A First Approach to Argument-based Recommender Systems based on Defeasible Logic Programming

Carlos Iván Chesñevar
Departament of Computer Science
Universitat de Lleida
Jaume II, 69 – 25001 Lleida, SPAIN
Email: cic@eup.udl.es

Ana Gabriela Maguitman
Computer Science Department
Indiana University
Bloomington, IN 47405-7104, USA
Email: anmaguit@cs.indiana.edu

Guillermo Ricardo Simari
Dept. of Computer Science and Eng.
Universidad Nacional del Sur
Alem 1253 – 8000 B.Blanca, ARGENTINA
Email: grs@cs.uns.edu.ar

Abstract

Recommender systems have evolved in the last years as specialized tools to assist users in a plethora of computer-mediated tasks by providing guidelines or hints. Most recommender systems are aimed at facilitating access to relevant items, a situation particularly common when performing web-based tasks. At the same time, defeasible argumentation has evolved as a successful approach in AI to model commonsense qualitative reasoning, with applications in many areas, such as agent theory, knowledge engineering and legal reasoning. This paper presents a first approach towards the integration of web-based recommender systems with a defeasible argumentation framework. The final goal is to enhance practical reasoning capabilities of current recommender system technology by incorporating argument-based qualitative inference.

KEYWORDS: Recommender systems, Defeasible Argumentation, Decision support systems, Practical reasoning

Introduction and motivations

Recommender systems (also known as *suggesters*) have evolved in the last years as specialized tools to assist users in a plethora of computer-mediated tasks by providing guidelines or hints (Resnick & Varian 1997; Konstan 2004). Most recommender systems are aimed at helping users to deal with the problem of information overload by facilitating access to relevant items (Maes 1994). A common technique adopted by many suggester systems is collaborative filtering, which infers preferences of individual users based on the behavior of multiple users (e.g., (Goldberg *et al.* 1992)). Collaborative filtering is based on the assumption that human preferences are correlated. Other recommender systems are content-based, which are driven by the premise that user's preferences tend to persist through time. Therefore, content-based recommender systems build on similarities between potential recommendations and the objects that the user liked in the past. A combination of collaborative-filtering and content-based recommendation gives rise to hybrid recommender systems (e.g., (Balabanovic, Shoham, & Yun 1995; Balabanović & Shoham 1997; Billsus & Pazzani 1999; Claypool *et al.* 1999)). Given the huge amount of information existing on the Web, it is not surprising that the great majority of the recommender systems have been built around content and resources available online (e.g., (Armstrong *et al.* 1995; Mladenic 1996; Lieberman 1995; Pazzani, Muramatsu, & Billsus 1996; Doorenbos, Etzioni, & Weld 1997)).

Although the effectiveness of existing recommender systems is remarkable, they still have serious limitations as they are unable to perform qualitative inference on the recommendations they offer and are incapable of dealing with the defeasible nature of users' preferences. In this context, defeasible argumentation frameworks (Chesñevar, Maguitman, & Loui 2000; Prakken & Vreeswijk 2002) have evolved to become a sound setting to formalize commonsense qualitative reasoning. Recent research has shown that argumentation can be integrated in a growing number of real-world applications such as multiagent systems (Parsons, Sierrra, & Jennings 1998; Amgoud, Maudet, & Parsons 2002; Parsons & McBurney), legal reasoning (Prakken & Sartor 2002), knowledge engineering (Carbogim, Robertson, & Lee 2000), analysis of news reports (Hunter 2001) clustering (Gomez & Chesñevar 2004), argumentation support systems (Verheij 2003), mediation systems and computer-supported collaborative argumentation (Maudet & Moore 1999; Reed & Walton 2001; Gordon & Karacapilidis 1997; Loui *et al.* 1997).

This paper presents a first approach to integrate recommender system technologies with a defeasible argumentation framework. The basic idea is to model the preference criteria associated with the active user and a pool of users by means of facts, strict rules and defeasible rules. These preference criteria are combined with additional background information and used by an argumentation framework to prioritize potential recommendations, thus enhancing the final results provided to the active user. The rest of the paper is structured as follows. Section briefly outlines the fundamentals of DeLP, a defeasible argumentation formalism based on logic programming. Section presents a generic characterization of recommender systems. Section discusses our proposal for characterizing argument-based recommender systems. Section presents a case study which illustrates how the proposed approach works. Finally, Section discusses related work and presents the main conclusions that have been obtained.

Modelling Argumentation in DeLP

Several defeasible argumentation frameworks have been developed on the basis of extensions to logic programming (see (Chesñevar, Maguitman, & Loui 2000; Prakken & Vreeswijk 2002; Kakas & Toni 1999)). *Defeasible logic programming* (DeLP) (García & Simari 2004) is one of such formalisms, combining results from defeasible argumentation theory and logic programming. DeLP is a suitable framework for building real-world applications that deal with incomplete and contradictory information in dynamic domains. In what follows we will present a brief overview of the DeLP framework. A more in-depth treatment can be found elsewhere (García & Simari 2004).

A defeasible logic program is a set (Π, Δ) of Horn-like clauses, where Π and Δ stand for sets of *strict* and *defeasible* knowledge, resp. The set Π involves *strict rules* of the form $p \leftarrow q_1, \ldots, q_k$ and *facts* (strict rules with empty body), and it is assumed to be *non-contradictory*. The set Δ involves *defeasible rules* of the form $p \mathrel{-\!\!\prec} q_1, \ldots, q_k$.The underlying logical language is that of extended logic programming (Gelfond & Lifschitz 1990), enriched with a special symbol "$\mathrel{-\!\!\prec}$" to denote defeasible rules. Both default and classical negation are allowed (denoted not and $\sim$, resp.).[1] DeLP rules are to be thought of as *inference rules* rather than implications in the object language. Deriving literals in DeLP results in the construction of *arguments*. Formally:

Definition 1 (Argument) *Given a DeLP program $\mathcal{P}$, an* argument $\mathcal{A}$ *for a query q, denoted $\langle \mathcal{A}, q \rangle$, is a subset of ground instances of defeasible rules in $\mathcal{P}$ such that:*

