

ORIGINAL RESEARCH

Providing personalized recommendation for attending events based on individual interest profiles

Boris Galitsky*

eBay, Inc. San Jose CA, USA

Received: June 14, 2015

DOI: 10.5430/air.v5n1p1

Accepted: August 2, 2015

URL: <http://dx.doi.org/10.5430/air.v5n1p1>

Online Published: September 24, 2015, 2015

ABSTRACT

In this article we present a framework to extract user interests from social network profiles such as Facebook to personalize recommendations about products and services. Matching users' interests as keywords with product attributes as keywords, performed by currently available personalization systems, has a very low recall, so more general category-based framework is needed. It turns out that substantial reasoning about products and their categories is required to match a taxonomy of the owner of products and services, with that of a user, as expressed in a public profile. To handle inconsistencies between these taxonomies, a mapping of one into another is expressed as a Defeasible Logic program (DeLP), where a potential mapping can be defeated by other ones if relevant information becomes available. Events and things to do are recommended at StubHub.com and www.facebook.com/StubHub/ so that the reader can observe the system at a scale. Also, we present content management system which supports personalized recommendation is outlined.

Key Words: Personalization, Recommendation of event attendance, Mapping categories of interest

1. INTRODUCTION

Personalized recommendations are becoming more and more popular to enable people to efficiently get products and services. Internet entrepreneurs have started to believe that personalization is one of the next big steps towards a more semantic web. Everything we "like" on sites like Facebook gives others information about the things users are interested in. If one gets enough of that kind of data, as well as similar data from the people he is connected to, he can effectively judge a person's tastes and interests.

Social data-based personalization is an important feature of context-aware systems which can both sense and react based on their environments. Although a high number of successful user interfaces employ user behavior, inter-personal trust and attempt to recognize user intentions,^[1-3] a context-aware

methodology of adjustment to prior knowledge about users is still to be developed.

A number of systems^[4-6] have been developed proposed to personalize the internet by getting to know users and then making smart recommendations about what they might like. There are systems functioning in horizontal domain, as well as specific areas like music and shopping. In the former case, the objective is to build a graph of users' tastes of the whole web, connecting web users with their affinities for all sorts of their everyday activities, from electronics to vacations.

A number of personalization systems have been developed over last few months. The New York Times rolled out a page of "Articles Recommended for You". To visits a page, one needs to log in, view several articles and wait a day for recommendations to appear. Users are reporting that

*Correspondence: Boris Galitsky; Email: bgalitsky@hotmail.com; Address: eBay, Inc. San Jose CA, USA.

a number of the recommendations are on target with their interests, however a top pick can be “Museum and Gallery Listings” even though the interests displayed on the right are technology, business and sports related, with not a hint of arts.

The Washington Post let out some details of a new site called Trove, citing that “the holy grail of online news is a service that tailors the experience to each reader as effectively as sites like Amazon and Pandora do for books and music.” It also noted that “news is more difficult than other products to gear to individual preferences.” The project reportedly has a development team of 20 people and the company is investing \$5 million to \$10 million.

Yahoo! announced of “Livestand”, described as a publishing platform for mobile devices that will be offered to other publishers as well as present Yahoo! content. Like other efforts, Yahoo! also claims it will be personalized based on the kinds of content you consume, much as the Yahoo! home page is tailored to each user’s interests. While this might work well for Yahoo! content, it is doubtful if other publishers want their articles mixed into a personalized blend of news from different sources or if they prefer to keep their content un-aggregated.

Although users are in the process of starting to appreciate the value of personalization and learn to adjust their profiles for efficient online personalization, the relevance and timeliness of personalization is still quite low. In this study we address the root problem of personalization quality, propose a solution for Zvents.com of eBay, and evaluate it for a vertical recommendation domain.

Nowadays, when integration and access control with social sites like Facebook has been mostly solved, the main reason for low relevance is the existence of inconsistent mappings between the categories of interests as specified in social sites like Facebook and Hi5 and the categories of providers of products/services/content. In fact, there is strong disagreement between how the set of user interests are stored in social sites and how such interests are associated with categories of product in a vertical domain of a recommendation system. In particular, Facebook stores user interests at individual level (user *likes*) and category level (categories of user likes) for the wide range of interests. Since our recommendation engine is focused on “things to do”, most of the existing Facebook categories are irrelevant, but those which are relevant are too coarse and provide limited and uneven coverage of our domain of events. Hence we need a systematic way to map horizontal domain categories and individual “likes” into the product attributes and categories in a vertical domain. In this paper, we use Defeasible Logic Programming,^[7] an

argumentative framework based on logic programming to define such mapping where categories expressed in the same worlds frequently have different meanings and are therefore inconsistent.

The main purpose of personalized recommendation delivery in dynamic domain as attending events includes:

- A user specifies her interests only once (in her Facebook profile) but thoroughly so that personalized recommendation can be produced in a wide variety of domains, from things to do to consumer products.
- Selecting an event for a given date, a user does not have to manually run queries for all kinds of events she is interested in; instead, she logs in with her personal profile and sees what is happening according to her interests.
- Personalization is expected to impress customers with unique treatment of interests of themselves and their friends supporting such social features as trust.

2. PERSONALIZATION ALGORITHMS AS INTERSECTION OF LIKES

We can define vertical personalization as finding a set of recommendations which are the overlap of two sets:

- *InterestSet*: all possible subjects of user interests (all events) we believe a user would be interested in according to what she specified;
- *LocationDateSet*: all events available at a specified location at a specific time.

In this setting, we can define a new set *Recommendation* = $InterestSet \cap LocationDateSet$. Since *InterestSet* is specified as two sets of $\langle Likes, Categories \rangle$, as long as *LocationDateSet* can be tagged with the same tags and categories, the problem is solved. If overlap of *likes* is too small (unimportant), events with *categories of likes* will be formed as the desired intersection. Note that $\langle Likes, Categories \rangle$ is frequently redundant: *Likes* derive *Categories* unambiguously but not the other way around.

Without personalization, using a conventional search engine, a user would have to explicitly search for each of her $\langle Likes, Categories \rangle$, or form a respective OR query, to find this intersection. This is happening today when a user is searching the web for “things to do” this weekend. However, not all Facebook likes are equally meaningful. Some of the likes were specified because the user is confident in her interests, whereas another set of likes is encouraged by various Facebook apps and therefore not as indicative of real user interest, and might be too unimportant. We use the following mechanism to differentiate between these two classes of likes (important/unimportant):

- (1) Using friends: all *likes* shared by friends
- (2) Using dynamics of how likes appeared: initial set of likes are assumed to be important, clusters of likes encouraged by various Facebook apps are unimportant, likes of weakly represented categories are unimportant as well, whereas well-represented categories of likes are important.

