

Construire le contexte d'une interface Web pour une base de données juridique

Paulo Quaresma and Irene Pimenta Rodrigues
Departamento de Matemática
Universidade de Évora
Largo dos Colegiais
7000 Évora
Portugal
{pq|ipr}@dmat.uevora.pt

Résumé

Dans ce travail nous proposons un système capable de construire et maintenir le contexte des interactions dans un interface web pour des bases de données avec des textes juridiques.

Nous utilisons quelques resultats du project *AustLII* (Australasian Legal Information Institute), un système d'inférence via le World Wide Web pour des bases de données avec des textes juridiques. En particulier, on se sert de SINO, un moteur de recherche pour bases de données avec des textes juridiques (Greenleaf, Mowbray and King 1997). Actuellement, nous utilisons SINO pour processor les textes de notre partenaire de project (PGR, Procuradoria Geral da República). La base de données qui est le resultat de ce processement est utilisée pour tester la capacité de notre système de générer des interrogations pour SINO à partir de questions en langue naturelle posées par les utilisateurs. Ces questions sont générées en tenant compte du contexte du dialogue qui est construit par les questions antérieures de l'utilisateur ainsi que les réponses données par le système.

Nous employons aussi YSH, un moteur d'inférence (Greenleaf, Mowbray, et Van Dijk 1995) pour établir des bases de règles modelant les connaissances légales. Ceci nous permet de tester encore plus de dispositifs avancés, à savoir les capacités de traduire les phrases en langage quasi naturel vers notre langage de représentation (un langage étendu de représentation en logique).

1. Introduction

Dans cet article nous présentons un système avec une interface intelligente Web pour une base de données avec les textes légaux. Le but principal de l'interface est de permettre la recherche et la visualisation des documents de la base de données. L'utilisateur pose une requête et il devrait obtenir comme réponse l'ensemble de documents qui satisfont la requête. Notre système est hybride dans le sens qu'il emploie différentes sources de connaissance afin de réaliser ses buts.

Nous utilisons certains des résultats du projet de l'AustLII (institut juridique de l'information d'Australie) avec son système légal d'inférence par l'intermédiaire de World Wide Web pour les bases de données légales des textes. À savoir, nous utilisons:

- SINO, un moteur de recherche pour bases de textes légaux (Greenleaf, Mowbray and King 1997). Actuellement nous utilisons SINO pour processor les textes de notre partenaire de project (PGR, Procuradoria Geral da República – Portuguese Attorney General). La base de textes qui resulte de ce processement est utilisée pour tester la capacité de notre system à générer des interrogations pour SINO, à partir de questions an langage naturel posées par les utilisateurs. Ces questions sont générées en tenant compte du contexte du dialogue qui est construit avec les questions antérieures de l'utilisateur ainsi que les reponses données par le système.
- YSH, un moteur d'inférence (Greenleaf, Mowbray, et Van Dijk 1995). Nous avons construit une base de règles modelant une certaine connaissance légale, définissant les conditions dans lesquelles "une personne a légalement le droit d'obtenir une pension pour prestation de services exceptionnels". Ceci nous permet de tester quelques autres dispositifs de notre système, à savoir les capacités de traduire les phrases de langage quasi naturel dans notre langage de représentation en logique (une DTR étendue). Notre système peut interpréter des phrases de langage naturel et les transformer en une forme de logique; nous avons prétraité certains des documents et avons augmenté leur texte avec la représentation logique de quelques parties. De cette façon nous pouvons tester notre base de règles avec les documents dans la base de données (voir la section 6).

L'interface intelligente de Web a plusieurs modules comme:

- Le moteur de recherche de SINO qui classe tous les mots dans les documents, y compris le texte et toutes les zones telles que la conclusion du document, les descripteurs et les données administratives. Les documents sont prétraités afin d'incorporer de l'information linguistique et sémantique laquelle peut être utilisée dans le procédé de recherche: mots et expressions étiquetées, et dans certains cas une formule de logique représentant la signification de parties du texte.
- Un module pour exécuter et représenter l'inférence des attitudes d'utilisateur, y compris un planificateur.
- Une taxonomie juridique de limites, un thesaurus, et un module qui nous permettent de rapporter des expressions de langage naturel avec des limites en taxonomie.
- Un dictionnaire comprenant des synonymes de langage.
- Un module pour l'analyse linguistique comprenant une analyse partielle, un module pour établir une représentation sémantique des expressions de langage naturel (le jeu rouleur-tambour aiment), et un module pour la traduction pragmatique (résolvant l'anaphore et établissant une structure de discours (dialogue)).