1. *there exists a* defeasible derivation *for q from $\Pi \cup \mathcal{A}$;*
2. *$\Pi \cup \mathcal{A}$ is non-contradictory (i.e, $\Pi \cup \mathcal{A}$ does not entail two complementary literals p and $\sim p$, nor does $\mathcal{A}$ contain literals s and* not *s, for any p, s in $\mathcal{P}$), and*
3. *$\mathcal{A}$ is the minimal set (with respect to set inclusion) satisfying (1) and (2).*

An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a sub-argument *of another argument $\langle \mathcal{A}_2, q_2 \rangle$ if $\mathcal{A}_1 \subseteq \mathcal{A}_2$. Given a DeLP program $\mathcal{P}$, $Args(\mathcal{P})$ denotes the set of all possible arguments that can be derived from $\mathcal{P}$.*

The notion of defeasible derivation corresponds to the usual query-driven SLD derivation used in logic programming, performed by backward chaining on both strict and defeasible rules; in this context a negated literal $\sim p$ is treated just as a new predicate name no_p. Minimality imposes the 'Occam's razor principle' (Simari & Loui 1992) on arguments.any superset $\mathcal{A}'$ of $\mathcal{A}$ can be proven to be 'weaker' than $\mathcal{A}$ itself, as the former relies on more defeasible information. The non-contradiction requirement forbids the use of (ground instances of) defeasible rules in an argument $\mathcal{A}$ whenever $\Pi \cup \mathcal{A}$ entails twocomplementary literals. It should be noted that non-contradiction captures the two usual approaches to negation in logic programming (viz. default negation and classic negation), both of which are present in DeLP and related to the notion of counterargument, as shown next.

[1] The definitions that follow summarize DeLP with default negation (see discussion in (García & Simari 2004, pages 30-33)).

Definition 2 (Counterargument – Defeat) *An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a* counterargument *for an argument $\langle \mathcal{A}_2, q_2 \rangle$ if and only if*

1. *There is an subargument $\langle \mathcal{A}, q \rangle$ of $\langle \mathcal{A}_2, q_2 \rangle$ such that the set $\Pi \cup \{q_1, q\}$ is contradictory, or*
2. *An extended literal* not q_1 *is present in some rule in $\mathcal{A}_2$.*[2]

A preference criterion $\preceq\, \subseteq Args(\mathcal{P}) \times Args(\mathcal{P})$ will be used to decide among conflicting arguments. An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a defeater *for an argument $\langle \mathcal{A}_2, q_2 \rangle$ if $\langle \mathcal{A}_1, q_1 \rangle$ counterargues $\langle \mathcal{A}_2, q_2 \rangle$, and $\langle \mathcal{A}_1, q_1 \rangle$ is preferred over $\langle \mathcal{A}_2, q_2 \rangle$ with respect to $\preceq$. For cases (1) and (2) above, we distinguish between* proper *and* blocking defeaters *as follows:*

- *In case (1), the argument $\langle \mathcal{A}_1, q_1 \rangle$ will be called a* proper defeater *for $\langle \mathcal{A}_2, q_2 \rangle$ if and only if $\langle \mathcal{A}_1, q_1 \rangle$ is strictly preferred over $\langle \mathcal{A}, q \rangle$ with respect to $\preceq$.*
- *In case (1), if $\langle \mathcal{A}_1, q_1 \rangle$ and $\langle \mathcal{A}, q \rangle$ are unrelated to each other with respect to $\preceq$, or in case (2), $\langle \mathcal{A}_1, q_1 \rangle$ will be called a* blocking defeater *for $\langle \mathcal{A}_2, q_2 \rangle$.*

Specificity (Simari & Loui 1992) is used in DeLP as a syntactic preference criterion among conflicting arguments, favoring those arguments that are *more informed* or *more direct* (Simari & Loui 1992; Stolzenburg *et al.* 2003). However, other alternative preference criteria could also be used.

An *argumentation line* starting in $\langle \mathcal{A}_0, q_0 \rangle$ (denoted $\lambda^{\langle \mathcal{A}_0, q_0 \rangle}$) is a sequence $[\langle \mathcal{A}_0, q_0 \rangle, \langle \mathcal{A}_1, q_1 \rangle, \langle \mathcal{A}_2, q_2 \rangle, \ldots, \langle \mathcal{A}_n, q_n \rangle \ldots]$ that can be thought of as an exchange of arguments between two parties, a *proponent* (evenly-indexed arguments) and an *opponent* (oddly-indexed arguments). Each $\langle \mathcal{A}_i, q_i \rangle$ is a defeater for the previous argument $\langle \mathcal{A}_{i-1}, q_{i-1} \rangle$ in the sequence, $i > 0$. In order to avoid *fallacious* reasoning, dialectical constraints are imposed on such an argument exchange to be considered rationally acceptable in light of a given program $\mathcal{P}$. An argumentation line satisfying such constraints is said to be *acceptable*, and can be proven to be finite (see (García & Simari 2004) for details).

Given a program $\mathcal{P}$ and an initial argument $\langle \mathcal{A}_0, q_0 \rangle$, the set of all acceptable argumentation lines starting in $\langle \mathcal{A}_0, q_0 \rangle$ accounts for a whole dialectical analysis for $\langle \mathcal{A}_0, q_0 \rangle$ (i.e., all possible dialogues rooted in $\langle \mathcal{A}_0, q_0 \rangle$), formalized as a *dialectical tree* $\mathcal{T}_{\langle \mathcal{A}_0, q_0 \rangle}$. Nodes in a dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, q_0 \rangle}$ can be marked as *undefeated* and *defeated* nodes (U-nodes and D-nodes, resp.): all leaves in $\mathcal{T}_{\langle \mathcal{A}_0, q_0 \rangle}$ will be marked U-nodes (as they have no defeaters), and every inner node is to be marked as *D-node* iff it has at least one U-node as a child, and as *U-node* otherwise. An argument $\langle \mathcal{A}_0, q_0 \rangle$ is ultimately accepted as valid (or *warranted*) with respect to a DeLP program $\mathcal{P}$ iff the root of its associated dialectical tree $\mathcal{T}_{\langle \mathcal{A}_0, q_0 \rangle}$ is labeled as *U-node*.

