprender como interagir com os formulários, como ligar tabelas, entre outras tarefas. Dada a sua fácil utilização, podemos classificar as ILNBD como sendo aplicações ideais para utilizadores ocasionais.

Destaca-se também a expressividade inerente à língua natural: há determinadas questões que não são facilmente representáveis através de linguagens formais, interfaces baseadas em formulários ou interfaces gráficas, sendo, no entanto, intuitivamente representadas em linguagem natural. Especificamente, as questões que envolvem quantificação e negação são as mais complexas. Por exemplo, as questões "Que departamentos não têm contabilistas?" e "Que departamentos têm igual número de receitas?" não são impossíveis de representar em linguagens formais, como o SQL, mas exigem expressões complexas cuja escrita não é acessível ao comum dos utilizadores.

Algumas ILNBD conseguem ainda responder a perguntas contendo anáforas[2] e elipses[3]. Este tipo de questões conseguem ser interpretadas porque se recorre ao contexto em que estão inseridas. Nas restantes interfaces analisadas, o recurso ao contexto não é possível.

1.2 Arquitectura

Nesta secção descreve-se a arquitectura da aplicação que, a partir de uma pergunta formulada em Português, obtém a sua resposta na base de dados. Os detalhes técnicos da arquitectura, bem como o seu

[2]Repetição sistemática da mesma palavra no princípio de diferentes frases ou de membros da mesma frase.
[3]Figura de sintaxe que consiste na omissão de uma ou mais expressões numa frase que podem, contudo, ser facilmente subentendidas.

esquema, estão disponíveis no apêndice A.

1.2.1 Visão Geral

A arquitectura está dividida em quatro fases:

- Reconhecimento de entidades mencionadas;
- Desambiguação;
- Processamento de língua natural;
- Acesso à base de dados.

O reconhecimento de entidades mencionadas é realizado através de interrogações *full-text* e será analisado detalhadamente na secção 4.2. As entidades que se pretendem reconhecer são os títulos de filmes e os nomes de pessoas.

A interface da aplicação é uma página *Web*: o utilizador escreve a sua questão sobre cinema e pressiona o botão "Submeter Pergunta". Consoante os nomes de pessoas e/ou títulos de filmes identificados na questão, poderá haver uma fase de desambiguação. Se houver mais que um filme com o título mencionado, o utilizador escolherá um deles com base no ano de estreia e elenco principal. Se houver mais que uma pessoa com o mesmo nome, o utilizador escolhe uma delas com base nos filmes em que participou.

Após a fase de reconhecimento de entidades e possível desambiguação, a questão é submetida a uma cadeia de processamento de língua natural em que, com base no padrão de questão encontrado, é escolhido o *script* que irá à base de dados obter a informação e pretendida.

1.2.2 Exemplo

Para uma melhor compreensão da arquitectura da aplicação, vai-se, a partir da questão "Em que filmes Meg Ryan contracena com Tom Hanks?", explicar os diversos passos desde a questão formulada pelo utilizador, até à resposta dada pela aplicação.

Nesta questão, seriam identificadas duas entidades do tipo "pessoa": "Meg Ryan" e "Tom Cruise". De seguida, verificar-se-ia se existe na base de dados mais que uma entidade desde tipo, ou seja, se existe mais que uma "Meg Ryan" e mais que um "Tom Cruise". Uma vez que são únicos na base de dados, passa-se ao registo dos seus *ID's* num ficheiro que depois será pesquisado na fase de consulta à base de dados. Abaixo, encontra-se o conteúdo desse ficheiro para este exemplo concreto.

```
tom cruise_person: 662253
meg ryan_person: 392405
```

A adição do sufixo *person* é importante para os casos em que existe uma entidade que é filme e pessoa simultaneamente. Por exemplo, existe o actor "Amadeus" e o filme "Amadeus" e, se houvesse uma questão que envolvesse essa entidade, teria que figurar no ficheiro o seguinte:

```
amadeus_person: 517345
amadeus_film: 28101
```

Voltando à questão "Em que filmes Meg Ryan contracena com Tom Hanks?", após a escrita dos *ID's*, segue-se a escrita da gramática local num ficheiro que posteriormente será usado pela cadeia de processamento da língua natural. De seguida, encontram-se as regras originadas pelo reconhecimento de entidades mencionadas nesta questão:

```
1> noun[actor=+] = ?[surface:tom], ?[surface:cruise].
1> noun[actriz=+] = ?[surface:meg], ?[surface:ryan].
```

A análise morfo-sintáctica efectuada durante o processamento de língua natural desta questão encontra a dependência *target_which_films_main_act_two* que preenche com os argumentos "Meg Ryan" e "Tom Cruise". No ficheiro XML que constitui a saída da cadeia de processamento de língua natural, figura o seguinte nó:

```
<DEPENDENCY name="TARGET_WHICH_FILMS_MAIN_ACT_TWO">
        <PARAMETER ind="0" num="23" word="meg ryan"/>
        <PARAMETER ind="1" num="24" word="tom cruise"/>
</DEPENDENCY>
```

O processamento do ficheiro XML por intermédio do XSLT dá origem à seguinte expressão:

```
get_from_BD/script-which-films-main-act-two.pl ACTOR 'meg ryan' ACTOR 'tom cruise'
```

Esta expressão constitui o *script* que será invocado juntamente com os seus argumentos. A execução deste *script* com estes argumentos realiza o acesso à base de dados e devolve o seguinte:

```
Meg Ryan e Tom Cruise contracenaram em:

Top Gun (1986)
```

Este bloco de texto é devolvido, sendo posteriormente exibido ao utilizador como resposta à pergunta efectuada, como é possível ver na imagem 1.2:

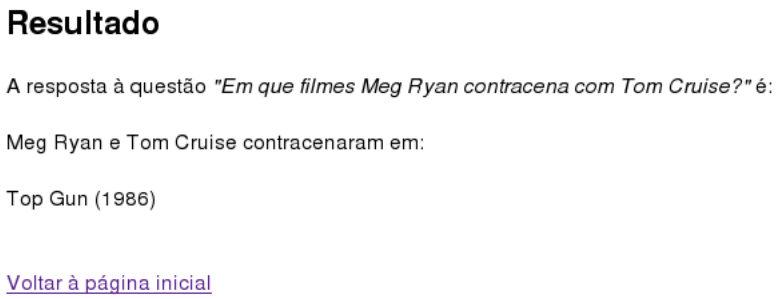

Figura 1.2: Ecrã com o resultado da questão "Em que filmes Meg Ryan contracena com Tom Hanks?"

1.2.3 Interface homem-máquina

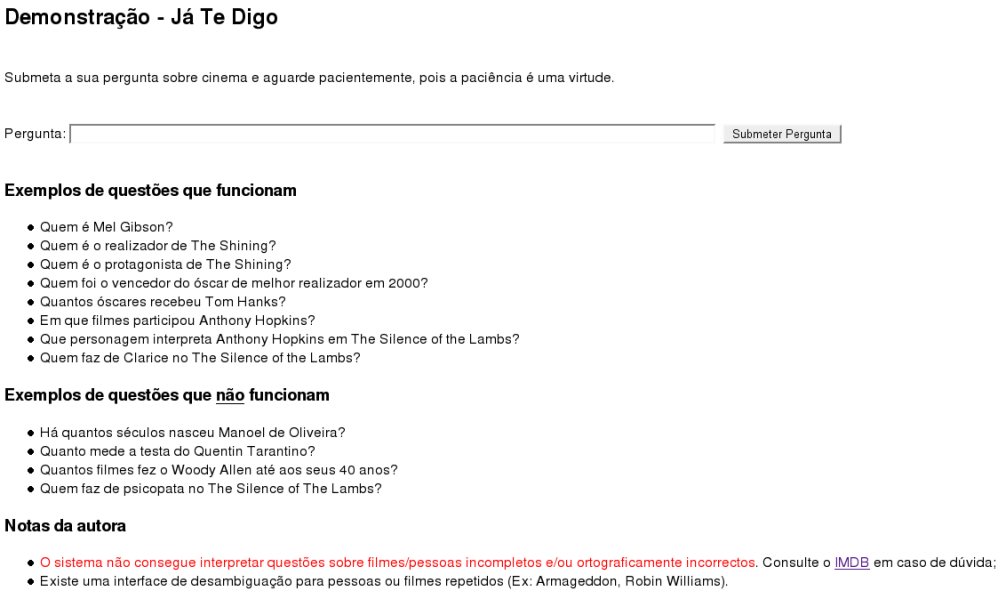

Figura 1.3: Interface homem-máquina para a aplicação.

Como é visível na figura 1.3, a interface com o sistema é uma página *Web* com um formulário que recebe a questão do utilizador e retorna uma resposta. Esta interface é semelhante à do sistema START[4], no sentido em que, na página inicial, figuram alguns exemplos de questões que o sistema consegue responder para que o utilizador perceba melhor o seu funcionamento. Como acrescento, figuram exemplos de questões que são, dada a sua complexidade, virtualmente impossíveis de responder. Desta forma, o utilizador pode melhor compreender as limitações da aplicação.

Os títulos e nomes de pessoas não são únicos na base de dados, podendo haver mais que um filme com o mesmo título e mais que uma pessoa com o mesmo nome. Se, na questão, figurar uma dessas entidades, é solicitada a escolha de uma delas com base em dados adicionais sobre cada uma para

[4]Disponível a 27 de Setembro de 2007 em `http://start.csail.mit.edu/`

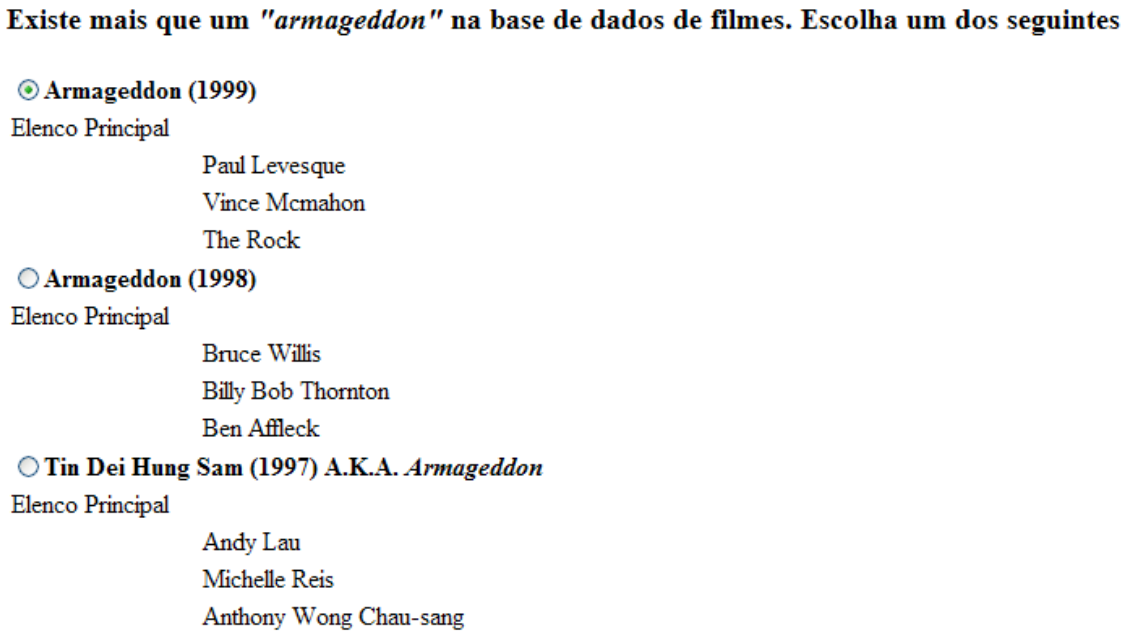

Figura 1.4: Ecrã de desambiguação para a questão "Quem é o realizador de Armageddon?"

facilitar a escolha. Por exemplo, se o utilizador questionasse o sistema sobre: "Quem é o realizador de Armageddon?", surgiria um ecrã com os diversos "Armageddons", o seu ano de estreia e elenco principal (figura 1.4).

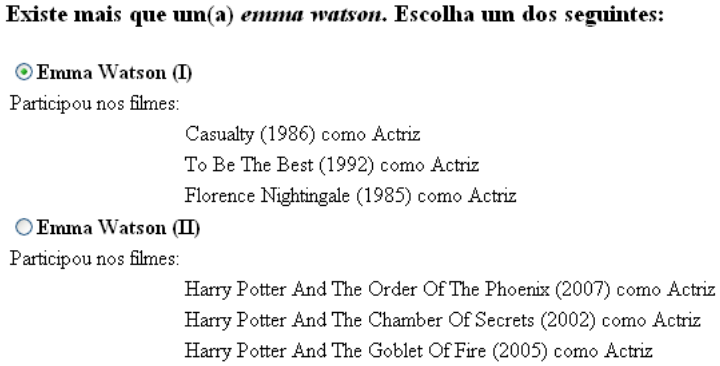

Figura 1.5: Ecrã de desambiguação para a questão "Em que filmes entra Emma Watson?".

A desambiguação é efectuada também ao nível dos nomes de pessoas. Dessa forma, se a questão for: "Em que filmes entra Emma Watson?", será mostrado um ecrã com as duas "Emmas Watson" existentes na base de dados e os filmes em que entraram para que o utilizador opte por aquela a que se refere (figura 1.5). É importante verificar que, neste caso, a "Emma Watson" mais popular é a *II*, reforçando a necessidade de desambiguação neste caso. Este exemplo prova que não se pode assumir que a pessoa mais popular é a primeira (*I*) a figurar na base de dados.

Se não houver necessidade de desambiguação, surge logo a resposta à questão formulada. Se o sis-

tema não compreender a questão (por não reconhecer as entidades ou por, simplesmente, não conseguir tratar aquele tipo de questão), informa que não a compreendeu e mostra as entidades reconhecidas. É ainda sugerida a consulta ao sítio do IMDB para verificar se os títulos de filmes e nomes de pessoas presentes na questão estão correctamente escritos.

1.3 Estrutura do documento

No capítulo 2 é analisada a evolução das ILNBD, desde os anos 60 até aos dias de hoje. São vistas as diversas abordagens existentes para o desenvolvimento de uma interface deste género, bem como as arquitecturas de algumas aplicações.

No capítulo 3 descrevem-se os passos necessários para construir a base de dados que está subjacente a esta aplicação.

O método utilizado para interpretar a questão formulada pelo utilizador é detalhadamente analisado no capítulo 4.

Para se determinar a eficácia da aplicação desenvolvida, procederam-se a diversos testes com utilizadores. A análise dos resultados obtidos é efectuada no capítulo 5.

No capítulo 6 é feito um sumário sobre a aplicação desenvolvida e analisado o trabalho futuro a desenvolver. É ainda visto a sua contribuição no panorama actual das ILNBD.

2 Estado da Arte

2.1 Introdução

Na secção 2.2 será feita uma perspectiva história sobre as ILNBD desde a sua génese até aos dias de hoje. De seguida, na secção 2.3, será analisada a sua problemática. O capítulo será finalizado na secção 2.4 com uma perspectiva das diferentes abordagens que podem ser seguidas para o desenvolvimento de uma ILNBD.

2.2 Perspectiva histórica

Os primeiros protótipos de ILNBD surgiram entre o final dos anos 60 e início dos anos 70. Segundo Hirchman (Hirschman & Gaizauskas, 2001), um dos primeiros sistemas foi o BASEBALL (Green, 1961) cujo propósito era responder a questões em língua natural sobre, justamente, *Baseball*. Contudo, o sistema LUNAR (W.A.Woods, 1972), uma interface em língua natural para uma base de dados de análises químicas de pedras lunares, foi o primeiro marco nesta área e acabou por influenciar bastante os sistemas subsequentes. Destacam-se quatro componentes segundo (R. Weischedel, 2006):

- analisador morfológico;
- analisador sintáctico (baseado em ATN — *Augmented Transition Networks*[1]);
- linguagem de representação de conhecimento que permite a representação do significado das questões;
- base de Dados com informação de Química.

A gramática utilizada pela analisador sintáctico é composta por diversas categorias gramaticais: artigo, nome, adjectivo, preposição, entre outras. É também usado um dicionário para determinar certos atributos (plural, tempo verbal, etc.). Os sintagmas identificados (sintagma nominal, verbal, preposicional) são processados pelo analisador semântico que determina se são ou não relevantes. Caso não sejam, são descartados.

[1]Tipo de grafo usado para definição de linguagens formais com ênfase para a análise de língua natural.

O sistema foi informalmente apresentado na *Second Annual Lunar Science Conference* em 1971. De acordo com (Hancox, n.d.), o seu desempenho era notável: em 1977 conseguiu responder correctamente a 78% das questões, subindo posteriormente para 90% após alguns melhoramentos. No entanto, estes valores acabam por ser ilusórios face ao verdadeiro desempenho do sistema. Na realidade, o LUNAR não tinha sido até então usado intensivamente. Posteriormente, chegou-se à conclusão que o sistema era pouco flexível devido ao uso do analisador baseado em ATN: apesar de ser muito eficiente (mesmo para grandes gramáticas), não conseguia responder bem a frases gramaticalmente incorrectas.

Outros protótipos foram surgindo nos anos 70 como é o caso do RENDEZVOUS (E.F.Codd, 1974). Este sistema caracteriza-se pela interacção constante com o utilizador como forma de obter mais informação sobre a questão que lhe estava a ser colocada. Assim, o sistema funciona como uma espécie de assistente que ajuda o utilizador a melhor formular a sua questão.

Os protótipos dessa altura foram concebidos para domínios específicos, logo caracterizam-se pela falta de modularidade e pouca portabilidade. Era indispensável o desenvolvimento de uma interface que fosse independente do SGBD — Sistema de Gestão de Base de Dados.

O LADDER (Hendrix et al., 1978) surgiu em 1978 com o objectivo de lidar com bases de dados de grande volume que, por isso, poderiam estar dispersas por vários computadores com diferentes SGBD, chegando a ser aplicado no sistema de base de dados da Marinha Norte-Americana. Este sistema não se limita a fazer a análise sintáctica da questão, existe também um processamento semântico que melhora os resultados. Contudo, apesar da inquestionável melhoria dos resultados, o sistema acaba por ser dependente do domínio a que é aplicado, apresentando uma portabilidade ainda muito limitada.

No fim dos anos setenta surgiram também os sistemas PLANES (Waltz, 1978) — Programmed LANguage-based Enquiry System — e PHILIPQA1 (Scha, 1977).

O processamento de uma questão pelo sistema PLANES está dividido em quatro fases (Waltz, 1978): análise, geração da interrogação, avaliação e resposta.

- Análise — É feito uma correspondência entre a entrada e alguns padrões pré-determinados. Os sintagmas em que é obtida correspondência são transformados em conjuntos de constituintes semânticos;
- Geração da interrogação — os conjuntos de constituintes semânticos são traduzidos para um interrogação em linguagem formal por forma a ser gerada a informação necessária para responder à questão do utilizador;
- Avaliação — nesta fase são usadas as expressões em linguagem formal geradas no fase anterior para encontrar a resposta na base de dados;

- Resposta — a informação proveniente da base de dados é passada para o gerador de respostas. A resposta pode ser disponibilizada em três formas diferentes: um simples número ou lista, um gráfico ou uma tabela.

A década de 80 constitui o período áureo das ILNBD. Nessa altura foram produzidos bastantes protótipos em que o grande objectivo era garantir a sua portabilidade. O sistema CHAT-80 (Warren & Pereira, 1982) acabou por ser dos ILNBD mais populares da década. É inteiramente implementado em Prolog e limita-se a transformar as perguntas em cláusulas que depois são avaliadas na base de dados também construída em Prolog. O código deste sistema serviu de inspiração a outras ILNBD como é o exemplo do MASQUE (Auxerre & Inder, 1986). O CHAT-80 foi também adaptado para a língua portuguesa (Lopes, 1984). O sistema LUSO corresponde a uma "tradução" do CHAT-80, mas com uma maior base de conhecimento (foi adicionado um módulo sobre Geografia) e um dicionário em Português.

Como já foi referido, o modo de operar do MASQUE é muito semelhante ao do CHAT-80 e apresenta tanto eficiência como portabilidade. Questões complexas são respondidas em poucos segundos e, nos casos em que o sistema não consegue responder, procura obter ajuda por parte do utilizador. Este sistema está restringido a operar sobre bases de dados em Prolog, não podendo por isso ser aplicado a bases de dados relacionais (que acabam por ser as mais populares e comercializadas).

