

# Avaliação de sistemas interactivos de recuperação de informação em bases de texto jurídicas

Paulo Quaresma and Irene Rodrigues  
Departamento de Informática  
Universidade de Évora, Portugal  
pq|ipr@di.uevora.pt

June 14, 2004

## Abstract

Neste artigo é apresentada uma arquitectura multi-agente para sistemas interactivos de recuperação de informação em bases de texto jurídicas. Em concreto, é descrito o sistema de pesquisa para os pareceres da Procuradoria Geral da República Portuguesa, desenvolvido no âmbito de projectos de investigação (PGR e ABC).

A arquitectura possui dois níveis: o primeiro utiliza técnicas de processamento de Língua Natural para gerir o acesso às bases de texto; o segundo nível efectua a gestão das interacções entre os utilizadores e o primeiro nível.

É efectuada uma proposta de avaliação do desempenho de um sistema interactivo de recuperação de informação, em termos da sua precisão e abrangência. São, ainda, apresentados resultados obtidos pela aplicação da proposta de avaliação efectuada ao sistema descrito.

## 1 Introdução

Com o aumento da complexidade e da dimensão das bases de texto, jurídicas ou não, surge a necessidade de incorporar-lhes novas capacidades (veja-se uma análise de um exemplo na área jurídica em [YS99]):

- Os sistemas de recuperação de informação (RI) devem poder agir de uma forma racional, autónoma e cooperativa, nas suas interacções com os diversos utilizadores;
- Os sistemas de RI devem poder representar conhecimento e efectuar inferências sobre esse conhecimento, de modo a melhorar o seu desempenho.

Neste artigo, descrevemos uma arquitectura de um sistema interactivo de recuperação de informação para bases de texto jurídicas [QR03] e propomos uma metodologia para avaliação do seu desempenho. Esta metodologia foi aplicada e alguns resultados são apresentados.

A arquitectura proposta, recorre à integração de diversas técnicas para tentar satisfazer os requisitos descritos:

- Procedimentos de recuperação de informação – Um motor de pesquisa adaptado à Língua Portuguesa é utilizado para suportar as operações básicas de recuperação de informação;
- Processamento de Língua Natural – São utilizados dicionários e ontologias jurídicas para melhorar o desempenho do sistema;
- Técnicas de agrupamento (clustering) – Algoritmos de agrupamento são aplicados para organizar a informação recuperada pelo motor de pesquisa.

A arquitectura proposta tem dois níveis:

- O nível de recuperação de informação (RI). Este nível integra o motor de pesquisa em texto com algumas técnicas de processamento de Língua Natural;
- O nível de gestão de interacções. Este nível é o responsável pela gestão da interacção entre os utilizadores e o primeiro nível.

Os dois níveis da arquitectura são implementados através de agentes autónomos, que comunicam entre si através de acções. Uma descrição detalhada desta implementação está fora do âmbito deste artigo, que pretende focar-se na avaliação do desempenho do sistema proposto.

No processo de avaliação do desempenho, ir-se-á analisar o impacto da interactividade do sistema, em termos da precisão e abrangência das respostas obtidas.

O sistema proposto foi aplicado aos pareceres da Procuradoria Geral da República Portuguesa e uma avaliação preliminar foi efectuada.

Na próxima secção, será descrita a arquitectura proposta para sistemas interactivos de recuperação de informação. Na secção 3 será detalhado o primeiro nível da arquitectura e na secção 4 analisaremos o segundo nível. Na secção 5 a proposta de avaliação será descrita e alguns resultados obtidos serão apresentados. Finalmente, na secção 6 algumas conclusões serão referidas.

## 2 Arquitectura

O sistema interactivo de recuperação de informação é baseado em dois níveis:

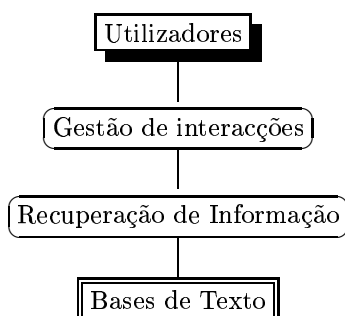


Figure 1: Arquitectura

### 2.1 Gestão de Interacções

O primeiro nível é o responsável pela gestão das interacções com os utilizadores. É constituído por dois sub-níveis compostos por agentes especializados em gerir a interacção e o modelo dos utilizadores.