Solving a query q with respect to a given program $\mathcal{P}$ accounts for determining whether q is supported by a warranted argument. Different doxastic attitudes are distin-

[2] The first notion of attack is borrowed from the Simari-Loui framework (Simari & Loui 1992); the second one is related to Dung's argumentative approach to logic programming (Dung 1993) as well as to other formalizations, such as (Prakken & Sartor 1997; Kowalski & Toni 1996).

guished when answering query q according to the associated status of warrant, in particular:

1. Believe q when there is a warranted argument for q that follows from $\mathcal{P}$.
2. Believe $\sim q$ when there is a warranted argument for $\sim q$ that follows from $\mathcal{P}$.
3. Believe q is *undecided* whenever neither q nor $\sim q$ are supported by warranted arguments in $\mathcal{P}$.

Recommender Systems: fundamentals

Recommender systems are programs that create a model of the user's preferences or user's task with the purpose of facilitating access to items (e.g. news, web pages, books, etc.) that the user may find useful (Resnick & Varian 1997; Konstan 2004). While in many situations the user explicitly posts a request for recommendations in the form of a query, many recommender systems attempt to anticipate the user's needs and are capable of proactively providing assistance (Rhodes & Maes 2000; Rhodes 2000; Budzik & Hammond 2000). In order to come up with recommendations for user queries, conventional recommender systems rely on *similarity measures* between users or contents, computed on the basis of methods coming either from the information retrieval or the machine learning communities. Recommender systems adopt mainly two different views to help predict information needs. The first approach is known as *user modeling* and relies on the use of a profile or model of the users, which is created by observing users' behavior (e.g., (Linton, Joy, & Schaefer 1999; Deshpande & Karypis 2004)). The second approach is based on *task modeling*, and recommendations are based on the context in which the user is immersed (e.g. (Budzik, Hammond, & Birnbaum 2001; Leake *et al.* 2000)). The context may consist of an electronic document the user is editing, web pages the user has recently visited, etc.

Two main techniques have been used to compute recommendations: *content-based* and *collaborative filtering*. Content-based recommenders frequently use machine-learning techniques to induce a profile of the active user. Typically, a model of the active user is stored as a list of rated items. In order to determine if a new item is a potentially good recommendation, content-based recommender systems rely on *similarity measures* between the new items and the rated items stored as part of the user model. On the other hand, recommender systems based on collaborative filtering maintain a pool of users' profiles. For a given active user, collaborative recommender systems find other similar users whose ratings strongly correlate with the current user. New items not rated by the active user can be presented as suggestions if similar users have rated them highly.

Some systems combine content-based recommendation and collaborative filtering giving rise to *hybrid recommender systems* (e.g., (Balabanovic, Shoham, & Yun 1995; Balabanović & Shoham 1997; Billsus & Pazzani 1999; Claypool *et al.* 1999)). Fig. 1 illustrates the main components of this approach. A hybrid recommender system typically generates a model of the active user by monitoring

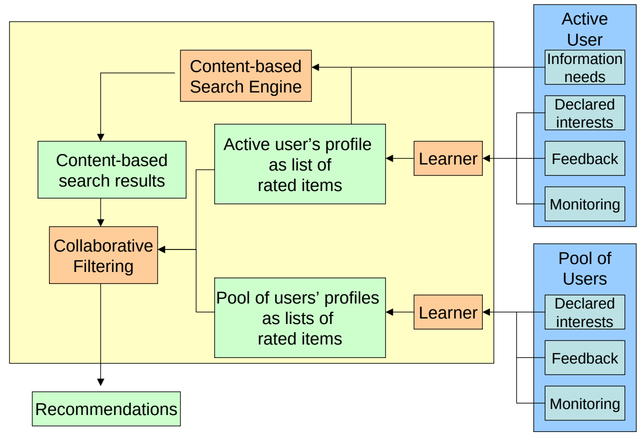

Figure 1: A generic framework for an Hybrid Recommender System

the user behavior or by analyzing his/her declared interest or feedback. The generated user model is usually combined with the user information needs and a request for recommendations is presented to a search engine. In addition, the system maintains a pool of profiles from other users, making possible the application of collaborative filtering to further refine the selected set of recommendations.

Although hybrid recommender systems are substantially more effective than the basic content-based and collaborative filtering approaches, existing systems are still limited. On the one hand, existing recommender systems are incapable of dealing formally with the defeasible nature of users' preferences in complex environments. Decisions about user preferences are mostly based on heuristics which rely on ranking previous user choices or gathering information from other users with similar interests. On the other hand, the quantitative approaches adopted by most existing recommender systems do not have a clean underlying model. This makes it hard to provide users with a clear explanation of the factors and procedures that led the system to come up with certain recommendations. As a result, serious trustworthiness issues may arise, especially in those cases when business interests are involved, or when external manipulation is possible.

We contend that defeasible argumentation can be integrated into existing recommender system technologies, paving the way to solve the above problems. We will analyze our proposal in the next section.

Argument-Based Recommender Systems

A fundamental problem addressed by recommender systems is determining which items are *relevant* to a user information needs (i.e., which items are worthwhile, given the user's preferences and user's task.) Recommendation results can be displayed to the user in different formats (e.g. using charts, colors, or some more specialized notation). In most cases, independently of the format used, the results shown are sorted according to some preference criterion (usually provided by the user). Thus, for example, when looking for

ALGORITHM Recommend_on_Query
INPUT: Query q,
DeLP program $\mathcal{P} = \mathcal{P}_{user} \cup \mathcal{P}_{pool} \cup \mathcal{P}_{domain}$.
OUTPUT: List L_{new} {*recommendation results wrt* $\mathcal{P}'$}
BEGIN
Let $L = [s_1, s_2, \ldots s_k]$ be the output of solving query q
wrt content-based search engine SE
{L *is the list of (the first* k*) results obtained from query* q *via* SE }
$\mathcal{P}_{search}$ = {facts encoding $info(s_1), info(s_2) \ldots info(s_k)$}
{$info(S_i)$ *stands for features associated with result* S_i }
$\mathcal{P}'$:= $\mathbf{Revise}$ ($\mathcal{P} \cup \mathcal{P}_{search}$).
{$\mathbf{Revise}$ *stands for a belief revision operator*
{*to ensure consistency in* $\mathcal{P}'$ }
Initialize S^w, S^u, and S^d as empty sets.
{S^w, S^u, *and* S^d *stand for the set of results* s_i*'s which are warranted as relevant, undecided and warranted as non-relevant, respectively* }
FOR EVERY $s_i \in L$
DO
Solve query $rel(s_i)$ using DeLP program $\mathcal{P}'$
IF $rel(s_i)$ is warranted **THEN** add s_i to S^w
ELSE
IF $\sim rel(s_i)$ is warranted **THEN** add s_i to S^d
ELSE add s_i to S^d
Return Recommendation L_{new} =
$[s_1^w, s_2^w, \ldots, s_{j1}^w, s_1^u, s_2^u, \ldots, s_{j2}^u, s_1^d, \ldots, s_{j3}^d]$
END

Figure 2: High-level algorithm for solving queries in an argument-based recommender system

a recommendation about books in a web-based bookstore, recommendations can be sorted in terms of price, availability, etc. In the sequel we will assume (without loss of generality) that recommendation results can be represented as a list [s_1, s_2, $\ldots s_k$], assuming that the earlier a result appears in the list, the earlier it is shown on the screen and the more relevant for the user it is.