O sistema TEAM (Grosz, 1983) — Transportable English Access Data Manager — surgiu em 1983 com o objectivo de ser facilmente configurável por administradores de bases de dados, dispensando detalhes sobre a implementação da ILNBD. Este sistema acaba por se destacar dos anteriores sistemas (LUNAR, LADDER, PLANES) por privilegiar a portabilidade em detrimento dos resultados inerentes a uma exploração efectiva do domínio da base de dados. Para esse efeito, TEAM separa a informação relativa à *linguagem*, da informação relativa ao *domínio* e ainda da informação relativa à *base de dados*.

O ASK (Thompson & Thompson, 1983) surge na mesma altura, embora com outra orientação. Destaca-se a capacidade de aprendizagem do sistema pela interactividade com os utilizadores, permitindo a introdução de novas palavras e novos conceitos. Este sistema permite a interacção com múltiplas bases de dados (sendo por isso independente do domínio) e ainda com programas de *e-mail*.

Em 1984 surgiu o PRE (Epstein, 1985) — Purposefully Restricted English — que foi concebido para ser um SGBD que conseguisse lidar com a utilização simultânea por parte de diversos utilzadores (concorrência) e com bases de dados de grandes dimensões. Ao contrário de outros sistemas cujo propósito é conseguir interpretar "tudo", o PRE tem uma abordagem minimalista que estabelece limites relativamente à forma como os utilizadores colocam as suas questões. Desta forma, consegue ser bastante eficiente e também flexível, facilitando a portabilidade entre domínios.

O JANUS (R. M. Weischedel, 1989) surgiu no final dos anos 80 e à semelhança do ASK, também permite a interacção com múltiplas aplicações (bases de dados, dispositivos gráficos, etc). O utilizador

consegue interagir com as diversas aplicações sem se aperceber da especificidade das camadas mais baixas. Este é também um dos poucos sistemas que consegue responder a questões temporais.

Outros sistemas marcaram década de 80 como o DATALOG (Hafner & Godden, 1985), EUFID (Burger, 1980), LDC (Ballard et al., 1984), TQA (Damerau et al., 1982) e TELI (Ballard, 1986). Sugere-se a consulta da bibliografia para mais informação sobre os sistemas referidos.

No início dos anos 90, mais precisamente em 1991, surge o SQUIRREL que está dividido em duas grandes secções: uma recebe a pergunta em língua natural e produz representações sintácticas e semânticas que depois mapeia para Lógica de Primeira Ordem; a outra traduz a representação lógica em SQL. Todas as representações são independentes do domínio da aplicação e do modelo da base de dados, o que lhe confere muita portabilidade.

Dois anos depois, surge o MASQUE/SQL (Androutsopoulos et al., 1993), uma versão modificada do MASQUE que pode ser usada em qualquer SGBD baseado em SQL. Este sistema gera instruções SQL a partir das perguntas e executa-as sobre a base de dados para obter as respostas. Tal como o sistema original, é facilmente configurável para alternar entre domínios diferentes.

Em Dezembro de 1993 surge o START (Katz & Lin, 2002) — SynTactic Analysis using Reversible Transformations — o primeiro sistema baseado em língua natural de *question-answering* disponível na *World Wide Web*. A primeira versão deste sistema continha apenas informação sobre os membros do *MIT Artificial Intelligence Laboratory* e seus trabalhos. Desde então, a base de conhecimento foi actualizada e foram construídas novas bases de conhecimentos. Actualmente, consegue responder a perguntas sobre vários domínios: Geografia, Ciência, Arte e Entretenimento, História e Cultura. É constituído por dois módulos que partilham a mesma gramática: o módulo de compreensão que processa texto em Inglês e produz uma base de conhecimento que contém a informação encontrada no texto; o módulo gerador que produz frases em Inglês de acordo com a informação retirada da base de conhecimento.

Ainda durante a década de 90 surgiram sistemas que foram comercializados como sendo o ELF (ELF *software Co.*, 1999) composto pelos módulos *English Query* e *English Wizard* que visavam gerar sistemas de processamento de língua natural "no momento", ou seja, de modo a que as novas bases de dados pudessem ser consultadas com um sistema adequado. Para além do ELF, outros sistemas foram comercializados:

- INTELLECT (Harris, 1988) pela empresa Trinzic que data de 1984 e vem no contexto do sistema ROBOT (Harris, 1977, 1978);
- PARLANCE (Systems & Technologies, 1989) da BBN decorrente do trabalho desenvolvido nos sistemas RUS (Bobrow, 1978) — Render Unto Syntax — e IRUS (Bates et al., 1986);
- LANGUAGEACESS (Sanamrad, 1992) da IBM, deixou de ser comercializado a Outubro de 1992;

- Q&A da Symantec;
- NATURAL LANGUAGE (Inc., 1986) da Natural Language Inc.;
- LOQUI (Technology, 1991) da IBM;

Sugere-se a consulta das referências bibliográficas para obter mais informação sobre os sistemas acima mencionados.

Ao nível da língua portuguesa destaca-se o sistema EDITE (Filipe, 1999; Marques, 1996), concebido com o objectivo de consultar uma base de dados de recursos turísticos. Em traços gerais, a arquitectura deste sistema é baseada numa sequência de processos (análise morfológica, análise sintáctica, análise semântica e tradução) a que a questão em língua natural é submetida. O resultado da análise morfológica é validado pelo analisador sintáctico. Posteriormente, é feita uma validação semântica, ou seja, é verificado se a pergunta se ajusta ao conceito subjacente à base de dados (neste caso, recursos turísticos). Se a interpretação semântica da pergunta for validada, é gerada uma interrogação na linguagem de representação intermédia designada por linguagem de interpretação lógica (LIL). Finalmente é realizada a tradução da interrogação em LIL para SQL gerando como resultado uma interrogação SQL.

Já no século XXI surgiu o sistema STEP (Minock, 2004) — Schema Tuple Expression Processor — que foi construído a partir da linguagem LISP. A sua arquitectura assemelha-se muito com a do sistema RENDEZVOUS (E.F.Codd, 1974). A grande diferença relativamente ao trabalho efectuado nesta área é o facto de restringir a representação das questões e respostas a *schema tuple expressions*, ou seja, a representação está limitada a conter uma única variável livre e quantificação existencial, admitindo também negação. A abordagem do STEP acaba por ser semelhante à de um problema de procura em espaço de estados: o estado inicial é a questão em língua natural e o estado final é a interrogação à qual corresponde. Os estados intermédios contêm uma interrogação que traduz uma parte da questão bem como o resto da questão em si. À semelhança do que acontece com o START (Katz & Lin, 2002), o sistema está disponível *on-line* para responder a perguntas sobre Geografia.

O sistema ORAKEL (Cimiano, 2004) aparece pouco depois e a sua motivação deriva da crescente importância dos SGBD orientados a objectos. Em linhas gerais, usa uma abordagem a nível semântico semelhante à existente no sistema JANUS (R. M. Weischedel, 1989) conseguindo responder a questões envolvendo quantificação, conjunção e negação. Por outro lado, o sistema gera automaticamente o domínio lexical para conseguir interpretar a pergunta efectuada. Nesse sentido, difere do TEAM (Grosz, 1983) em que o léxico era desenvolvido pelo administrador da base de dados ou, no caso do RENDEZVOUS (E.F.Codd, 1974) em que era construído incrementalmente pela interacção com o utilizador.

O NALIX (Li, 2005) é um dos mais recentes desenvolvimentos nesta área e distingue-se dos anteriores por estar orientado para uma base de dados em XML.

Recentemente, tem havido interesse por parte da indústria dos telemóveis nas ILNBD, nomeadamente por parte da Nokia que, em colaboração com investigadores do MIT, está a tentar implementar o conceito subjacente ao sistema START nos sistema operativo dos seus telemóveis (Bourzac, n.d.).

Na figura 2.1 encontra-se a cronologia dos sistemas referidos nesta secção.

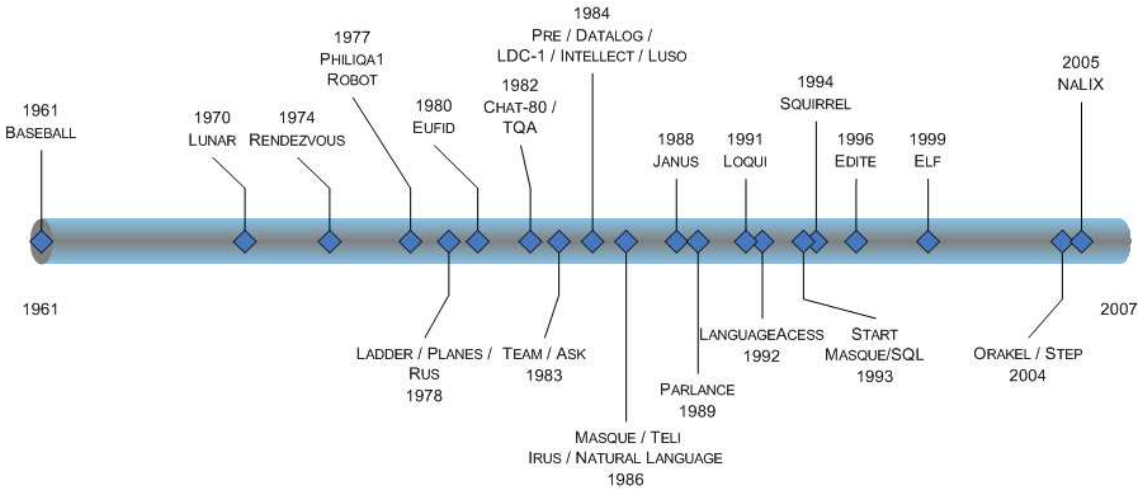

Figura 2.1: Cronologia das interfaces em língua natural para bases de dados.

2.3 Problemática

As ILNBD são, do ponto de vista teórico, interfaces "perfeitas" para interacção com bases de dados. Contudo, também acarretam alguns problemas face a outras interfaces, o que acaba por destruir essa utopia. Na secção 2.3.1 são analisadas alguns dos problemas inerentes à utilização destas interfaces.

O processamento da língua natural não é uma tarefa trivial pois a língua natural é rica em ambiguidade quer a nível sintáctico, quer a nível semântico. Para além disso, a língua natural contempla o recurso a figuras de estilo como a anáfora ou a elipse, que acabam por tornar a sua interpretação ainda mais complexa. Na secção 2.3.2 são analisados os principais obstáculos à correcta interpretação das questões em língua natural.

2.3.1 Dificuldades de utilização

Um dos problemas frequentemente associados às ILNBD é a obscuridade relativamente às suas capacidades linguísticas. O utilizador tem tendência a humanizar o sistema, pensando que, por aceitar perguntas em língua natural, pode "conversar" com o sistema. Os filmes de ficção científica acabam por reforçar esta tendência, uma vez que fazem uso da língua natural de forma irrealista (ou não fosse

ficção científica). Com toda esta perspectiva "fantástica", os utilizadores acabam por ficar decepcionados quando se apercebem que o sistema "apenas" consegue responder a questões directas.

Ralativamente às linguagens formais, interfaces baseadas em formulários e interfaces gráficas não há qualquer ilusão. As linguagens formais estão normalmente bem documentadas e qualquer erro sintáctico pode ser resolvido através da consulta da documentação. No caso das interfaces baseadas em formulários e interfaces gráficas, o utilizador consegue discernir que tipo de perguntas pode efectuar consultando as opções disponíveis na interface.

Quando uma ILNBD não responde a uma questão, o utilizador não consegue distinguir se se trata de uma limitação linguística ou conceptual. Por vezes, os utilizadores tentam reformular as suas questões sem se aperceberem que o problema não reside na interpretação da questão, mas sim na informação que o sistema tem. Por exemplo, se utilizador tentar questionar uma base de dados de recursos turísticos sobre a localização de farmácias, o sistema não lhe conseguirá responder. Noutros casos, o utilizador não tenta sequer reformular uma questão que o sistema consegue responder conceptualmente, mas que linguisticamente não conseguiu interpretar. A não obtenção de resposta é uma grande frustração para o utilizador que, perante essa situação, pode deduzir que o sistema é "burro" e desistir da interacção. Algumas ILNBD tentam resolver este problema dando mensagens elucidativas sobre a falha do sistema — mensagens de diagnóstico.

Existem dúvidas relativamente à adequação da língua natural enquanto intermediário na interacção com um sistema informático (JL Binot, 1991). É argumentado que a língua natural é demasiado rica e ambígua para a interacção homem-máquina. Os utilizadores das ILNBD precisam de escrever questões longas enquanto que nas interfaces baseadas em formulários basta preencher os campos necessários e nas interfaces gráficas basta usar o rato. As questões em língua natural são frequentemente ambíguas enquanto que em questões formulados em línguas formais, interfaces gráficas ou interfaces baseadas em formulários, não há múltiplos significados.

2.3.2 Obstáculos à interpretação das questões

A língua natural tem propriedades que dificultam o desenvolvimento de interfaces. Analisam-se de seguida algumas dessas propriedades com particular relevância para exemplos relacionados com as ILNBD.

2.3.2.1 Ambiguidade

Consideremos a questão "Quais os funcionários do departamento de Informática que têm carta de condução?". Em termos linguísticos, tanto *do departamento de Informática* como *que têm carta de condução*

são modificadores, isto é, modificam o significado dos outros sintagmas. Um humano rapidamente percebe que *do departamento de Informática* se refere a *funcionários* enquanto que *que têm carta de condução* se refere a *funcionários do departamento de Informática*. O sistema poderia associar erroneamente o modificador *carta de condução* a *departamento de Informática* deturpando o sentido da questão.

Para que o sistema consiga interpretar correctamente as questões efectuadas tem que ter conhecimento sobre o domínio da aplicação. Neste exemplo, podemos constatar que a ambiguidade da língua natural é um grande obstáculo para as ILNBD.

Alguns sistemas usam heurísticas para tentar resolver ambiguidades. Por exemplo, o sistema PRE (Epstein, 1985) usa o conceito de "associação mais provável à direita", ou seja, assume-se que os modificadores afectam o sintagma mais à direita. Para a questão "Quais são os funcionários contratados por um recrutador cujo salário é maior que 2000 euros?", é assumido que o salário se refere ao recrutador e não aos funcionários. Existem contudo outras técnicas de desambiguação (Whittemore et al., 1990):

- Anexação mínima — a anexação é feita de modo a que sejam empregues o número mínimo de regras sintácticas;
- Preferência lexical — os sintagmas modificadores são associados aos verbos que os costumam acompanhar. Por exemplo, o verbo "viver" está normalmente associado ao sintagma proposicional de local (como "em Lisboa").

Outra abordagem possível para resolver este problema, é a seguida pelo sistema EDITE (Marques, 1996) que recorre ao modelo conceptual para determinar à correcta associação do sintagma modificador. Como exemplo, imagine-se o pedido "Indique-me um hotel com piscina que tenha suite nupcial" — há duas interpretações possíveis:

- O sintagma modificador "que tenha suite nupcial" refere-se ao hotel;
- O sintagma modificador "que tenha suite nupcial" refere-se à piscina.

O modelo conceptual implementado "sabe" que os hotéis é que têm suite nupcial e não as piscinas, sendo por isso fácil determinar a interpretação correcta. Nos casos em que não é possível discernir o verdadeiro intuito da pergunta, pode-se optar por mostrar as interpretações possíveis e solicitar a escolha de uma.

2.3.2.2 Conjunção e Disjunção

Por vezes, é difícil perceber se uma determinada questão está formulada em termos de disjunção ou de conjunção. A palavra "e" é muitas vezes utilizada para disjuntar e não para conjugar como, à partida,

parece óbvio. Este factor introduz um tipo de ambiguidade que é difícil de resolver. Os seguintes exemplos retirados de (Templeton & Burger, 1983) ilustram esta problemática.

- "Quais os clientes que habitam em Lisboa e Coimbra?".

Qualquer humano interpreta facilmente que se pretendem os clientes que vivem em Lisboa *ou* Coimbra, uma vez que dificilmente residem em duas cidades simultaneamente. Contudo, o sistema pode não conseguir deduzir essa informação pois o conceito de não viver simultaneamente em dois lugares é conhecimento semântico.

- "Que programadores sabem Cobol e Fortran?"

E difícil, até para um humano, saber se se pretende os que sabem ambas as linguagens, ou se, pelo contrário, a pergunta se pode desdobrar em "Que programadores sabem Cobol e que programadores sabem Fortran?".

Há duas abordagens possíveis para resolver este problema: uma é dar conhecimento semântico ao sistema de forma a que ele perceba quando um "e" é efectivamente conjuntivo, outra é dar a resposta para as duas situações (conjunção e disjunção).

2.3.2.3 Anáfora

A utilização de anáforas é bastante frequente em língua natural e representa uma grande dificuldade para o seu processamento. É comum a utilização de pronomes como "ele/ela", "seu/sua", "dele/dela", "ali/aqui" e muitos outros no nosso dia-a-dia. Relativamente às ILNBD, este fenómeno é problemático pois obriga à manutenção de contexto para conseguir resolver as referências pronominais. Segue-se um exemplo retirado do sistema ASK para ilustrar o problema.

>Is there a ship whose destination is unknown?
`Yes`
>What is it?
`What is [the ship whose destination is unknown]?`
`Saratoga`

O sistema conseguiu determinar que o pronome "it" se referia a "whose destination is unknown". Para que o utilizador não seja induzido em erro, é mostrado explicitamente a dedução que foi feita.

O LOQUI (Technology, 1991) também consegue resolver a anáfora, como exemplo segue-se um pequeno diálogo com o sistema:

>Who leads TPI?

`E. Feron`

>Who reports to him?

`C. Leonard, C. Willems, E. Bidonet, P. Cayphas, J.P. Van Loo`

Uma das abordagens para lidar com a anáfora é manter uma listagem de todas as entidades mencionadas no diálogo. Quando é encontrado um pronome, é examinada essa listagem começando nas entradas mais recentes e associa-se o pronome à entidade mais recente que satisfaça as restrições de ordem gramatical e semântica. No caso da pergunta "Who reports to him" o sistema conseguiu associar "him" a "E. Feron" ao invés de associar a "TPI" — tal é possível porque o pronome é de género masculino enquanto que "TPI" é de género indefinido.

Outra abordagem possível, é o sistema pedir ao utilizador para, em caso de dúvida, indicar a quem se refere numa determinada questão. Segue-se um exemplo de um diálogo no sistema SQUIRREL retirado de (Barros & DeRoeck, 1994).

Who is Edna's boss?

`malcolm`

Who is Sylvia's boss?

`Edna`

Who works for her?

```
** USER: Please choose one substitute for the pronoun 'her':
1 - Sylvia
2 - Edna
3 - none above       number:
```

Como é visível, o sistema não consegue desambiguar a informação, tendo por isso que solicitar o esclarecimento por parte do utilizador. De notar que nem um humano conseguiria perceber se a questão se referia à Edna ou à Sylvia.

2.3.2.4 Uso da elipse

A elipse, tal como a anáfora, é usada de forma habitual no nosso discurso. Pode ser encontrada no habitual "Com ou sem açúcar?", quando se serve chá — embora o alvo do açúcar não esteja explícito na

frase, é facilmente subentendido. "No mar, tanta tormenta e tanto dano" — neste trecho d'Os Lusíadas também está presente a elipse pela omissão do verbo "haver".

Se por um lado, a anáfora pode não ser contemplada numa ILNBD pois o utilizador tendencialmente não necessita de a usar, já a elipse se revela de uma grande utilidade para estas interfaces. Imagine-se que seguinte pergunta era efectuada sobre uma base de dados sobre cinema:

- *Em que filmes é George Clooney o actor principal?*

Imagine-se que o utilizador pretende saber a mesma informação sobre outros actores, o "natural" seria perguntar logo de seguida:

- *E o Johnny Depp?*
- *E o Ethan Hawke?*

Se a ILNBD não contemplar elipse, o utilizador terá que escrever a questão na sua totalidade o que acaba por ser maçador. Para se evitar o processamento da elipse, pode-se facultar a edição de perguntas anteriormente realizadas. Para ser possível o processamento da elipse é necessária uma manutenção do contexto ao longo da interacção. Os PLANES e o IRUS são exemplo de sistemas que contemplam elipse.

2.3.2.5 Uso de abreviaturas e erros ortográficos

O processamento da língua natural já não é, por si só, uma tarefa trivial. A linguagem usada no dia-a-dia não é sintáctica e gramaticalmente correcta, bem pelo contrário, está recheada de erros . A Internet e os telemóveis acabaram por agravar essa tendência devido à proliferação de abreviaturas e palavras escritas de modo "xpexial".