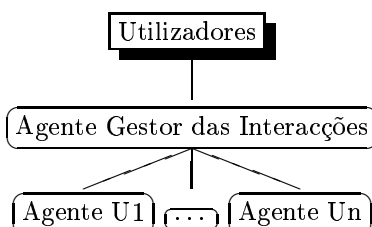


Figure 2: Sub-nível gestor das interacções

O agente gestor das interações recebe todas as acções executadas pelos utilizadores (frases, selecção de botões, etc.) e redirecciona-as para os agentes especializados em gerir o modelo de cada utilizador. De facto, cada utilizador possui um agente especializado em gerir o seu modelo, tentando inferir as suas intenções e ajudá-lo a satisfazer os seus objectivos. A figura 3 apresenta, para cada agente  $i$ , a sua ligação com os outros agentes.

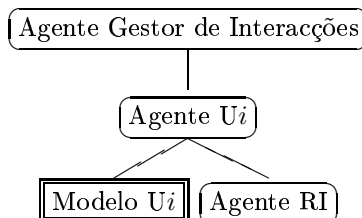


Figure 3: Sub-nível Agente Utilizador  $i$

O modelo de cada utilizador é baseado num sistema BDI (beliefs, desires, intentions) em que as suas crenças, objectivos e intenções são representadas através de factos num ambiente de programação em lógica (ver mais detalhes em [QR01]).

## 2.2 Recuperação de Informação

O segundo nível é o responsável pelo acesso às bases de texto (jurídicas) e por enviar as respostas aos agentes do primeiro nível. É composto pelos seguintes agentes:

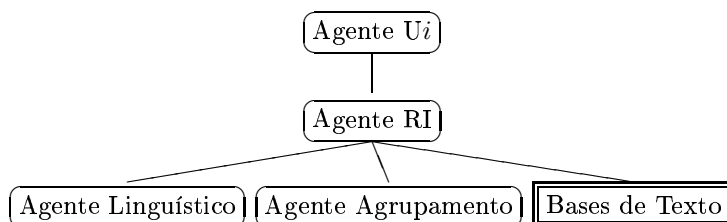


Figure 4: Recuperação de Informação

O agente RI recebe interrogações dos agentes específicos dos utilizadores e acede às bases de texto para lhes responder. De modo a melhorar o seu desempenho, pode recorrer a agentes linguísticos para obter formas canónicas de palavras, sinónimos e termos mais gerais e mais específicos. Finalmente, pode utilizar os serviços do agente agrupador (cluster) que agrupa conjuntos de documentos, de acordo com os seus temas. Na próxima secção será efectuada uma descrição mais detalhada deste processo.

## 3 O nível “Recuperação de Informação”

Como se pode observar na figura 4 o nível de recuperação de informação é constituído por três agentes – RI, linguístico e agrupamento – e pelas bases de texto.

### 3.1 Bases de texto

No âmbito deste trabalho, as bases de texto são compostas pelos pareceres da Procuradoria Geral da República Portuguesa. Esta base tem cerca de 7.000 documentos produzidos desde 1940 e cerca

de 10.000.000 de palavras. Os documentos possuem uma estrutura concreta e encontram-se num formato compatível com XML. Neste formato, cada secção do documento, está associada a uma etiqueta XML. Alguns exemplos são: Número, Título, Tópicos, Data, Autor, Conclusões, Texto Integral

Os documentos foram pré-processados, de modo a ser-lhes acrescentada informação linguística acerca da sua forma canónica.

A forma canónica de cada palavra é obtida através do acesso à base de dados lexical POLARIS<sup>1</sup>, que possui cerca de 900.000 palavras com informação acerca das suas formas canónicas e possíveis etiquetas morfo-sintácticas. Esta base de dados lexical foi construída no contexto de um projecto anterior no Centro de Inteligência Artificial da Universidade Nova de Lisboa.