L'interface Web intelligente a trois étapes:

1. Elle essaye de déterminer quelles sont les intentions de l'utilisateur avec sa requête. Dans cette étape nous établissons un modèle de l'utilisateur en essayant de déduire ses attitudes (croyance et intentions). Chaque fois que l'ensemble des intentions de l'utilisateur ne peut pas être accompli, le système va à la deuxième étape.
2. Elle essaye de clarifier les intentions de l'utilisateur afin d'atteindre un positionnement qui puisse être accompli.
3. Elle essaye d'accomplir l'intention de l'utilisateur.

Notre système nous permet également de tester les règles-bases qui étaient construites en utilisant les données des textes dans le système de SINO. Certains de nos textes légaux ont une section avec une description du cas et une section avec les conclusions. Étant donné une base de règles qui modèle une connaissance légale, nous pouvons la tester suivant l'approche suivante:

1. Nous traitons la base de règles pour extraire les mots appropriés à partir des règles de langage.
2. Nous augmentons l'ensemble de mots appropriés en utilisant le thesaurus des termes juridiques établi par nos associés (PGR).
3. Nous produisons des requêtes de SINO pour choisir un ensemble de textes appropriés à partir de notre base de données des textes.

Pour chaque texte choisi nous activons le système de base de règles répondant aux questions de YSH en transformant la question de langage quasi-naturel DRT étendue, et puis nous voyons si le DRT qui représente le texte le contient.

Ainsi, un utilisateur peut pouvoir voir si une description de cas satisfait la base de règles. En comparant ces résultats avec la conclusion des textes l'utilisateur peut conclure sur l'exactitude de sa règle-base.

1. Le system de interface à WEB

Notre interface est multimodale; elle permet à l'utilisateur de poser des questions avec des expressions booléennes du type de SQL (à l'aide de menus et de boutons) ou avec des expressions en langage naturel. De toute façon le système essaye de garder le contexte d'interrogation, c.-à-d. afin d'établir une réponse qui utilise toutes les questions d'utilisateur

et réponses précédentes de système.

La première question de l'utilisateur

Une question placée par un utilisateur peut être une expression en langage naturel, comme:

Q: En quels documents est-ce qu'une personne obtient une pension pour des services exceptionnels?

Cette question, après sa traduction en DRT donne:

$\$x, p, t, e$: $person(x), pension_exceptional_services(p), evt(e, has(x,p)), occurs(e,t)$.

Pour répondre à la question de l'utilisateur il faut chercher des documents qui satisfassent cette expression logique.

- Mais il y a beaucoup de documents qui ne parlent pas de l'attribution de pensions, on perdra beaucoup de temps si on cherche tous les documents dans la base de textes.
- Pour le moment on n'a pas la représentation logique pour tous les documents.

Alors, ce qu'on fait c'est:

- Mètre en rapport l'expression logique de la question avec une question dans le langage SQL (la requête SINO) qui probablement apparaît dans un document qui parle de l'attribution de pensions, on sélectionne tous les documents qui ont cette expression. (Dans l'exemple, l'expression peut être: favorable et descripteur = pension pour des services exceptionnels)
- Si un document a une expression logique associée, on vérifie s'il satisfait l'expression logique.

User questions may also have different formats:

- pension and exceptional (meaning that user want documents where the word "pension" and the word "exceptional" are present).
- descriptor = pension for exceptional services (meaning that the user wants the documents where the field descriptor has the value "pension for exceptional services").

These are already SINO queries, and what the system does is to transform them using our juridical thesaurus (see section 5).

Amélioration de la Requête

Après obtention de la requête de l'utilisateur finale, le système a l'ensemble de documents choisis par la requête et il peut accomplir l'intention de l'utilisateur, à moins qu'il y ait des intentions contradictoires comme:

- 100 est le nombre de documents choisis
- l'utilisateur ne veut pas en voir plus de 20

L'identification des intentions d'utilisateur est faite par les analyses des actes d'utilisateur. En fait chaque acte d'utilisateur peut être traduit en formule logique qui est employée pour mettre à jour un programme de logique décrivant le modèle d'utilisateur. Dans ce modèle d'utilisateur nous avons des règles associer des actes et des intentions et des buts, nous permettant d'impliquer des intentions et des plans d'utilisateur. Cependant, cette inférence peut créer un état contradictoire entre les intentions réelles et précédentes. Dans cette situation, un procédé de déplacement de contradiction est doit être commencé.