Se o objectivo das ILNBD é ajudar o utilizador, então têm que estar preparadas para as suas limitações de ordem linguística. Uma abordagem possível é a utilização de um dicionário para conseguir corrigir os erros ortográficos. O sistema ASK usa essa abordagem: "What is the crago of the Orient Clipper" — é detectado o erro em "crago" e é corrigido para "cargo". O START e o EDITE também detectam erros e sugerem possíveis correcções:

What is the capitel of Portugal?

```
The word CAPITEL may be misspelled. Please choose one of the
following:
Capitol
```

```
Capital
Accept Word
Abort
```

2.4 Diferentes abordagens

Nesta secção pretende-se descrever sucintamente as abordagens mais comuns no desenvolvimentos de ILNBD. Optou-se por uma divisão de acordo com as técnicas de processamento da questão em língua natural, tais como descritas em (Androutsopoulos et al., 1995). Consideram-se os sistemas de emparelhamento, os sistemas baseados em sintaxe, os sistemas baseados em semântica e os que recorrem a linguagens de representação intermédia.

2.4.1 Sistemas de emparelhamento

Os primeiros sistemas baseavam-se em técnicas de emparelhamento para responder às questões dos utilizadores. Tome-se como exemplo a tabela 2.1.

TABELA_PAÍSES

PAÍS	CAPITAL	LÍNGUA
Alemanha	Berlim	Alemão
Espanha	Madrid	Espanhol
...	...	...

Tabela 2.1: Tabela de uma base de dados contendo informação sobre países.

Um sistema básico de emparelhamento pode conter regras como as que se seguem:

- `pattern: ... ''capital'' ... <country>`
- `action: Report CAPITAL of row where COUNTRY = <country>`
- `pattern: ... ''capital'' ... ''country''`
- `action: Report CAPITAL and COUNTRY of each row`

A primeira regra serve para tratar o caso em que a questão do utilizador contém a palavra "capital" seguida de um nome de um país. O sistema tenta encontrar a entrada da tabela que contém o nome do país e imprimir a sua capital. Se, por exemplo, a pergunta for "Qual é a capital da Alemanha?", o sistema usa a primeira regra de emparelhamento e responde "Berlim".

De acordo com a segunda regra, qualquer questão contendo a palavra "capital" seguida da palavra "país" deve devolver a capital de cada país existente na tabela. A questão "Qual é a capital de cada país?" é então tratada pela segunda regra.

A grande vantagem desta abordagem é a sua simplicidade. Para além disso, este tipo de sistemas consegue fornecer respostas aceitáveis mesmo quando a pergunta "foge" ao âmbito das regras. Por exemplo, a pergunta "É verdade que a capital de todos os países é Atenas?" seria respondida com uma listagem das capitais de todos os países, o que acaba por ser uma resposta indirecta à questão efectuada.

Os sistemas baseados em técnicas de emparelhamento têm um desempenho aceitável em algumas aplicações de linguagem controlada, ou seja, em que o domínio e a sintaxe das questões estão bem definidos. No entanto, acaba por falhar demasiado devido à sua superficialidade. Como exemplo tem-se a pergunta "A Alemanha tem que capital?": apesar de este tipo de formulação ser bastante frequente, dadas as regras já enunciadas, não consegue ser respondida por um destes sistemas.

2.4.2 Sistemas baseados em sintaxe

Nos sistemas baseados em sintaxe a questão do utilizador é analisada sintacticamente e o seu resultado é directamente mapeado numa linguagem de interrogação de base de dados (SQL, por exemplo).

Estes sistemas usam uma gramática que descreve as estruturas sintácticas que poderão figurar nas questões dos utilizadores. De seguida, tem-se o exemplo de uma gramática bastante simplificada para um sistema destes, baseado no LUNAR (W.A.Woods, 1972).

$$
\begin{aligned}
S &\rightarrow NP\ VP \\
NP &\rightarrow Det\ N \\
Det &\rightarrow \text{“what”} \mid \text{“which”} \\
N &\rightarrow \text{“rock”} \mid \text{“specimen”} \mid \text{“magnesium”} \mid \text{“radiation”} \mid \text{“light”} \\
VP &\rightarrow V\ \ N \\
V &\rightarrow \text{“contains”} \mid \text{“emits”}
\end{aligned}
$$

A gramática diz que uma frase, representada por S (*sentence*) consiste num NP (*noun phrase*) seguido de um VP (*verbal phrase*); um NP, por sua vez, consiste num determinante (*Det*) seguido de um nome (N); um determinante pode ser "what" ou "which"; o VP pode ser constituído por um verbo (V) e por um N.

Recorrendo a esta gramática, o sistema consegue construir a estrutura sintáctica da questão *"Which rock contains magnesium"* através de uma árvore de análise sintáctica como se pode ver na figura 2.2.

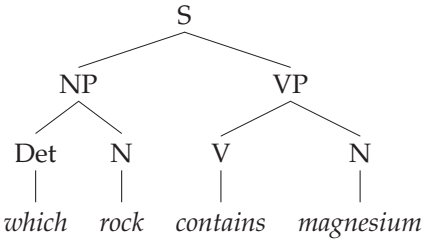

Figura 2.2: Árvore de análise sintáctica para a questão "Which rock contains magnesium?".

Posteriormente, o sistema mapeia a árvore de análise sintáctica na seguinte interrogação à base de dados:

```
(for_every X (is_rock X)
    (contains X magnesium)
    (printout X))
```

Este mapeamento é concretizado por regras e baseado na íntegra na informação sintáctica proveniente da gramática. Para o exemplo em questão, é deduzido que:

- O mapeamento de "which" é `for_every X`;
- O mapeamento de "rock" é `(is_rock X)`;
- O mapeamento de um *NP* é *Det' N'* em que *Det'* e *N'* são os mapeamentos do determinante e nome respectivamente. Desta forma, o mapeamento da sub-árvore *NP* é `for_every X (is_rock X)`;
- O mapeamento de "contains" é `contains`;
- O mapeamento de "magnesium" é `magnesium`;
- O mapeamento de um *VP* é *(V' X N')* em que *V'* é o mapeamento do verbo e *N'* o do complemento directo. O mapeamento de sub-árvore *VP* do exemplo é `(contains X magnesium)`;
- O mapeamento de *S* é *(NP' VP' (printout X))* em que *NP'* e *VP'* são os mapeamentos das árvores *NP* e *VP* respectivamente.

2.4.3 Sistemas baseados em semântica

Tal como nos sistemas baseados em sintaxe, nos sistemas baseados em semântica é feito um processamento da questão (*parsing*) e um mapeamento para uma interrogação à base de dados. A diferença reside nas categorias gramaticais usadas que, neste caso, podem não corresponder necessariamente a conceitos sintácticos.

A informação semântica sobre o domínio da base de dados pode ser explicitamente incluída nas regras gramaticais de modo a forçar restrições de ordem conceptual. Por exemplo, se tivermos a pergunta "Que carro conduz Steven Spielberg?" sobre o domínio de cinema, será identificado que o nome "carro" e o verbo "conduzir" não pertencem ao domínio da base de dados. No entanto, a pergunta "Que filmes realizou Steven Spielberg?" já será respondida, pois contém termos que pertencem ao domínio cinematográfico.

$$
\begin{aligned}
S &\rightarrow \textit{Specimen\_question} \mid \textit{Spacecraft\_question} \\
\textit{Specimen\_question} &\rightarrow \textit{Specimen\_spec Emits\_info} \mid \textit{Specimen\_spec Contains\_info} \\
\textit{Specimen\_spec} &\rightarrow \textit{“which rock”} \mid \textit{“which specimen”} \\
\textit{Emits\_info} &\rightarrow \textit{“emits” Radiation} \\
\textit{Radiation} &\rightarrow \textit{“radiation”} \mid \textit{“light”} \\
\textit{Contains\_info} &\rightarrow \textit{“contains” Substance} \\
\textit{Substance} &\rightarrow \textit{“magnesium”} \mid \textit{“calcium”} \\
\textit{Spacecraft\_question} &\rightarrow \textit{Spacecraft Depart\_info} \mid \textit{Spacecraft Arrive\_info} \\
\textit{Spacecraft} &\rightarrow \textit{“which vessel”} \mid \textit{“which spacecraft”} \\
\textit{Depart\_info} &\rightarrow \textit{“was launched on” Date} \mid \textit{“departed on” Date} \\
\textit{Arrive\_info} &\rightarrow \textit{“returns on” Date} \mid \textit{“arrives on” Date}
\end{aligned}
$$

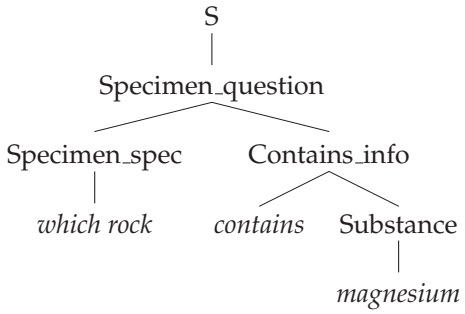

Figura 2.3: Árvore de análise semântica para a questão “Which rock contains magnesium?”.

Recordando o exemplo dado na secção anterior, tem-se na figura 2.3 a análise da mesma questão na figura 2.2, mas, desta vez, recorrendo à seguinte gramática semântica:

Contudo, uma vez que as gramáticas semânticas contêm informação explícita sobre o domínio das aplicações, torna-se difícil transportá-las entre aplicações devido à sua especificidade. Em contraste, as gramáticas puramente sintácticas podem ser aplicáveis em diferentes domínios.

O PLANES (Waltz, 1978), o LADDER (Hendrix et al., 1978) e o EUFID (Burger, 1980) são exemplo de sistemas que usam gramáticas semânticas.

2.4.4 Linguagens de representação intermédia

As ILNBD mais actuais transformam a questão em língua natural numa interrogação lógica intermédia. Esta linguagem intermédia tem como propósito representar a semântica da questão em termos de conceitos de alto nível que são independentes da estrutura da base de dados. A interrogação lógica é traduzida posteriormente para uma expressão na linguagem que manipula a base de dados. A grande vantagem deste tipo de abordagem é a separação entre a componente linguística e o domínio da base de dados.

Como exemplos deste tipo de arquitectura têm-se os sistemas MASQUE/SQL (Androutsopoulos et al., 1993), IRUS (Bates et al., 1986), TEAM (Grosz, 1983), LOQUI (Technology, 1991), JANUS (R. M. Weis-

chedel, 1989) e EDITE (Filipe, 1999; Marques, 1996). Na figura 2.4 encontra-se um esquema simplificado da arquitectura que engloba a abordagem acima descrita.

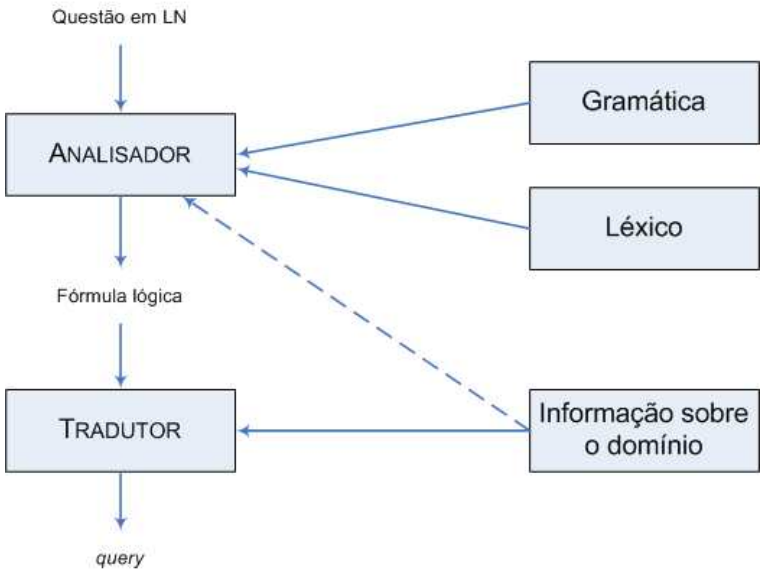

Figura 2.4: Arquitectura típica de um sistema que usa linguagem de representação intermédia.

Os detalhes da forma lógica produzida pelo analisador varia entre sistemas. Para que a linguagem de representação intermédia seja tão poderosa quanto uma linguagem formal para base de dados relacional, é essencial que haja forma de especificar quantificação universal e existencial. Muitas ILNBD usam uma linguagem de representação bastante mais poderosa que as linguagens formais de interrogação, pois a língua natural requer mais expressividade.

3 Constituição da BD

3.1 Introdução

A estruturação da base de dados constitui um passo importante para o sucesso da aplicação. Uma vez que se trata de um grande volume de informação, na ordem dos milhões de entradas para algumas tabelas, é importante que a ligação entre tabelas seja feita de forma a que as interrogações à base de dados devolvam resultados rapidamente. Nesse sentido, decidiu-se que o tempo máximo aceitável para a resposta a uma questão deveria ser cinco segundos.

O IMDB não contém apenas informação relativa a filmes, contemplando também séries televisivas, concursos, noticiários e outros programas televisivos. A aplicação está, no entanto, direccionada para a consulta de informação sobre cinema, pelo que a base de dados não engloba a totalidade da informação disponível no IMDB — *Internet Movie Database*. A informação recolhida visa ilustrar o funcionamento de uma ILNDB, não sendo seu objectivo facultar todos os dados disponíveis no IMDB.

3.2 Processamento da informação

3.2.1 Proveniência

Toda a informação proveniente do IMDB está disponível numa colecção de ficheiros de texto que pode ser obtida em diversos *sites* FTP cujos elos estão disponíveis na área *Interfaces*[1]. Os dados que figuram actualmente na base de dados datam de *30 de Julho de 2007* e foram obtidos no seguinte endereço FTP `ftp://ftp.fu-berlin.de`. Até à data mencionada, não figurava na listagem de ficheiros informação relativa aos Óscares obtidos por filmes e pessoas. Devido a este facto, foi necessário recorrer ao *Web Site* da Academia dos Óscares[2] para completar os dados recolhidos no IMDB.

Apesar da interface *Web* do IMDB permitir a procura por títulos em Português, essa informação não se encontra disponível nos ficheiros de texto disponibilizados. Verificou-se, em testes efectuados durante a fase de desenvolvimento, que essa informação era vital pois muitas questões não eram respondidas porque os utilizadores não sabiam escrever correctamente o título original do filme. Para além

[1] Disponível a 27 de Setembro de 2007 em `http://www.imdb.com/interfaces`.
[2] Disponível a 27 de Setembro de 2007 em `http://awardsdatabase.oscars.org/ampas_awards/BasicSearch`.

disso, apesar de estar referenciado que a aplicação apenas interpretava os títulos originais, alguns utilizadores ignoravam esta restrição e usavam o título em português. Posta esta situação, contactaram-se duas entidades: o IGAC — Inspecção Geral das Actividades Culturais — e o *Web Site* Cinema PTGate[3]. A última enviou todos os títulos em Português que detinha (cerca de 5000) e o correspondente título original.

3.2.2 Informação disponível

A base de dados não engloba alguma informação sobre os filmes, como a banda sonora, o conteúdo que figura em *Fun Stuff*, os comentários dos utilizadores, *Tagline*, *Plot Outline* e *Plot Keywords*. No que concerne às pessoas ligadas ao cinema (sejam actores, realizadores, argumentistas, etc..), estão disponíveis as datas e locais de nascimento e óbito, nome completo e alcunha (*nickname*). A restante informação não foi incluída pois considerou-se que não faria sentido devolver informação biográfica, ou mesmo curiosidades acerca dos filmes, em blocos de texto escritos na língua inglesa. Está assim disponível apenas a informação que pode ser devolvida como resposta directa a uma questão do utilizador.

Relativamente aos prémios e nomeações recebidas, apenas podem ser consultados os Óscares. Os restantes prémios não foram obtidos pois assumiu-se que o alvo principal das questões formuladas seriam os prémios da Academia.

As categorias destes prémios que foram consideradas são as seguintes:

- Melhor Filme
- Melhor Filme Estrangeiro
- Melhor Actor Principal
- Melhor Actor Secundário
- Melhor Actriz Principal
- Melhor Actriz Secundária
- Melhor Realizador

As restantes categorias não foram contempladas por dificuldades em processar os ficheiros obtidos no *Web Site* dos óscares. Alguns títulos, nomes de pessoas e nomes de personagens, não correspondiam com os existentes no IMDB, obrigando a uma correcção manual dos dados. Este problema será abordado mais detalhadamente na secção 3.2.3.5.

[3]Disponível a 27 de Setembro de 2007 em `http://www.cinema.ptgate.pt`.

3.2.3 Carregamento da base de dados

O carregamento da base de dados é efectuado através de *scripts* na linguagem Perl que processam cada um dos ficheiros. Estes *scripts*, por intermédio de expressões regulares, recolhem a informação relevante e inserem-na na base de dados. Alguns dos *scripts* utilizados já haviam sido desenvolvidos anteriormente, tendo, no entanto, que ser adaptados ao pretendido para esta interface; outros, porém, tiveram que ser criados de raiz.

O IMDB resulta da contribuição de diversas pessoas e, apesar de estar explícito qual o formato de dados pretendido para cada ficheiro, algumas entradas não lhe obedecem, dificultando o processamento dos dados. Actualmente, o IMDB já tem formulários para contribuidores bastante completos que impedem a introdução de dados no formato errado, contudo, há ainda trabalho a fazer no que diz respeito à limpeza dos dados.

O carregamento é efectuado em 5 passos distintos que serão seguidamente analisados individualmente.

3.2.3.1 *Aka-titles*

Os *aka-titles* correspondem aos títulos alternativos que um determinado filme pode ter. Podem assim encontrar-se as diferentes designações para alguns filmes que, na maioria dos casos, corresponde ao nome dado ao filme numa língua diferente da do seu título original.

```
Goodfellas (1990)
   (aka GoodFellas (1990))        (USA) (promotional title)
   (aka GoodFellas - Drei Jahrzehnte in der Mafia (1990))        (Austria)
   (aka GoodFellas - Drei Jahrzehnte in der Mafia (1990))        (West Germany)
   (aka Quei bravi ragazzi (1990))        (Italy)
   (aka Wise Guy (1989))        (USA) (working title)
```

O exemplo acima, retirado do ficheiro que alberga informação sobre os títulos alternativos, significa que o filme cujo título original é "Goodfellas" é conhecido por outros cinco títulos, bem como o país em que é conhecido como tal.

Os títulos alternativos ficam associados na base de dados aos títulos originais, permitindo obter dados sobre um filme em questão a partir do seu título alternativo.

Nesta fase são também inseridos os títulos em Português. Nesse ficheiro, os títulos estão dispostos no seguinte formato <título original> <título traduzido>. Abaixo encontram-se alguns dos títulos que figuram nesse ficheiro:

```
One Night with the King   One Night with the King
One Point O               Um Ponto Zero
One True Thing            Podia-te Acontecer
The One                   Força Explosiva
Ong-bak                   Ong-bak - O Guerreiro
Only You                  Só Tu
```

Como é visível, não há informação sobre o ano de estreia do filme. Por esse motivo, é necessário "ligar" todos os filmes cujo título seja o original com o título traduzido. Por exemplo, como existem 6 filmes com o título "Only You", é necessário ligar o título "Só Tu" a todos eles.

3.2.3.2 Filmes

Nesta fase, é processado o ficheiro que contém todos os filmes e alguma informação adicional, como sendo o ano em que foi editado e a sua versão.

Os títulos começados pela categoria gramatical *determinante artigo* nas diversas línguas, estão num formato diferente. Para exemplificar, recorra-se ao filme "The Silence of the Lambs" — seria de esperar que figurasse desta forma no ficheiro, no entanto aparece como "Silence of the Lambs, The". O mesmo acontece com os filmes iniciados pelos artigos: *A, As, An, O, Os, La, Le, L', Les, Las, El, Die, De, Het ,Il, Das, Der*. São excepção a esta regra os artigos portugueses *um* e *uma* e os franceses *un*, *de* e *des*. Para facilitar o processo de procura, estas situações são detectadas e o título do filme é inserido com o artigo no início do título.

Para ilustrar o formato pretendido, vejam-se os seguintes exemplos de títulos de filmes:

```
Pulp Fiction (1994)
Schindler's List (1993)
Blaue Engel, Der (1930)
Uma Vida Normal (1994)
Strada, La (1954)
Enfants du paradis, Les (1945)
```

Entre parênteses curvos encontra-se o ano da primeira exibição do filme. No caso de não se saber o ano de exibição do filme, deverá estar entre parênteses curvos quatro pontos de interrogação, *(????)*. Ao longo do ficheiro, alguns títulos não obedeciam ao formato estabelecido, interrompendo o processamento. Numa primeira abordagem, tentou-se corrigir esses títulos, mas dada a grande dimensão do ficheiro de texto (cerca de 45 *Megabytes*), era dispendido muito tempo a editar e gravar o ficheiro a cada erro. Posteriormente, conseguiu-se contornar esse problema através de uma condição no *script* de carregamento.