### 3.2 Agente “Recuperação de Informação”

O agente recuperador de informação é o responsável por responder aos agentes específicos dos utilizadores acerca dos documentos que satisfazem as diversas interrogações.

Como motor de pesquisa em texto, foi utilizado uma aplicação já existente, SINO [GMK97], do AustLII Instituto (Australasian Legal Information Institute). O SINO utiliza ficheiros invertidos como estrutura de dados e suporta pesquisa com operadores *booleanos*, de *vizinhança*, *wildcards* e de *ordenação*.

O SINO foi alterado e adaptado à Língua Portuguesa, de modo a poder indexar os documentos utilizando a forma canónica das palavras (através de uma ligação à base de dados lexical POLARIS).

É, ainda, possível efectuar ligações entre o SINO e o agente linguístico, obtendo, para cada termo, sinónimos e termos mais genéricos e/ou mais específicos.

### 3.3 Agente Linguístico

O agente linguístico recebe pedidos do agente de recuperação de informação para obter sinónimos ou termos mais específicos e/ou termos mais gerais.

Esta informação pode ser obtida através de duas possíveis abordagens:

- Utilização de um “thesaurus” jurídico existente
- Extração automática de “thesaurus”

A primeira abordagem, utilização de um thesaurus jurídico, é baseada num thesaurus criado pelo Gabinete Jurídico da Procuradoria Geral da República Portuguesa. Este thesaurus é constituído por cerca de 6.000 conceitos, com as seguintes relações:

- equivalência  
ex: “lei” é equivalente a “norma”
- generalização  
ex: “primeiro-ministro” é generalizado por “ministro”
- especificidade  
ex: “acidente de trânsito” é mais específico do que “acidente”
- relacionamento  
ex: “fuga” está relacionado com “acidente de trânsito”

---

<sup>1</sup>Portuguese Lexicon Acquisition and Retrieval Interaction System

Na segunda abordagem, tenta-se extrair automaticamente relações entre os conceitos existentes nos documentos. Uma abordagem possível é o recurso a técnicas estatísticas descritas em [GGA<sup>+</sup>01]. A ideia base é medir semelhanças entre palavras que possuem funções idênticas nos documentos (mesmas etiquetas morfo-sintácticas, complementos, modificadores, ...). No entanto, é importante realçar que esta abordagem permite identificar relações entre conceitos mas tem dificuldades em identificar essas relações (distinguir entre sinonímia e antonímia, por exemplo).

### 3.4 Agente de agrupamento “clustering”

O agente de agrupamento (clustering) é o responsável por calcular conjuntos de documentos relacionados entre si através dos seus conceitos.

Agrupamento de documentos é um processo complexo [Sal89] que envolve a necessidade de representar documentos e de os agrupar, recorrendo a algum algoritmo. Existem muitas abordagens possíveis, desde técnicas de aprendizagem não supervisionada até à utilização de heurísticas que pretendem modelar determinadas regras gerais [CDRKT92, CKP93, HP96].

A implementação utilizada actualmente para o agrupamento de documentos é baseada num algoritmo que recebe como “input” um conjunto de documentos, representados pelas listas de tópicos que os caracterizam (esta lista foi efectuada manualmente por peritos jurídicos). O algoritmo tenta dividir o conjunto inicial em sub-conjuntos associados a tópicos distintos, de modo a que: A união dos sub-conjuntos é o conjunto inicial;

Este algoritmo é um algoritmo de pesquisa “best first” com uma função de avaliação admissível que limita a complexidade temporal e espacial.

## 4 O nível “Gestor de Interações”

Conforme referido na secção 2 o nível gestor de interações é baseado em dois tipos de agentes [QR01]:

- Um gestor de interações que recebe os pedidos dos utilizadores e que os redirecciona para os agentes específicos de cada utilizador;
- Agentes especializados que têm como responsabilidade gerir os modelos dos diversos utilizadores, as suas crenças e intenções, bem como as suas interações anteriores.

### 4.1 O agente gestor de interações

O agente gestor de interações recebe os pedidos dos utilizadores, analisa-os e redirecciona-os para os respectivos agentes especializados.