A common problematic situation occurs when hundreds or thousands of recommendation results are available, so that a detailed user analysis of the whole search space becomes extremely expensive. Experienced users of recommender systems rely many times on the combination of different (mostly implicit) preference criteria to build and evaluate alternative *hypotheses* for filtering recommendation results. In this context, meta-information associated with recommendation results turns out to be particularly helpful. Thus, as an example, particular features from URL's and HTML pages (e.g. web domain, year, author, etc.) may help the user discard some recommendation results he/she does not find useful.

Since users' preference criteria provide incomplete and potentially inconsistent knowledge about the search domain, our proposal is to model the users' preference criteria in terms of a DeLP program built on top of a content-based search engine. The resulting framework is an *argument-based recommender system,* in which recommendations are provided on the basis of arguments built upon information from the active user, the pool of users and domain (background) knowledge (see Figure 3(a)). The above aspects are to be encoded as a DeLP program $\mathcal{P} = \mathcal{P}_{user} \cup \mathcal{P}_{pool} \cup \mathcal{P}_{domain}$. Sets $\mathcal{P}_{user}$ and $\mathcal{P}_{pool}$ represent preferences and behavior of the active user and the pool of users, respectively. In the case of the active user, his/her profile can be encoded as facts and rules in DeLP. In the case of the pool of users, rule induction techniques are in order[3] resulting in defeasible rules characterizing trends and general preference criteria (e.g., *normally if a given user likes X then she also likes Y*). The set $\mathcal{P}_{domain}$ represents the domain (background) knowledge, encoded using facts and rules in DeLP.

The user's information needs are presented to a content-based search engine, which returns a list of search results $[s_1, s_2, \ldots, s_k]$. In a typical hybrid recommender system, such results are contrasted against the active user's profile and the pool of users' profiles to obtain personalized recommendations as a final output. Our proposal is based on properly encoding the list of search results as DeLP facts. We can assume that s_i is a unique name characterizing a piece of information $info(s_i)$, in which a number of associated features (meta-tags, filename, URL, etc.) can be identified. We assume that such features can be identified and extracted from $info(s_i)$ by some specialized tool, as suggested by Hunter (Hunter 2001) in his approach to dealing with structured news reports (see discussion in Section). Such features will be encoded as a set $\mathcal{P}_{search}$ of new DeLP facts, extending thus the original program $\mathcal{P}$ into a new program $\mathcal{P}'$. A special operator $\mathbf{Revise}$ deals with possible inconsistencies found in $\mathcal{P}_{search}$ with respect to $\mathcal{P}'$, ensuring $\mathcal{P} \cup \mathcal{P}_{search}$ is not contradictory.[4]

At this point the obtained search results can be analyzed in the context of $\mathcal{P}'$. We will consider a distinguished predicate name rel for analyzing the *relevance* of every recommendation result s_i. In this setting, the existence of a warranted argument $\langle A, rel(s_i) \rangle$ built on the basis of DeLP program $\mathcal{P}$ will allow to conclude that s_i is a candidate recommendation relevant to the user's information needs. We will classify the elements in the original list L of content-based search results into three sets, namely:

- S^w (warranted results): those results s_i for which there exists at least one warranted argument supporting $rel(s_i)$ based on $\mathcal{P}'$.
- S^u (undecided results): those results s_i for which there is no warranted argument for $rel(s_i)$, neither there is a warranted argument for $\sim rel(s_i)$ on the basis of $\mathcal{P}'$, and
- S^d (defeated results): those results s_i such that there is a warranted argument supporting $\sim rel(s_i)$ on the basis of $\mathcal{P}'$.

Figure 3(a) presents an outline of the proposed approach. Note that the above classification has a direct correspondence with the doxastic attitudes associated with answers to DeLP queries. The final output presented to the user will

[3]An approach for inducing defeasible rules from association rules can be found in (Governatori & Stranieri 2001).

[4]E.g contradictory facts may be found on the web; a simple belief revision criterion is that those facts with newer timestamp are preferred over older ones.

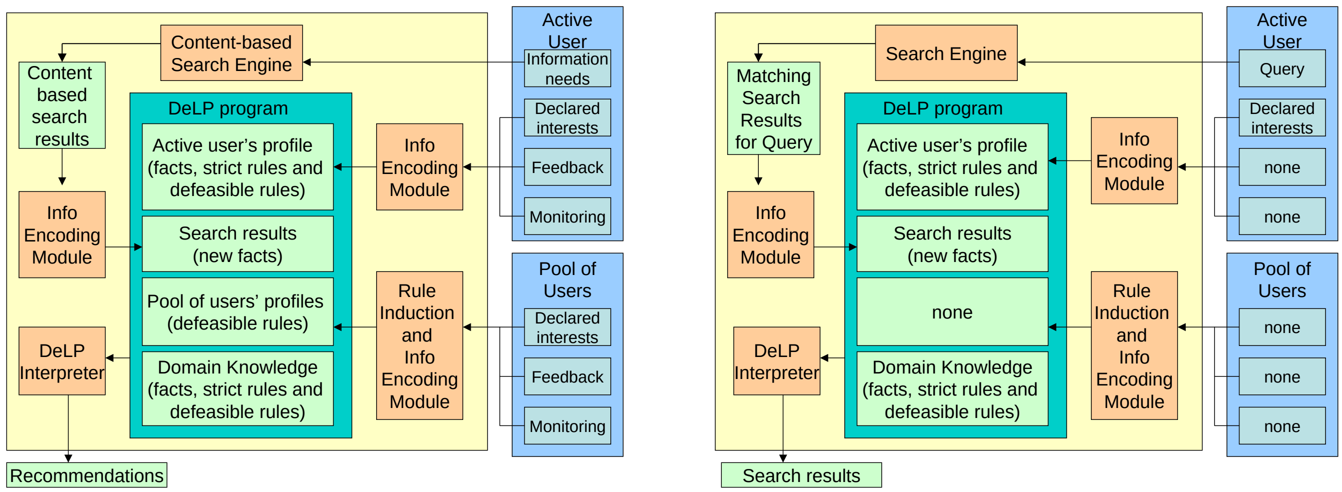

Figure 3: (a) A generic framework for an Argument-Based Recommender System; (b) Enhanced search engine as a particular instance of an Argument-Based Recommender System

