ArgueNet: An Argument-Based Recommender System for Solving Web Search Queries

Carlos Iván Chesñevar, Ana Gabriela Maguitman

Abstract—**In the last years several specialized techniques for improving web search have been developed. Most existing approaches are still limited, mainly due to the absence of qualitative criteria for ranking results and insensitivity to user preferences for guiding the search. At the same time, defeasible argumentation 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 ArgueNet, a recommender system that classifies search results according to preference criteria declaratively specified by the user. The proposed approach integrates a traditional web search engine with a defeasible argumentation framework.**

Index Terms—**Recommender systems, Defeasible Argumentation, Decision support systems**

I. Introduction

Despite the many benefits that Internet is bringing to its users, the huge amount of data reachable by querying a conventional search engine is rapidly becoming overwhelming. In the face of this issue, there have been several proposals to prioritize search results in an efficient and reliable way. The success of search engines like Google is due not only to the large volume of web pages indexed, but also to the quality of the search results returned.

Several specialized techniques for improving web search have been developed, ranging from the use of powerful ranking algorithms [1], [2] to the so-called *special syntaxes* [3] that can be used to search specific parts of web pages (e.g., title, text body, anchor text) or specific types of information (e.g., file type, date range, phone numbers.) To a certain degree, the combination of ranking methods and special syntaxes empowers users to successfully direct their searches to the information they want to see.

Although the effectiveness and value of the current web search engines is remarkable, the existing approaches are still limited due to a number of barriers:

- **Absence of qualitative criteria for solving search queries:** search engines do not apply qualitative criteria for guiding meaningful searches and ranking results. They rely instead on a variety of syntactic criteria for pruning the search space (e.g., by excluding certain web domains) and on quantitative measures for ranking search results (e.g., by counting occurrences of matching keywords or by assessing sites popularity.)
- **Insensitivity to user preference criteria:** search engines perform searches independently of the user's preferences. Only the terms that explicitly appear in a query are used to describe the user's information needs. In addition, information sources that the user considers reliable cannot be prioritized over those considered unreliable.
- **Obscure query syntaxes:** special syntaxes are powerful but follow hard-to-memorize rules that are difficult to master by the ordinary user: certain syntaxes cannot be mixed, whereas others may result in too narrow queries, or even cancel each other.

For an increasing number of situations, the key to success is access to high-quality relevant information guided by a simple specification of the information needs and some preference criteria, without excessive distraction. Consider, for example, the case of a journalist investigating certain events and searching for relevant online published information. As the journalist browses the results returned by a conventional search engine, she will apply some preference criteria to manually select the most valuable results (e.g., those articles published during a specific date range will be preferred over others.) Much of the process of selecting such material according to some preference criteria could be effectively automatized. However, a full-spectrum analysis such as the one described above is beyond the power of traditional web search engines like Google or Altavista.

Recommender systems [4] are aimed at helping users to deal with the problem of information overload by facilitating access to relevant items. These systems attempt to generate a model of the user or user's task and apply diverse heuristics to anticipate what information may be of interest to the user. Recommender systems can be collaborative, which build on similarities between users with respect to the objects they interact with, or content-based, which build on similarities between potential recommendations and the objects that the user liked in the past. However, current approaches do not perform qualitative inference on the potential recommendations and are incapable of dealing with the defeasible nature of users' preferences.

In this paper we present ArgueNet, a Web recommender system that addresses the above-described problems by integrating a traditional web search engine with a defeasible argumentation framework. ArgueNet evaluates and ranks search results based on the user's declared preference criteria. The proposed system abstracts the user away from the obscure special syntax necessary to construct queries that reflect his or her preferences. As an alternative, user preferences are captured as a set of rules and facts, which can be made explicit in a more intuitive manner than by the use of query special commands. Such set of rules

Research partially supported by Projects CICYT TIC2001-1577-C03-03, TIC2003-00950 and Ramón y Cajal Program. Ministery of Science and Technology, Spain.