En fait, quand le système atteint des intentions contradictoires il entre dans un processus d'amélioration de documents. L'amélioration est faite en groupant des ensembles de textes en utilisant le thesaurus juridique et les valeurs du descripteur de zone. Nous établissons l'ensemble de toutes les valeurs de descripteur de documents et employons le thesaurus afin de nous effondrer l'ensemble de valeurs plus petit en jeu (voir la section 5).

Comme exemple, supposez que l'ensemble de documents choisis avec la requête "accident" après l'exécution d'effondrement provoque les valeurs suivantes de descripteur:

- accident de travail
- accident de trafic
- accident de maison
- autre accident

Comme une réponse à la requête notre système fournit l'utilisateur:

- Le nombre de documents a choisi
- La possibilité d'amélioration de requête qui inclut l'ensemble ci-dessus de descripteur évalué
- La possibilité de relâcher l'intention de ne pas voir beaucoup de documents (par exemple puis 20)
- La possibilité de présenter une nouvelle contrainte, en langage naturel ou comme expression booléenne qui sera interprétée dans le contexte du premier.

Contexte d'interprétation

Notre système tient compte du contexte de traduction de la requête d'utilisateur dans les deux genres de requêtes, de phrases de langage naturel et d'expressions booléennes.

Quand l'utilisateur agit l'un sur l'autre en utilisant des phrases de langage naturel il est nécessaire de garder le contexte d'interrogation afin de nous permettre de résoudre des phénomènes de discours tels que temporel pronominal et nominal et l'anaphore; et capturer l'utilisateur a destiné la signification de leurs phrases.

Par exemple si la première question d'utilisateur est:

Q1: En quels documents est-ce qu'une personne obtient une pension pour des services exceptionnels?

Après que la réponse du système que l'utilisateur peut demander:

Q2: Dans lesquels de ces documents la personne est-elle un pompier?

Afin d'obtenir la signification de la deuxième phrase le système doit résoudre l'anaphore: " ces documents " et " la personne ".

Si, au lieu de Q2, l'utilisateur utilisait Q2 ':

Q2 ': Dans quels documents la personne est-elle un pompier?

Le système impliquera que l'intention d'utilisateur avec cette question est de demander:

Q: En quels documents est-ce qu'un pompier obtient une pension pour des services exceptionnels?

Mais si la deuxième question était Q2'':

Q2'': Dans quels documents est-ce qu'une personne a sa demande d'une pension pour des services exceptionnels rejetés?

Le système peut détecter que l'utilisateur commence un nouveau contexte, c.-à-d. cette phrase ne a pas besoin de l'information de Q2 pour être interprétée.

Quand l'utilisateur questionnent d'abord est une expression booléenne telle que Q1:

Q1: pension pour des services exceptionnels

Cela est suivi de Q2:

Q2: favorable

Le système de dialogue peut impliquer que les intentions possibles d'utilisateur sont:

1. Documents avec la séquence de mots "pension pour services exceptionnels" et le mot "favorable"
2. Documents avec le mot "favorable"

Puisque le système implique ces deux intentions possibles il donnera deux réponses, une pour chaque intention et lui présente l'intention avant la réponse.

Si l'utilisateur pose une nouvelle interrogation comme Q3:

Q3: rejeté

Le système de dialogue peut déduire que maintenant les intentions possibles d'utilisateur sont:

1. Documents avec la séquence de mots "pension pour services exceptionnels" et le mot "rejeté".
2. Documents avec la séquence de mots "pension pour des services exceptionnels" et le mot "favorable" et le mot "rejeté"
3. Documents avec le mot "rejeté"

Si le système découvre que pour la deuxième intention il y a les 0 documents, il supposera que ce n'est pas une intention valide et il présentera juste les réponses pour les deux autres intentions.

La forme comme le contexte est pris en considération par notre système donne à l'utilisateur l'idée qu'il interagit avec un agent intelligent.