be a sorted list L' in which the elements of L are ordered according to their epistemic status with respect to $\mathcal{P}$'. This process can be characterized in terms of the high-level algorithm shown in Figure 2. We must remark that it is always possible to ensure that the computation of warrant cannot lead to contradiction (García & Simari 2004): if there exists a warranted argument $\langle A, h\rangle$ on the basis of a program $\mathcal{P}$, then there is no warranted argument $\langle B, \sim h\rangle$ based on $\mathcal{P}$.

A Case Study: Solving Web Search Queries

In this section we will outline an example (adapted from (Chesñevar & Maguitman 2004)) of how the proposed approach works in the context of solving web search queries. In this context, the recommendation system aims at providing an enriched web search engine which categorizes results, as outlined in Fig. 3(b). Thus, the resulting framework can be seen as a particular instance of an argument-based recommendation system, where the user's needs correspond to strings to be searched on the web. The content-based engine is a conventional search engine (e.g. GOOGLE). Final recommendation results for a query q are prioritized according to domain background knowledge and the user's declared preferences. It must be remarked that the ARGUENET system (Chesñevar & Maguitman 2004) is an instance of this particular argument-based recommender framework.

Consider a journalist who wants to search for news articles about recent outbreaks of bird flu. A query q containing the terms *news*, *bird*, and *flu* will return thousands of search results. Our journalist may have some implicit knowledge to guide the search, such as: (1) she always considers relevant the newspaper reports written by Bob Beak; (2) she usually considers relevant the reports written by trustworthy journalists; (3) Reports written by trustworthy journalists which are out of date are usually not relevant; (4) Knowing that a journalist has not faked reports provides a tentative reason to believe he or she is trustworthy. By default, every journalist is assumed to be trustworthy. (5) Japanese and Thailandian newspapers usually offer a biased viewpoint on bird flu outbreaks; (6) The *"Japanese Times"* (http://jpt.jp) is a Japanese newspaper which she usually considers non biased; (7) Chin Yao Lin is known to have faked a report. Such rules and facts can be modeled in terms of a DeLP program $\mathcal{P}$ shown in Fig. 4. Note that some rules in $\mathcal{P}$ rely on "built in" predicates computed elsewhere and not provided by the user.[5]

For the sake of example, suppose that the above query returns a list of search results L=[s_1, s_2, s_3, s_4]. Most of these results will be web pages annotated with a number of HTML or XML references (e.g. author, date, URL, etc.). Such references can be encoded as a collection of DeLP facts as shown in Fig. 4(b). Following the algorithm shown in Fig. 2 we can now analyze s_1, s_2, s_3 and s_4 in the context of a new DeLP program $\mathcal{P}'$=$\mathcal{P}\cup Facts$, where $Facts$ denotes the set corresponding to the collection discussed above and $\mathcal{P}$ corresponds to domain knowledge and the user's preference theory about the search domain.[6] For each s_i, the query $rel(s_i)$ will be analyzed in light of this new program $\mathcal{P}'$.

Consider the case for s_1. The search for an argument for $rel(s_1)$ returns $\langle \mathcal{A}_1, rel(s_1)\rangle$: s_1 should be considered relevant since it corresponds to a newspaper article written by Chin Yao Lin who is considered a trustworthy author (note that every journalist is considered to be trustworthy by default.) Here we have [7] $\mathcal{A}_1$={ $rel(s_1) \prec author(s_1, chin\_yao\_lin)$, $trust(chin\_yao\_lin)$; $trust(chin\_yao\_lin) \prec \text{not}\ faked\_news(chin\_yao\_lin)$ }. Search for defeaters for $\langle \mathcal{A}_1, rel(s_1)\rangle$ will result in finding a proper defeater $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$: s_1 is not relevant as

[5] E.g., determining the country of origin corresponding to a specific web domain can be found querying Internet directory services such as WHOIS.

[6] In this particular context, note that $\mathcal{P} = \mathcal{P}_{domain} \cup \mathcal{P}_{user}$.

[7] For the sake of clarity, we use semicolons to separate elements in an argument $\mathcal{A} = \{e_1 ; e_2 ; \ldots ; e_k \}$.

$$
\begin{array}{rcl}
rel(X) & \prec & author(X, A), trust(A). \\
\sim rel(X) & \prec & author(X, A), trust(A), \\
 & & outdated(X). \\
trust(A) & \prec & \text{not } faked\_news(A). \\
\sim rel(X) & \prec & address(X, Url), biased(Url). \\
biased(Url) & \prec & thailandian(Url). \\
biased(Url) & \prec & japanese(Url). \\
\sim biased(Url) & \prec & japanese(Url), domain(Url, D), \\
 & & D = \text{“}jpt.jp\text{”}. \\
rel(X) & \leftarrow & author(X, bob\_beak). \\
oudated(X) & \leftarrow & date(X, D), getdate(Today), \\
 & & (Today - D) > 100. \\
thailandian(X) & \leftarrow & \text{[Computed elsewhere]} \\
japanese(X) & \leftarrow & \text{[Computed elsewhere]} \\
domain(Url, D) & \leftarrow & \text{[Computed elsewhere]} \\
getdate(T) & \leftarrow & \text{[Computed elsewhere]} \\
faked\_news(chin\_yao\_lin) \leftarrow & &
\end{array}
$$

$author(s_1, chin\_yao\_lin).$ $address(s_1, \text{“}jpt.jp/...\text{”}).$ $date(s_1, 20031003).$
$author(s_2, jen\_doe).$ $address(s_2, \text{“}news.co.uk/...\text{”}).$ $date(s_1, 20001003).$
$author(s_3, jane\_truth).$ $address(s_3, \text{“}jpt.jp/...\text{”}).$ $date(s_3, 20031003).$
$author(s_4, bob\_beak).$ $address(s_4, \text{“}mynews.com/...\text{”}).$ $date(s_4, 20031003).$