Once we have $\langle Likes, Categories \rangle$ of *InterestSet*, we first try to find important likes, then unimportant likes, and finally different likes but from Categories in *LocationDateSet*.

The remaining problem is to map two set of categories for *Likes, CategoriesSrc* for source and *CategoriesDest* for destination. For this problem we will apply argumentation techniques for dealing with potentially inconsistent and contradictory information.

3. MAPPING CATEGORIES OF INTEREST / TAXONOMIES

What is available for Facebook likes for the domain of entertaining events are as follows: *CategorySrc* = {Retail, Consumer_products, Sports_athletics, Sports_teams, Athlete, Television, Comedian, Clubs, Food_beverage, Musicians, Health_beauty, Actor, Writer, Restaurants, Fashion, Comedian, Musician/band, Games, Musicians, Movie, Tv show, Television, Album, Actor/director, Film, Bars, Education, Nonprofit, Song}. As the reader can see these categories are overlapping, and belong to various level of generalization and granularity. These categories have to be mapped into types of events, venues such as restaurants and theaters, and particular music genres: *CategoryDest* = {Arts & Crafts, Community, Business & Tech, Dance, Fairs & Festivals, Food & Dining, Music, Performing Arts, Sports & Outdoors, Visual Arts} (higher-level categories) \cap {Fairs & Festivals / {sport festivals} excluding other kinds of festivals} \cap {sub-categories including Jazz, R&B and Soul, Rock, Techno & Dance, Country, Classical, Folk & Traditional ...} Mapping between categories can be described as *Sports_athletics* \rightarrow *Sports & Outdoors*/{soccer, hiking ...} excluding {camping, bird-watching} \cap *Dance/gymnastics* excluding other kinds of dance \cap *Fairs & Festivals*/{sport festivals} excluding other kinds of festivals}

As an essentially deterministic categorization, we would avoid applying statistical and fuzzy mapping here; instead we prefer a systematic way to handle inconsistencies between source and target categorizations. Deterministic mapping better fits current software development methodology, making this mapping fully controllable and therefore well-suited for commercial environments (compare with approaches to reasoning related to argumentation in Ref.^[8-10]).

The rules (clauses) for the target category above would look like: *sports_outdoors*:-*sports_athletics* OR (*outdoors, camping, bird-watching*) OR (*dance, gymnastics*) OR (*fairs_festivals & sport_festival*). We now proceed to the systematic treatment of inconsistencies among such rules using Defeasible Logic Programming (DeLP), an argumentative framework based on logic programming.^[7]

4. DEFEASIBLE LOGIC PROGRAM FOR CATEGORY MAPPING

Next we will present an overview of the main concepts associated with DeLP.^[7] Then we will show an example of how categories can be mapped in order to determine recommendation categories from social profile categories.

DeLP is a general argumentation based system for knowledge representation and reasoning. Its proof theory is based on a dialectical analysis where arguments for and against a literal interact in order to determine whether this literal is believed by a reasoning agent. Its operational semantics is based on a dialectical analysis where arguments for and against a literal interact in order to determine whether this literal is believed by a reasoning agent. DeLP is a newly developed extension to conventional logic programming that captures some aspects of common-sense reasoning hard to model within the classical approach. Particularly, defeasible logic programs are expressive enough to represent incomplete and potentially contradictory information, applying ideas from defeasible argumentation in order to decide between conflicting goals. Nowadays, several architectures designed for executing DLPs are based on abstract machines.^[11-14]

A *Defeasible logic program* (delp program) is a set of facts, strict rules Δ of the form (A:-B), and a set of defeasible rules Δ of the form A-<B, whose intended meaning is “if B is the case, then usually A is also the case”. A *Category mapping* delp program includes facts which are formed from *likes*, and strict and defeasible clauses where the heads and bodies corresponds to the sets *Category1* and *Category2*. A given delp includes a part from a social profile that contains facts (likes), and a fixed set of mapping rules which include positive and negative occurrences of categories.

Let $P = (\Pi, \Delta)$ be a delp program and L a ground literal. A *defeasible derivation* of L from P consists of a finite sequence $L_1, L_2, \dots, L_n = L$ of ground literals, such that each literal L_i is in the sequence because:

- (1) L_i is a fact in Π , or
- (2) There exists a rule R_i in P (strict or defeasible) with head L_i and body B_1, B_2, \dots, B_k and every literal of the body is an element L_j of the sequence appearing before $L_j (j < i)$.

Let h be a literal, and $P = (\prod, \Delta)$ a delp program. We say that $\langle A, h \rangle$ is an argument for h , if A is a set of defeasible rules of Δ , such that:

- (1) There exists a defeasible derivation for h from $=(\prod \cup A)$
- (2) The set $(\prod \cup A)$ is non-contradictory, and
- (3) A is minimal: there is no proper subset A_0 of A such that A_0 satisfies conditions (1) and (2).

Hence an argument $\langle A, h \rangle$ is a minimal non-contradictory set of defeasible rules, obtained from a defeasible derivation for a given literal h associated with a program P .

We say that $\langle A_1, h_1 \rangle$ attacks $\langle A_2, h_2 \rangle$ iff there exists a sub-argument $\langle A, h \rangle$ of $\langle A_2, h_2 \rangle$ ($A \subseteq A_1$) such that h and h_1 are inconsistent (i.e. $\prod \cup \{h, h_1\}$ derives complementary literals). Our analysis will be focused on those attacks corresponding to defeaters. When comparing attacking arguments, we will assume a partial preference ordering on arguments (given e.g. by specificity as in Ref.^[11]).

Specificity, for example, is a syntactic criterion preferring those arguments that are more direct (i.e. Requiring less inference steps) or more informed (those based on more premises are preferred over those based on less information). This preference criterion is modular in DeLP, and in fact other criteria are possible (see e.g. Ref.^[15]), where numerical values are propagated via modus ponens and used for comparing arguments).

We will say that $\langle A_1, h_1 \rangle$ defeats $\langle A_2, h_2 \rangle$ if $\langle A_1, h_1 \rangle$ attacks $\langle A_2, h_2 \rangle$ at a subargument $\langle A, h \rangle$ and $\langle A_1, h_1 \rangle$ is strictly preferred (or not comparable to) $\langle A, h \rangle$. In the first case we will refer to $\langle A_1, h_1 \rangle$ as a *proper defeater*, whereas in the second case it will be a *blocking defeater*. Defeaters are arguments can be on their turn attacked by other arguments, as is the case in a human dialogue. An *argumentation line* is a sequence of arguments where each element in a sequence defeats its predecessor. In the case of DeLP, there is a number of *acceptability* requirements for argumentation lines^[7] in order to avoid fallacies (such as circular reasoning by repeating the same argument twice).