1. Utilisation du thesaurus de termes juridique

Le thesaurus de termes juridiques est une taxonomie avec les valeurs de descripteur qui a les relations:

- est équivalent à
ex: la loi est équivalente à la norme
- est généralisé par
ex: le premier ministre est généralisé par le ministre
- est spécifié par
ex: l'accident est spécifié par accident de trafic
- est associé à
ex: l'abandon est associé à l'accident de trafic

Cette connaissance est utilisée pour:

- Augmenter les requêtes.

Chaque fois qu'une requête qui indique la valeur d'un descripteur est faite, nous l'augmentons avec toutes les valeurs qui sont: équivalent ou plus spécifique ou connexe, avec la valeur initiale de descripteur.

- Réduire les ensembles de valeurs de descripteurs

Chaque fois que nous avons un grand ensemble de documents nous rassemblons toutes les paires possibles de: valeur de descripteur / numéro de document. Il est possible de les joindre et réduire en utilisant la relation de généralisation afin d'obtenir un plus petit ensemble de paires avec la forme: valeur de descripteur / ensemble de numéros de document.

Exemple

Supposons qu'un utilisateur veut être informé au sujet des textes légaux au sujet des "accidents":

- Documents au sujet des accidents?

Le système augmente la requête en utilisant le thesaurus et il recherche toutes les valeurs connexes et plus spécifiques. Par exemple, il recherchera "accident ET accident de trafic ET abandon ET".

Puis, la réponse est employé pour s'effondrer l'ensemble de textes dans des classes des réponses groupées par les limites de thesaurus:

- X documents au sujet d'accidents;
- Y documents au sujet d'accidents de trafic;
- Z documents au sujet d'abandon...

Comme décrit dans la section 2, il est possible que l'utilisateur raffine cette question en utilisant le contexte de l'interaction.

1. Le moteur d'Inférence

Un autre but important de notre projet est de pouvoir modeler la connaissance légale. Afin de traiter ce problème que nous avons besoin de:

1. Un moteur d'inférence légal;
2. Un formalisme de représentation de bases de connaissance;
3. Une interface utilisateur;
4. Une base de règles décrivant la connaissance légale modelée.

Comme un moteur d'inférence légal nous utilisons YSH, un moteur d'inférence du projet d'AustLLI (Greenleaf, Mowbray, et Van Dijk, 1995). YSH met en application inferencing basé sur les règles, avec la capacité pour modeler l'enchaînement vers l'avant et en arrière. D'ailleurs il utilise une représentation ' de langage naturel quasi ' avec des entités avouées pour être une PERSONNE, une CHOSE, ou des PERSONTHING. Les dialogues inferencing sont produits dynamiquement.

Afin de représenter les textes légaux nous employons la DRS étendue – structures de représentation de discours (Rodrigues, 1995; Kamp 1990). Pour le moment nous ne représentons pas les textes complets mais seulement les sections où les cas sont décrits. Un autre problème relatif est la construction de chaque DRS: car nous n'avons pas un programme d'analyse syntaxique complet pour la langue portugaise, la construction de DRS est faite semi-automatiquement. C'est un problème qui limite la généralisation de notre système et elle devrait être manipulée en tant que travaux futurs. Les DRS sont représentés en utilisant un environnement de programmation en logique (Pereira et autres 1996) ce raisonnement non-monotonique de supports, à savoir, le raisonnement par défaut, ce qui permet obtenir le modèle bien fondé du programme en logique.

Comme l'interface utilisateur nous utilisons un environnement de WWW avec WYSH – l'interface utilisateur de Web-ysh d'AustLLI (Greenleaf, Mowbray, roi, pente, et Chung, 1997) – et une interface avec XSB-Prolog – le cadre de programmation de logique nécessaire pour des inférences au-dessus du jeu rouleau-tambour. En conclusion, nous devons décrire la connaissance légale en utilisant les règles de YSH. En raison de la complexité de cette tâche, parce que du moment nous avons choisi seulement quelques domaines spécifiques, à savoir, la législation qui définit quand une personne a un droit pour une pension pour des services exceptionnels.