Figure 4: (a) DeLP program modeling preferences of a journalist; (b) Facts encoded from original web search results

it comes from a Japanese newspaper, which is by default assumed to be biased about bird flu. In this case we have $\mathcal{A}_2$={ $\sim rel(s_1) \prec address(s_1, \text{“}jpt.jp...\text{”}), biased\ (\text{“}jpt.jp...\text{”})$; $biased(\text{“}jpt.jp...\text{”}) \prec japanese\ (\text{“}jpt.jp...\text{”})$ }. Note that we also have an argument $\langle \mathcal{A}_3, \sim biased(\text{“}jpt.jp...\text{”})\rangle$ which defeats $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$, reinstating $\langle \mathcal{A}_1, rel(s_1)\rangle$: Usually articles from the “Japanese Times” are not biased. In this case we have $\mathcal{A}_3$={ $\sim biased(\text{“}jpt.jp...\text{”}) \prec japanese(\text{“}jpt.jp...\text{”}), domain(\text{“}jpt.jp...\text{”}, \text{“}jpt.jp\text{”}), (\text{“}jpt.jp\text{”} = \text{“}jpt.jp\text{”})$ }. Finally, another defeater for $\langle \mathcal{A}_1, rel(s_1)\rangle$ is found, namely $\langle \mathcal{A}_4, faked\_news(chin\_yao\_lin)\rangle$, with $\mathcal{A}_4 = \emptyset$. No other arguments need to be considered. The resulting dialectical tree rooted in $\langle \mathcal{A}_1, rel(s_1)\rangle$ as well as its corresponding marking is shown in Figure 5a (left). The root node is a D-node (defeated), and hence $\langle \mathcal{A}_1, rel(s_1)\rangle$ is not warranted. Carrying out a similar analysis for $\sim rel(s_1)$ results in the dialectical tree shown in Figure 5a (right). The root node $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$ is marked as D-node. There are no other candidate arguments to consider; hence s_1 is deemed as ***undecided***.

The case of s_2 is analogous. The argument $\langle \mathcal{B}_1, rel(s_2)\rangle$ can be built, with $\mathcal{B}_1$={ $rel(s_2) \prec author(s_2, jen\_doe), trust(jen\_doe)$; $trust(jen\_doe) \prec \text{not } faked\_news\ (jen\_doe)$ }. This argument is defeated by a proper defeater $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$, with $\mathcal{B}_2$={$\sim rel(s_2) \prec author(s_2, jen\_doe), trust(jen\_doe), outdated(s_2)$; $trust(jen\_doe) \prec \text{not } faked\_news(jen\_doe)$}. There are no more arguments to consider, and $\langle \mathcal{B}_1, rel(s_2)\rangle$ is deemed as non warranted (the resulting marked dialectical tree is shown in Fig. 5b (left)). The analysis of $\sim rel(s_2)$ results in an single argument. Consequently, its associated dialectical tree has a single node $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$ and it is ***warranted***.

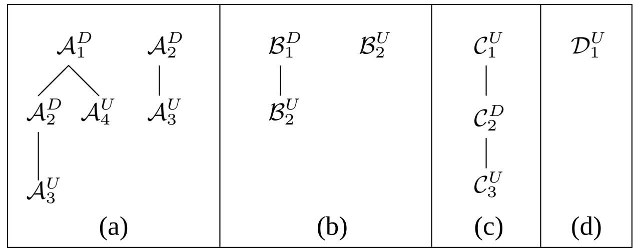

Figure 5: Dialectical trees associated with (a) $\langle \mathcal{A}_1, rel(s_1)\rangle$ and $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$; (b) $\langle \mathcal{B}_1, rel(s_2)\rangle$ and $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$; (c) $\langle \mathcal{C}_1, rel(s_3)\rangle$ and (d) $\langle \mathcal{D}_1, rel(s_4)\rangle$

Following the same line of reasoning used in the case of s_1 we can analyze the case of s_3. An argument $\langle \mathcal{C}_1, rel(s_3)\rangle$, with $\mathcal{C}_1$={ $rel(s_3) \prec author(s_3, jane\_truth), trust(jane\_truth)$; $trust(jane\_truth) \prec \text{not } faked\_news(jane\_truth)$ } can be built supporting the conclusion $rel(s_3)$ (a newspaper article written by Jane Truth is relevant as she can be assumed to be a trustworthy author). A defeater $\langle \mathcal{C}_2, \sim rel(s_3)\rangle$ will be found: s_1 is not relevant as it comes from a Japanese newspaper, which by default is assumed to be biased about bird flu. Here we have $\mathcal{C}_2$={ $\sim rel(s_3) \prec address(s_3, \text{“}jpt.jp...\text{”}), biased\ (\text{“}jpt.jp...\text{”})$; $biased(\text{“}jpt.jp...\text{”}) \prec japanese\ (\text{“}jpt.jp...\text{”})$ }. But this defeater in its turn is defeated by a third argument $\langle \mathcal{C}_3, biased(s_3)\rangle$, as usually articles from the “Japanese Times” are not biased. In this case we have $\mathcal{C}_3$={ $\sim biased(\text{“}jpt.jp...\text{”}) \prec japanese(\text{“}jpt.jp...\text{”}), domain(\text{“}jpt.jp...\text{”}, \text{“}jpt.jp\text{”}), (\text{“}jpt.jp\text{”} = \text{“}jpt.jp\text{”})$ }. The resulting dialectical tree for $\langle \mathcal{C}_1, rel(s_3)\rangle$ is shown in Fig. 5c (left)). The original argument $\langle \mathcal{C}_1, rel(s_3)\rangle$ can be thus deemed as ***warranted***.

Finally let us consider the case of s_4. There is an argument $\langle \mathcal{D}_1, rel(s_4)\rangle$ with $\mathcal{D}_1 = \emptyset$, as $rel(s_4)$ follows directly from the strict knowledge in $\mathcal{P}$. Clearly, there is no defeater for an empty argument, and therefore $rel(s_4)$ is ***warranted***. The associated dialectical tree is shown in Fig. 5d.