C. Chesñevar is with the Department of Informatics, University of Lleida, C/Jaume II, 69 – 25001 Lleida, Spain (email: cic@eup.udl.es)

A. Maguitman is with the Computer Science Department, Indiana University. Bloomington, IN 47405-7104, USA (email: anmaguit@cs.indiana.edu).

and facts will provide a knowledge base upon which a *qualitative* analysis of the results returned by a search engine will be performed.

The rest of the paper is structured as follows. In section II we present the fundamentals of our argumentation framework. Section III introduces ARGUENET, a framework that integrates traditional web search and defeasible argumentation. Next, in section IV, we present a worked example that illustrates how the proposed approach works. In Section V we briefly overview implementation issues and discuss related work. Finally, Section VI concludes.

II. Modeling Defeasible Argumentation in DeLP

Defeasible argumentation [5], [6] has evolved in the last decade as a successful approach to formalize defeasible, commonsense reasoning. Argument-based applications have been developed in many areas, such as agent theory, knowledge engineering and legal reasoning [7], [8] *Defeasible logic programming* (DeLP) [9] is a defeasible argumentation formalism based on logic programming. A defeasible logic program is a set $K = (\Pi, \Delta)$ of Horn-like clauses, where Π and Δ stand for sets of strict and defeasible knowledge, respectively. The set Π of strict knowledge 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 of defeasible knowledge involves *defeasible rules* of the form $p \prec q_1, \ldots, q_k$, which stands for "$q_1, \ldots q_k$ provide a *tentative reason* to believe p." The underlying logical language is that of extended logic programming, enriched with a special symbol "$\prec$" to denote defeasible rules. Both default and classical negation are allowed (denoted not and $\sim$, resp.). Syntactically, the symbol "$\prec$" is all that distinguishes a *defeasible* rule $p \prec q_1, \ldots q_k$ from a *strict* (non-defeasible) rule $p \leftarrow q_1, \ldots, q_k$. DeLP rules are thus Horn-like clauses to be thought of as *inference rules* rather than implications in the object language.

Deriving literals in DeLP results in the construction of *arguments*. An argument $\mathcal{A}$ is a (possibly empty) set of ground defeasible rules that together with the set Π provide a logical proof for a given literal h, satisfying the additional requirements of *non-contradiction* and *minimality.*

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 $\mathcal{P}$ and a (possibly empty) set of default ground literals "not L", 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 minimal with respect to set inclusion.

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 a kind of 'Occam's razor principle' [10] 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 two complementary 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.

Definition 2 (Counterargument – Defeat) An argument $\langle \mathcal{A}_1, q_1 \rangle$ is a *counterargument* for an argument $\langle \mathcal{A}_2, q_2 \rangle$ iff
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.
2. A literal not q_1 is present in some rule in $\mathcal{A}_1$.

A partial order $\preceq \subseteq Args(\mathcal{P}) \times Args(\mathcal{P})$ will be used as a *preference criterion* 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 $\langle \mathcal{A}_2, q_2 \rangle$ wrt $\preceq$. 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 $\langle \mathcal{A}_2, q_2 \rangle$ iff $\langle \mathcal{A}_1, q_1 \rangle$ is strictly preferred over $\langle \mathcal{A}, q \rangle$ wrt $\preceq$.
- In case 1, if $\langle \mathcal{A}_1, q_1 \rangle$ and $\langle \mathcal{A}, q \rangle$ are unrelated to each other, 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 [10] is used in DeLP as a syntax-based criterion among conflicting arguments, preferring arguments which are *more informed* or *more direct* [10], [11]. However, other alternative orders could also be used.

An *argumentation line* starting in an argument $\langle \mathcal{A}_0, Q_0 \rangle$ (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* (even-indexed arguments) and an *opponent* (odd-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, dialectics imposes additional constraints on such an argument exchange to be considered rationally acceptable in a program $\mathcal{P}$.