É de salientar que este agente necessita de manter um serviço tipo “yellow pages”, que lhe permita saber quais os agentes existentes no serviço, quais estão activos e quais é necessário activar.

### 4.2 Os agentes dos utilizadores

Os agentes dos utilizadores são os responsáveis por processar os seus pedidos e por os interpretar no contexto das interrogações anteriores.

Para tal, é necessário manter modelos dos utilizadores e efectuar inferências sobre as atitudes modeladas (crenças, intenções e objectivos). Este processo é efectuado recorrendo a um ambiente de programação em lógica e a uma arquitectura BDI (beliefs, desires, intentions) [QR01]

Após terem sido inferidas as “reais” intenções dos utilizadores, o agente comunica com o sistema de recuperação de informação, de modo a obter os documentos pretendidos.

Em conclusão, o agente deve:

- Possuir o modelo do utilizador. Este modelo é basicamente um programa em lógica dinâmica que permite representar regras de comportamento, racionalidade e eventos.

- Interpretar as diversas acções dos utilizadores, tendo em conta o contexto da interacção e o modelo do agente.
- Comunicar com o agente de recuperação de informação.
- Responder às acções do utilizador.

#### 4.2.1 Comportamento

De modo a serem cooperativos os agentes necessitam de inferir as atitudes dos utilizadores (crenças, intenções e objectivos). Esta tarefa, tal como foi referido anteriormente, é baseada em regras de programação em lógica dinâmica.

O “estado mental” dos agentes pode ser decomposto em diversos módulos (ver [QL95] para uma descrição detalhada destes módulos):

- Descrição dos efeitos e das pré-condições de cada acção, em termos de crenças e intenções;
- Definição de regras de comportamento que representam o modo como as atitudes se relacionam entre si e como são transferidas entre os agentes (cooperatividade).

Após cada evento, o modelo do agente é actualizado com a descrição desse evento e um processo de inferência é iniciado. Esse processo permite obter as atitudes do utilizador e planear as próximas acções.

#### 4.2.2 Planos e acções

As acções dos agentes dos utilizadores são o resultado de um processo de inferência abductiva sobre as atitudes desses utilizadores. Este processo tenta identificar as acções que permitam satisfazer os objectivos de cada utilizador.

Por exemplo, se  $I$  for o conjunto de intenções de um agente num dado instante de tempo, o processo de planeamento abductivo tenta identificar o conjunto  $A$  de acções que tenha como consequência satisfazer as intenções identificadas. Por exemplo, se um utilizador tem como intenção ser informado sobre documentos com determinadas características, então uma acção possível de ser abduzida e executada é redireccionar esse pedido ao agente de recuperação de informação.

## 5 Avaliação do sistema de gestão das interacções

O sistema a avaliar interage com o utilizador fornecendo dois tipos de sugestões:

**Tipo 1 — Sugestões similares** Lista de termos equivalentes ou conceitos mais abrangentes (termos relacionados) que os termos da pergunta do utilizador.

Exemplo - na pergunta: *costas*:

O sistema fornece seguinte lista de termos:

- Coluna
- Corpo humano
- Marítimo
- Lesão
- ...

Estes termos são mais gerais que os usados na pergunta e permitem ao utilizador aumentar o número de textos seleccionados.

**Tipo 2 — Sugestões para refinamento** Lista de termos para refinar a pergunta. Este termos correspondem a etiquetas de subgrupos dos documentos seleccionados.

Exemplo - na pergunta *costas*

- Ambiente
- Incapacidade geral de ganho
- Tráfico
- Risco Agravado
- ...

Os termos nesta lista não são necessariamente equivalentes ou semelhantes aos termos usados na pergunta. Esta lista de termos tem termos que classificam alguns subconjuntos do conjunto de documentos seleccionados.

Para avaliar a qualidade das sugestões do nosso sistema vamos considerar dois parâmetros: a *abrangência* (recall) e a *precisão* (precision). Para um conjunto documentos  $t_1$ : a *abrangência*,  $recall_{t_1}$  é a percentagem de documentos relevantes no conjunto  $t_1$  em relação ao total de documentos relevantes,  $t_0$ ,  $recall_{t_1} = \frac{\#(t_1 \cap t_0)}{\#t_0}$ ; a *precisão*,  $precision_{t_1}$ , é a percentagem de documentos relevantes em  $t_1$ ,  $precision_{t_1} = \frac{\#(t_1 \cap t_0)}{\#t_1}$ .