Antes de ser inserido na base de dados, o título é convertido para letras minúsculas, garantindo, desta forma, um emparelhamento mais fácil entre os títulos incluídos nas perguntas efectuadas pelos utilizadores e os títulos inseridos na base de dados.

Os diversos títulos são distinguidos de acordo com o formato em que foram inseridos nos ficheiros de texto. Os títulos entre aspas ("") dizem respeito a séries de televisão enquanto que os titulos de filmes de televisão são seguidos por *(TV)*. Os títulos referentes a edições em vídeo são seguidos por *(V)*, os de vídeo-jogos são seguidos por *(VG)*. Com base nesta nomenclatura, separaram-se os títulos de filmes de cinema dos restantes. Esta separação é importante, pois evita que, a questões como "Em que filmes entra Jodie Foster?" se responda "The 46th Annual Academy Awards", "The Making of 'Panic Room", entre outros, juntamente com títulos de filmes em que Jodie Foster tenha participado.

Seguem-se alguns exemplos de títulos de televisão e vídeo-jogos para ilustrar esta distinção.

```
"Seinfeld" (1990) {The Puffy Shirt (#5.2)}
"Daily Show, The" (1996) {(2001-04-03)}
"Office, The" (2001) {(#2.1)}
"Monty Python's Flying Circus" (1969) {Michael Ellis}
Everest: The Death Zone (1998) (TV)
AFI's 100 Years... 100 Heroes & Villains (2003) (TV)
Midnight Club: Street Racing (2000) (VG)
```

3.2.3.3 Pessoas

Nesta fase são processados os ficheiros relativos às pessoas ligadas ao cinema e televisão. O IMDB faz a distinção por cargos, havendo um ficheiro para cada um deles. Nesse sentido temos um ficheiro para cada um dos seguintes grupos: actores, actrizes, compositores, *designers* de guarda-roupa, realizadores, editores, *designers* de produção, produtores, argumentistas e aquilo que o IMDB denomina por *miscellaneous crew*, ou seja, todos os que não se incluem nas categorias anteriores. As únicas categorias contempladas foram os actores, as actrizes, os realizadores, os argumentistas e os produtores. Tal decisão deve-se ao facto de, à partida, as pessoas não terem interesse em saber informação sobre os *designers* de guarda-roupa ou de produção, etc.. As pessoas que mais suscitam interesse são os actores e actrizes, seguidos dos realizadores, argumentistas e produtores. Esta opção permitiu reduzir o volume de dados na tabelas de pessoas e diminuir a probabilidade de coincidência de nomes. Por exemplo, se se tivessem incluído os *designers* de produção e se se fizesse uma questão sobre o Matthew Broderick, o sistema pediria que se desambiguasse entre o "Matthew Broderick I - actor e realizador" e o "Matthew Broderick - designer de produção". Desta forma, poupa-se a fase de desambiguação em alguns casos.

Os diferentes ficheiros são processados individualmente através do mesmo *script* que coloca todas as pessoas numa só tabela, embora distinguidas pela sua profissão. Os nomes estão colocados na forma

apelido, nome próprio. Uma vez que, na língua portuguesa, se costuma colocar o nome próprio seguido do apelido, optou-se por colocar os nomes na base de dados no formato *nome apelido* por dois motivos: para facilitar a correspondência entre o nome de uma pessoa inserido na questão do utilizador e para, numa resposta devolvida ao utilizador, devolver o nome de uma dada pessoa no segundo formato referido.

A nível do ficheiro de texto, há distinção entre os actores, actrizes e os restantes. Segue-se um extracto da informação disponível para o actor James Dean:

```
Dean, James (I)        'Giant' Stars Are Off to Texas (1955)  (uncredited)  [Himself]
                        101 Most Shocking Moments in Entertainment (2003) (TV) (archive footage) [Himself]
                        72nd Annual Academy Awards, The (2000) (TV) (archive footage) [The Rebel]
                        ABC 2000: The Millennium (1999) (TV)  (archive footage)
                        America at the Movies (1976)  (archive footage)  [Himself]  <23>
                        Death Scenes 2 (1992) (V)  (archive footage) (uncredited)  [Himself]
                        East of Eden (1955)  [Cal Trask]  <2>
                        "Danger" (1950) {Death Is My Neighbor (#3.45)}  [J.B.]
                        "Danger" (1950) {No Room (#3.26)}
                        "Danger" (1950) {Padlocks (#5.10)}  [Augie]
                        "Danger" (1950) {The Little Woman (#4.27)}
```

Verifica-se então que na primeira linha está o nome do actor seguido do número romano *I*. Isto significa que há mais que um James Dean na base de dados e este, em particular, é identificado com este número. De seguida, está a indentificação dos títulos em que participou como actor (nome seguido do ano de exibição como distinção) e, entre parênteses rectos [ ], o nome do papel que foi desempenhado. Por exemplo, o actor James Dean representou o personagem "Cal Trask" no filme "East of Eden" de 1955. Entre os símbolos <> encontra-se a ordem na qual o actor aparece no genérico. No filme atrás referido, o actor James Dean aparece em segundo lugar. Este dado é importante para conseguir encontrar o actor (ou actriz) principal de um filme.

Relativamente aos restantes cargos, o formato é distinto, pois não contempla informação relativa à posição nos créditos e à personagem desempenhada. Segue-se um extracto da informação presente no ficheiro dos realizadores para Woody Allen:

```
Allen, Woody    Alice (1990)
                Annie Hall (1977)
                Another Woman (1988)
                Anything Else (2003)
                Bananas (1971)
                Broadway Danny Rose (1984)
                Bullets Over Broadway (1994)
                Celebrity (1998)
                        (...)
```

Como é visível, aparece o nome seguido dos títulos e o ano de exibição em que esteve envolvido enquanto realizador. Para os restantes cargos, a estrutura é idêntica.

3.2.3.4 Informação biográfica

A informação biográfica do IMDB vai desde os dados mais básicos, como sendo datas e locais de nascimento e óbito, até detalhes como o nome dos cônjuges, altura, citações, salário auferido em alguns filmes, etc.. No entanto, estes detalhes estão disponíveis para um reduzido número de pessoas, apenas as mais conhecidas, não fazendo sentido preparar uma base de dados para albergar tantos detalhes que só seriam preenchidos na totalidade por um número restrito de pessoas. Decidiu-se que os dados mais relevantes a recolher são: o nome completo, a alcunha, a data e local de nascimento e, no caso da pessoa ter falecido, a sua data e local de óbito.

O formato do ficheiro de texto é um pouco diferente dos restantes e é ilustrado de seguida com um excerto da informação biográfica de Courteney Cox:

```
-----------------------------------------------------------
NM: Cox, Courteney

RN: Cox, Courteney Bass

NK: CeCe

DB: 15 June 1964, Birmingham, Alabama, USA

HT: 5' 5"

BY: David Ross

SP: * 'Arquette, David' (qv) (12 June 1999 - present); 1 child
-----------------------------------------------------------
```

A fila de caracteres "hífen" separa as diferentes pessoas. As siglas no início de cada linha indicam o tipo de informação que se segue:

- NM - Nome pelo qual é conhecida;
- RN - Nome completo (*Real Name*);
- NK - Alcunha (*Nickname*);

- DB - Data de nascimento (*Date of Birth*) seguida do local;
- HT - Altura (*Height*);
- SP - Cônjuge (*Spouse*).

3.2.3.5 Óscares

Como já foi referido, a informação sobre os óscares foi obtida a partir de outra fonte uma vez que não estava disponível no IMDB. A partir de pesquisas no portal dos óscares, foi criado um ficheiro de texto para cada categoria: Melhor Actor Principal e Secundário, Melhor Actriz Principal e Secundária, Melhor Filme e Melhor Filme Estrangeiro e Melhor Realizador. A informação presente nesses ficheiros é obtida a partir de *scripts* em Perl criados para o efeito que, posteriormente, inserem a informação na base de dados. Distinguem-se três diferentes tipos de ficheiros a nível do seu formato: os ficheiros para Melhor Filme e Melhor Filme Estrangeiro, os ficheiros para as categoria de representação e os restantes. Segue-se um extracto do ficheiro para Melhor Filme:

```
2001 (74th)
BEST PICTURE
*
A Beautiful Mind -- Brian Grazer and Ron Howard, Producers
Gosford Park -- Robert Altman, Bob Balaban and David Levy, Producers
In the Bedroom -- Graham Leader, Ross Katz and Todd Field, Producers
The Lord of the Rings: The Fellowship of the Ring -- Peter Jackson,
                        Fran Walsh and Barrie M. Osborne, Producers
Moulin Rouge -- Martin Brown, Baz Luhrmann and Fred Baron, Producers
```

O símbolo $*$ significa que a linha seguinte corresponde a um filme premiado. São adicionados também os nomeados, mas é assinalado aquele que, efectivamente, recebeu o prémio. A informação relativa aos produtores é desprezada uma vez que a informação que interessa para a aplicação é somente o título do filme.

No caso de haver mais do que um filme com o mesmo título em base de dados, como é o caso do filme "Moulin Rouge", é escolhido aquele que mais se aproxima do ano de realização da cerimónia. Assim, entre o "Moulin Rouge" do ano de 2001, 1952, 1940, 1934 e 1928, obviamente seria escolhido o de 2001. Se houver mais um título com o mesmo ano, é escolhido aquele com a "menor" versão, ou seja, se houvesse um "Moulin Rouge I" e um "Moulin Rouge II" seria escolhido o primeiro.

Os ficheiros para as categorias de representação apresentam características diferentes. Segue um extracto do ficheiro com os óscares para Melhor Actor:

```
1992 (65th)
ACTOR IN A LEADING ROLE
Robert Downey Jr. -- Chaplin {"Charles Chaplin"}
Clint Eastwood -- Unforgiven {"William 'Bill' Munny"}
*
Al Pacino -- Scent of a Woman {"Lt. Col. Frank Slade"}
Stephen Rea -- The Crying Game {"Fergus"}
Denzel Washington -- Malcolm X {"Malcolm X"}
```

Para os ficheiros de representação, o formato é <*nome do actor*> – <*filme*> {<*personagem interpretada*>}. Um dos grandes obstáculos ao correcto carregamento destes dados foi a nomenclatura para as personagens. Uma vez que os dados dos Óscares não provieram do IMDB, havia distinção nos nomes das personagens, especialmente para aquelas com cargos militares. É o caso do exemplo acima para o personagem "Lt. Col. Frank Slade" que, no IMDB, é designada como "Lieutenant Colonel Frank Slade". Esta distinção obrigou a que todas as distinções entre nomes de personagens fossem corrigidas para conseguir relacionar os dados dos óscares de representação (tabela *oscaracting*), com os dados do IMDB. Para além disso, alguns nomes de actores e actrizes não correspondiam aos que figuravam no IMDB e, pelo mesmo motivo, tinham que ser corrigidos.

Para os restantes ficheiros, o formato é semelhante ao dos ficheiros de representação, exceptuando na informação relativa ao personagem, que não tem aplicação em cargos que não os de representação. Tal como para as outras categorias, havia diferenças nos nomes dos vencedores que impediam a correcta ligação com os nomes já existentes. O esforço de corrigir manualmente os ficheiros de todas as categorias para obter uma correcta correspondência era demasiado elevado, logo obtou-se por processar apenas o ficheiro relativo à categoria de Melhor Realizador, a mais popular entre os cargos de não-representação.

3.3 Estrutura da base de dados

A base de dados é constituída por 11 tabelas que albergam toda a informação angariada através do processamento dos ficheiros de texto. Para uma melhor compreensão da sua estrutura, aconselha-se a consulta do esquema entidade-relação na figura 3.1.

As tabelas mais importantes são a *films* e a *persons* que, respectivamente, contêm todos os títulos de filmes e nomes de pessoas ligadas ao cinema. Ambas as tabelas são criadas com índices *full-text* no nome das pessoas e títulos de filmes para permitir esse tipo de procuras. Desta forma, é possível fazer interrogações *full-text* sobre essas tabelas para conseguir determinar se uma determinada questão contém um título de filme e/ou um nome de pessoa. Mais detalhes sobre esse procedimento serão dados no capítulo 4.2. Nas tabelas 3.1 e 3.2 figuram exemplos de entradas das tabelas *persons* e *films*.

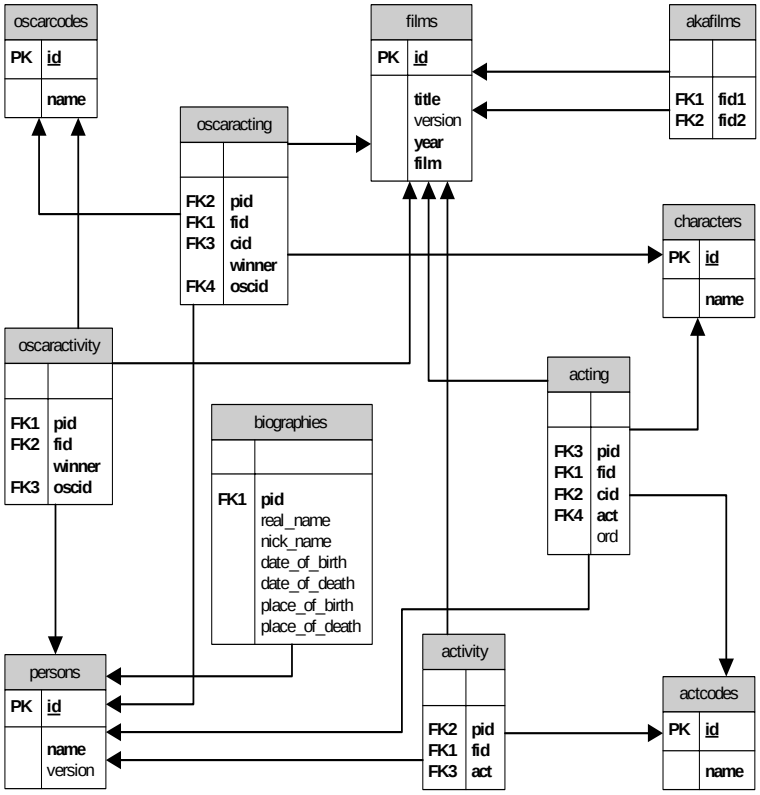

Figura 3.1: Esquema entidade-relação da base de dados com informação de cinema.

id	name	version
223267	andré glatti	0
223268	joe castagnoli	I
223269	johnny hanna	0
223270	cornelius glatzl	0
223271	óscar david gómez	0
223272	benjamin castaldi	0

Tabela 3.1: Tabela *persons*.

Uma vez que a relação entre pessoas e filmes é de *muitos-para-muitos*, ou seja, uma pessoa pode figurar em um ou mais filmes e, num filme, figuram uma ou mais pessoas, é necessária uma tabela intermédia que estabeleça essa relação. Tal é assegurado, neste caso, por duas tabelas: a tabela *acting* e a tabela *activity*. A necessidade de duas tabelas para esta função prende-se com a distinção de dados entre pessoas com cargos de representação (actores e actrizes) e outros cargos (para uma melhor compreensão dessa distinção, sugere-se a consulta à secção 3.2.3.3). A tabela *acting* tem duas colunas a mais que a tabela *activity*, uma para indicar a ordem pela qual aparece no genérico do filme, *ord*, e outra, *cid*, que indica a chave primária da personagem que interpreta (tabela *characters*). Como exemplo, a primeira entrada na tabela 3.3 significa que, a pessoa cujo *id* na tabela persons é o 325564, participou no filme cujo *id* é o 90658, interpretando a personagem cujo *id* é o 75782, aparece em segundo lugar no genérico desse filme e participa no filme como actriz (consultar a tabela 3.4).

id	title	year	version	film
257799	on the nickel	1980	0	1
257800	científicamente perfectos	1996	0	1
257801	salassel min harir	1963	0	1
257802	perfect witness	1989	0	1
257803	scientifically perfect	1996	0	1
257804	the mary kay letourneau story	2000	0	1

Tabela 3.2: Tabela *films*.

pid	fid	cid	ord	act
325564	90658	75782	2	1
325564	315638	75783	9	1

Tabela 3.3: Tabela *acting*.

Com as tabelas *acting*, *persons* e *films* é possível saber, por exemplo, em que filmes participou um determinado actor. Por exemplo, "em que filmes participou Johnny Depp?" seria traduzido para a seguinte interrogação SQL:

```
SELECT films.title FROM films, persons, acting WHERE
        persons.name="Johnny Depp" AND acting.pid=persons.id AND acting.fid=films.id;
```

O campo *film* da tabela *films* permite a diferenciação entre conteúdos de cinema e outros conteúdos. Se, porventura, se pretendesse apenas os filmes de cinema desse actor, teria que se adicionar a condição "films.film=1". Neste exemplo, não foi necessário recorrer à tabela *actcodes*, mas para a questão "Que filmes realizou Steven Spielberg", já seria necessário:

```
SELECT films.title FROM films, persons, activity, actcodes
        WHERE persons.name="Steven Spielberg" AND activity.pid=persons.id AND
        activity.fid=films.id AND actcodes.name="realizador" AND
        activity.act=actcodes.code;
```

O campo *act* tem também utilidade na tabela *acting*. Por exemplo, no caso de se pretender "o protagonista" ou "a actriz principal" pode-se obter o primeiro da lista de créditos, ou seja, a entrada na tabela *acting* em que o campo *ord* seja menor para "actor" ou "actriz". Supondo que a pergunta é "Quem é a protagonista de The Silence of the Lambs?", usar-se-ia a interrogação que se segue:

```
SELECT persons.name, acting.ord FROM films, persons, acting, actcodes
   WHERE films.title="The Silence of the Lambs" AND acting.pid=persons.id
   AND acting.fid=films.id AND actcodes.name="actriz" AND acting.act=actcodes.code
   ORDER BY acting.ord limit 1;
```

id	code	name
1	0	actor
2	1	actriz
3	2	realizador
4	3	produtor
5	4	argumentista

Tabela 3.4: Tabela *actcodes*.

Relativamente aos óscares, existem 3 tabelas que albergam toda a informação: *oscarfilms*, *oscaracting* e *oscaractivity*. Na primeira, estão os óscares para Melhor Filme e Melhor Filme Estrangeiro, na segunda, as categorias de representação, Melhor Actor Principal e Secundário e Melhor Actriz Principal e Secundária e, na terceira, a de Melhor Realizador. Todas têm em comum o campo *fid* que é o identificador do filme que ganhou o óscar em questão.

Se, por exemplo, se pretender saber quem ganhou o Óscar de Melhor Filme em 1997, basta realizar a seguinte query:

```
SELECT films.title FROM films, oscarfilms, oscarcodes WHERE oscarfilms.year=1997
   AND oscarfilms.fid=films.id AND oscarcodes.name="óscar de melhor filme"
   AND oscarfilms.oscid=oscarcodes.code AND oscarfilms.winner=1;
```

Para saber os nomeados para esse óscar, bastaria retirar a condição "oscarfilms.winner=1". Esta estruturação da informação relativa aos óscares facilita a obtenção de dados para questões como: "Quantos óscares recebeu <filme>?" ou então, "Quantos óscares recebeu <pessoa>", sendo o mesmo válido se o pretendido for o número de nomeações. Por exemplo, se pretendêssemos saber que nomeações para óscar já recebeu Woody Allen, executaríamos duas interrogações distintas: uma sobre a tabela *oscaracting*, para saber as nomeações em categorias de representação, e outra sobre a tabela *oscaractivity* para saber as nomeações para outras categorias.

```
SELECT oscarcodes.name, oscaractivity.year FROM oscarcodes, persons, oscaractivity
    WHERE persons.name="woody allen" AND oscaractivity.pid=persons.id
   AND oscarcodes.code=oscaractivity.oscid;
```

A interrogação para as categorias de representação seria idêntica, sendo apenas consultada a tabela "oscaracting" ao invés da "oscaractivity".

```
SELECT oscarcodes.name, oscaracting.year FROM oscarcodes, persons, oscaracting
   WHERE persons.name="woody allen" AND oscaracting.pid=persons.id
   AND oscarcodes.code=oscaracting.oscid;
```

4 Interpretação da Questão

4.1 Introdução

A interpretação da questão é uma etapa bastante complexa. Um dos seus aspectos capitais é o correcto reconhecimento de entidades mencionadas sendo estas, neste caso específico, os nomes de pessoas do mundo do cinema e os títulos de filmes. Infelizmente, o analisador morfo-sintáctico da cadeia de processamento de língua natural não consegue processar correctamente a questão quando esta contém nomes de pessoas e/ou títulos de filmes, tal como é visível na figura 4.1 em que se apresenta a análise sintáctica da questão "Quem é o realizador de eyes wide shut?". O motivo prende-se com o facto de esses títulos e nomes não estarem presentes no léxico do analisador.