- **Non-contradiction:** given an argumentation line λ, the set of arguments of the proponent (resp. opponent) should be *non-contradictory* wrt $\mathcal{P}$. Non-contradiction for a set of arguments is defined as follows: a set $S = \bigcup_{i=1}^{n} \{\langle \mathcal{A}_i, Q_i \rangle\}$ is *contradictory* wrt $\mathcal{P}$ iff $\Pi \cup \bigcup_{i=1}^{n} \mathcal{A}_i$ is contradictory.
- **No circular argumentation:** no argument $\langle \mathcal{A}_j, Q_j \rangle$ in λ is a sub-argument of an argument $\langle \mathcal{A}_i, Q_i \rangle$ in λ, $i < j$.
- **Progressive argumentation:** every blocking defeater $\langle \mathcal{A}_i, Q_i \rangle$ in λ is defeated by a proper defeater $\langle \mathcal{A}_{i+1}, Q_{i+1} \rangle$ in λ.

The first condition disallows the use of contradictory information on either side (proponent or opponent). The second condition eliminates the "circular reasoning" fallacy. The last condition enforces the use of a stronger argument to defeat an argument which acts as a blocking defeater. An argumentation line satisfying the above restrictions is called *acceptable*, and can be proven to be finite [9].

Given a DeLP 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 $\langle \mathcal{A}_0, Q_0 \rangle$ (ie., all possible dialogues rooted in $\langle \mathcal{A}_0, Q_0 \rangle$), formalized as a *dialectical tree.*

Definition 3 (Dialectical Tree) Let $\mathcal{P}$ be a DeLP program, and let $\langle \mathcal{A}_0, Q_0 \rangle$ be an argument in $\mathcal{P}$. A *dialectical tree* for $\langle \mathcal{A}_0, Q_0 \rangle$, denoted $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$, is a tree structure defined as follows:

1. The root node of $\mathcal{T}_{\langle \mathcal{A}_0, Q_0 \rangle}$ is $\langle \mathcal{A}_0, Q_0 \rangle$.
2. $\langle \mathcal{B}', H' \rangle$ is an immediate children of $\langle \mathcal{B}, H \rangle$ iff there exists an acceptable argumentation line $\lambda^{\langle \mathcal{A}_0, Q_0 \rangle}$ $= [\langle \mathcal{A}_0, Q_0 \rangle, \langle \mathcal{A}_1, Q_1 \rangle, \ldots, \langle \mathcal{A}_n, Q_n \rangle$] such that there are two elements $\langle \mathcal{A}_{i+1}, Q_{i+1} \rangle = \langle \mathcal{B}', H' \rangle$ and $\langle \mathcal{A}_i, Q_i \rangle$ $= \langle \mathcal{B}, H \rangle$, for some $i = 0 \ldots n-1$.

□

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.). A dialectical tree will be marked as an AND-OR tree: 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*) wrt 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*.

Given a DeLP program $\mathcal{P}$, solving a query q wrt $\mathcal{P}$ accounts for determining whether q is supported by a warranted argument. Different doxastic attitudes are distinguished when answering that query q according to the associated status of warrant, in particular:

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

III. The ArgueNet Framework: Fundamentals

A fundamental problem addressed by web search engines is how to determine which web documents are *relevant* to a query q When providing a list of search results $[s_1,\ s_2,\ \ldots s_k]$ in response to a query q, it is common to assume 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. This is specially problematic when thousands of results are available, so that a detailed analysis of the whole search space becomes extremely expensive.

Experienced users of search engines rely on the combination of different (mostly implicit) preference criteria to build and evaluate alternative *hypotheses* for filtering search results. In this context, meta-information associated with search results turns out to be particularly helpful, as search results are mostly links to HTML pages which have a number of associated features (e.g. filename, timestamp or date in which the document was created, URL, etc.). In particular, the recent evolution of the Semantic Web has favored the incorporation of additional features to semantically characterize the content of web documents.

Consider for example a journalist who wants to search for news articles about the Iraq war. A query containing the terms *news*, *iraq*, and *war* will return thousands of search results. Our journalist may have some implicit, *tentative* knowledge that she could use to guide the search, such as:

- She considers most news appeared in American and Iraqi newspapers as too biased with respect to the Iraq war.
- She thinks that the American newspaper "The New York Times" (NYT) is usually not biased and trustworthy with respect to the Iraq war.
- She considers trustworthy every journalist who never faked a report. However, she knows that John Doe, who works for the NYT, has faked news reports about the Iraq war.