Based on the previous notions, DeLP queries are solved in terms of a dialectical tree, which subsumes all possible argumentation lines for a given query. The definition of dialectical tree provides us with an algorithmic view for discovering implicit self-attack relations in users' claims. Let $\langle A_0, h_0 \rangle$ be an argument from a program P . A *dialectical tree* for $\langle A_0, h_0 \rangle$ is defined as follows:

- (1) The root of the tree is labeled with $\langle A_0, h_0 \rangle$.

- (2) Let N be a non-root vertex of the tree labeled $\langle A_n, h_n \rangle$ and $\Lambda = [\langle A_0, h_0 \rangle, \langle A_1, h_1 \rangle, \dots, \langle A_n, h_n \rangle]$ the sequence of labels of the path from the root to N . Let $[\langle B_0, q_0 \rangle, \langle B_1, q_1 \rangle, \dots, \langle B_k, q_k \rangle]$ all attack $\langle A_n, h_n \rangle$. For each attacker $\langle B_i, q_i \rangle$ with acceptable argumentation line $[\Lambda, \langle B_i, q_i \rangle]$, we have an arc between N and its child N_i .

A marking on the dialectical tree can be then performed as follows:

- (1) All leaves are to be marked as U-nodes (undefeated nodes).
- (2) Any inner node is to be marked as U-node whenever all its associated children nodes are marked as D-nodes.
- (3) Any inner node is to be marked as D-node whenever at least one of its associated children nodes is marked as U-node.

After performing this marking, if the root node of the tree is marked as a U-node, the original argument at issue (and its conclusion) can be deemed as *justified or warranted*.

Let us now build an example of a dialectical tree of category mapping. Imagine we have a following set of mapping clauses and available categories obtained from Table 1.

Table 1. An example of a Defeasible Logic Program for modeling category mapping

<p><i>Defeasible Rules</i></p> <p><i>sports_outdoors</i> -< <i>sports_athletics</i></p> <p><i>sports_athletics</i> -< <i>exercise_facility</i>.</p> <p>\neg <i>sports_athletics</i> -< <i>exercise_facility, yoga</i>.</p> <p>\neg <i>sports_athletics</i> -< <i>chess</i>.</p> <p>\neg <i>community</i> -< <i>food_dining, music</i>. (<i>commercial, not a community event</i>)</p> <p><i>music</i> -< <i>rock</i>.</p> <p>\neg <i>sports_athletics</i> -< <i>exercise_facility, community</i> (<i>where people do stuff different from sport</i>)</p> <p><i>community</i> -< <i>food_dining</i>.</p> <p>\neg <i>community</i> -< <i>business_tech</i>.</p> <p>\neg <i>music</i> -< <i>visual_arts</i>.</p> <p><i>Facts (facts are obtain from explicit likes)</i></p> <p><i>exercise_facility</i>.</p> <p><i>yoga</i>.</p> <p><i>chess</i>.</p> <p><i>rock</i>.</p> <p><i>business_tech</i>.</p> <p><i>food_dining</i>.</p> <p><i>visual_arts</i>.</p>

In this category mapping to DeLP, the literal *sports_outdoors*

is supported by $\langle A, sports_outdoors \rangle = \langle \{(sports_outdoors - \langle sports_athletics \rangle), (sports_athletics - \langle exercise_facility \rangle)\}, sports_outdoors \rangle$ and there exist three defeaters for it with three respective argumentation lines: $\langle B_1, sports_athletics \rangle = \langle \{(sports_athletics - \langle exercise_facility, yoga \rangle)\}, sports_athletics \rangle$. $\langle B_2, sports_athletics \rangle = \langle \{(sports_athletics - \langle exercise_facility, community \rangle), (community - \langle food_dining \rangle)\}, sports_athletics \rangle$. $\langle B_3, _athletics \rangle = \langle sports_athletics - \langle chess, sports_athletics \rangle$. The first two are proper defeaters and the last one is a blocking defeater. Observe that the first argument structure has the counter-argument, $\langle \{sports_athletics - \langle exercise_facility \rangle, sports_athletics \rangle$, but it is not a defeater because the former is more specific.

Thus, no defeaters exist and the argumentation line ends there.

B_3 above has a blocking defeater $\langle (sports_athletics - \langle exercise_facility \rangle), sports_athletics \rangle$ which is a disagreement sub-argument of $\langle A, sports_outdoors \rangle$ and it cannot be introduced since it gives rise to an unacceptable argumentation line. B_2 has two defeaters which can be introduced: $\langle C_1, community \rangle$, where $C_1 = (community - \langle food_dining, music \rangle), (music - \langle rock \rangle)$, a proper defeater, and $\langle C_2, community \rangle$, where $C_2 = (community - \langle business_tech \rangle)$ is a blocking defeater. Hence one of these lines is further split into two; C_1 has a blocking defeater that can be introduced in the line $\langle D_1, music \rangle$, where $D_1 = (music - \langle visual_arts \rangle)$. D_1 and C_2 have a blocking defeater, but they cannot be introduced, because they make the argumentation line not acceptable. Hence the target category $sports_outdoor$ cannot be accepted for the given user, as the argument supporting the literal $sports_outdoor$ is not warranted. The dialectical tree for A is shown in Figure 1.

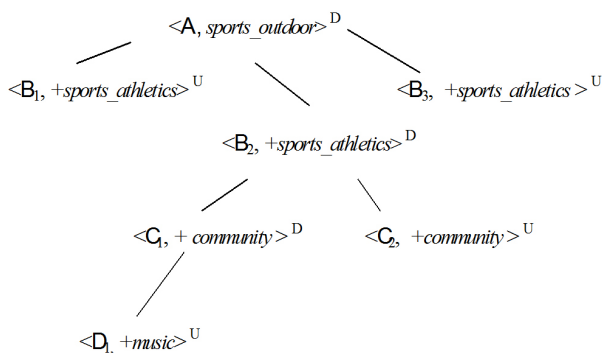


Figure 1. Dialectical tree for category $sports_outdoor$ using the DeLP Category Mapping

Having shown how to build dialectic tree, we now ready to outline the algorithm for category mapping:

- (1) Get the list of likes from the social profile, and their list of categories $Categories1$.
- (2) Filter out unimportant categories and likes following criteria outlined in Section 2.
- (3) Add resultant set of $facts$ to the fixed set of defeasible rules for category mappings.
- (4) Build a dialectic tree for each expected target category and accept/reject it based of defeasibility criterion.
- (5) Form the list of resultant target categories $Categories2$.