Comme décrit dans la section 1, donnée une règle-base qui modèle de la connaissance légale nous peut la tester l'utilisation de l'approche suivante:

1. Extraire les mots appropriés à partir des règles.
2. Utiliser le thesaurus des termes juridiques.
3. Produisez des requêtes de SINO pour choisir un ensemble de textes appropriés.
4. Pour chaque texte choisi, exploiter le système de règle-base répondant aux questions de YSH en transformant la question en DRT étendu et puis en contrôlant si le DRT des textes nécessite le DRT de la question (utilisant l'environnement de programmation de logique).

Cette approche nous permet de voir si la description de cas satisfait la règle-base. En comparant les résultats à la section de conclusion des textes nous concluons au sujet de l'exactitude des règle-bases.

Exemple

Dans cette section nous montrerons un exemple sur la législation qui définit quand une personne a le droit à une pension pour des services exceptionnels. Cependant, en raison de son extension et complexité, nous ferons quelques simplifications sur la législation:

GOAL RULE Pension for exceptional service art3 PROVIDES
A person has a legally right under art3 to obtain a pension for exceptional services ONLY IF
art3(1) applies AND art3(2) applies.

RULE Art3(1) applies ONLY IF
A person has made an action that is considered exceptional and relevant to Portugal
(exceptional_action) AND (art3(1a) applies OR art3(1b) applies)

RULE Art3(1a) applies ONLY IF
A person has made an exceptional action in a war place (war_place) OR
A person has made an abnegated and courageous action OR
A person has made a high service to his country or to the Humanity

RULE Art3(1b) applies ONLY IF
(A person has made an humanity act OR
A person has made an act of dedication to the public cause)
AND a person has been injured or has deceased.

RULE Art3(2) applies ONLY IF
A person has constantly showed respect for the individual and collectives rights and freedom AND
A person has constantly showed respect for the prestige and dignity of the country.

RULE exceptional_action applies ONLY IF
A person has made an action which benefits the country AND
A person has made an action with a correct typology (typology) AND
A person has made an action without any remuneration AND
A person has made an action that goes beyond the duty of his functions.

RULE typology applies ONLY IF
A person has made an action that serves the national interests AND
A person has made an action that presupposes a capacity of high-availability.

RULE war_place applies ONLY IF
A person has made an action in a war place AND
A person has made an action that goes beyond the standard military actions
(beyond_standard_military_actions).

RULE beyond_standard_military_actions applies ONLY IF
A person has made an action which is considered beyond the standard military actions by the
Military Administration OR
A person has made an action which places the defense of others lives above the defense of his own
live OR
A person has made an action which places the defense of others lives above his own injures.

Supposons que nous voulons tester ces règles d'inférence sur des textes légaux:

1. Nous traitons les règles pour extraire les mots appropriés à partir des règles de langage quasi-naturel. Par exemple, un mot est considéré approprié si c'est un nom ou une expression de nom dans la règle de but. Dans cet exemple nous obtiendrions la pension, services exceptionnels, personne, correctement.
2. Nous augmentons l'ensemble de mots appropriés en utilisant notre thesaurus des termes juridiques. Dans cet exemple nous avons: personne -> citoyen
3. Nous produisons des requêtes de SINO pour choisir un ensemble de textes appropriés à partir de notre base de connaissance des textes: 189 dans notre exemple.
4. Pour chaque texte choisi nous exploitons le système de règle-base répondant aux questions de YSH en transformant la question en DRT étendu qui est testé pour voir s' il est nécessité par le DRT qui représente chaque texte. Ce processus est fait par l'utilisation d'un cadre de programmation en logique mis en application avec XSB-Prolog. Par exemple, la question "a la personne fait une action dans un endroit de guerre?" peut être représenté par le DRS suivant:

$\$x, e1, e2, p, t1, t2$: $person(x), occurs(e1, t1), holds(e2, t2), t1 \dot{\bar{I}} t2, agent(e1, x), war_place(p), evt(e2, in(x, p))$.

1. En utilisant le processus d'inférence, nous pouvons avoir un des trois résultats suivants: la personne a droit à la pension, la personne n'a pas droit à la pension, ou le résultat est inconnu.
2. Comme point final, les résultats impliqués sont comparés contre les textes légaux en utilisant la section de conclusion. La requête suivante de SINO peut être utilisée:

search pension for exceptional services;
(conclusions(has right), conclusions(satisfies))
NOT (conclusions(not satisfies), conclusions(not has right))

1. Analysant les résultats globaux nous pouvons conclure au sujet de l'exactitude des règle-bases.

En utilisant ces règles d'inférence nous avons détecté qu'au commencement la plupart des réponses étaient " inconnu ". Le problème est associé avec le besoin de représentation plus complète et plus puissante de la connaissance de domaine. Par exemple, la question " a la personne faite une action qui dépasse le devoir de ses fonctions? " a une grande probabilité de l réponse en tant que " inconnu " s' il n'y a aucune règle décrivant les actions qui dépassent le devoir.