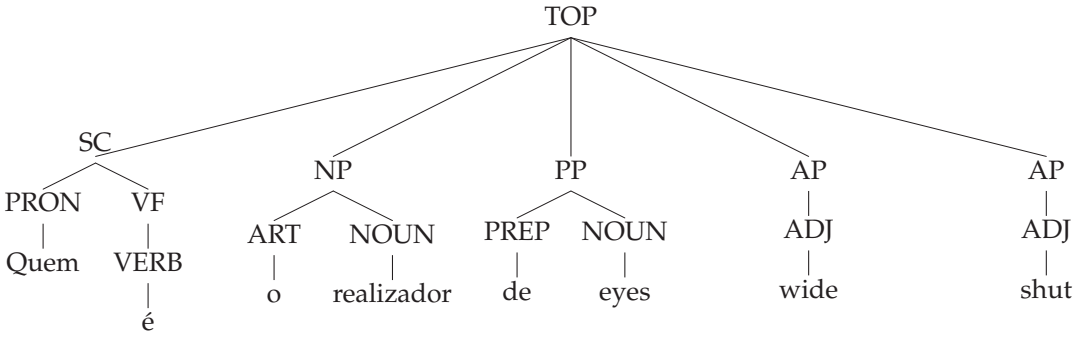

Figura 4.1: Árvore de análise sintáctica para a questão "Quem é o realizador de eyes wide shut?" sem reconhecimento de entidades mencionadas.

O analisador divide o título do filme em diferentes grupos sintácticos pois não consegue identificar o título do filme como um só. Poder-se-ia assumir que o título do filme está no fim da questão e aglomerar todas as palavras desde "de" até ao ponto de interrogação (?). Contudo, nem sempre o título do filme está no fim da questão, basta que esta esteja na passiva para tal não se verificar — "Eyes wide shut foi realizado por quem?". Pode ainda estar no meio da questão, como se pode ver em "Quem contracena em eyes wide shut com tom cruise?" cuja análise sintáctica está visível na figura 4.2.

Sendo que, muitas das vezes, não é possível distinguir um título ou nome dos restantes componentes da questão, é essencial que estas entidades sejam identificadas previamente para facilitar o seu processamento. Analisando as figuras 4.2 e 4.3, é visível a diferença na análise sintáctica no caso das entidades mencionadas (Nicole Kidman e Eyes Wide Shut) estarem previamente identificadas (figura

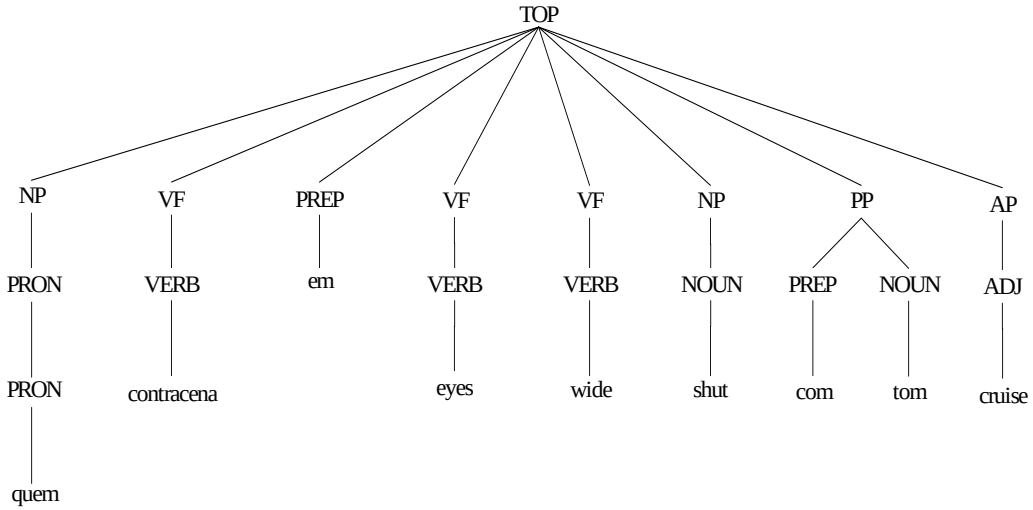

Figura 4.2: Árvore de análise sintáctica para a questão "Quem contracena em eyes wide shut com tom cruise?" sem reconhecimento prévio de entidades mencionadas.

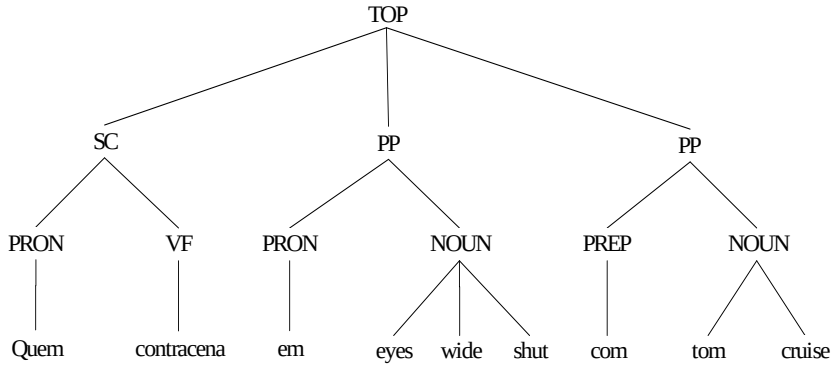

Figura 4.3: Árvore de análise sintáctica para a questão "Quem contracena em eyes wide shut com tom cruise?" com reconhecimento de entidades mencionadas.

4.3) e não estarem (figura 4.2).

O método utilizado para reconhecer as entidades mencionadas é descrito detalhadamente na secção 4.2, enquanto que na secção 4.3 será descrito o processo utilizado no caso de haver ambiguidade nas entidades reconhecidas. Na secção 4.4 vai ser descrita a cadeia de processamento da língua natural que permite interpretar a questão do utilizador.

4.2 Reconhecimento de Entidades mencionadas

4.2.1 Estratégias Consideradas

Uma das estratégias consideradas para o reconhecimento de entidades mencionadas foi a criação de uma gramática local que contivesse todos os títulos de filmes e todos os nomes de pessoas ligadas ao cinema. Abaixo encontram-se os exemplos de quatro regras que garantem a classificação de "eyes wide shut" e "the bridges of madison county" como sendo "noun", juntamente com o traço adicional "culture". Adicionalmente, "marlon brando" e "tom cruise" seriam identificados como sendo "noun" com o traço adicional "actor".

```
1> noun[culture=+] = ?[surface:eyes], ?[surface:wide], ?[surface:shut].
1> noun[culture=+] = ?[surface:the], ?[surface:bridges], ?[surface:of], ?[surface:madison], ?[surface:county].
2> noun[actor=+] = ?[surface:marlon], ?[surface:brando].
2> noun[actor=+] = ?[surface:tom], ?[surface:cruise].
```

A estratégia inicial para gerar as regras consistiu então em, a partir dos nomes e títulos em base de dados, criar um ficheiro que fosse carregado pelo analisador, garantindo a correcta identificação destas entidades. Gerou-se um ficheiro com cerca de 500 000 regras e tentou-se a execução do analisador, mas o volume de regras era demasiado grande para que o analisador pudesse executar em tempo útil (em 30 minutos não conseguiu terminar o processamento de uma simples questão).

Dado que, até 2000 regras, o analisador processava uma questão em cerca de 4 segundos (tempo aceitável para a aplicação), surgiu uma nova estratégia: arranjar um critério de selecção para apurar os 2000 títulos e nomes mais "populares".

Um critério possível era a selecção dos títulos mais recentes, por exemplo, apenas os que estrearam no século XXI. Há, no entanto, mais de 100 000 filmes estreados após o ano 2000, logo esse critério não era o mais adequado, pois continuava a ter-se um número demasiado elevado de filmes. Para além disso, estar-se-ia a privar o utilizador de consultar informação acerca de muitos filmes populares anteriores a essa data. No que diz respeito aos nomes de pessoas mais populares, um critério possível era, por exemplo, seleccionar os nomes que estiveram envolvidos em mais de 10 filmes. No entanto, nem sempre os mais requisitados para actuar em filmes são os mais populares, basta pensar no caso de "James Dean" que, embora tenha participado em 3 filmes de cinema, é dos nomes mais conhecidos.

Uma vez que nenhuma das estratégias referidas anteriormente era adequada para resolver o problema do reconhecimento das entidades mencionadas, optou-se pela utilização de interrogações *full-text* sobre a tabela de títulos (*films*) e de pessoas (*persons*), processo que se descreve de seguida.

4.2.2 Estratégia seguida

4.2.2.1 Interrogações *full-text*

A realização de interrogações *full-text* é uma funcionalidade disponível nas bases de dados MySQL desde a versão 3.23.23 [1]. Para exemplificar a sua utilização, segue-se o exemplo do código SQL para uma interrogação deste género sobre a tabela *films*:

```
SELECT DISTINCT films.title FROM films
        WHERE match(title) AGAINST ("Quem é o realizador de Forrest Gump");
```

Na interrogação acima, está-se a fazer uma procura pela expressão "Quem é o realizador de Forrest Gump?" sobre a coluna *title* da tabela *films*. Na tabela 4.1, podem ser vistos os dez primeiros resultados dessa interrogação.

TITLE
forrest gump
black forrest gump
die welt des forrest gump
through the eyes of forrest gump
vida, pasión y muerte de un realizador iracundo
gump & co.
foreskin gump
forrest
andy gump for president
the amazing mr. forrest

Tabela 4.1: Tabela de resultados para interrogação *full-text* "Quem é o realizador de Forrest Gump?" sobre a tabela "films".

Os resultados estão ordenados por ordem de maior relevância sendo "forrest gump" o mais relevante e "the amazing mr. forrest" o 10º mais relevante.

4.2.2.2 Método de emparelhamento

Analisando a tabela 4.1 verificamos que as primeiras quatro entradas contêm "forrest gump". No entanto, apenas a primeira de todas está completamente contida na questão do utilizador. Para classificar entidades na questão, é requisito que estas estejam integralmente presentes na questão. Se assim não fosse, haveria inúmeras entradas a considerar. Pode igualmente acontecer que mais do que uma entrada esteja nessa situação: é o caso de "forrest" e "forrest gump'. Entendeu-se que a única entidade reconhecida seria "forrest gump", pois "forrest" está contido em "forrest gump". Existem, contudo, excepções a esta regra. Analise-se, por exemplo, a questão "Que personagem interpreta John Malkovich em Being

[1] Informação disponível a 27 de Setembro em `http://www.onlamp.com/pub/a/onlamp/2003/06/26/fulltext.html`.

John Malkovich?". Nesta questão devem ser reconhecidas duas entidades distintas: "Being John Malkovich" como filme e "John Malkovich" como nome de pessoa. Nesse caso, apesar de "John Malkovich" estar contido em "Being John Malkovich", não é descartado, pois aparece mais que uma vez na questão.

4.2.2.3 Procura limitada

Uma vez que estas procuras são realizadas em tabelas com um número elevado de entradas (1 502 517 nomes de pessoas e 672 048 títulos de filmes), são retornados muitos resultados. Por exemplo, a interrogação *full-text* da questão "Quem é o realizador de The Last Boy Scout" sobre a tabela *films* originaria mais de 1000 entradas. As entradas devolvidas pelas interrogações *full-text* estão ordenadas por ordem de maior relevância, no entanto, não foi encontrada informação sobre o método utilizado para a determinar. As entradas consideradas mais relevantes, não são, por vezes, aquilo que se estaria à espera. Por exemplo, a interrogação que se segue daria origem aos resultados disponível na tabela 4.2.

```
SELECT DISTINCT films.title FROM films
        WHERE match(title) AGAINST ("apocalypse now") LIMIT 10;
```

TITLE
10.5: apocalypse
the apocalypse
apocalypse joe
apocalypse now
apocalypse oz
apocalypse bop
apocalypse
the little apocalypse
apocalypse!
after the apocalypse

Tabela 4.2: Tabela de resultados para interrogação *full-text* "apocalypse now" sobre a tabela *films*.

Seria expectável que "apocalypse now" fosse a entrada mais relevante e, por isso, aparecesse em primeiro lugar, no entanto, foi considerada a 4ª entrada mais relevante. De salientar que a interrogação está a ser feita só e apenas com o nome do filme, não havendo outras palavras a "interferir". Se, mesmo com a entidade exacta, os resultados mais relevantes não são os esperados, ao fazer-se a interrogação com a questão formulada pelo utilizador, esse desvio é ainda maior. Por esse motivo, devem ser testadas bastantes entradas por forma a encontrar a entidade mencionada. Seria demasiado "pesado" verificar se cada uma das mais de 1000 entradas estão contidas na frase, por isso decidiu-se que seria razoável analisar as 150 consideradas mais relevantes.

Note-se ainda que, como pode haver mais que uma entidade mencionada na questão, não se pode ficar pela primeira entidade encontrada. Veja-se a questão "Em que filmes contracenam Glenn Close e

John Malkovich?". Os 10 primeiros resultados da interrogação desta questão sobre a tabela "persons" está visíveis na tabela 4.3. As primeiras duas entradas estão presentes na questão e não estão contidas uma na outra, logo deverão ser reconhecidas como entidades mencionadas. Se não se prosseguisse, apenas "Glenn Close" seria identificado, ignorando "John Malkovich" como outra entidade presente na questão.

NAME
glenn close
john malkovich
john close
gary malkovich
matt malkovich
erik malkovich
claudia malkovich
kara malkovich
becky malkovich
kent malkovich

Tabela 4.3: Tabela de resultados para interrogação *full-text* "Em que filmes contracenam Glenn Close e John Malkovich?" sobre a tabela "persons".

4.2.2.4 Principais Problemas

Como foi dito anteriormente, o reconhecimento de entidades mencionadas através da base de dados, exige que a entidade esteja correctamente escrita, de outra forma, não será possível o emparelhamento exacto da entidade com a questão formulada. Isto obriga a que o utilizador escreva os nomes das pessoas e os títulos *ipsis verbis*, o que nem sempre é fácil, especialmente se o título for escrito numa língua que não a nativa do utilizador.

A grande dificuldade em resolver esta situação, reside na dificuldade de "isolar" os nomes e títulos de filmes (aliás, é o problema que desencadeou o reconhecimento de entidades mencionadas antes da análise morfo-sintáctica). Uma maneira de o fazer, seria exigir ao utilizador a colocação de aspas ("") nos títulos de filmes e nomes de pessoas. Desta forma, não seria necessário interrogar a base de dados com a pergunta na sua totalidade, far-se-ia apenas com as entidades entre aspas. Assim, seria possível, através de algoritmos de detecção de erros, encontrar o título de filme ou nome de pessoa mais parecido com o escrito pelo utilizador. Outra alternativa, seria a criação de tabelas com os títulos e nomes mais populares do cinema. Inicialmente, seriam constituídas pelas pessoas e filmes que foram nomeados para os Óscares da Academia e iria sendo actualizada progressivamente conforme as entidades reconhecidas em questões formuladas pelo utilizador.

4.3 Desambiguação de entidades mencionadas

Após o reconhecimento de entidades mencionadas, pode verificar-se que os títulos de filme e/ou nomes de pessoa reconhecidos não são únicos na base de dados. Sendo esse o caso, é necessária uma desambiguação. Há, no entanto, critérios que permitem à própria aplicação fazer a desambiguação sem ter que maçar o utilizador com essa tarefa.

4.3.1 Pré-desambiguação

Através da informação presente na questão, é possível desambiguar uma entidade sem que seja necessário recorrer ao utilizador. Por exemplo, na questão "Quem contracena com Bruce Willis no filme Armageddon?" estão presentes duas entidades, a pessoa "Bruce Willis" e o filme "Armageddon". Este título não é único, existindo vários filmes com o mesmo título. No entanto, uma vez que temos presente na questão o nome de um actor, à partida estarão ambos relacionados. Para todos os filmes com o mesmo título, verifica-se se "Bruce Willis" participa em algum deles e, em caso afirmativo, parte-se para a interpretação da questão com esse filme já escolhido. No caso de não participar, o utilizador terá de desambiguar.

O processo inverso também se verifica, ou seja, o título de um filme também pode servir para desambiguar entre nomes de actores. Por exemplo, a questão "Quem contracena com Robin Williams em Good Will Hunting?" tem ambiguidade no que diz respeito à pessoa "Robin Williams" pois existem vários actores com esse nome. No entanto, só um "Robin Williams" participou nesse filme, sendo este o escolhido.

4.3.2 Ambiguidade entre títulos e nomes

Pode haver o caso de uma entidade ser identificada quer como filme quer como nome de pessoa. Por exemplo, para a questão: "Quem é o realizador de Troy?", a palavra Troy é reconhecida como sendo um nome de filme e como sendo um nome de pessoa. Existe um filme de 2004 e 7 pessoas com esse nome. No entanto, não faz sentido pedir ao utilizador para desambiguar entre os diversos actores uma vez que, dado o "formato" da pergunta ("Quem é o realizador de Troy?"), Troy deverá ser encarado como nome de filme. Para garantir isso, é feito um pré-processamento com expressões regulares que, consoante alguns padrões encontrados na questão, resolve se a entidade encontrada deve ser encarada como filme ou como nome de pessoa. Seguem-se exemplos de padrões que garantem que uma entidade é um filme:

- (...) filme <entidade>?

- (...) em <entidade>?
- quem fez/protagoniza/realiza <entidade>?

Da mesma forma, há padrões que descartam a possibilidade de uma entidade ser um título de filme:

- quem é <entidade>?
- (...) que filmes entra/protagoniza/fez <entidade>?

Há, no entanto, situações em que não é possível distinguir se a entidade é um filme ou um nome de pessoa. Por exemplo, "Quantos óscares ganhou Amadeus?", pode tanto ser sobre o filme "Amadeus" ou sobre o actor "Amadeus". Nestes casos, em que não é possível uma pré-desambiguação, o sistema responde tanto para o actor como para o título de filmes. Não é muito comum haver entidades que sejam simultaneamente títulos de filmes e nomes de pessoas, no entanto, é importante esta detecção antes da desambiguação. Se o utilizador faz uma questão sobre um filme e se o sistema lhe pedir para desambiguar entre nomes de pessoas, o utilizador perde a confiança no sistema e duvida da sua "inteligência".

4.4 Cadeia de Processamento da Língua Natural

Nesta secção será feita uma breve explicação da cadeia de processamento da língua natural utilizada, que inclui o analisador XIP®(*Xerox Incremental Parser*) e vários módulos desenvolvidos no L^2F INESC-ID. A arquitectura completa da cadeia de processamento da língua natural pode ser consultada na figura 4.4 e será descrita na secção seguinte.

4.4.1 Arquitectura

A primeira fase é a de "tokenização" da entrada, ou seja, do texto que é fornecido. Nessa fase, são identificados endereços electrónicos, números ordinais, cardinais e romanos, abreviaturas, sinais de pontuação, entre outras coisas. Resumindo, o texto é dividido em palavras. Na fase de etiquetação *Part of Speech* (POS), as palavras identificadas previamente são categorizadas morfologicamente, sendo que cada palavra pode ter mais que uma etiqueta. Posteriormente, é feita uma divisão em frases pela identificação dos sinais de pontuação (!,? e .). A saída dessa fase é convertida para XML e passada para o RuDriCo (*rule-driven converter*) que realiza tarefas como:

- Descontracção das palavras (ex: nas = em + as);
- Alterações à lematização efectuada na etiquetação;

- Identificação de locuções adverbiais, prepositivas, etc..

Há, novamente, uma conversão para XML e os dados são passados para o Marv que selecciona uma das etiquetas atribuídas na fase de etiquetação. O RuDriCo é novamente executado para fazer a identificação de algumas entidades mencionadas. A saída do RuDriCo é então passada para o XIP, responsável pela análise sintáctica/semântica.

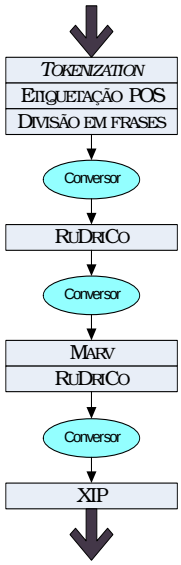

Figura 4.4: Arquitectura do analisador morfo-sintáctico analisado.