5. SYSTEM ARCHITECTURE OF PERSONALIZED RECOMMENDATIONS COMPONENT

Besides the category mapping component which requires AI technology, there is a number of web mining and data processing components required to produce personalized recommendations (see Figure 2).

The system mines both private Facebook account information if such permission is granted, or attempts to mine a public data on user, including public part of Facebook profile, otherwise. Yahoo and Bing APIs are used to identify a URL for Facebook profile for a given user, given his name, if such profile is unique.

Ontologies for categories mapping and tags/ $likes$ matching are needed so the system can generalize from $likes$ where direct matches are unavailable. For example, last.fm ontology is used to assign tags to $likes$ and to categorize $likes$ with respect to local categories and tags of recommendation system.

Substantial search engineering efforts are required to convert a conventional keyword-based and facet-based search to processing $likes$ as search terms. Multiple hypotheses are formed at the query run time to assure most specific (and therefore accurate) results are obtained. In particular, $likes$ are first matched against event titles, then event tags, event categories and bodies of event description for finding most relevant events with the highest search rank. At the high level, personalization is implemented as a search with controllable generality. If the current version of search query is too broad / too general, we have too many search results (products to recommend). Otherwise, if the current search query is too specific / too constrained, we get too few or zero results. Naturally, a desired situation is somewhere in between, and to find the query with the proper generality we have to iterate through various indexed fields with various forms of query (Boolean and span, taking into account mutual positions of words in product description document).

Defeasible logic programming has been implemented in a procedural way following the algorithm of dialectic tree

building described above. The logic program for category search component fixed. mappings was built, manually and subject to few iterations in the process of relevance improvement, having the real-time

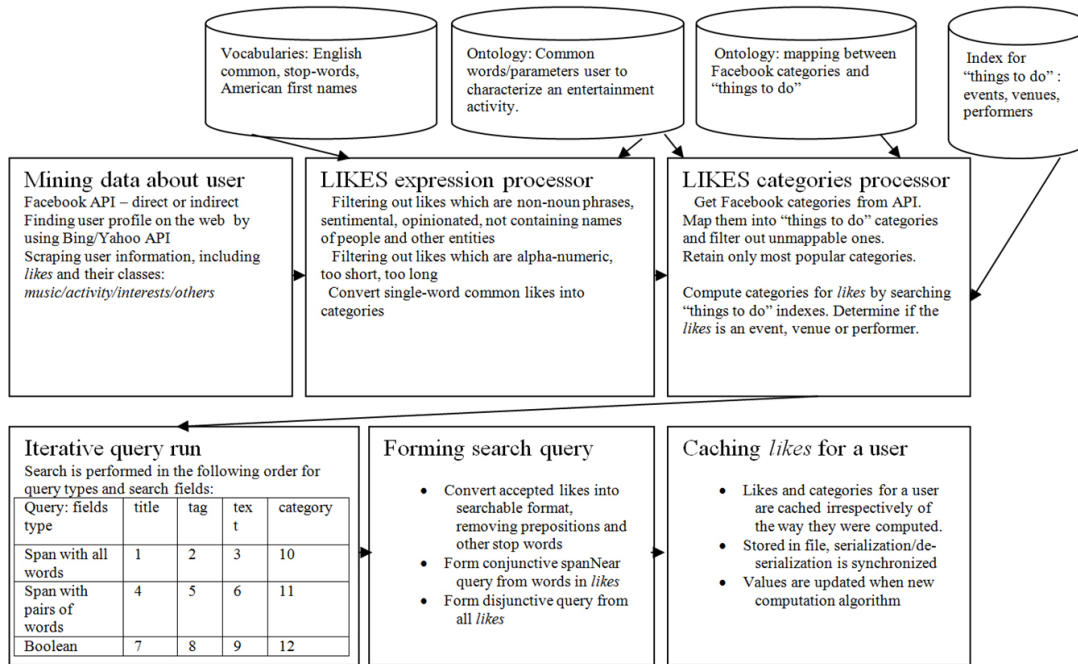


Figure 2. Architecture of personalization system

6. EVALUATION OF PERSONALIZED RECOMMENDATIONS

For evaluation of personalization we split the set of personalization users into the following five groups with respect to how complete their Facebook profile, how many likes they have and how representative they are of user interests:

- Novice or inactive user.
- Intermediate user with some relevant categories (music, outdoor).
- Intermediate users with a number of categories and likes.
- Advanced user accumulating many likes and systematically managing them.
- Advanced user accumulating many likes and not managing them.

For each above group, we conduct evaluation of the portion of relevant events suggested by the recommendation system. We use two recommendation scenarios: user specifies search query for a certain type of Events (see Table 3) or does not specify it (see Table 2). Each user produced a set of twenty requests and received ten events on average for each recommendation. User interface of apps.facebook.com/discover_zvents is shown in Figure 3.

Table 2. Evaluation of increase of the % of relevant recommendation without initial search query

Satisfaction	Satisfaction without personalization (%)	Satisfaction with personalization (%)
Group 1	67	58
Group 2	64	76
Group 3	63	82
Group 4	71	89
Group 5	69	73

Table 3. Evaluation of increase of the % of relevant recommendation with a search query for specific kind of events

Satisfaction	Satisfaction without personalization (%)	Satisfaction with personalization (%)
Group 1	64	59
Group 2	68	74
Group 3	66	81
Group 4	69	87
Group 5	68	80

The left column indicates the percentage of events found satisfactory when most popular (for everyone) events were recommended, and the right column shows events personalized for given user.

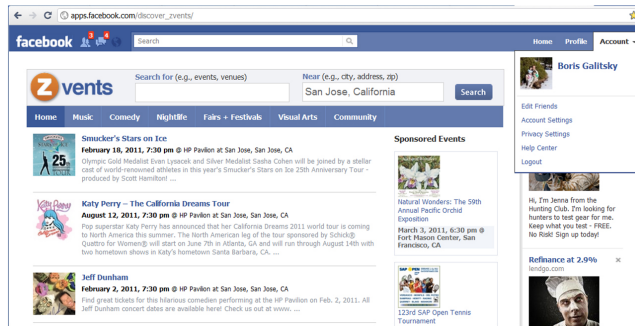


Figure 3. User interface of personalization system. It gets user geo-location from her IP and its preferences from her Facebook profile. List of recommended events for a given user changes as the location changes (see Figure 5). The current URL is https://www.facebook.com/Stubhub/app_267091300008193.