O nosso sistema com os dois tipos de sugestões descritos acima, procura auxiliar o utilizador a:

- Aumentar a *abrangência*. Com as sugestões do tipo 1 o sistema procura sugerir ao utilizador termos para construir a sua pergunta (acrescentados à sua pergunta inicial) com o objectivo de aumentar a abrangência, seleccionar mais textos relevantes.
- Aumentar a *precisão*. Com as sugestões do tipo 2 o sistema procura sugerir ao utilizador termos para refinar a sua pergunta com o objectivo de aumentar a precisão. O resultado na nova pergunta construída com as sugestões dadas deverá seleccionar menos textos irrelevantes.

Para calcular a *abrangência* e a *precisão* do conjunto de documentos seleccionados por uma pergunta,  $t_1$ , é preciso determinar o conjunto de documentos relevantes para a pesquisa,  $t_0$ .

Nos testes para avaliação do desempenho do sistema marcamos um conjunto de documentos como relevantes para a pesquisa, e construímos a pergunta inicial  $p_1$  de forma a obtermos para o conjunto de documentos,  $t_1$ , seleccionado por  $p_1$ :  $0\% < recall_{t_1} < 100\%$  e  $0\% < precision_{t_1} < 100\%$ .

Suponhamos que numa pesquisa onde se pretende encontrar documentos sobre “acidentes em saltos de pára-quedas efectuados por militares que tenham causado lesões nas costas”, marcamos os documentos que consideramos relevantes (15 documentos na base de pareceres da PGR), e construímos a primeira pergunta,  $p_1$  com o termo “costas”. O conjunto de documentos seleccionados por  $p_1$ ,  $t_1$  tem 320 documentos dos quais 7 são relevantes (pertencem a  $t_0$ ). A abrangência de  $t_1$  é 46% e a precisão é 2.1%.

As sugestões do sistema para este caso estão listadas no início da secção e a avaliação das sugestões é feita:

- Calculando os valores da abrangência para a pergunta que se pode construir com as sugestões do tipo 1 (sugestões similares).

Para este exemplo obtêm-se:

- Coluna - abrangência de 75%, taxa de melhoramento potencial é de  $63\% = \frac{recall_{p_{coluna}} - 1}{recall_{p_1}}$
- Corpo humano - abrangência de 46%, taxa de melhoramento potencial é de 0%
- Marítimo - abrangência de 47%, taxa de melhoramento potencial é de 2%
- Lesão - abrangência de 77%, taxa de melhoramento potencial é de 67%

A taxa de melhoramento da lista de sugestões similares é a melhor taxa de melhoramento dos termos, que neste exemplo é de 67%.

Desta forma podemos medir a qualidade das sugestões de forma quantitativa.

- Calculando os valores da precisão para a pergunta que se pode construir com as sugestões do tipo 2 (sugestões de refinamento)

Para as sugestões de refinamento do exemplo obtêm-se:

- Ambiente - precisão de 0%, taxa de melhoramento potencial -1%.
- Incapacidade geral de ganho - 12%, taxa de melhoramento potencial 471%
- Tráfico - precisão de 0%, taxa de melhoramento potencial -1%.
- Risco Agravado - 48%, taxa de melhoramento potencial 2185%.

Vamos escolher como a taxa de melhoramento das sugestões de refinamento, a melhor taxa da lista de termos, neste caso é de 2185%.

A forma como fazemos a avaliação das sugestões do sistema não tem em conta as escolhas do utilizador, no entanto permite avaliar a qualidade das sugestões feitas ao utilizador. Este tipo de avaliação permite comparar o desempenho de diferentes sistemas de pesquisa de informação interactivos ao mesmo tempo que é importante no desenvolvimento de um sistema em particular na afinação dos métodos usados para calcular as sugestões ao utilizador.