4.4.2 Estrutura do XIP

O XIP é um analisador que recebe como entrada um segmento de texto e fornece informação linguística sobre o mesmo. A sua arquitectura facilita a adição de informação lexical, sintáctica e semântica. Uma das suas funcionalidades é a extracção de dependências — ligações linguísticas entre conjuntos de palavras.

O XIP é composto por 3 módulos principais:

- Módulo de desambiguação contextual – as palavras são categorizadas de acordo com o contexto em que estão inseridas;
- Módulo de segmentação – agrupa as palavras em unidades linguísticas como por exemplo: NP-*Noun phrase*, VP-*Verbal Phrase*, PP-*Prepositional Phrase* e produz uma árvore de segmentação como as apresentadas nas figuras 4.1, 4.2 e 4.3;
- Módulo de dependência – usa regras pré-definidas para identificar dependências presentes no texto.

A maior parte do trabalho realizado na configuração do XIP para a interpretação das questões sobre cinema foi feito ao nível das definição de dependências, embora tenham sido necessárias algumas adições ao nível da segmentação.

4.4.2.1 Agrupamento

Uma das tarefas importantes nesta fase é o agrupamento de entidades mencionadas: títulos de filmes e nomes de pessoas. Para tal, para cada questão formulada, as entidades reconhecidas são adicionadas a um ficheiro de gramática local por forma a que sejam identificadas como um só segmento. Há, no entanto, outras expressões que devem ser agrupadas como é o caso dos óscares:

```
46> noun[ofe=+, oscar=+] @= ?[lemma:óscar];?[lemma:oscar], (?[lemma:de]); (?[lemma:para]),
                            ?[lemma:melhor], ?[lemma:filme], ?[lemma:estrangeiro] .
47> noun[of=+, oscar=+]  @= ?[lemma:óscar];?[lemma:oscar], (?[lemma:de]); (?[lemma:para]),
                            ?[lemma:melhor], ?[lemma:filme] .
```

A primeira regra pretende "apanhar" expressões como "óscar (de/para) melhor filme estrangeiro", sendo "de" ou "para" opcionais neste caso. O "lemma" representa a raiz da palavra, por isso, se a expressão for "óscares de melhores filmes estrangeiros", a regra também é aplicada. A inclusão de "oscar" como alternativa a "óscar" prende-se com o facto de algumas pessoas escreverem sem acento salvaguardando-se essa situação.

A segunda regra tem como objectivo detectar o "óscar (de/para) melhor filme". Como esta regra é mais genérica que a regra "óscar (de/para) melhor filme estrangeiro", é necessário colocá-la depois da regra mais específica para evitar que seja apenas agrupado "óscar de melhor filme".

Seguem-se as regras utilizadas para os restantes óscares:

```
48> noun[oactsec=+, oscar=+]  @= ?[lemma:óscar];?[lemma:oscar], (?[lemma:de]); (?[lemma:para]),
                                 ?[lemma:melhor], ?[lemma:actor], ?[lemma:secundário] .
49> noun[oactp=+, oscar=+]    @= ?[lemma:óscar];?[lemma:oscar], (?[lemma:de]); (?[lemma:para]),
                                 ?[lemma:melhor], ?[lemma:actor], (?[surface:principal]) .
50> noun[oactzsec=+, oscar=+] @= ?[lemma:óscar];?[lemma:oscar], (?[lemma:de]); (?[lemma:para]),
                                 ?[lemma:melhor], ?[lemma:actriz], ?[lemma: secundário] .
51> noun[oactzp=+, oscar=+]   @= ?[lemma:óscar];?[lemma:oscar], (?[lemma:de]); (?[lemma:para]),
                                 ?[lemma:melhor], ?[lemma:actriz], (?[lemma:principal]) .
52> noun[oreal=+, oscar=+]    @= ?[lemma:óscar];?[lemma:oscar], (?[lemma:de]); (?[lemma:para]),
                                   ?[lemma:melhor], ?[lemma:realizador] .
```

Estas cinco regras dizem respeito, respectivamente, às categorias de "Melhor Actor Secundário", "Melhor Actor Principal", "Melhor Actriz Secundária", "Melhor Actriz Principal" e "Melhor Realizador". Durante as fases intermédias de teste, verificou-se que as categorias "Melhor Actriz/Actor Principal" eram, por vezes, apelidadas de "Melhor Actor/Actriz" justificando a colocação de "principal"

como opcional nas regras para estas duas categorias. Todas as categorias são etiquetadas com "oscar", sendo cada uma delas adicionalmente etiquetada com um acrónimo para a sua categoria.

Para facilitar a escrita de regras de dependência, que são abordadas na próxima secção, criaram-se regras para agrupar tipos de questões. Por exemplo, para as questões que se iniciam com "Como se chama", criou-se a seguinte regra:

```
1> pron[whatHisName=+,interrog=+] @= ?[lemma:como], ?[lemma:se], ?[lemma:chamar] | ?*, punct[lemma:"?"] |.
```

4.4.3 Regras de dependência

As regras de dependência permitem, mediante determinados padrões de questões, a criação de um predicado com um ou mais argumentos. Embora haja a intuição de alguns padrões de questões que poderão surgir, como: "Quem é o realizador de <filme>", "Quem ganhou o óscar de melhor filme em <ano>", entre outras, seria difícil albergar os tipos de questões mais populares só com base na opinião da autora. Nesse sentido, antes de qualquer desenvolvimento, solicitou-se a 10 pessoas que fizessem questões sobre cinema, resultando num *corpus* de 150 perguntas. Com base na análise dessas questões, determinou-se que questões eram mais populares, e teriam que ser tratadas pela aplicação.

Os predicados criados a partir das dependências extraem a informação relevante da questão para que lhe seja dada uma resposta. Alguns predicados têm um número variável de argumentos. Na tabela 4.4 encontra-se a totalidade dos predicados definidos cujo número de argumentos é fixo. Por sua vez, na tabela 4.5 encontram-se os predicados com um número variável de argumentos. Na secção 4.4.3.2 será explicada a necessidade de ter predicados com um número de argumentos variável.

4.4.3.1 Método de emparelhamento

Na figura 4.5 pode-se ver a análise sintáctica da questão "Quem é o realizador de forrest gump?" e abaixo, a regra de dependência utilizada para extrair a informação relevante da questão e colocá-la no predicado respectico, *target_who_directed*.

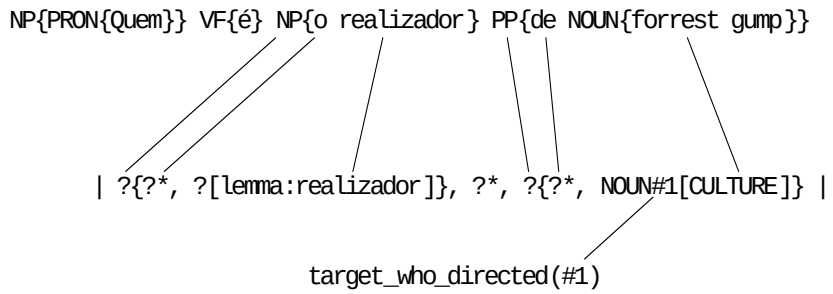

Figura 4.5: Emparelhamento efectuado para a questão "Quem é o realizador de forrest gump?".

O ponto de interrogação (?) significa que pode emparelhar com qualquer tipo de nó: *NP*, *PRON*, *VF*, entre outros. Se estiver seguido de um asterisco (*) significa que é opcional. A parte `?{?*, ?[lemma:realizador]}` emparelha com `NP{o realizador}`, pois está presente uma palavra cujo lema é "realizador" e o artigo "o" emparelha com *?**. Relativamente a `PP{de NOUN{forrest gump}}`, pode-se ver que a palavra "de" encaixa como opcional em *?**; por sua vez, "forrest gump", previamente identificado como *CULTURE*, é emparelhado e colocado na variável *#1*.

Existem outros padrões de perguntas para questionar o sistema acerca do realizador de um filme. Por exemplo, a questão pode ser colocada na passiva da seguinte forma: "forrest gump foi realizado por quem?", ou, em alternativa, "forrest gump foi dirigido por quem?". Na figura 4.6 está a correspondência entre a primeira questão e a seguinte regra de dependência:

```
| NP{?*, NOUN#1[culture]}, VERB[lemma:ser]},
VCPART{PASTPART[lemma:realizar];PASTPART[lemma:fazer];
PASTPART[lemma:dirigir]},
PP{?*, pron[who]} |
target_who_directed(#1)
```

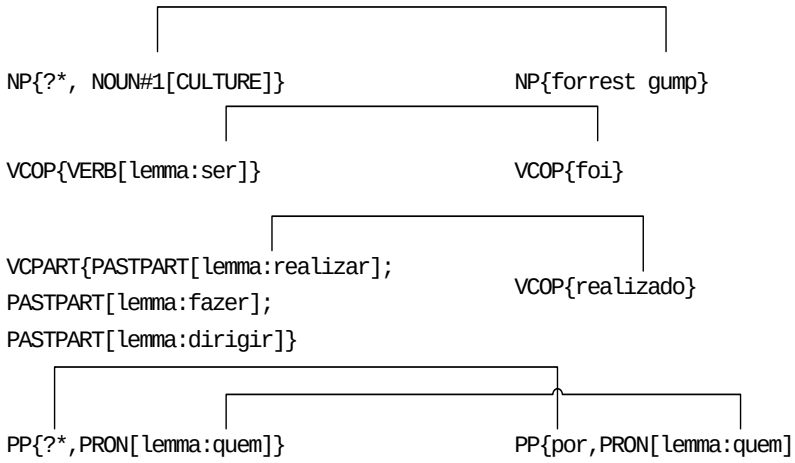

Figura 4.6: Aplicação da regra de dependência para a questão "forrest gump foi realizado por quem?".

Com esta regra de dependência, qualquer variância no tempo verbal de "ser" é válida, ou seja, a questão "forrest gump é realizado por quem?" também seria válida. Outra possibilidade, é a inclusão do artigo "o" antes da questão, estando essa hipótese salvaguardada pelo *?** dentro do nó *NP*.

Analisando agora o predicado *target_who_acts_like_in*, tome-se como exemplo a questão "Quem faz de Clarice Starling no The Silence of the Lambs?". Nela estão presentes duas entidades: o nome do filme "The Silence of the Lambs" e a personagem "Clarice Starling". No entanto, nesta aplicação, apenas são reconhecidos os títulos dos filmes e os nomes de actores, actrizes, realizadores, etc., não sendo, por isso, "Clarice Starling" reconhecido como um só nó. Segue-se a análise sintáctica desta questão e a regra de dependência que a trata:

```
{NP{PRON{quem}} VF{VERB{faz de}}
NP{NOUN{clarice}} AP{starling}
PP{em NOUN{the silence of the lambs}} ?}

| NP{pron[who]}, ?{verb[act_like]}, ?{#1}, ?*,
(PP{?*, NOUN[lemma:filme]}), ?{?*, NOUN#2[culture]} |
target_who_acts_like_in(#1,#2)
```

Na análise sintáctica, "clarice" não está agrupado com "starling". O resultado do processamento da questão será portanto `target_who_acts_like_in(clarice,the silence of the lambs)`. Apesar do nome da personagem não estar completo, é possível responder à questão só com o primeiro nome: basta extrair da base de dados o nome do intérprete da personagem cujo nome inclui "clarice" para o filme "the silence of the lambs". Este tipo de abordagem tem uma grande vantagem: o utilizador não tem que saber o nome completo da personagem, e pode referir-se a esta pelo primeiro ou último nome, ou seja, a questão "quem faz de starling em the silence of the lambs" também seria correctamente respondida. A desvantagem é, no caso de haver mais que uma personagem cujo nome inclua "clarice", serem devolvidos todos esses intérpretes. É o que sucede para o nome "clarice" — existem as personagens: Clarice Starling, Young Clarice Starling e Clarice's Father. Mesmo que o utilizador escreva na questão "Clarice Starling", são disponibilizados os intérpretes das outras duas personagens. Uma vez que não é comum a existência de mais que um personagem num filme com o mesmo nome, optou-se por esta abordagem, pois maximiza a probabilidade da questão ser respondida.

4.4.3.2 Predicados com número variável de argumentos

Abordando agora a necessidade de predicados com número de argumentos variável, tome-se como exemplo a questão "Quem protagoniza The Silence of the Lambs?". Neste caso, pretende-se o actor/actriz que aparece em primeiro lugar na lista de créditos. Neste caso, seria respondido Jodie Foster, uma vez que é considerada a actriz principal. No entanto, a pergunta podia ser formulada como: "Quem é o protagonista de The Silence of The Lambs" e, nesse caso, não faria qualquer sentido responder Jodie Foster, pois o utilizador pretende *o* protagonista. Não deve ser ignorada a especificação do género, pois, se assim for, a resposta pode não corresponder ao pretendido pelo utilizador. Para resolver esta situação, para este padrão de questão, é adicionado o artigo que antecede "protagonista" ao predicado. A questão "Quem é o protagonista de The Silence of the Lambs" seria sintetizada desta forma: `target_who_main_act(o,the silence of the lambs)`. Já para a questão "Quem protagoniza The Silence of the Lambs?", em que o género não é especificado, o predicado resultante seria `target_who_main_act(the silence of the lambs)`.

Esta situação verifica-se para outro tipos de perguntas cujo objectivo seja a obtenção do nome de uma ou mais pessoas. Por exemplo, a questão "Que actor contracena com Tom Hanks em Forrest Gump?" é diferente da questão "Quem contracena com Tom Hanks em Forrest Gump?". A primeira

PREDICADO	OBJECTIVO
target_who_directed(x)	realizador do filme x
target_who_wrote(x)	argumentista(s) do filme x
target_who_produced(x)	produtor(es) do filme x
target_who_acts_in_like(x, y)	intérprete de personagem x no filme y
target_who_won(x,y)	pessoa vencedora da categoria x no ano y
target_which_won(x,y)	filme vencedor da categoria x no ano y
target_who_nominated(x,y)	nomeados para a categoria x no ano y
target_how_many_oscars(x)	número de óscares recebidos por pessoa x
target_how_many_oscars_film(x)	número de óscares recebidos por filme x
target_how_many_nominations(x)	número de nomeações recebidos por pessoa x
target_how_many_nominations_film(x)	número de nomeações recebidas por filme x
target_which_films_directed(x)	filmes realizados por x
target_which_act(x)	filmes em que participa x
target_which_films_main_act(x)	filmes protagonizados por x
target_which_films_main_act_two(x,y)	filmes protagonizados por x e y
target_last_film(x)	último filme de x
target_which_year_film(x)	ano de estreia do filme x
target_which_character(x, y)	personagem interpretado por x no filme y
target_main_character(x)	personagem principal no filme x
target_who_is(x)	informação biográfica de x
target_born_when(x)	data de nascimento de x
target_born_where(x)	local de nascimento de x
target_died_when(x)	data de óbito de x
target_born_where(x)	local de óbito de x

Tabela 4.4: Tabela de predicados com número de argumentos fixo.

procura explicitamente o nome de um actor, enquanto que a segunda é genérica. Para distinguir estas duas situações, o predicado *target_who_acts_with_in* pode ter dois ou três argumentos. No caso da primeira questão, ter-se-ia `target_who_acts_with_in(actor,tom hanks,forrest gump)`, no caso da segunda, ter-se-ia apenas `target_who_acts_with_in(tom hanks,forrest gump)`.

O predicado *target_who_act* sintetiza questões como: "Quem entra no filme Forrest Gump?". Nesse caso, em particular, o predicado seria concretizado da seguinte forma: `target_who_act(forrest gump)`. Contudo, a questão "Que actrizes entram em forrest gump?" é tratada diferenciadamente, sendo o predicado para este caso: `target_who_act(actrizes,forrest gump)`.

As questões "Que actor entra em Sleepless in Seattle e Joe Versus the Volcano?" e "Quem entra em Sleepless in Seattle e Joe Versus the Volcano?" também são diferenciadas no predicado *target_who_acts_in_two*. A primeira deverá ser respondida apenas com actores (género masculino) que entrem em ambos os filmes, neste caso, Tom Hanks. A segunda, deverá ser respondida com todos os actores que entrem em ambos os filmes, ou seja, Tom Hanks e Meg Ryan. Da mesma maneira, a questão "Que actriz entra em Sleepless in Seattle e Joe Versus the Volcano?" teria como resposta Meg Ryan.

PREDICADO	OBJECTIVO
target_who_main_act(x)	protagonista do filme x
target_who_main_act(x,y)	protagonista do filme y em que x especifica o género
target_who_acts_with_in(x,y)	quem contracena com x no filme y
target_who_acts_with_in(x,y,z)	quem contracena com x no filme y em que z especifica o género
target_who_act(x)	intérpretes do filme x
target_who_act(x,y)	intérpretes do filme y em que x especifica o género
target_who_acts_in_two(x,y,z)	quem entra no filme x e no filme y
target_who_acts_in_two(x,y,z)	quem entra no filme y e no filme z em que x especifica o género

Tabela 4.5: Tabela de predicados com número de argumentos variável.

4.4.4 Extracção de informação

O XIP pode devolver o resultado do processamento de texto em diversas formas, uma delas, é a apresentada nas figuras 4.1, 4.2 e 4.3. O resultado pode também ser devolvido num ficheiro XML onde figurem as etiquetas atribuídas às palavras dadas como entrada, a análise sintáctica do texto e também as dependências encontradas. Por exemplo, no ficheiro XML gerado pela questão "Quem contracena com Brad Pitt em Fight Club?" estaria presente o seguinte nó:

```
<DEPENDENCY name="TARGET_WHO_ACTS_WITH_IN">
        <PARAMETER ind="0" num="20" word="brad pitt"/>
        <PARAMETER ind="1" num="21" word="fight club"/>
</DEPENDENCY>
```

Verifica-se que está presente o nome da dependência (ou predicado) encontrada, juntamente com os seus argumentos. Nesse ficheiro XML consta também a etiqueta atribuída a "Brad Pitt" (ACTOR) e a "Fight Club" (CULTURE).

O ficheiro é processado com recurso ao XSLT que, mediante os resultados dos testes efectuados, retorna o nome do *script* que vai tratar o acesso à base de dados, bem como os argumentos que deverá receber. Esta é como que uma fase intermédia entre a análise sintáctica e o acesso à base de dados. Encontra-se abaixo, um excerto do ficheiro XSLT que corresponde a uma condição para detectar dependências no ficheiro XML.

```
<xsl:when test="$frase-actual/DEPENDENCY[@name='TARGET_WHO_ACTS_WITH_IN']">
  <xsl:text>get_from_BD/script-who-acts-in-with.pl TARGET </xsl:text>
  <xsl:choose>
   <xsl:when test="$frase-actual/DEPENDENCY[@name='TARGET_WHO_ACTS_WITH_IN']/PARAMETER[@word='actriz']">
    <xsl:text>"actriz" </xsl:text>
   </xsl:when>
   <xsl:when test="$frase-actual/DEPENDENCY[@name='TARGET_WHO_ACTS_WITH_IN']/PARAMETER[@word='actor']">
    <xsl:text>"actor" </xsl:text>
   </xsl:when>
   <xsl:otherwise>
```

```
    <xsl:text>"indef"</xsl:text>
   </xsl:otherwise>
  </xsl:choose>
  <xsl:call-template name="entidades">
     <xsl:with-param name="phrase" select="$frase-actual"/>
  </xsl:call-template>
</xsl:when>
```

Existe uma condição para cada dependência definida. Esta, em particular, devolveria para a questão "Quem contracena com Brad Pitt em Fight Club?", a seguinte expressão:

```
get_from_BD/script-who-acts-in-with.pl TARGET "indef" ACTOR 'brad pitt ' CULTURE 'fight club '
```

A primeira *string* é o nome do *script* que deverá ser executado, "indef" significa que o objectivo da resposta é indefinido.

Se a questão fosse "Quem é o actor que contracena com Brad Pitt em Fight Club?", a dependência estaria explícita no ficheiro XML da seguinte forma:

```
<DEPENDENCY name="TARGET_WHO_ACTS_WITH_IN">
        <PARAMETER ind="0" num="2" word="actor"/>
        <PARAMETER ind="1" num="22" word="brad pitt"/>
        <PARAMETER ind="2" num="23" word="fight club"/>
</DEPENDENCY>
```

Neste caso, o "TARGET" já não é indefinido, pelo contrário, está explícito que se pretende um actor. Assim sendo, o XSLT devolveria:

```
get_from_BD/script-who-acts-in-with.pl TARGET "actor" ACTOR 'brad pitt ' CULTURE 'fight club '
```

4.5 Problemas de interpretação

Durante a fase de desenvolvimento e testes, surgiram alguns problemas ao nível da interpretação das questões, alguns dos quais referidos na secção 2.3.2. A análise dos problemas surgidos será feita nas subsecções seguintes.