What we can see in general is that the more detailed Facebook profile is, the higher the improvement in the percentage of relevant events. This is true for both sessions with search and without search. However, when there is a high number of *likes* in diverse categories which are not managed, it is hard to assess the validity of each *likes* and one can observe a decrease of relevance improvement by means of personalization (last rows of both columns).

Overall, one can conclude that personalization delivers recommendations for products which would not be discovered by users otherwise: they would have to manually run multiple searches per each of their current *likes*/interests to discover if a relevant event is happening at their location.

One of the advantages of social network-based personalization is that a user becomes aware of much more events she would discover otherwise. We evaluate the proportion of events which would be exposed to a user, and call it event accessibility measure:

- (1) Using email notification (passive approach, users get email notifications with events they would potentially attend);
- (2) Using search (active approach, users try to find events they might want to attend);
- (3) Using personalization (passive, but expected to be a high-relevance approach).

For each user we build a total set T of events we believe are of interest to a person, using means other than personalization-related. We selected a ticket purchase data and user click data as most relevant and averaged through users with similar interest to derive T for the total set of potentially interesting events for a given class of users. Then we evaluate the size

of E_1 , E_2 and E_3 as subsets of T according to our definition above.

We selected 15 major metropolitan areas and 5 averaged users with their favorite categories of events. For each of these users, we calculated E_1 value based on search result by location and then filtering out events with foreign categories for a given user. E_1 is calculated assuming average category-based search session of 5 queries, and E_3 is obtained as a result of personalization to the selected averaged customer profiles.

One can see that personalization gives increase of 37% over the set of events which is being sent to an average user by email. A search session gives less than a quarter of events of potential interest offered by personalization (see Table 4).

Table 4. Accessibility of events by email notification, active search and personalized recommendations

Name	T	E_1	E_2	E_3
New York	13,092	120.5008	41.80151	245.0025
San Francisco	5,522	57.52668	18.76983	105.9543
Las Vegas	4,282	47.99049	15.02309	40.57705
Los Angeles	4,102	43.14766	14.01716	51.11965
Boston	3,798	41.84619	12.52423	59.65823
Chicago	3,515	41.03406	11.60954	40.69687
Houston	3,075	38.4163	10.84504	32.02998
Atlanta	2,757	27.67643	9.053296	36.87898
Nashville	2,693	27.96139	9.374129	30.10135
Austin	2,574	24.21604	9.135278	66.29651
Denver	2,518	26.31187	8.022806	32.02995
Lexington	2,140	23.17031	6.879982	18.03972
Charleston	2,131	23.32734	7.239115	18.01083
Philadelphia	2,062	17.99938	6.714275	27.12133
San Diego	1,930	23.17136	6.062084	21.06925
St Louis	1,910	17.58314	6.349103	17.23174
Washington	1,875	17.30389	6.53234	9.099314
Fresno	1,867	20.01329	6.526486	14.70334
Seattle	1,861	15.6457	5.52391	30.42235
Average	3,352.8	34.4	11.1	47.1
Percentage personalization improves the number of discovered events		137%	423%	100%

Different event categories have different popularities across the metropolitan areas. Most popular categories are shown with locations and counts of events in these categories in Table 5. It demonstrates that availability of events in various

categories significantly varies, as well as categories of *likes* (data is not shown) across metropolitan areas in US.

7. CONTENT MANAGEMENT SYSTEM FOR RECOMMENDATION SUPPORT

To successfully provide personalized recommendation, there are certain requirements to a content management system which maintains the content to be delivered to users in a context-aware form. Events need to be properly categorized, including music genre, each performer needs to have an associated list of similar performers which can substitute a desired one in each location, textual data must be rich enough to provide sufficient information for go/not go decision making. Images, vides, blogs, reviews are expectation to provide a user with an argument why she would enjoy an events. In this section we outline the content management system and its web mining capabilities to support personalized recommendations and their argumentation.

7.1 Mining for tail events, blogs, forums, images and videos

Providing detailed information on local events which are less popular on a global scale but fairly important to locals is a key competitive advantage of proposed personalization technology. The system runs a web search for event theme keywords and location-based keywords, and then finds web search results which looks like events, according to certain rules (shown as the third component from the left on the top of Figure 4). These rules include time and date constraints, price information, and some linguistic patterns specific to event descriptions. We have formed the training dataset, and

machine learning helps to filter out search results (and web domains) which are not events. Using this technology, we can add local events not available via feeds or the event fetching procedure, where we extract events from the page with known format. This component is capable of harvesting tail events since we don't need to know specific web domains or web page formats.

Since the events are obtained by the content acquisition system from multiple sources, they are frequently presented to users apart from their context. When content comes from feed or manually entered, frequently background knowledge on events, venues and performers is missing or limited. It is sometimes hard for a user to judge how interesting and appealing a given event first just by looking at title, short description, and a single image of a performer (if available). To enhance the user experience, we harvest additional visual data to provide more information about similar events, performers and locations. We mine for images and videos from major search engines and apply additional relevance verification, making sure entities from image captions are the same as performers, locations and other entities of events. 5-10 thumbnail images assist as well in the go / no-go decision. Entity verification is based on linguistic technology of syntactic matching, which assesses the similarity of phrases for entities at the level of syntactic parse trees. Once the syntactic tree for event title and each sentence in event description on one side, and image caption on the other side has been built, we try to find a substantial common sub-tree including a noun phrase for the common entity to confirm the relevance.

Table 5. Categories of most popular events in cities with the highest numbers of events

City	Event category	Count	City	Event category	Count(*)
New York	Theater	3811	Nashville	Museums	565
New York	Musicals	3301	Portland	Theater	563
New York	Performing Arts	3271	Denver	Comedy	553
Las Vegas	Performing Arts	2106	Mesa	Galleries	552
Houston	Theater	1317	Los Angeles	Food & Dining	549
Las Vegas	Theater	1150	San Francisco	Theater	545
New York	Comedy	1047	Calgary	Museums	542
San Francisco	Museums	920	Las Vegas	Comedy	542
New York	Clubs	895	Boston	Arts & Crafts	501
Chicago	Theater	839	Los Angeles	Theater	484
Austin	Comedy	696	London	Dance	478
New York	Museums	669	San Francisco	Clubs	477
San Diego	Theater	651	New York	Music	468
Milwaukee	Theater	632	Los Angeles	Musicals	436

7.2 Content enhancement for personalized recommendations

De-duplication is an essential component of any Content Management System. All entities like performers, events

and venues are subject to de-duplication, being careful not to merge entities which should not be merged (Left-middle area of Figure 4).