Un exemple d'une telle règle pourrait être:

RULE A person has made an action that goes beyond the duty of his functions applies IF
A person is a fireman AND
A person has made an action during an off duty time AND
The action presupposes the risk of his live.

D'ailleurs, il est nécessaire de définir des règles pour "risk of live":

RULE An action presupposes the agent's risk of live IF
The agent enters a burning house in order to rescue somebody AND
The agent has no backup support.

Clairement, ce processus doit être raffiné encore: il est nécessaire de définir ce qui est une exécution de délivrance et ce qui est un support de sauvegarde.

En fait la création de la base de connaissance de domaine est un des tâches principales de notre projet. Au moment où nous sommes bâtiment immobile cette base de connaissance et nous utilisons la suivante approche:

1. Identifier les textes légaux où le résultat est inconnu;
2. Identifier la raison du résultat;
3. Ajouter le domaine nécessaire afin de décrire amplement la situation;

Après traitement de tous les textes, nous comptons avoir une bonne représentation du domaine et nous pourrions commencer la phase suivante:

Use the system to suggest the result of new cases

La suggestion de la réponse peut être expliquée et appuyée par les règles légales appliquées et par la connaissance de domaine utilisée.

1. Conclusions

Comme décrit le cadre proposé a les caractéristiques principales suivantes:

- Il peut rechercher de grandes bases de données de textes en utilisant l'information linguistique et sémantique, faisant la traduction pragmatique d'une partie des documents;
- Il peut établir et garder le contexte des interactions;
- Il représente les intentions et la croyance de l'utilisateur; Il peut mettre à jour des attitudes et impliquer des nouvelles chaque fois que nécessaire, notamment s' il y a des incohérences;
- Il emploie une taxonomie juridique afin de classifier les documents;
- Il permet à la définition des règles de modeler la connaissance légale;
- Il peut appliquer les règles définies à la base de données légale et il peut comparer les résultats.

Cependant, beaucoup de travail a reste à effectuer:

- Afin d'améliorer le module linguistique et sémantique de traduction nous devons améliorer les outils de traitement de langage naturel (les analyseurs de taggers, de corpus, syntactiques et sémantiques...)
- Le gestionnaire de dialogue devrait être généralisé pour manipuler des interactions plus puissantes;
- La connaissance de domaine doit être complètement représentée.

1. Bibliographie

- [AKPT91] James Allen, Henry Kautz, Richard Pelavin, and Josh Tenenber. Reasoning about Plans. Morgan Kaufman Publishers, Inc., 1991.
- [AP96] José Júlio Alferes and Luís Moniz Pereira. Reasoning with Logic Programming, volume 1111 of Lecture Notes in Artificial Intelligence. Springer, 1996.
- [Bra90] Michael Bratman. What is Intention?, in Intentions in Communication. MIT, 1990.
- [CL90a] P. Cohen and H. Levesque. Intention is choice with commitment. Artificial Intelligence, 42(3), 1990.
- [CL90b] Philip Cohen and Hector Levesque. Persistence, Intention, and Commitment, in Intentions in Communication, pages 33—70. MIT, 1990.
- [Car88] Sandra Carberry. Modeling the user's plans and goals. Computational Linguistics,

14(3):23—37, 1988.