The above preference criteria will help our journalist to classify some search results as potentially irrelevant (e.g. by skipping certain links associated with URLs corresponding to American and Iraqi newspapers) whereas some others would be deemed as particularly interesting (e.g. those links corresponding to the domain `nyt.com`). Note that preference criteria provide incomplete knowledge about the search domain. Since user preference criteria can be inconsistent, such kind of knowledge cannot be modeled through traditional rule-based approaches.

Our proposal is to model the user's preference criteria in terms of a DeLP program $\mathcal{P}$. A distinguished predicate name *rel* will be used for analyzing the *relevance* of every search result s_i with respect to the user's preference criteria. The existence of a warranted argument $\langle A, rel(s_i) \rangle$ built on the basis of the DeLP program $\mathcal{P}$ will allow to conclude that s_i is a search result relevant to the user's query. As stated before, for a given query q a typical search engine will return a list of (probably thousands of) search results $S = [s_1,\ s_2,\ \ldots s_k]$, and every search result s_i will be characterized by 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 (see discussion in Section V). Such features will be encoded as DeLP facts, extending the original program $\mathcal{P}$ into a new program $\mathcal{P}$'. A special operator **Revise** deals with possible inconsistencies found in S wrt $\mathcal{P}$, ensuring $\mathcal{P}$' is not contradictory.[1] Every search result $s_i \in S$ will be then automatically analyzed in the context of $\mathcal{P}$' by solving the query $?-rel(s_i)$ using the DeLP interpreter. We will classify the elements in the original list L of search results in three sets, namely:

- S^w (warranted search results): those search results s_i for which there exists at least one warranted argument supporting $rel(s_i)$ based on $\mathcal{P}$'.

[1]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.

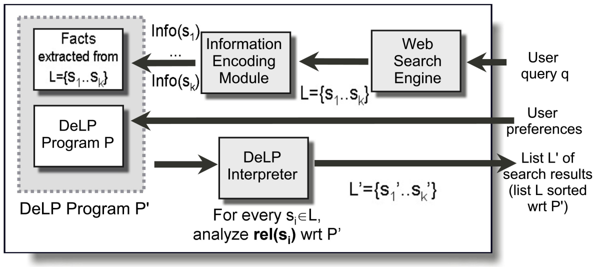

Fig. 1. The ArgueNet Framework: Outline

ALGORITHM SolveBrowserQuery
INPUT: Query q, DeLP program $\mathcal{P}$ modeling user preferences
OUTPUT: List L_{new} $\{$*search results sorted according to* $\mathcal{P}\}$
BEGIN
Let $L = [s_1, s_2, \dots s_k]$ be the output of solving query q using a web search engine.
$\{L$ *is the list of (the first k) results obtained from query* q $\}$
$\mathcal{P}_{search} = \{$facts encoding $info(s_1)$, $info(s_2) \dots info(s_k)\}$.
$\mathcal{P}$' := **Revise** $(\mathcal{P} \cup \mathcal{P}_{search})$.
Initialize S^w, S^u, and S^d as empty sets.
$\{S^w$*,* S^u*, and* S^d *stand for the set of warranted as relevant, undecided and warranted as non-relevant results, resp.*$\}$
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 $L_{new} = [s_1^w, s_2^w, \dots, s_i^w, s_1^u, s_2^u, \dots, s_j^u, s_1^d, s_1^d, s_k^d]$
END

Fig. 2. High-level algorithm for solving queries in ArgueNet

• S^u (undecided search results): those results s_i for which there is no warranted argument for $rel(s_i)$ but there is not warranted argument for $\sim rel(s_i)$ either on the basis of $\mathcal{P}$', and

• S^d (defeated search results): those results s_i such that there is a warranted argument supporting $\sim rel(s_i)$ on the basis of $\mathcal{P}$'.

Figure 1 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 be a sorted list L' in which the elements of L are ordered according to their epistemic status wrt $\mathcal{P}$' (e.g. first all search results warranted to be relevant, then all search results which are undecided wrt their relevance and finally all those search results which are warranted to be non-relevant according to the user's preferences.) 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 [9]: if there exists a warranted argument $\langle A, h \rangle$ on the basis of a DeLP program $\mathcal{P}$, then there is no warranted argument $\langle B, \sim h \rangle$ based on $\mathcal{P}$.