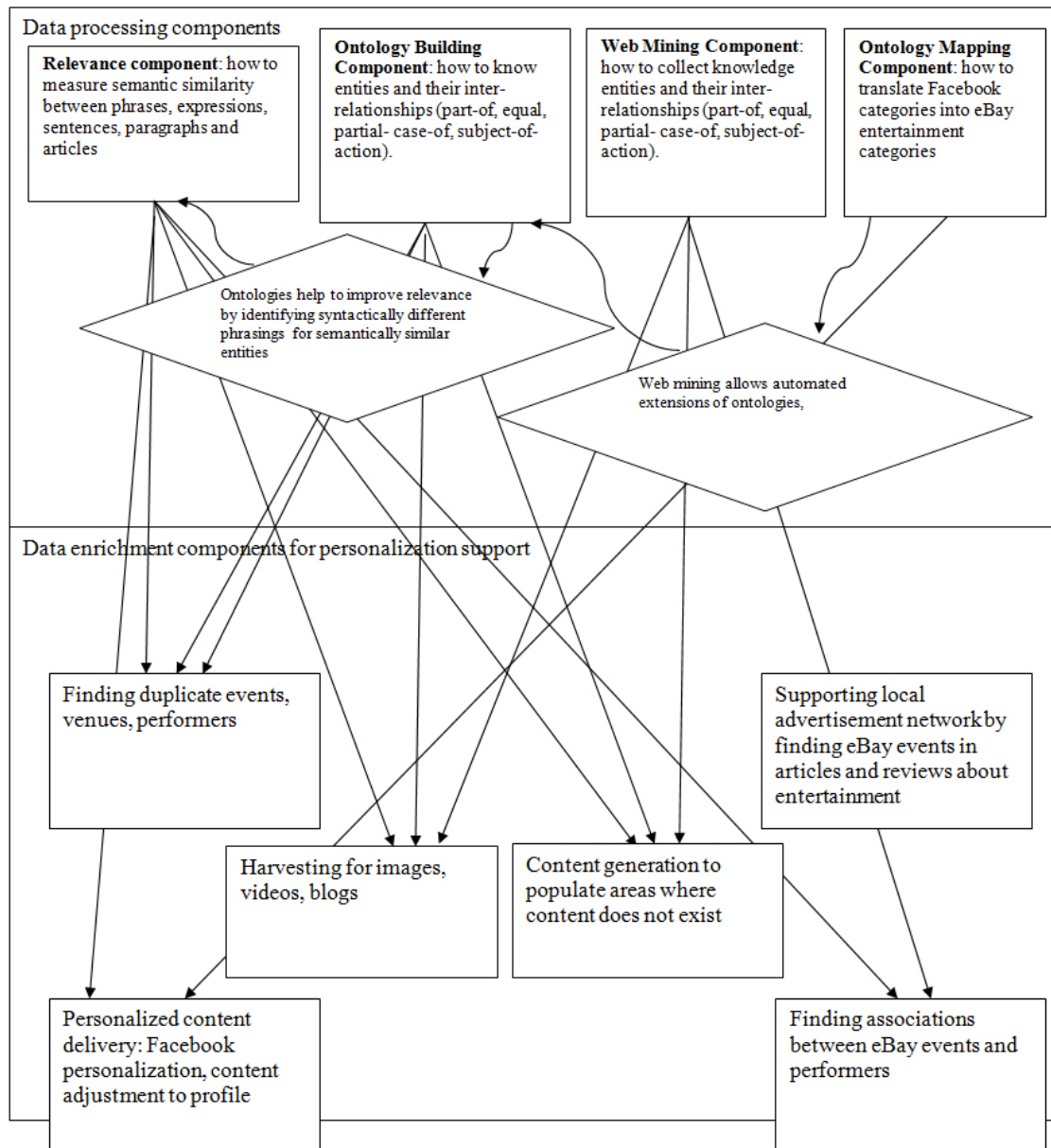


Figure 4. Technologies and high-level architecture of recommendation system

The De-duplication process has undergone thorough test-driven development on hundreds of cases and the system is always being adjusted to handle new problematic cases. We use a rather sophisticated rule system with multiple layers of exceptions, implemented via nonmonotonic reasoning-based architecture, machine learning and web mining, taking advantage of the experience web search engines have

accumulated on how people refer to performers, events and venues in various ways. We also look up the entities being de-duped at various entertainment domain-specific sites and social network sites to make sure most possible phrasings to express each entity is covered. We compute so called “similarity in web search space” among the search results for a pair of entities to reject/confirm. We also compute the

“similarity in web search space” among the search results for a pair of entities to reject/confirm that they are identical.

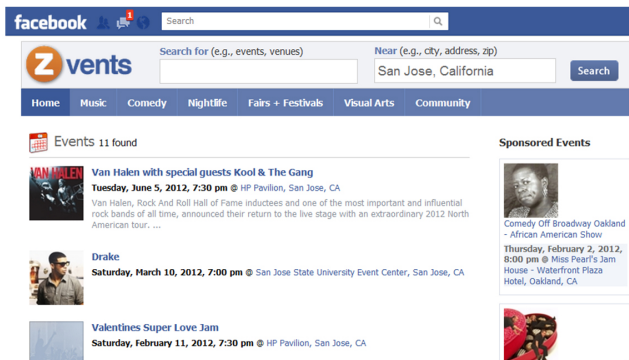


Figure 5. Once location changes, the personalization system finds other events for the same set of likes for this new location

For venues, the system extracts the relationship of being a sub-venue (like a room in a building) or a sibling between venues, which is important to provide a precise location for events and clarify how multiple events occurring in a given location are related to each other. A number of rules which are based on taxonomy of terms in venue titles, as well as address normalization rules, are applied, and being constantly machine learned to improve the accuracy and integrity of content.

The algorithms are available for generic objects at http://code.google.com/p/relevance-based-on-parse-trees/source/browse/src/main/java/opennlp/tools/apps/object_dedup

Matching algorithm 1: Finding common expression and confirming that it constitutes an entity

- (1) Get a set of candidate pairs
- (2) Compare entity names: select overlap keywords
 - See if common keywords are common words, or entity-specific words
 - Retain overlap words which are entity specific, subtract stop words for given duplicate-item type (event, venue, performer).
 - Normalize remaining words.
- (3) Filter out cases of too short list of overlap words, or those including all common English words.
- (4) Verify that “Normalize remaining words” is an entity by searching for it in web space
 - Collect all title of search results and see how the candidate searched entity occur in title
 - Filter out search results so that the candidate searched entity occurs in them in a “distorted”

way: there is no alignment between searched entity and obtained title

- Filter out cases where words other than {nouns, adjectives, gerunds} occur in the sub-title which corresponds to searched entity.
- (5) Count the number of all accepted search results titles for a formed entity and compare with threshold for the minimum number of such titles. If it is above the threshold, confirm that the overlap words constitute an entity
 - (6) If overlap words constitute an entity confirm duplication, otherwise confirm that candidate pair are different entities