4.5.1 Ambiguidades

O problema da ambiguidade marcou presença ao longo do desenvolvimento da aplicação. Por exemplo, a questão "Que filmes fez Woody Allen?" era, inicialmente, interpretada como "Que filmes realizou Woody Allen?", contudo, durante a fase de testes surgiu, por diversas vezes, essa mesma questão mas referindo-se a actores. A resposta dada era "Nenhum filme.", o que desiludia os utilizadores. A solução

encontrada para este problema foi devolver todos os filmes em que determinada pessoa participou e a sua função. Por exemplo, a questão "Que filmes fez Woody Allen?" passou a ser respondida da seguinte forma:

```
Como actor:
        Scoop (2006)
        Anything Else (2003)
        Hollywood Ending (2002)
        (...)
Como realizador:
        Woody Allen Spanish Project (2008)
        Cassandra's Dream (2007)
        Scoop (2006)
        (...)
(...)
```

Outra ambiguidade surgiu com a questão: "Quantos óscares ganhou Amadeus?" que, na altura, foi respondida com "Nenhum óscar.". Efectivamente, o filme "Amadeus" de 1984 venceu diversos óscares, logo a resposta não estava correcta. A explicação reside no facto de "Amadeus" ser não só um título de filme, mas também o nome de um actor. Se a questão fosse "Quantos óscares ganhou o filme Amadeus?", o sistema não contemplaria "Amadeus" enquanto nome de actor. Uma vez que, para a questão inicial, não é possível saber se o utilizador se refere ao filme ou à pessoa, optou-se por dar a resposta tanto para a entidade enquanto pessoa, como para entidade enquanto filme.

A questão "Quem é o autor de Gladiator?" surgiu durante a fase de testes e, na altura, não foi respondida pois não havia dependência que a tratasse. Esta questão é ambígua, no sentido em que "o autor" tanto pode ser considerado "o realizador" como o "argumentista". Contudo, uma vez que "autor" remete para a escrita, decidiu-se que a interpretação mais correcta é a segunda.

Quando a pergunta é "Quem interpreta x no filme y?", o que é x? Quererá o utilizador saber o nome do actor que interpreta x no filme y ou, ao invés, quer saber o nome do personagem interpretado pelo actor x no filme y? Este problema surgiu durante a fase de testes quando um utilizador formulou a questão "quem interpreta Wladyslaw Szpilman no filme The Pianist?" e não obteve resposta. A aplicação encarou "Wladyslaw Szpilman" como sendo o nome de um actor e respondeu "Wladyslaw Szpilman não entra no filme The Pianist". O utilizador procurava sim saber o nome do actor que interpretava "Wladyslaw Szpilman" no filme "The Pianist". Uma forma de resolver este problema era simplesmente escolher uma das interpretações, mas entendeu-se que qualquer das duas estaria correcta, logo, o mais adequado seria suportar ambas. Assim, se o nome a seguir ao verbo "interpretar" for reconhecido como sendo o nome de um actor, verifica se o actor faz parte do elenco do filme e devolve

o nome do personagem interpretado. Se não for reconhecido, o sistema procura por aquele nome de personagem ligado àquele filme, e devolve o nome do actor (ou actriz) que o interpreta. Desta forma, a questão "quem interpreta Wladyslaw Szpilman no filme The Pianist?" devolve "Adrien Brody" e a questão "quem interpreta Adrien Brody no filme The Pianist?" devolve "Wladyslaw Szpilman".

4.5.2 Conjunção e Disjunção

O problema da conjunção e disjunção surgiu em questões como "Quem entra em Big e Philadelphia?": pretende o utilizador saber o elenco do filme "Big" *e* o elenco do filme "Philadelphia" (caso da disjunção) ou, pelo contrário, pretende saber quem são as pessoas que entram em ambos os filmes (caso da conjunção)?. Ambas as interpretações são válidas, no entanto, assumiu-se que a interpretação correcta para este padrão de questão seria a segunda. Considerou-se que não é comum querer obter os elencos de dois filmes na mesma questão, logo, quando são referidos dois títulos com a conjunção copulativa "e", faz-se o cruzamento entre os dois elencos e facultam-se os intervenientes em comum.

Como trabalho futuro, poderá ser dada a possibilidade de se escolher uma das interpretações possíveis.

4.5.3 Erros ortográficos

Por vezes não é possível interpretar determinadas questões pela existência de erros ortográficos. Nalguns casos, os erros não afectam a interpretação, no entanto, se ocorreram em palavras essenciais para o objectivo da questão (como "realizador", "protagonista", "actor", "óscar", entre outras), podem impedir o seu processamento.

Um dos erros mais frequentas tem que ver com a (falta de) acentuação. A palavra "óscar" é uma das "vítimas" mais frequentes. Como descrito na secção 4.4.2.1, tentou-se prevenir esse facto aceitando as expressões "oscar de melhor...", no entanto, não se preveniu essa situação para as questões do tipo "Quantos oscares recebeu...". O padrão definido para tratar esse tipo de questões, espera a palavra "óscar" e não "oscar", falhando a interpretação por esse motivo.

5 Avaliação

5.1 Introdução

Durante o desenvolvimento da aplicação, foram recolhidas 355 questões. Contudo, após filtragem das questões idênticas e eliminação de questões irrelevantes para os testes (como "Quem fez este sistema?", "Quem és tu?"), resultaram 198 questões. O método utilizado para a recolha foi a disponibilização da aplicação através da *Web* e o armazenamento das questões formuladas e respostas dadas pelo sistema em base de dados. Dessa forma, foi possível detectar erros na aplicação em todas as fases de desenvolvimento, permitindo o seu constante aperfeiçoamento.

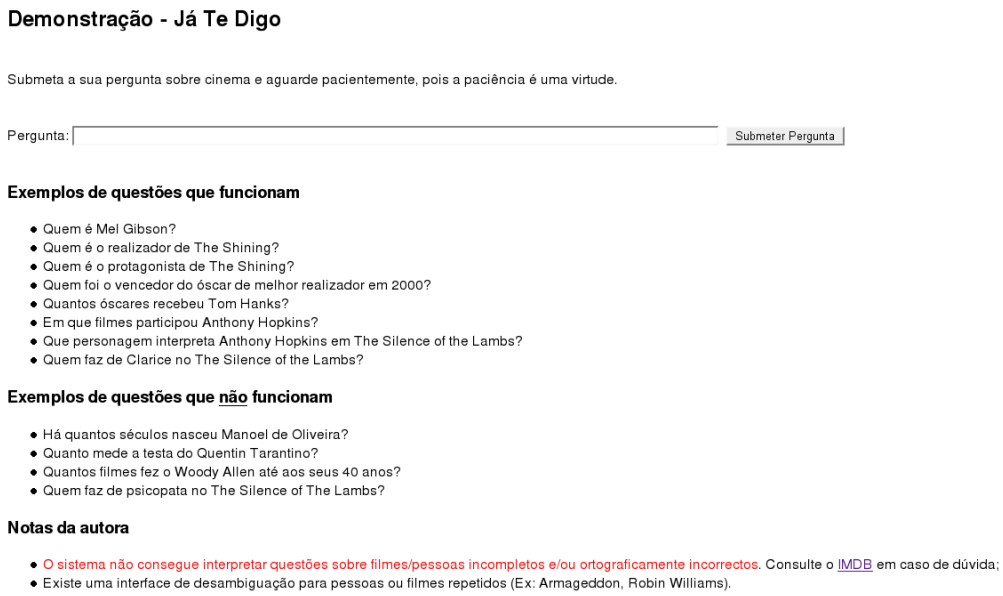

Figura 5.1: Interface com questões-exemplo.

Se se avaliasse a sua eficácia com base apenas nas questões recolhidas durante a fase de desenvolvimento, não se estaria a conhecer o verdadeiro desempenho da aplicação, pois, de certa forma, o desenvolvimento acabou por estar "orientado" a essas questões. Assim, após a definitiva conclusão do desenvolvimento, solicitou-se a realização de 10 questões a 5 utilizadores, perfazendo um total de 50 questões. Para esse teste, utilizou-se a interface *Web* em que estão presentes alguns exemplos de perguntas às quais a aplicação consegue responder (figura 5.1). Esses exemplos acabam por "guiar" o utilizador, ajudando-o a perceber como deve formular as suas questões. Por esse motivo, realizou-se um

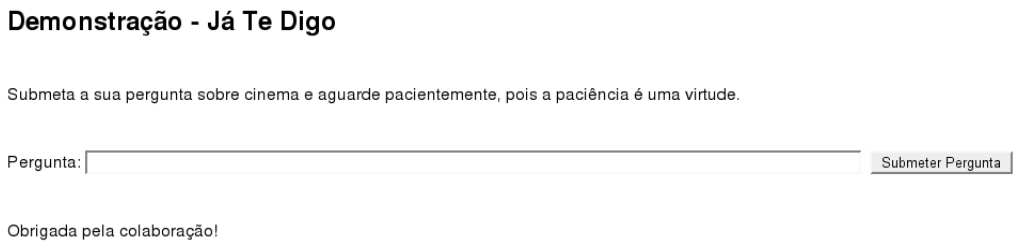

Figura 5.2: Interface sem questões exemplo.

outro teste: a realização de 10 questões por parte de 5 outros utilizadores, sem qualquer contacto com a aplicação na fase de desenvolvimento, e perante uma interface sem quaisquer exemplos de questões (figura 5.2). Dessa forma, foi possível perceber o impacto dessas questões-exemplo na eficácia do sistema.

Nas secções seguintes serão analisados os resultados para as questões angariadas durante a fase de desenvolvimento, seguido de uma comparação entre os resultados para a interface com questões-exemplo e os resultados para a interface sem qualquer tipo de "ajudas".

5.2 Resultados para as questões recolhidas durante a fase de desenvolvimento

Aquando da finalização do desenvolvimento da aplicação, realizou-se um teste final utilizando todas as questões recolhidas durante a fase de desenvolvimento. Dessas 198 questões, 41 não foram interpretadas pela aplicação. Segue-se a análise das razões pelas quais foi não possível a sua interpretação, seguida de alguns exemplos.

5.2.1 Ausência de tratamento

A ausência de tratamento a algumas questões formuladas foi responsável por 23 questões não respondidas. Infelizmente, a aplicação não está preparada para tratar todo o tipo de questões de cinema que podem ser feitas. Como é natural, o tratamento de novas questões faz parte do trabalho futuro a desenvolver, logo o leque de questões a tratar pode ser facilmente alargado. Há, no entanto, algumas questões que são demasiado específicas, cujo tratamento é difícil de implementar, por exemplo, "Que filmes realizaram os irmãos Cohen?", ou então, "Que filmes realizaram os irmãos Wachowski?". O conceito de "irmãos" não é abrangível pela base de dados. "Quem é o herói de The Matrix?" é, também, uma questão difícil de tratar. A palavra "herói" não é necessariamente sinónimo de protagonista, da mesma forma que o "vilão" não é antónimo de protagonista, nem sinónimo de actor secundário. Outro problema comum surgido nesta fase foi o tratamento de personagens de filmes de animação não pelo seu nome, mas pelo seu "tipo". Por exemplo, "Quem faz de peixe-palhaço no Finding Nemo?" ou "Quem

faz de burro no Shrek?" são questões impossíveis de responder, pois os nomes das personagens não são, muitas vezes, o animal que encarnam. De qualquer forma, a designação "oficial" da personagem, no caso de ser um animal, só será a portuguesa no caso do filme ser falado em Português, o que diminui ainda mais a probabilidade de uma questão deste género ser respondida.

Há, no entanto, alguns padrões que poderão ser implementados, como o tratamento de questões relacionadas com: duração dos filmes, nacionalidades de pessoas, locais de filmagem, entre outros dados. Toda essa informação está disponível no IMDB, por isso pode facilmente ser realizada como trabalho futuro.

5.2.2 (Não) reconhecimento de entidades mencionadas

A limitação de conseguir responder apenas a questões que contenham os títulos de filmes e nomes de pessoas escritos correctamente, é responsável por 13 questões não interpretadas. Por exemplo, a questão "Que filmes fez Jesen Akles?" não foi respondida pois não existe esse nome na base de dados, existe sim "Jensen Akles". Outro exemplo é a questão "Quem contracena com David Duchovny em X-Files?" que, à partida, consegue ser respondida, mas como o título do filme é "X Files" e não "X-Files", não é interpretada.

Não só a incorrecção ortográfica impede o reconhecimento de entidades mencionadas. Se o nome da pessoa ou título de filme estiver incompleto, também não é reconhecido. É o que acontece na questão "Que filme realizou Wachowski?": se o utilizador tivesse escrito "Andy Wachowski" ou "Larry Wachowski", a questão teria sido respondida, como não existe ninguém na base de dados cujo nome seja somente "Wachowski", a questão não foi respondida.

Pelo facto de haver na interface um aviso quanto à impossibilidade de interpretar a questão no caso das entidades estarem incorrectamente escritas e/ou incompletas, o número de questões não respondidas devido a este facto, acabou por ser inferior do que se estaria à espera.

5.2.3 Outros motivos

A incorrecção ortográfica e também a ocorrência de erros tipográficos (*typo*) são responsáveis pelos restantes erros da aplicação. Por exemplo, "quem é o personagem princípal de the grudge?" é uma questão que, aparentemente, está correcta. Contudo, por "princípal" estar mal escrito, o analisador não consegue emparelhar com o padrão que trata este tipo de questões pois está à espera de "principal" e não "princípal".

5.2.4 Questões incorrectamente respondidas

Algumas questões não foram correctamente respondidas pela aplicação, ou, pelo menos, não foram respondidas de acordo com o que seria esperado. Como exemplo, várias questões foram formuladas acerca de "Hoody Allen":

- Quando estreou o último filme do Hoody Allen?;
- Qual é o último filme do Hoody Allen?;
- Quantos filmes fez o Hoody Allen?;
- O Hoody Allen participou em que filmes?.

Todas estas questões foram respondidas para o actor "Allen", pois foi a correspondência exacta que a aplicação conseguiu encontrar. No entanto, apesar de ter havido uma resposta, o pretendido não era a informação sobre o actor "Allen", mas sim sobre o "Woody Allen". Considera-se assim, que as respostas dadas às questões acima estavam erradas.

Como outro exemplo de resposta errada, tem-se a questão "em que filmes participou dan ackroyd?" que foi respondida para o actor "Dan", uma vez que "Dan Aykroyd" seria o nome correcto do actor.

No caso da questão "quem faz de indiana jones no indiana jones?", mais uma vez a resposta dada estava errada. A aplicação interpretou que se pretendia o intérprete da personagem "indiana" no filme "jones", uma vez que foi o título que foi possível identificar na base de dados. Com efeito, não existe nenhum filme denominado "Indiana Jones" — os filmes dessa saga denominam-se "Indiana Jones e a grande cruzada" e "Indiana Jones e o templo perdido" — sendo impossível responder de acordo com as expectativas.

Da totalidade das 157 questões respondidas, 7 foram respondidas incorrectamente.

5.2.5 Síntese

A tabela 5.1 representa a síntese dos resultados obtidos nesta fase de testes.

5.3 Comparação entre resultados para interface com e sem questões-exemplo

Os exemplos de questões às quais a aplicação consegue responder correctamente, bem como um "aviso" quando à incapacidade de responder a questões em que os títulos de filmes e nomes de pessoas estejam

			Totais	
Respondidas	Correctamente	150 (∼95,5%)	157 (∼79,3%)	198
	Incorrectamente	7 (∼4,5%)		
Não Respondidas	Inexistência de Tratamento	23 (∼56,1%)	41 (∼20,7%)	
	Incorrecto NER	13 (∼31,7%)		
	Outros Motivos	5 (∼12,2%)		

Tabela 5.1: Tabela de resultados para as questões recolhidas durante a fase de desenvolvimento

incorrectamente escritos, tem grande influência face à sua eficácia. Os resultados obtidos reflectem isso mesmo: das 50 questões formuladas na interface com questões-exemplo, 33 foram respondidas; para a interface sem questões-exemplo, o resultado é 20.

Os pricipais motivos por detrás da incapacidade de interpretar algumas questões mantém-se relativamente aos analisados para as questões recolhidas durante a fase de desenvolvimento. Para discernir o impacto dos exemplos de questões na eficácia da aplicação, vejam-se as tabelas 5.2 e 5.3.

			Totais	
Respondidas	Correctamente	29 (∼87,9%)	33 (66,0%)	50
	Incorrectamente	4 (∼12,1%)		
Não Respondidas	Sem Tratamento	5 (∼19,4%)	17 (34,0%)	
	Incorrecto NER	0 (∼0,0%)		
	Outros Motivos	12 (∼70,6%)		

Tabela 5.2: Tabela de resultados para as questões realizadas na interface com questões-exemplo.

			Totais	
Respondidas	Correctamente	18 (90,0%)	20 (40,0%)	50
	Incorrectamente	2 (10,0%)		
Não Respondidas	Sem Tratamento	15 (50,0%)	30 (60,0%)	
	Incorrecto NER	9 (30,0%)		
	Outros Motivos	6 (20,0%)		

Tabela 5.3: Tabela de resultados para as questões realizadas na interface sem questões-exemplo.

Verifica-se que a eficácia global para a interface com questões-exemplo é bastante maior que para a interface simples. Para a primeira, a aplicação conseguiu responder a 33 questões, enquanto que na segunda, apenas 20 foram respondidas. É também visível, ao nível das questões não interpretadas por não estar previsto o seu tratamento, bastante diferença percentual de um para outro teste — para a interface com questões exemplo, a percentagem é de 19,4%, enquanto que para a outra interface o resultado é de 50%. Verifica-se que os utilizadores são "influenciados" pelas questões dadas como exemplo, no sentido

em que formulam questões iguais ou semelhantes às apresentadas. Os exemplos dados são também importantes na prevenção de erros ortográficos que, por vezes, impedem a interpretação da questão. A palavra "óscares" figura numa das questões-exemplo e poderia evitar as 5 questões não interpretadas devido à sua incorrecção ortográfica ("oscars", "óscars" e "oscares" foram as variantes apresentadas).

6 Conclusão e Trabalho Futuro

6.1 Resumo

O JáTeDigo é uma interface em Português para uma base de dados de cinema. A sua arquitectura envolve diversos passos pela seguinte ordem: reconhecimento de entidades mencionadas, desambiguação, processamento de língua natural e acesso à base de dados. Tendo em conta as várias abordagens para o desenvolvimento de ILNBD descritas na secção 2.4, considera-se que a aplicação foi desenvolvida tendo por base a utilização (e concepção) de uma linguagem de representação intermédia.

A interface utilizada é uma página *Web* simples, com uma caixa de texto onde é inserida a questão em língua natural. Após a submissão da questão, é devolvida a resposta ao utilizador, podendo haver uma fase intermédia de desambiguação de entidades no caso de existir, em base de dados, repetição dos nomes encontrados. Na interface figuram ainda exemplos de questões que a aplicação consegue interpretar para que o utilizador tenha a intuição, por um lado, das suas potencialidades, e por outro, das suas limitações.

O desempenho da aplicação está bastante restringido pela necessidade de escrever as entidades correctamente. Contudo, face aos resultados obtidos, verifica-se que a grande parte das questões não respondidas se devem a erros ortográficos e questões cujo tratamento não está previsto. O problema dos erros ortográficos pode ser resolvido recorrendo a um corrector ortográfico. A adição do tratamento para novas questões é um trabalho progressivo que poderá ser realizado no futuro.

O impacto das questões-exemplo na interface é notório ao nível dos resultados. Para uma aplicação deste género, é importante que figurem alguns exemplos de utilização, de forma a que o utilizador perceba como a aplicação deve ser utilizada.

6.2 Contribuições

As contribuições principais deste trabalho são as seguintes:

- Proposta de uma arquitectura genérica para ILNBDs;
- Concretização dessa arquitectura para uma aplicação no domínio de cinema e sua avaliação;

- Utilização e análise da adequabilidade de ferramentas de processamento de língua natural a este tipo de aplicações;
- Desenvolvimento de *scripts* que garantem a limpeza e coerência da informação na base de dados.

A arquitectura desenvolvida pode, facilmente, ser utilizada para outro domínio de aplicação: Literatura, Geografia, História, entre outros. A partir de uma dada base de dados com informação sobre um destes domínios, basta criar as dependências que irão captar os padrões de questões e, também, criar os *scripts* que obtêm a informação pretendida na base de dados. Para os domínios da Literatura e História, pode ser aplicado o método de reconhecimento de entidades mencionadas através da base de dados.