IV. A Worked Example

Consider again a journalist who is searching for news reports concerning the Iraq war. She has some preference criteria which could help her guide the search, namely: 1) she always considers relevant the newspaper reports written by Bob Doll; 2) she usually considers

$$
\begin{array}{rcl}
rel(X) & \prec & author(X,A), trust(A). \\
\sim rel(X) & \prec & author(X,A), trust(A), \\
 & & outdated(X). \\
trust(A) & \prec & \text{not } faked\_news(A). \\
\sim rel(X) & \prec & address(X, Url), biased(Url). \\
biased(Url) & \prec & iraqi(Url). \\
biased(Url) & \prec & american(Url). \\
\sim biased(Url) & \prec & domain(Url, D),\ D = \text{“}nyt.com\text{”}. \\
rel(X) & \leftarrow & author(X, bob\_doll). \\
oudated(X) & \leftarrow & date(X, D), curr\_date(Today), \\
 & & (Today - D) > 100. \\
iraqi(X) & \leftarrow & \text{[Computed elsewhere]} \\
american(X) & \leftarrow & \text{[Computed elsewhere]} \\
domain(Url, D) & \leftarrow & \text{[Computed elsewhere]} \\
curr\_date(T) & \leftarrow & \text{[Computed elsewhere]} \\
faked\_news(john\_doe) & \leftarrow &
\end{array}
$$

Fig. 3. A DeLP program modeling the preferences of a journalist

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) Iraqi and American viewpoints on the war are usually considered biased; 6) The New York Times is an American newspaper which she usually considers non biased; 7) John Doe is known to have faked a report.

The above tentative rules and facts can be modeled in terms of a DeLP program $\mathcal{P}$ shown in Figure 3. Note that some rules in $\mathcal{P}$ rely on "built in" predicates computed elsewhere and not provided by the user (e.g., determining the country of origin corresponding to a specific web domain can be found querying Internet directory services such as Whois).

Suppose that the query containing the terms *news*, *iraq*, and *war* is presented to a traditional search engine, which returns a list of search results L=$\{$ s_1, s_2, s_3, s_4 $\}$. Most of these results will be associated with news articles and will contain a number of features (e.g. author, date, URL, etc.). Such features can be encoded as a collection of DeLP facts as follows:

$author(s_1, john\_doe)$.
$address(s_1, \text{“}http://www.nyt.com/...\text{”})$.
$date(s_1, 20031003)$.

$author(s_2, jen\_oldie)$.
$address(s_2, \text{“}http://www.britishnews.co.uk/...\text{”})$.
$date(s_1, 20001003)$.

$author(s_3, jane\_truth)$.
$address(s_3, \text{“}http://www.nyt.com/...\text{”})$.
$date(s_3, 20031003)$.

$author(s_4, bob\_doll)$.
$address(s_4, \text{“}http://www.mynewspaper.com/...\text{”})$.
$date(s_4, 20031003)$.