Applying the criterion given in the algorithm shown in Fig. 2, the initial list of search results [s_1, s_2, s_3, s_4] will be shown as [s_3, s_4, s_1, s_2] (as $\langle \mathcal{C}_1, rel(s_3)\rangle$ and $\langle \mathcal{D}_1, rel(s_4)\rangle$ are warranted, $\langle \mathcal{A}_1, rel(s_3)\rangle$ is undecided and $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$ is warranted (i.e., s_2 is warranted to be a non-relevant result).

Related work. Conclusions

In this paper we have outlined a computational framework that provides a novel way of enhancing recommendation technologies through the use of qualitative analysis using argumentation.We have shown how DeLP provides a suitable tool for carrying on such analysis. It must be remarked that an abstract machine for an efficient implementation of DeLP has been developed, based on an extension of the WAM (Warren's Abstract Machine) for Prolog. Features concerning an efficient implementation of DeLP and a comparison to other logic programming formalisms have been recently studied (Stolzenburg *et al.* 2003; Chesñevar *et al.* 2003).

The proposed framework operates on top of a conventional search engine, providing a powerful abstraction for solving queries on the basis of the users' information. Many personalized Web recommender systems that operate on top of Internet services have been proposed over the past years (e.g., (Armstrong *et al.* 1995; Mladenic 1996; Lieberman 1995; Pazzani, Muramatsu, & Billsus 1996)). Existing Web recommender tools take into account the user's interests to rank or filter web pages, but differ from our proposal in that they do not attempt to perform a qualitative analysis to warrant recommendations. There are currently many ambitious projects to facilitate automatic qualitative reasoning by relying on the realization of the Semantic Web vision (Berners-Lee 1998; Berners-Lee, Hendler, & Lassila 2001). Although the concretization of such a vision is still underway, the use of defeasible argumentation for qualitative analysis can also be naturally integrated into such approaches.

One important issue in our proposal is the need to extract relevant features from Web search results, encoding them as part of a DeLP program. Although HTML tags associated to Web documents are not intended to convey a formal semantics, these tags can be usefully exploited to extract meaningful content (Doorenbos, Etzioni, & Weld 1997; Ashish & Knoblock 1997; Kushmerick, Weld, & Doorenbos 1997). On the other hand, the emergence of XML as a standard for data representation on the Web contributes to further simplify the above problem. In this context, the approach proposed by Hunter (Hunter 2001; 2002a; 2002b) to represent semi-structured text through logical formulas is particularly relevant for enhancing the capabilities of the framework outlined in this paper. We think that in future developments this process could be complemented by additional techniques, such as defeasible rule discovery (Governatori & Stranieri 2001) and specialized argument assistance tools (Verheij 2003).

We contend that the evolution of recommender systems will lead to more efficient search environments, where both quantitative and qualitative analysis will play important roles. In this context, defeasible argumentation is a powerful tool that can help fulfill this long-term goal.

Acknowledgments

This research was partially supported by Projects CICYT TIC2001-1577-C03-03, TIC2003-00950 and Ramón y Cajal Program funded by the Ministerio de Ciencia y Tecnología (Spain). The authors would like to thank anonymous reviewers for providing helpful comments to improve the final version of this paper.

References

Amgoud, L.; Maudet, N.; and Parsons, S. 2002. An argumentation-based semantics for agent communication languages. In *Proc. of the 15th. European Conference in Artificial (ECAI), Lyon, France*.

Armstrong, R.; Freitag, D.; Joachims, T.; and Mitchell, T. 1995. Webwatcher: A learning apprentice for the world wide web. In *AAAI Spring Symp. on Information Gathering*, 6–12.

Ashish, N., and Knoblock, C. A. 1997. Semi-automatic wrapper generation for internet information sources. In *Conference on Cooperative Information Systems*, 160–169.

Balabanović, M., and Shoham, Y. 1997. Fab: content-based, collaborative recommendation. *Communications of the ACM* 40(3):66–72.

Balabanovic, M.; Shoham, Y.; and Yun, Y. 1995. An adaptive agent for automated web browsing. *Journal of Visual Communication and Image Representation* 6(4).

Berners-Lee, T.; Hendler, J.; and Lassila, O. 2001. The semantic web. *Scientific American*.

Berners-Lee, T. 1998. Semantic web road map. Technical report, W3C Design Issues.

Billsus, D., and Pazzani, M. J. 1999. A hybrid user model for news classification. In *In Kay J. (ed.), UM99 User Modeling - Proceedings of the Seventh International Conference*, 99–108. Springer-Verlag.

Budzik, J., and Hammond, K. J. 2000. User interactions with everyday applications as context for just-in-time information access. In *Proceedings of the 2000 International Conference on Intelligent User Interfaces*. New Orleans, Louisiana: ACM Press.

Budzik, J.; Hammond, K. J.; and Birnbaum, L. 2001. Information access in context. *Knowledge based systems* 14(1–2):37–53.

Carbogim, D.; Robertson, D.; and Lee, J. 2000. Argument-based applications to knowledge engineering. *The Knowledge Engineering Review* 15(2):119–149.

Chesñevar, C., and Maguitman, A. 2004. ARGUENET: An Argument-Based Recommender System for Solving Web Search Queries. In *Proc. of Intl. IEEE Conference on Intelligent Systems (IS-2004). Varna, Bulgaria. (to appear)*.

Chesñevar, C. I.; Dix, J.; Stolzenburg, F.; and Simari, G. R. 2003. Relating Defeasible and Normal Logic Programming through Transformation Properties. *Theoretical Computer Science* 290(1):499–529.

Chesñevar, C.; Maguitman, A.; and Loui, R. 2000. Logical Models of Argument. *ACM Computing Surveys* 32(4):337–383.

Claypool, M.; Gokhale, A.; Miranda, T.; Murnikov, P.; Netes, D.; and Sartin, M. 1999. Combining content-based and collaborative filters in an online newspaper. In *Proceedings of ACM SIGIR Workshop on Recommender Systems*.

Deshpande, M., and Karypis, G. 2004. Item-based top-n recommendation algorithms. *ACM Trans. Inf. Syst.* 22(1):143–177.

Doorenbos, R. B.; Etzioni, O.; and Weld, D. 1997. A scalable comparison-shopping agent for the world-wide web.