Matching algorithm 2: finding performers for events with-out performers

- (1) Get a selected or full set of performers
 - Select all noun phrases
 - Select all expressions in quotes
 - Normalize all POS except nouns and adjectives (remove or turn into a normal form)
- (2) For the normalized input performer name, go to SOLR index of events and search for event without performer which has keywords from this performer:
 - First search for exact occurrence in “name” field
 - If no result, then look for occurrence in any field
- (3) For each set of SOLR search results, verify that found event name is included in the performer name from (1).
 - Verify that found performer name is word subset of event name
 - Verify that a candidate person name in found event name is the same person from the performer name
 - Verify the first/last name logic on both sides
- (4) Perform comparison in web search space: “dedupe.Matching algorithm 2: comparing possibly identical entities in web search space”
- (5) If verification steps 3)&4) are successful, confirm the found association between the input event and found event, otherwise no association for found event.

Also, it is of utmost importance that the content includes the latest and most accurate information on performers and venues. For the latter, we use web mining and search result filtering to confirm that currently stored information for a venue is correct. We filter out search engine-optimized results trying to find an original source of venue data, avoiding aggregator sites. A number of machine-learned rules is in action to verify venue address from the most authentic source found.

7.3 Content enrichment

For some events, there is a lack of textual information, and for most events all textual information available on the web is beneficial to improve the user experience familiarizing with event content. We have developed a technology of storytelling, based on factual information mined on the web by major search engines. Fragments of text mined on the web and determined to be relevant to the event, are combined based on the selected discourse model to improve the coherence of the composed synthetic text. If we generate a story on a particular performer, we search for information related to “performer-inspired”, then “performer-announced”, then “performer-produced”, followed by “performer’s audience-impressed, excited, shared with friends, inspired new performance”, merging fragments of text obtained as results of respective searchers, and accessing plausibility of obtained discourse, filtering out irrelevant and/or incoherent text fragments. The results do not read as well as professional writers but matches the quality of writing of an average college student:

Britney Spears “Femme Fatale” in North SF Bay Area:

“I was surprised to discover that sales for ‘Femme Fatale’ were relatively low for a superstar like Ms. Spears. I learned this week that Larry Rohter talks to Ben Ratliff about the new wave of Fado music from Portugal; Jon Caramanica on Britney Spears’ new album *Femme Fatale*; and Nate Chinen on jazz trumpeter Ambrose Akinmusire. A new wave of old-style Portuguese music including a new album from Britney Spears. Britney Spears is straddling the top of the pop charts, riding another smash album”.

I watched Britney Spears performed on Good Morning America Tuesday morning to celebrate the release of her new album, *Femme Fatale*. A comprehensive Britney Spears fan site which pays tribute to Britney with the most active message board, daily news, many pictures, and desktop media.

The latest news from Sony Artists including Westlife, Britney Spears, JLS & Alexandra Burke. Fans can look forward to seeing Britney Spears perform a whopping 22 songs when the pop diva launches her *Femme Fatale* tour on June 16. Details of the hit-packed set list were leaked after a handful of media members were treated to an ultra-private dress rehearsal for the tour, held Friday (June 10).

Britney Spears reveals “shocking” tour Singer also admits she gets nervous when she steps onstage. Example: “Hollywood inspired”, “Changed The Way You Kiss Me”, Watch the Capital Summertime Ball star talking about his chart topping hit Ne-Yo: “I’d love to play a comic character’ Singer also reveals release date for his forthcoming album. I don’t

know what Will Britney Spears Wear On Her *Femme Fatale* Toura. Will Tracy Morgan Recover From Anti-Gay Slur? Not sure about that. Mannie Fresh Loves And Hates Cash Money’s New Sound.”

In addition to improvement of the integrity of internal data, we focus on making sure a web resource which mentions available event can be automatically linked to this event. The entity extraction mechanism is similar to internal event-performer matcher, but it is also augmented with crawled, web domain manager and web page parser which ensures we apply the matching algorithm to a proper area on the webpage.

8. CONCLUSIONS

Recommender systems are intended in general to help users search for products and services in overloaded information domains. Recommender tools automatically select items that may be appealing to each user taking into account his/her personal preferences, in particular, in social networks like Facebook. The personalization strategies which match user preferences against the available items suffer from well-known word/syntactic matching issues that reduce the quality of the recommendations. It is well known that syntactic matching techniques miss a lot of useful information during recommendation process. Ref.^[16] proposed a personalization strategy that overcomes these drawbacks by applying inference techniques borrowed from the Semantic Web. Similar to the current paper, the approach reasons about the semantics of items and user preferences to discover complex associations between them. These semantic associations provide additional knowledge about the user preferences, and permit the recommender system to compare them with the available items in a more effective way. The authors illustrate its use in AVATAR, a tool that selects appealing audiovisual programs from among the myriad available in Digital TV. In the current paper the similar comparison framework is supported by defeasible reasoning, making the recommendation results more prone to inconsistencies in available information about users and domain knowledge. Ref.^[17] describe the profiling features that are used as an extension to a news indexing system, PerSSonal, enhancing the personalization algorithm that the system utilizes with various features derived from the user’s profile, such as the list of viewed articles and the time spent on them. The authors also analyze the system’s interconnection channels that are used with the client-side desktop application that was developed and evaluated.

Personalization and recommendation have their own flavors in multiagent settings. The user’s intention should be detected as soon as possible so that the agent can define a way to collaborate with the user. Plan recognition can be applied

to identify the user's goal based on her actions in the environment, such as event attendance selection process. However, classical approaches in plan recognition, lacking personalization and lacking considerations of the transition between different goals pursued by the user, fail.

Our study^[18] describes the travel recommendation system where a concept based model of mental world has been proposed. The broader issues of personalization and recommendation has been addressed in our earlier studies of simulation of human reasoning about mental world,^[19,20] explainability of recommendations,^[21] graph-based learning of conflict scenarios^[22] and formalization of speech act theory for the purpose of conflict resolution recommendations. The common idea behind these papers is that an adequate model of human reasoning, in particular about mental attitudes, is required to provide adequate personalized recommendations, in addition to the domain model itself (which is the focus of the current paper).

In evaluation of this study we demonstrated how Facebook users of various level of activity and sophistication benefits from personalization in a variety of degrees. Users with limited number of Facebook categories, or a high number of adjusted set of Facebook likes, leverages personalization the most. In this paper we argue that the hardest personalization component is to map Facebook categories into ones of the system providing recommendation, given its content and set of products/services. We tackled this problem by introducing defeasibility relation between category mapping rules, which allowed for inconsistent rules to be defeated and retain only rules which deliver more relevant recommendations.