## 5.1 Algoritmo de avaliação de listas de sugestões ao utilizador

O algoritmo para a quantificação das listas de sugestões do sistema pode ser resumido nos seguintes passos:

1. Construir  $t_0$ , conjunto de documentos da base de documentos que constituem os documentos relevantes para a avaliação de uma pesquisa.
2. Construir  $p_1$ , uma pergunta para o sistema de pesquisa que deve seleccionar um conjunto de documentos  $t_1$  que deve ser diferente de  $t_0$ .
3. Calcular a taxa de melhoramento potencial (abrangência) da lista de sugestões do tipo 1.  
Para cada termo na lista de sugestões fazer:
  - (a)  $p_2 = p_1 OR$  termo
  - (b)  $t_2$  é o conjunto de textos seleccionado por  $p_2$
  - (c) Calcular a abrangência e a taxa de melhoramento potencial de  $t_2$
  - (d) Guardar o termo e a taxa de melhoramento se a taxa for superior à dos termos já calculados.
4. Calcular a taxa de melhoramento potencial (precisão) da lista de sugestões do tipo 2, sugestões de refinamento.
  - (a)  $p_2 = p_1 AND$  termo
  - (b)  $t_2$  é o conjunto de textos seleccionado por  $p_2$
  - (c) Calcular a precisão e a taxa de melhoramento potencial de  $t_2$
  - (d) Guardar o termo e a taxa de melhoramento se a taxa for superior à dos termos já calculados.
5. Retornar um tuplo com as taxas calculadas em 3 e 4 e os termos (sugestões) com esses valores. ( Para o exemplo acima este procedimento deve retornar o tuplo ('risco agravado' - 2185, 'lesão' - 67). )



**Convergência** Outro aspecto importante num sistema interactivo com sugestões ao utilizador é avaliar o potencial de convergência. O potencial de convergência mede a capacidade de o sistema estabilizar o conjunto de textos seleccionados ao fim de algumas iterações escolhendo as “melhores” sugestões do sistema de pesquisa.

O facto de um sistema convergir de acordo com a esta definição não garante que um utilizador não se perca numa pesquisa, pois o utilizador nem sempre seguirá as melhores sugestões.

No entanto, esta medida pode ser utilizada para comparar sistemas diferentes e para comparar os algoritmos usados nos métodos de cálculo das sugestões para o utilizador.

A convergência pode ser avaliada de 3 formas diferentes:

- **Convergência da abrangência.** Calcula o número de passos até que a abrangência do conjunto de documentos que vai sendo seleccionado não varia.

O resultado é o número de passos até convergir e o valor da abrangência. Na comparação dos resultados dos algoritmos devem-se preferir os algoritmos que têm menos passos.

- **Convergência da precisão.** Calcula o número de passos até que a precisão do conjunto de documentos que vai sendo seleccionado não varia.

Tal como acima o resultado é número de passos até convergir e o valor final da precisão

- **Convergência intercalada.** Em 10 passos intercala a aceitação de sugestões de termos similares (que devem aumentar a abrangência) com termos de refinamento (que devem aumentar a precisão).

O resultado deste calculo deve ser um par com o valor da precisão e da abrangência ao fim dos 10 passos.

**Testes realizados** Atendendo à dimensão da base de documentos (7000 documentos) e à sua especificidade optou-se por fazer um conjunto de testes fixando o conjunto de documentos relevantes com duas cardinalidades, 1 e 10. Para cada cardinalidade fizeram-se 20 experiências independentes. Na tabela em abaixo apresenta-se para cada cardinalidade do conjunto de documentos relevantes: o valor médio (sobre 20 testes) para as medidas de convergência, o melhor valor das medidas nos 20 testes e o pior valor das medidas nos 20 testes.

	nº textos relevantes na base de docs	Convergência da precisão	Convergência da abrangência	Convergência intercalada
valor médio	1	3.2	1.8	(67.8%,59.3%)
melhor resultado	1	1	0	(100%,100%)
pior resultado	1	7	3	(0%,0%)
valor médio	10	4.8	3.5	(62.2%,47.6%)
melhor resultado	10	3	2	(100%,80%)
pior resultado	10	8	5	(0%,0%)

Os 40 exemplos estudados permitem testar o método de avaliação mas para retirar conclusões sobre o sistema é necessário realizar testes sobre um conjunto de exemplos mais amplo.