We can now analyze s_1, s_2, s_3 and s_4 in the context of the user's preference theory about the search domain by considering the DeLP program $\mathcal{P}'$=$\mathcal{P}$$\cup$$Facts$, where $Facts$ denotes the set corresponding to the collection of facts given above. For each s_i, the query $rel(s_i)$ will be analyzed wrt 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 John Doe who is considered a trustworthy author (note that every journalist is considered to be trustworthy by default.) Here we have [2] $\mathcal{A}_1$={ $rel(s_1) \prec author(c_1, john\_doe)$, $trust(john\_doe)$; $trust(john\_doe) \prec$ not $faked\_news(john\_doe)$ }. The DeLP inference engine will then search for defeaters for $\langle \mathcal{A}_1, rel(s_1)\rangle$. A defeater $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$ will be found: s_1 is not relevant as it comes from an American newspaper, which is by default assumed to be biased about Iraq war. Here we have $\mathcal{A}_2$={ $\sim rel(c_1) \prec address(c_1,$ "nyt.com..."$)$, *biased*("nyt.com...") ; *biased*("nyt.com...") $\prec$ *american*("nyt.com...") }. However, there exists in its turn another defeater $\langle \mathcal{A}_3, \sim$ *biased*("nyt.com...")$\rangle$ for $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$, reinstating the first argument $\langle \mathcal{A}_1, rel(s_1)\rangle$: Usually articles from the NYT are not biased. Here we have $\mathcal{A}_3$={ $\sim$ *biased*("nyt.com...") $\prec$ *domain*("nyt.com...", "nyt.com"), ("nyt.com" = "nyt.com") }. Note that the definition of dialectical tree (Def. 3) does not allow the use of $\langle \mathcal{A}_1, rel(s_1)\rangle$ to defeat $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$, as this would imply falling into fallacious, circular argumentation. Note however that $\langle \mathcal{A}_1, rel(s_1)\rangle$ has another defeater besides $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$, namely $\langle \mathcal{A}_4, faked\_news(john\_doe)\rangle$, with $\mathcal{A}_4 = \emptyset$. No other arguments need to be considered. The resulting dialectical tree rooted $\langle \mathcal{A}_1, rel(s_1)\rangle$ as well as its corresponding marking is shown in Figure 4a (left). The root node is marked as D-node (defeated), which implies that the argument $\langle \mathcal{A}_1, rel(s_1)\rangle$ is not warranted. Carrying out a similar analysis for $\sim rel(s_1)$ results in the dialectical tree shown in Figure 4a (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*.

Similarly we can analyze the case of s_2. An argument $\langle \mathcal{B}_1, rel(s_2)\rangle$ can be built supporting the conclusion $rel(s_2)$, with $\mathcal{B}_1$={ $rel(s_2) \prec author(s_2,)$, $trust(jen\_oldie)$; $trust(jen\_oldie) \prec$ not $faked\_news$ (jen_oldie) }. This argument has a proper defeater[3] $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$ which defeats $\langle \mathcal{B}_1, rel(s_2)\rangle$, with $\mathcal{B}_2$={$\sim rel(s_2) \prec author(s_2,)$, $trust(jen\_oldie)$, $outdated(s_2)$; $trust(jen\_oldie) \prec$ not $faked\_news(jen\_oldie)$}. 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. 4b (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*.

Following the same line of reasoning used in the case of s_1 we can analyze the case of s_3. An argument $\langle \mathcal{C}_1, rel(s_3)\rangle$ 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 autor). A defeater $\langle \mathcal{C}_2, \sim rel(s_3)\rangle$ will be found: s_1 is not relevant as it comes from an American newspaper, which by default is assumed to be biased about Iraq war. But this defeater in its turn is defeated by a third argument $\langle \mathcal{C}_3$, *biased*$(s_3)\rangle$. The resulting dialectical tree for $\langle \mathcal{C}_1, rel(s_3)\rangle$ is shown in Fig. 4c (left)). The original argument $\langle \mathcal{C}_1, rel(s_3)\rangle$ can be thus deemed as *warranted*.

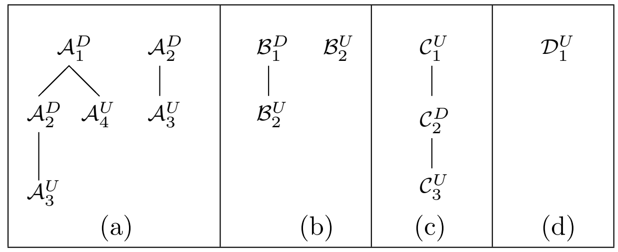

Fig. 4. Dialectical trees associated with (a) $\langle \mathcal{A}_1, rel(s_1)\rangle$ and $\langle \mathcal{A}_2, \sim rel(s_1)\rangle$; (b) $\langle \mathcal{B}_1, rel(s_2)\rangle$ and $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$; (c) $\langle \mathcal{C}_1, rel(s_3)\rangle$ and (d) $\langle \mathcal{D}_1, rel(s_4)\rangle$

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 (there is no defeasible knowledge involved). Hence $rel(s_4)$ is *warranted*. The associated dialectical tree is shown in Fig. 4d.