- [Ebe92] Kurt Eberle. On representing the temporal structure of a natural language text. In Proceedings of the COLING'92, pages 288–294, 1992.
- [Esh88] Kave Eshghi. Abductive planning with event calculus. In Proceedings of the International Conference on Logic Programming, 1988.
- [GMD95] G. Greenleaf, A. Mowbray, and P. van Dijk. Representing and using legal knowledge in integrated decision support systems. In *Artificial Intelligence & Law*, 2(97–142), 1995
- [GMK97] G. Greenleaf, A. Mowbray, and G. King. Law on the Net via AustLII – 14 M hypertext links can't be right? In *Information Online and On Disk'97 Conference*, Sydney, 1997.
- [GMKCC95] G. Greenleaf, A. Mowbray, G. King, S. Cant, and P. Chung. More than wishful thinking: AustLII's legal inference via the World Wide Web. In *6th International Conference On Artificial Intelligence and Law*, Melbourne, 1997.
- [GS86] Barbara Grosz and Candice Sidner. Attention, intention, and the structure of discourse. *Computational Linguistics*, 12(3):175—204, 1986.
- [Hob90] Jerry Hobbs, Mark Stickel, Douglas Appelt, and Paul Martin. Interpretation as abduction. Technical Report 499, SRI International, December 1990.
- [KR93] Hans Kamp and Uwe Reyle. *From Discourse to Logic: An Introduction to Modeltheoretic Semantics of Natural Language, Formal Logic and Discourse Representation Theory*. Dordrecht: D. Reidel., 1993.
- [KS86] Robert Kowalski and Marek Sergot. A logic-based calculus of events. *New Generation Computing*, 4:67—95, 1986.
- [LA87] D. Litman and J. Allen. A plan recognition model for subdialogues in conversations. *Cognitive Science*, (11):163—200, 1987.
- [LA91] Alex Lascarides and Nicholas Asher. Discourse relations and defeasible knowledge. In Proceedings of the 29th Annual Meeting of ACL, pages 55—62, 1991.
- [LA92] Alex Lascarides, Nicholas Asher, and Jon Oberlander. Inferring discourse relations in context. In Proceedings of the 30th Annual Meeting of ACL, pages 55—62, 1992.
- [LAD93] L.M. Pereira, C. Damásio, and J.J. Alferes. Diagnosis and debugging as contradiction removal. In Luís Moniz Pereira and Anil Nerode, editors, *Proceedings of the Second International Workshop on Logic Programming and Non-monotonic Reasoning*, pages 316—330. The MIT Press, 1993.
- [LO93] Alex Lascarides and Jon Oberlander. Temporal connectives in a discourse context. In Proceedings of the 6th European chapter of ACL, EACL'93, 1993.
- [MS88] Mark Moens and Mark Steedman. Temporal ontology and temporal reference. *Computational Linguistics*, 14(2):15—28, June 1988.
- [Mis91] Lode Missiaen. *Localized Abductive Planning with the Event Calculus*. PhD thesis, Univ. Leuven, 1991.
- [PR93] J.Pinto and R.Reiter. Temporal reasoning in logic programming: A case for the situation calculus. In D.S. Warren, editor, *Proceedings of the 10th ICLP*. MIT Press, 1993.
- [Per90] Raymond Perrault. An Application of Default Logic to Speech Act Theory, in *Intentions in Communication*, chapter 9, pages 161—186. MIT, 1990.
- [Per91] F.Pereira and M.Pollack. Incremental interpretation. *Artificial Intelligence*, 50:40—82, 1991.
- [Pol90] Martha Pollack. Plans as Complex Mental Attitudes, in *Intentions in Communication*, chapter 5, pages 77—104. MIT, 1990.
- [QL95] Paulo Quaresma and José Gabriel Lopes. Unified logic programming approach to the abduction of plans and intentions in information-seeking dialogues. *Journal of Logic Programming*, (54), 1995.
- [Qua97] Paulo Quaresma. *Inference of Attitudes in Dialogues*. PhD Thesis, Universidade Nova de Lisboa, 1997. In Portuguese.
- [RL92] Irene Pimenta Rodrigues and José GabrielPereira Lopes. Discourse temporal structure. In Proceedings of the COLING'92, 1992.
- [RL93a] Irene Pimenta Rodrigues and José Gabriel Lopes. Building the text temporal structure. In *Progress in Artificial Intelligence: 6th Portuguese Conference on AI*. Springer-Verlag, 1993.
- [RL93b] Irene Pimenta Rodrigues and José Gabriel Lopes. Text processing in a logic programming framework: The interpretation of tense and aspect. In Proceedings of the fourth International Workshop on Natural Language Understanding and Logic Programming, NLULP4, 1993.
- [Son91] F.Song. *A Processing Model for Temporal Analysis and its Application to Plan Recognition*. PhD thesis, University of Waterloo, Waterloo, Ontario, Canada, 1991.