A utilização assídua da ferramenta de processamento da língua natural presente na arquitectura da aplicação, permitiu a detecção de alguns erros contribuindo, assim, para a sua melhoria. A grande vantagem desta ferramenta é a análise linguística profunda que consegue fazer, bem como a possibilidade de ser facilmente alterada por forma a satisfazer os requisitos da interface desenvolvida. Tem, no entanto, a desvantagem de "adivinhar" a classificação morfológica das palavras originando alguns "fenómenos" estranhos . Por exemplo, na questão "Quem realiza Eyes Wide Shut?", "eyes" é considerado uma forma verbal do verbo "eyar" e "wide" como forma verbal de "widir". Contudo, é inegável que esta ferramenta constituiu um forte aliado e não um estorvo.

A constituição da base de dados com informação de cinema foi um trabalho árduo pois envolveu muito tempo dada a grande quantidade de dados. Mas, nesta altura, foram feitas adições e correcções aos *scripts* que permitem, no futuro, manter uma base de dados mais coerente e "limpa".

6.3 Trabalho Futuro

A aplicação pode ser melhorada a diversos níveis, sendo cada um deles detalhado nas seguintes subsecções.

6.3.1 Disponibilização de mais dados

Actualmente, só estão disponíveis informações sobre o elenco dos filmes, prémios recebidos e informação biográfica. O IMDB disponibiliza mais informação sobre cinema que vale a pena ser incluída: locais de filmagem, duração dos filmes, classificação etária, linguagem em que são falados, etc.. A inclusão desta informação é simples, pois os *scripts* que adicionam esta informação na base de dados estão prontos. O trabalho a realizar nesta área reside na interpretação da questão, em que terão que ser adicionadas as dependências que tratam padrões de questões como:

- Qual é a duração de <filme>?

- Em que língua é falado <filme>?
- Qual é a nacionalidade de <pessoa>?

Deverão também ser adicionados os vencedores e nomeados para os restantes óscares: "Óscar de Melhor Guarda-Roupa", "Óscar de Melhores Efeitos Especiais", "Óscar de Melhor Argumento Original", "Óscar de Melhor Argumento Adaptado", entre outros. Desta forma, a resposta a questões como "Que óscares ganhou <filme>?" ou "Que óscares ganhou <pessoa>?" será mais completa.

Actualmente, só está a ser dada informação sobre filmes de cinema. Essa opção foi tomada, pois, por vezes, ao questionar a aplicação acerca dos filmes em que entrava determinada pessoa, se obtinha como resposta "37th Oscar Academy Awards" ou "10th MTV Movie Awards" em conjunto com os outros filmes. No entanto, desta forma exclui-se a informação sobre séries televisivas. Numa altura em que as séries televisivas ganham particular importância, esta opção deve ser reformulada.

6.3.2 Correcção ortográfica

Algumas questões não foram respondidas por haver erros ortográficos na questão. Palavras como "realisar", "oscars", "portagonista" não são reconhecidas pelo analisador morfo-sintáctico, boicotando a possibilidade de interpretar a questão. A (falta de) acentuação é, também, responsável por algumas questões não interpretadas.

A integração de um corrector ortográfico ajudaria a resolver este problema, contribuindo para uma maior eficácia da aplicação.

6.3.3 Exactidão na escrita de títulos e nomes

Não é fácil escrever com total exactidão os títulos de filmes e nomes de pessoas, especialmente se estiverem escritos numa linguagem que não a nativa do utilizador. Pensava-se que a inclusão dos títulos em Português pudesse mitigar um pouco esta situação, mas mesmo quando se trata de títulos em Português, os utilizadores não os sabem escrever com exactidão. Um dos erros frequentes é a omissão do artigo no início do título, ou seja, os utilizadores escrevem "Lista de Schindler" e não "A Lista de Schindler", ou então "Aeroplano" em vez de "O Aeroplano".

Justifica-se uma nova abordagem para resolver este problema. Uma forma possível é solicitar ao utilizador que coloque os nomes de pessoas e, em especial, os títulos de filmes entre aspas (" "). Assim, é possível isolar as entidades na frase e dar sugestões ao utilizador de outros títulos (ou nomes) no caso das entidades entre aspas não estarem na base de dados. Adicionalmente, poderão ser construídas tabelas com os títulos e nomes mais populares com base, justamente, nos títulos e nomes premiados e

nomeados nos óscares. Desta forma, se uma determinada entidade entre aspas não se encontrar na base de dados, consulta-se primeiramente essas tabelas onde, dada a sua pequena dimensão comparativamente com as tabelas com todos os títulos e todas as pessoas, mais rapidamente se pode encontrar o título ou nome a que o utilizador se pretendia referir.

6.3.4 Tratamento de elipse e anáfora

Como foi dito na secção 2.3.2.3 e 2.3.2.4, estas figuras de estilo são frequentemente usadas no nosso dia-a-dia, logo, justifica-se o seu tratamento para uma interface deste género. Das perguntas recolhidas, nenhuma fez uso de elipse ou anáfora, talvez porque os utilizadores na altura não tinham grande confiança na aplicação, logo não experimentaram essa vertente. De qualquer forma, é uma melhoria a fazer no futuro uma vez que acrescenta bastante valor à aplicação.

6.3.5 Integração com um sistema de QA

Para as questões cuja resposta não se encontra em base de dados, ou que, simplesmente não se conseguem interpretar, pode recorrer-se a uma "entidade externa", por exemplo, um sistema de QA como o *QA@L²F* (Mendes, 2007). Seria uma mais-valia para a aplicação, pois maximizaria a possibilidade de encontrar a resposta à questão do utilizador.

6.4 Observações Finais

As ILNBD são interfaces para bases de dados caracterizadas pela sua fácil utilização e expressividade. Existem diversas abordagens para o seu desenvolvimento: o emparelhamento, o recurso à sintaxe, à semântica e ainda a linguagens de representação intermédia. Os sistemas mais actuais acabam por usar um pouco de cada uma das abordagens.

Houve um grande desenvolvimento nesta área nos anos 80 mas, posteriormente, verificou-se um certo desinteresse por parte da comunidade científica. A aplicação comercial destas interfaces não foi muito bem sucedida, havendo inúmeros sistemas resultantes de investigação académica, mas poucos desenvolvidos com aplicação no mercado. Contudo, o interesse relativamente a esta área tem crescido, nomeadamente por parte da indústria dos telemóveis. Devido à pouca área disponível nos ecrãs destes dispositivos móveis, as interfaces em língua natural constituem uma alternativa apelativa como forma de navegação. Como exemplo disso, há a recente parceria entre o MIT e a NOKIA para o desenvolvimento de uma interface em língua natural para telemóveis com base no sistema START.

Para além da potencialidade destas interfaces no contexto dos telemóveis, existem outras oportunidades. Tem havido um crescente interesse por parte da comunidade sénior na Internet. Para os mais

idosos, existe alguma dificuldade em lidar com interfaces como o rato ou o *touch pad*. O preenchimento de formulários ou o recurso a interfaces gráficas não é linear para quem não domina o uso destas duas interfaces pessoa-máquina. Já o uso do teclado é mais intuitivo, pois é baseado na metáfora da máquina de escrever. Pode-se assim pensar nas interfaces em língua natural como sendo os meios de interacção ideais para os mais idosos.

Apesar de este documento se basear no processamento da língua escrita, não se pode cingir o conceito de interface em língua natural a uma interacção pessoa–teclado. Assim que se consiga transpôr a língua falada para texto escrito com extrema precisão, as interfaces em língua natural tornar-se-ão mais facilmente utilizáveis. Dadas as vastas potencialidades destas interfaces e ao contínuo interesse pelo processamento da língua falada, são previsíveis maiores desenvolvimentos nesta área no futuro.

Uma aplicação como a desenvolvida nunca está terminada. Há sempre melhoramentos a fazer aqui e ali, sempre pequenos detalhes que podem ser aperfeiçoados.

A própria natureza da língua natural implica que o trabalho a desenvolver nesta aplicação nunca esteja terminado. A toda a hora surgem novas palavras, novas expressões, novas maneiras de elaborar questões. Nunca se chegará ao ponto de dizer: "Não há questão a que o sistema não consiga responder", pois surgirá sempre uma qualquer questão que destrone essa presunção.

Apêndice I

A Detalhes da Arquitectura

Na figura A.1 encontra-se o esquema da arquitectura concebida para esta aplicação.

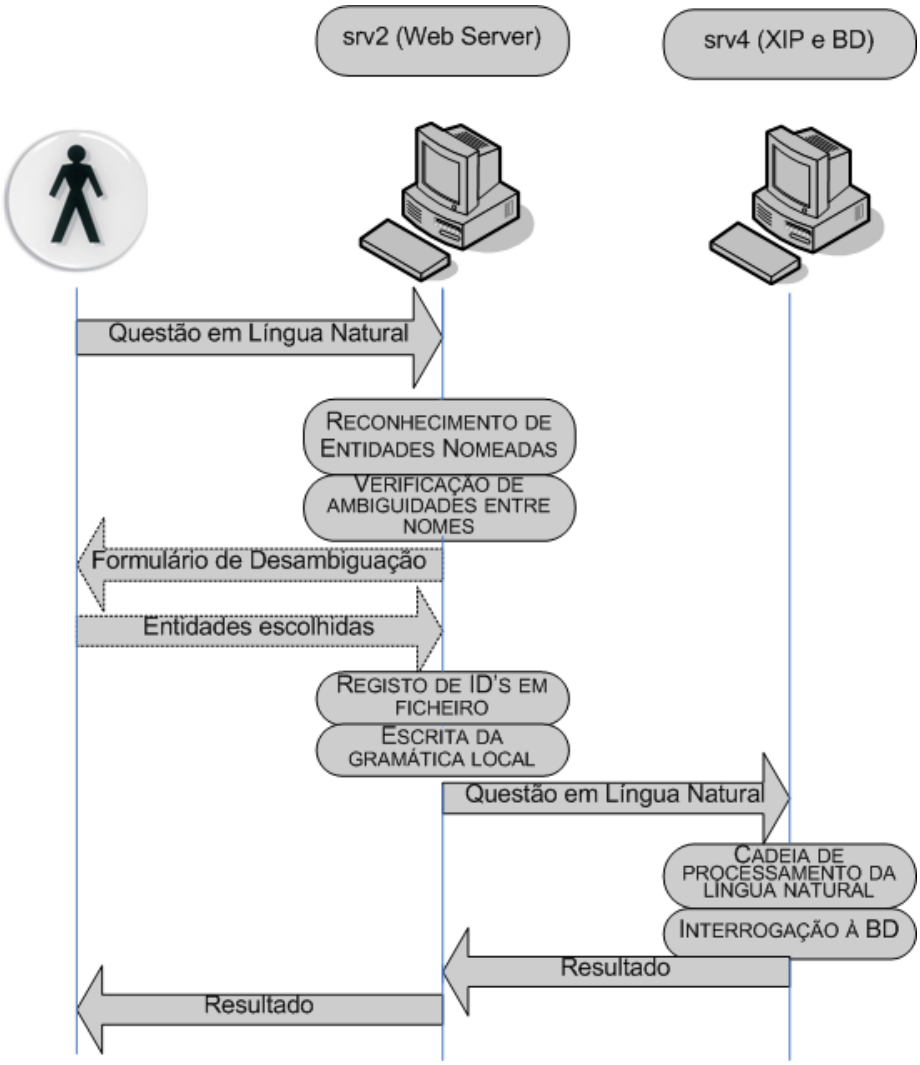

Figura A.1: Arquitectura global da aplicação.

A arquitectura envolve duas máquinas distintas: a srv2 e a srv4. A primeira cumpre a função de *Web Server* enquanto que a segunda executa o processamento da língua natural e consulta a base de dados. Não haveria qualquer impedimento para tudo ser feito na mesma máquina, no entanto, não se quis sobrecarregar o *Web Server* com a execução da cadeia de processamento de língua natural.

O preenchimento do formulário na página *Web* seguido da sua submissão, desencadeia todo o processo. Inicialmente, são identificadas as entidades mencionadas na questão formulada, sejam estas títulos de filmes ou nomes de pessoas. A fase de desambiguação pode ou não ocorrer, dependendo da questão formulada. Em qualquer dos cenários, são registados num ficheiro os identificadores únicos (*ID's*) dos filmes e pessoas reconhecidos. Este procedimento simplifica as interrogações SQL realizadas para obter a resposta.

As entidades identificadas são também registadas num ficheiro que vai ser usado como gramática local na fase de processamento da língua natural.

Após estas fases, é passado o controlo para a outra máquina: a srv4. Para tal, recorre-se ao protocolo SOAP para invocar uma função definida na srv4 que recebe como argumento a questão e, após o processamento da questão e consulta à base de dados, retorna a resposta. A função invocada executa a cadeia de processamento de língua natural cujo *output* é um ficheiro XML que contém informação sintáctica e semântica sobre a questão e um nó especial que identifica o tipo de pergunta e os argumentos necessários para a responder. Para obter esse nó, recorre-se à linguagem XSLT que, mediante o nome do nó encontrado, devolve o nome do script que deverá obter a resposta e os argumentos que necessita para o conseguir. Esta fase será analisada com mais detalhe no capítulo 4.

Bibliography

Androutsopoulos, I., Ritchie, G., & Thanisch, P. (1993). MASQUE/SQL: An efficient and portable natural language query interface for relational databases. In *Proc. of the sixth international conference on industrial and engineering applications of artificial intelligence and expert systems iea/aie-93.* Edinburgh, Scotland.

Androutsopoulos, I., Ritchie, G., & Thanisch, P. (1995). Natural language interfaces to databases–an introduction. *Journal of Language Engineering*, *1*(1), 29–81.

Auxerre, P., & Inder, R. (1986). MASQUE: Modular answering system for queries in english - user's manual. In A. A. Institute (Ed.), *Tech. rep. aiai/sr/10.*

Ballard, B. W. (1986). User specification of syntactic case frames in TELI, a transportable, user-customized natural language processor. In *Proceedings of the 11th coling* (pp. 454–460).

Ballard, B. W., Lusth, J. C., & Tinkham, N. L. (1984). LDC-1: a transportable, knowledge-based natural language processor for office environments. *ACM Trans. Inf. Syst.*, *2*(1), 1–25.

Barros, F. A., & DeRoeck, A. (1994). Resolving anaphora in a portable natural language front end to databases. In *Proceedings of the fourth conference on applied natural language processing* (pp. 119–124). San Francisco, CA, USA: Morgan Kaufmann Publishers Inc.

Bates, M., Moser, M. G., & Stallard, D. (1986). The IRUS transportable natural language database interface. In *Proceedings from the first international workshop on expert database systems* (pp. 617–630). Redwood City, CA, USA: Benjamin-Cummings Publishing Co., Inc.

Bobrow, R. (1978). The RUS system. In *Research in natural language understanding.* BBN Report Nº 3837.

Bourzac, K. (n.d.). (`http://www.technologyreview.com/read\_article.aspx?id\=16745\&ch\=infotech/`)

Burger, J. F. (1980). Semantic database mapping in EUFID. In *Sigmod '80: Proceedings of the 1980 acm sigmod international conference on management of data* (pp. 67–74). New York, NY, USA: ACM Press.

Cimiano, P. (2004). ORAKEL: A natural language interface to an f-logic knowledge base. In *Proceedings of the 9th international conference on applications of natural language to information systems* (p. 401-406). Springer.

Damerau, F. J., Petrick, S. R., Pivovonsky, M., & Plath, W. J. (1982). Transformational question-answering (TQA) system: IBM, yorktown heights. *SIGART Bull.*(79), 62–64.

E.F.Codd. (1974). Seven steps to RENDEZVOUS with the casual user. In *IFIP working conference database management*.

Epstein, S. S. (1985). Transportable natural language processing through simplicity, the PRE system. *ACM Trans. Inf. Syst.*, *3*(2), 107–120.

Filipe, P. P. (1999). *Sistema de interrogações em língua natural para bases de dados: Modelo conceptual, aquisição de vocabulário e tradução*. Unpublished master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Green, W. A. K. C. C. L. K., Bert F. Jr. (1961). Baseball: an automatic question answerer. In (pp. 39–45).

Grosz, B. J. (1983). TEAM, a transportable natural language interface system. In *ACL proceedings, conference on applied natural language processing* (pp. 39–45).

Hafner, C. D., & Godden, K. (1985). Portability of syntax and semantics in DATALOG. *ACM Trans. Inf. Syst.*, *3*(2), 141–164.

Hancox, P. (n.d.). (`http://www.cs.bham.ac.uk/~pjh/sem1a5/pt1/pt1\_history.html`)

Harris, L. R. (1977). User oriented data base query with the ROBOT natural language query system. *International Journal of Man-Machine Studies*, *9*(6), 697-713.

Harris, L. R. (1978). The ROBOT system: Natural language processing applied to data base query. In *Acm annual conference (1)* (p. 165-172).

Harris, L. R. (1988). Experience with INTELLECT: artificial intelligence technology transfer. 468–475.

Hendrix, G. G., Sacerdoti, E. D., Sagalowicz, D., & Slocum, J. (1978). Developing a natural language interface to complex data. *ACM Trans. Database Syst.*, *3*(2), 105–147.

Hirschman, L., & Gaizauskas, R. (2001). Natural language question answering: the view from here. *Nat. Lang. Eng.*, *7*(4), 275–300.

Inc., N. L. (1986). *Natural language retrieval of database information.*

JL Binot, D. S. B. V., L Debille. (1991). *Natural language interfaces: A new philosophy.* SunExpert Magazine.

Katz, B., & Lin, J. (2002). Annotating the semantic web using natural language. In *Nlpxml '02: Proceedings of the 2nd workshop on nlp and xml* (pp. 1–8). Morristown, NJ, USA: Association for Computational Linguistics.

Li, Y. (2005). *NaLIX: an interactive natural language interface for querying XML.*

Lopes, G. P. (1984). Transforming english interfaces to other natural languages: an experiment with portuguese. In *Proceedings of the 22nd annual meeting on association for computational linguistics* (pp. 8–10). Morristown, NJ, USA: Association for Computational Linguistics.

Marques, L. (1996). *Edite – um sistema de acesso a base de dados em língua natural – análise morfológica, sintáctica e semântica*. Unpublished master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Mendes, A. C. (2007). *Clefomania, QA: primeiros passos*. Unpublished master's thesis, Instituto Superior Técnico, Universidade Técnica de Lisboa.

Minock, M. (2004). *Natural language access to relational databases through STEP.*

Sanamrad, M. A. (1992). IBM SAA LanguageAccess: A large-scale commercial product implemented in prolog. In *The first international conference on the practical applications of prolog.*

Scha, R. (1977, Fevereiro). Philips question answering system PHILIQA1. In *Sigart newsletter, nº61.* New York: ACM.

Systems, B., & Technologies. (1989). *BBN parlance interface software - system overview.*

Technology, B. I. (1991). *LOQUI: An open natural query system – general description.*

Templeton, M., & Burger, J. (1983). Problems in natural-language interface to DBMS with examples from EUFID. In *Proceedings of the first conference on applied natural language processing* (pp. 3–16). Morristown, NJ, USA: Association for Computational Linguistics.

Thompson, B. H., & Thompson, F. B. (1983). Introducing ASK, a simple knowledgeable system. In *Proceedings of the first conference on applied natural language processing* (pp. 17–24). Morristown, NJ, USA: Association for Computational Linguistics.

Waltz, D. (1978, Julho). An english language question answering system for a large relational database. In *Communications of the acm.*

Warren, D., & Pereira, F. (1982, Julho-Dezembro). An efficient easily adaptable system for interpreting natural language queries. In *Computational linguistics.*

W.A.Woods, R. K. e. B. W. (1972). The lunar sciences natural language information system: Final report. In (Vol. BBN Report 2378). Cambridge, Massachussets: Bolt Beranek and Newman Inc.

Weischedel, R. (2006). Natural-language understanding at BBN. *IEEE Ann. Hist. Comput.*, *28*(1), 46–55.

Weischedel, R. M. (1989). A hybrid approach to representation in the JANUS natural language processor. In *Meeting of the association for computational linguistics* (p. 193-202).

Whittemore, G., Ferrara, K., & Brunner, H. (1990). Empirical study of predictive powers of simple attachment schemes for post-modifier prepositional phrases. In *Proceedings of the 28th annual meeting on association for computational linguistics* (pp. 23–30). Morristown, NJ, USA: Association for Computational Linguistics.