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).

V. Implementation Issues and Related Work

Performing defeasible argumentation is a computationally complex task. 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. An interpreter of DeLP was also implemented in Prolog. Several features leading to the efficient implementation of DeLP have also been recently studied, mainly those related to comparing conflicting arguments by specificity [11] and to pruning the search space [12]. In particular, the search space associated with dialectical trees is reduced by applying $\alpha - \beta$ pruning. Thus, in Fig. 4 (a), the left branch of the tree does not need to be computed if the right branch has been computed first (as in that case the root node can be already deemed as ultimately defeated).

ArgueNet operation relies on the user declaring his or her preference criteria, which the system codifies as facts and rules. This process could be complemented by the application of techniques for defeasible rule discovery as described in [13]. Another important issue is the need to extract relevant features from the search results and to codify them as DeLP facts. Web documents are usually represented using HTML, a document markup language that uses predefined tags for presentation purposes and not to convey semantics. In spite of that, HTML tags can be usefully exploited to extract meaningful content [14], [15], [16]. The emergence of XML as a standard for data representation on the Web contributes to further simplify the extraction of facts from web pages.

Work on query languages for semistructured data (e.g., [17], [18], [19]) is mostly based on the metaphor of

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

[3]In terms of specificity, $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$ is based on more information (is more specific) than $\langle \mathcal{B}_1, rel(s_2)\rangle$. Thus $\langle \mathcal{B}_2, \sim rel(s_2)\rangle$ is preferred over $\langle \mathcal{B}_1, rel(s_2)\rangle$.

the Web as a database. Some of these approaches provide rich syntax and semantics that allow for expressing powerful queries and to reuse user's partial knowledge but do not attempt to perform any kind of qualitative inference to support the returned answers.

Our system operates on top of a conventional search engine, providing a powerful abstraction for solving queries based on a user's preference criteria. In that sense, our proposal shares motivations with the Internet agents called SoftBots [20], which, upon a user's request, use planning technology to select Web services by taking into consideration a person's declared interest. Many personalized Web recommender systems that operate on top of Internet services have been proposed over the past years (e.g., [21], [22], [23]). Existing Web recommender tools take into account the user's interests (either declared by the user or conjectured by the system) 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.

More ambitious projects to facilitate automatic qualitative reasoning on the Web rely on the realization of the Semantic Web vision [24]. The content of the Semantic Web is expected to be meaningful and tractable by autonomous systems, which will facilitate the implementation of qualitative reasoning tools. However, the concretization of such a vision is still underway. A recent discussion on issues and perspectives of adding deduction capability to search engines through the use of fuzzy logic is presented in [25]. As discussed in that presentation, a web question-answering system with deductive capabilities is still far from becoming a reality.

VI. Conclusions

Search engine technology has evolved rapidly in the last years, leading to very efficient and reliable algorithms. Nevertheless, current approaches still have serious limitations due to the absence of qualitative criteria for solving search queries.

In this paper we have presented an integrated framework for web search results recommendation based on defeasible argumentation. The proposed framework provides a novel way of enhancing web searching through the use of qualitative analysis to prioritize search results. The proposed system preserves the simplicity of traditional web search engines for posing queries, while abstracting the user away from special syntaxes to reflect his or her preferences.

We contend that the evolution of recommender systems will result in efficient and reliable web search environments, where both quantitative and qualitative analysis will play important roles. We believe our proposal is a realistic and do-able approach to help fulfill this long-term goal.

References

[1] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer Networks and ISDN Systems*, vol. 30, no. 1–7, pp. 107–117, 1998.

[2] J. Kleinberg, "Authoritative sources in a hyperlinked environment," *Journal of the ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[3] T. Calishain and R. Dornfest, *Google Hacks. 100 Industrial-Strength Tips and Tools*, O'Reilly, 2003.

[4] P. Resnick and H. R. Varian, "Recommender systems," *CACM*, vol. 40, no. 3, pp. 56–58, 1997.