## 6 Conclusões

Um sistema interactivo para recuperação em bases de texto jurídicas foi descrito neste artigo. O sistema é baseado numa arquitectura em dois níveis: gestão das interacções e recuperação de informação.

Sobre este sistema foi proposta uma metodologia de avaliação da melhoria do seu desempenho em termos do potencial de melhoramento da precisão e abrangência dos resultados das sugestões dos sistema ao utilizador. Alguns resultados foram apresentados, mas uma avaliação sistemática necessita de ser efectuada.

O método de avaliação proposto é importante para a avaliação de sistemas de pesquisa interactivos quando os métodos usados para calcular as listas de sugestões não são facilmente modelados e permite comparar o desempenho deste sistema com o de outros sistemas interactivos de recuperação de informação.

O método proposto é independente das opções e apreciações do utilizador, esta é uma vantagem na medida em que pode ser facilmente aplicado no desenvolvimento de sistemas semelhantes. No entanto este tipo de avaliação não dispensa uma avaliação mais complexa e dispendiosa por parte de um conjunto de utilizadores que possa reflectir as impressões do utilizador.

## References

- [CDRKT92] D. R. Cutting, J. O. Pedersen D. R. Karger, and J. W. Tukey. Scatter/gather: A cluster-based approach to browsing large document collections. In *Proc. 15th Annual Int'l ACM SIGIR Conf. on R&D in IR*, June 1992.
- [CKP93] D. R. Cutting, D. Karger, and J. Pedersen. Constant interaction-time scatter/gather browsing of very large document collections. In *Proc. of the 16th Annual Int. ACM/SIGIR Conf.*, Pittsburgh, PA, 1993.
- [GGA<sup>+</sup>01] P. Gamallo, C. Gasperin, A. Agustini, G. Lopes, and V. Lima. The use of syntactic context for measuring word similarity. In *ESSLI – Proceedings of the Workshop on Semantic Knowledge Acquisition and categorization*", 2001.
- [GMK97] G. Greenleaf, A. Mowbray, and G. King. Law on the net via austlii - 14 m hypertext links can't be right? In *In Information Online and On Disk'97 Conference, Sydney*, 1997.
- [HP96] M. A. Hearst and J. O. Pedersen. Reexamining the cluster hypothesis:scatter/gather on retrieval results. In *Proceedings of the Nineteenth Annual International ACM SIGIR Conference*, Zurich, June 1996.
- [Lew91] D. D. Lewis. Evaluating Text Categorization. In *Proceedings of Speech and Natural Language Workshop*, pages 312–318. Morgan Kaufmann, 1991.
- [QL95] P. Quaresma and J. G. Lopes. Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. *Journal of Logic Programming*, 54, 1995.
- [QR01] Paulo Quaresma and Irene Rodrigues. Using logic programming to model multi-agent web legal systems – an application report. In *Proceedings of the ICAIL'01 - International Conference on Artificial Intelligence and Law*, St. Louis, USA, May 2001. ACM.
- [QR03] Paulo Quaresma and Irene Pimenta Rodrigues. PGR: Portuguese attorney general's office decisions on the web. In Bartenstein, Geske, Hannebauer, and Yoshie, editors, *Web-Knowledge Management and Decision Support*, Lecture Notes in Artificial Intelligence LNCS/LNAI 2543, pages 51–61. Springer-Verlag, 2003.
- [Sal89] Gerard Salton. *Automatic text processing: the transformation, analysis, and retrieval of information by computer*. Addison-Wesley, 1989. Reading, MA.
- [VR79] C. J. Van Rijsbergen. *Information Retrieval, 2nd edition*. Dept. of Computer Science, University of Glasgow, 1979.
- [YS99] John Yearwood and Andrew Stranieri. The integration of retrieval, reasoning and drafting for refugee law: a third generation legal knowledge based system. In *Proceedings of the ICAIL'99 – 7th International Conference on Artificial Intelligence and Law*, pages 117–125. ACM, June 1999.