In Johnson, W. L., and Hayes-Roth, B., eds., *Procs. of the First Intl. Conf. on Autonomous Agents (Agents'97)*, 39–48. Marina del Rey, CA, USA: ACM Press.

Dung, P. M. 1993. On the Acceptability of Arguments and its Fundamental Role in Nomonotonic Reasoning and Logic Programming. In *Proc. of the 13th. International Joint Conference in Artificial Intelligence (IJCAI), Chambéry, Francia*, 321–357.

García, A., and Simari, G. 2004. Defeasible Logic Programming: An Argumentative Approach. *Theory and Practice of Logic Programming* 4(1):95–138. Preliminary version available from `cs.uns.edu.ar/~grs`.

Gelfond, M., and Lifschitz, V. 1990. Logic programs with classical negation. In *Proceedings of the 7th International Conference on Logic Programming. Jerusalem*.

Goldberg, D.; Nichols, D.; Oki, B. M.; and Terry, D. 1992. Using collaborative filtering to weave an information tapestry. *Communications of the ACM* 35(12):61–70.

Gomez, S., and Chesñevar, C. 2004. A Hybrid Approach to Pattern Classification Using Neural Networks and Defeasible Argumentation. In *Procs. of the Intl. FLAIRS Conference (to appear), Florida, USA*. AAAI.

Gordon, T., and Karacapilidis, N. 1997. The Zeno Argumentation framework. In *Proc. of the Int. Conf. on Artificial Intelligence and Law (ICAIL-97), Melbourne, Australia*, 10–18.

Governatori, G., and Stranieri, A. 2001. Towards the application of association rules for defeasible rules discovery. In et. al, B. V., ed., *Legal Knowledge and Information Systems*, 63–75. Amsterdam: JURIX.

Hunter, A. 2001. Hybrid argumentation systems for structured news reports. *Knowledge Engineering Review* (16):295–329.

Hunter, A. 2002a. Logical fusion rules for merging structured news reports. *Data and Knowledge Engineering* 42:23–56.

Hunter, A. 2002b. Merging structured text using temporal knowledge. *Data and Knowledge Engineering* 41:29–66.

Kakas, A. C., and Toni, F. 1999. Computing argumentation in logic programming. *Journal of Logic and Computation* 9(4):515–562.

Konstan, J. A. 2004. Introduction to recommender systems: Algorithms and evaluation. *ACM Trans. Inf. Syst.* 22(1):1–4.

Kowalski, R., and Toni, F. 1996. Abstract Argumentation. *Artificial Intelligence and Law* 4(3-4):275–296.

Kushmerick, N.; Weld, D. S.; and Doorenbos, R. B. 1997. Wrapper induction for information extraction. In *IJCAI'97*, 729–737.

Leake, D. B.; Bauer, T.; Maguitman, A.; and Wilson, D. C. 2000. Capture, storage and reuse of lessons about information resources: Supporting task-based information search. In *Proceedings of the AAAI-00 Workshop on Intelligent Lessons Learned Systems. Austin, Texas*, 33–37. AAAI Press.

Lieberman, H. 1995. Letizia: An agent that assists web browsing. In *IJCAI'95*, 924–929. Morgan Kaufmann.

Linton, F.; Joy, D.; and Schaefer, H.-P. 1999. Building user and expert models by long-term observation of application usage. In *Proceedings of the seventh international conference on User modeling*, 129–138. Springer-Verlag New York, Inc.

Loui, R.; Norman, P.; Altepeter, J.; Pinkard, D.; Craven, D.; Lindsay, J.; and Foltz, M. 1997. Progress on Room 5: A testbed for public interactive semi-formal legal argumentation. In *Proc. of the Int. Conf. on Artificial Intelligence and Law (ICAIL-97), Melbourne, Australia*, 207–214.

Maes, P. 1994. Agents that reduce work and information overload. *Communications of the ACM* 37(7):30–40.

Maudet, N., and Moore, D. J. 1999. Dialogue games for computer supported collaborative argumentation. In *Proceedings of the 1st Workshop on Computer supported collaborative argumentation (CSCA99)*.

Mladenic, D. 1996. Personal webwatcher: Design and implementation. Technical report ijs-dp-7472, School of Computer Science, Carnegie-Mellon University, Pittsburgh, USA.

Parsons, S., and McBurney, P. Argumentation-based Dialogues for Agent Coordination. *Group Decision and Negotiation (to appear)*.

Parsons, S.; Sierrra, C.; and Jennings, N. 1998. Agents that Reason and Negotiate by Arguing. *Journal of Logic and Computation* 8:261–292.

Pazzani, M. J.; Muramatsu, J.; and Billsus, D. 1996. Syskill & webert: Identifying interesting web sites. In *AAAI/IAAI'96*, 54–61.

Prakken, H., and Sartor, G. 1997. Argument-based extended logic programming with defeasible priorities. *Journal of Applied Non-classical Logics* 7:25–75.

Prakken, H., and Sartor, G. 2002. The role of logic in computational models of legal argument - a critical survey. In Kakas, A., and Sadri, F., eds., *Computational Logic: Logic Programming and Beyond*. Springer. 342–380.

Prakken, H., and Vreeswijk, G. 2002. Logical Systems for Defeasible Argumentation. In Gabbay, D., and F.Guenther., eds., *Handbook of Phil. Logic*. Kluwer. 219–318.

Reed, C., and Walton, D. 2001. Applications of argumentation schemes. In *Conference of the Ontario Society for the Study of Argument (OSSA2001), Windsor, Canada*.

Resnick, P., and Varian, H. R. 1997. Recommender systems. *CACM* 40(3):56–58.

Rhodes, B., and Maes, P. 2000. Just-in-time information retrieval agents. *IBM Systems Journal special issue on the MIT Media Laboratory* 39(3-4):685–704.

Rhodes, B. J. 2000. *Just-In-Time Information Retrieval*. Ph.D. Dissertation, MIT Media Lab.

Simari, G., and Loui, R. 1992. A Mathematical Treatment of Defeasible Reasoning and its Implementation. *Artificial Intelligence* 53:125–157.

Stolzenburg, F.; García, A.; Chesñevar, C.; and Simari, G. 2003. Computing Generalized Specificity. *Journal of Non-Classical Logics* 13(1):87–113.

Verheij, B. 2003. Artificial argument assistants for defeasible argumentation. *Artificial Intelligence Journal* 150:291–324.