[5] C. Chesñevar, A. Maguitman, and R. Loui, "Logical Models of Argument," *ACM Computing Surveys*, vol. 32, no. 4, pp. 337–383, Dec. 2000.

[6] H. Prakken and G. Vreeswijk, "Logical Systems for Defeasible Argumentation," in *Handbook of Phil. Logic*, D. Gabbay and F.Guenther, Eds., pp. 219–318. Kluwer, 2002.

[7] D. Carbogim, D. Robertson, and J. Lee, "Argument-based applications to knowledge engineering," *The Knowledge Engineering Review*, vol. 15, no. 2, pp. 119–149, 2000.

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

[9] A. García and G. Simari, "Defeasible Logic Programming: An Argumentative Approach," *Theory and Practice of Logic Programming*, vol. 4, no. 1, pp. 95–138, 2004,

[10] G. Simari and R. Loui, "A Mathematical Treatment of Defeasible Reasoning and its Implementation," *Artificial Intelligence*, vol. 53, pp. 125–157, 1992.

[11] F. Stolzenburg, A. García, C. Chesñevar, and G. Simari, "Computing Generalized Specificity," *Journal of Non-Classical Logics*, vol. 13, no. 1, pp. 87–113, 2003.

[12] C. Chesñevar, G. Simari, and A. García, "Pruning Search Space in Defeasible Argumentation," in *Proc. of the Workshop on Advances and Trends in AI*. XX Intl. Conf. of the SCCC, Santiago, Chile, Nov. 2000, pp. 46–55.

[13] G. Governatori and A. Stranieri, "Towards the application of association rules for defeasible rules discovery," in *Legal Knowledge and Information Systems*, B. Verheij et. al, Ed., Amsterdam, 2001, JURIX, pp. 63–75, IOS Press.

[14] R. B. Doorenbos, O. Etzioni, and D. Weld, "A scalable comparison-shopping agent for the world-wide web," in *Procs. of the First Intl. Conf. on Autonomous Agents (Agents'97)*, W. Lewis Johnson and Barbara Hayes-Roth, Eds., Marina del Rey, CA, USA, 1997, pp. 39–48, ACM Press.

[15] N. Ashish and C. A. Knoblock, "Semi-automatic wrapper generation for internet information sources," in *Conference on Cooperative Information Systems*, 1997, pp. 160–169.

[16] N. Kushmerick, D. S. Weld, and R. B. Doorenbos, "Wrapper induction for information extraction," in *IJCAI'97*, 1997, pp. 729–737.

[17] L. V. S. Lakshmanan, F. Sadri, and I. N. Subramanian, "A declarative language for querying and restructuring the WEB," in *International Workshop on Research Issues in Data Engineering, RIDE '96*, 1996, pp. 12–21.

[18] A. O. Mendelzon, G. A. Mihaila, and T. Milo, "Querying the world wide web," *Intl. Journal on Digital Libraries*, vol. 1, no. 1, pp. 54–67, 1997.

[19] S. Abiteboul, D. Quass, J. McHugh, J. Widom, and J. L. Wiener, "The Lorel query language for semistructured data," *Intl. Journal on Digital Libraries*, vol. 1, no. 1, pp. 68–88, 1997.

[20] O. Etzioni, "Moving up the information food chain: Deploying softbots on the world wide web," in *AAAI/IAAI'96*, Menlo Park, 4–8 1996, pp. 1322–1326, AAAI Press / MIT Press.

[21] R. Armstrong, D. Freitag, T. Joachims, and T. Mitchell, "Webwatcher: A learning apprentice for the world wide web," in *AAAI Spring Symp. on Information Gathering*, 1995, pp. 6–12.

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

[23] H. Lieberman, "Letizia: An agent that assists web browsing," in *IJCAI'95*. 1995, pp. 924–929, Morgan Kaufmann.

[24] T. Berners-Lee, J. Hendler, and O. Lassila, "The semantic web," *Scientific American*, May 2001.

[25] L. A. Zadeh, "Invited talk: Web intelligence and fuzzy logic the concept of web iq," in *Intelligent Agent Technology - IAT 2003*. 2003, IEEE.