Aggregating likes from friends is another important area supporting personalization, where adequate treatment of product categories is the key. The value of enables retailers/service providers and buyers/users alike to utilize the influence of trusted friends and family within the shopping experience has been demonstrated. Similar to the presented system, retailers and manufacturers using (iGoDigital 2011) product recommendation platform, can leverage a consumers' social network to provide an added layer of personalization and relevancy within their shopping experience. Consumers benefit from immediate access to product recommendations and opinions from their social network as they research, browse, and complete purchases, adding relevance and authenticity.

Defeasible reasoning is a rule-based approach for efficient reasoning with incomplete and inconsistent information. Such reasoning is, among others, useful for ontology integration, where conflicting information arises naturally; and for the modeling of business rules and policies, where rules with exceptions are often used. Category mapping is an example of such domain, where we need to systematically treat exceptions. Ref.^[23,24] describe these scenarios with rules with

exceptions in more detail, and reports on the implementation of a system for defeasible reasoning on the Web. The system (1) is syntactically compatible with RuleML; (2) features strict and defeasible rules and priorities; (3) is based on a translation to logic programming with declarative semantics; and (4) is flexible and adaptable to different intuitions within defeasible reasoning.

It must be remarked that integrating argumentation and recommender systems is a recent research topic. DeLP has proven to be an efficient tool for achieving this integration, as exemplified in Ref.^[12] Other successful applications of DeLP involve news analysis,^[14] intelligent processing of web-based forms,^[13] and intelligent robotic soccer,^[25] among many others.

DeLP has also been used for modeling ontology reasoning. Standard approaches to reasoning with description logics ontologies require them to be consistent. However, as ontologies are complex entities and sometimes built upon other imported ontologies, inconsistencies can arise. Ref.^[26] presents δ -ontologies, a framework for reasoning with inconsistent ontologies, expressing them as defeasible logic programs. Given a query posed w.r.t. an inconsistent ontology, a dialectical analysis is performed on a DeLP program obtained from such an ontology, where all arguments in favor and against the final answer of the query will be taken into account. The current paper presents an industrial system for handling a special case of ontology inconsistencies by using a different mechanism of mapping, but similar underlying logic of argumentation.

Our evaluation demonstrates that using personalized instead of traditional recommendations, we significantly increase:

- Overall user satisfaction with recommendation system, because users have to deal much less with irrelevant recommendations
- The number of attended events, including ones requiring ticket purchase.

Hence personalized recommendation dramatically improves efficiency and effectiveness of the user decision process on which events to attend.

The system presented in this paper is oriented to work with Facebook and developed as a Facebook App and mobile app; however, other social network profiles can be handled in a similar manner, including international social profiles in various languages.

ACKNOWLEDGEMENTS

The author is grateful to Amit Nithianandan and Gordon Rios for fruitful discussions and help in preparation of this manuscript.

REFERENCES

- [1] Varshavsky R, *et al.* Group Recommendations in Social Networks US Patent App 20110270774, Microsoft.
- [2] Folino G, Mastroianni C, Mostaghim S, *et al* (eds). BADS. 2011: Proceedings of the 3rd IEEE Internatioal Workshop on Bio-Inspired and Self-* Algorithms for Distributed Systems, ACM Press, June 2011.
- [3] Anderson RM, Christian HB, Jennifer TC, *et al.* Trust-based recommendation systems US Patent 7991841 Microsoft 2011.
- [4] Amazon.com 2015.
- [5] last.fm 2015.
- [6] Hunch.com 2015.
- [7] Garcia A, Simari G. Defeasible logic programming: An argumentative approach. *Theory and Practice of Logic Programming*. 2004; 4: 95-138. <http://dx.doi.org/10.1017/S1471068403001674>
- [8] Rahwan I, Amgoud L. An argumentation based approach for practical reasoning. In *International Joint Conference on Autonomous Agents and Multi Agent Systems*. 2006: 347-54.
- [9] Bordini RH, Braubach L. A survey of programming languages and platforms for multi-agent systems. In *Informatica*. 2006: 33-44.
- [10] Bordini RH, Wooldridge M, Hübner JF. *Programming Multi-Agent Systems in AgentSpeak using Jason*. John Wiley & Sons. 2007.
- [11] González, Sebastian Gottifredi MP, Alejandro J, *et al.* Towards Argument Representational Tools for Hybrid Argumentation Systems. *HCI*. 2011; 12: 236-45.
- [12] Chesñevar, Maguitman CA, González MP. Empowering Recommendation Technologies Through Argumentation. In: *Argumentation in Artificial Intelligence*. I.Rahwan, G.Simari (eds.). Springer Verlag, 2009. ISBN 978-0-387-98196-3.
- [13] Gómez SC, Chesñevar G. Defeasible Reasoning in Web-Based Forms through Argumentation. *International Journal of Information Technology and Decision Making (IJITDM)*. 2008 March; 7(1): 71-101. <http://dx.doi.org/10.1142/S021962200800282X>
- [14] Sagui F, Maguitman A, Chesñevar C, *et al.* Modeling News Trust: A Defeasible Logic Programming Approach. *Iberoamerican Journal of Artificial Intelligence*. 2009; 12(40): 63-72.
- [15] Alsinet T, Chesñevar CI, Godo L, *et al.* A logic programming framework for possibilistic argumentation: Formalization and logical properties. *Fuzzy Sets and Systems*. 2008; 159(10): 1208-28. <http://dx.doi.org/10.1016/j.fss.2007.12.013>
- [16] Blanco-Fernández Y, Pazos-Arias JJ, Gil-Solla A, *et al.* A flexible semantic inference methodology to reason about user preferences in knowledge-based recommender systems. *Knowledge-Based Systems*. 2008 May; 21(4): 305-20. <http://dx.doi.org/10.1016/j.knsys.2007.07.004>
- [17] Bouras C, Tsogkas V. Noun retrieval effect on text summarization and delivery of personalized news articles to the user's desktop. *Data & Knowledge Engineering*. 2010 July; 69(7): 664-77. <http://dx.doi.org/10.1016/j.datak.2010.02.005>
- [18] Galitsky B, Rosa J. Concept-based learning of human behavior for customer relationship management. *Special Issue on Information Engineering Applications Based on Lattices*. *Information Sciences*. 2011 May; 181(10): 2016-35. <http://dx.doi.org/10.1016/j.ins.2010.08.027>
- [19] Galitsky B, *et al.* Discovering common outcomes of agents' communicative actions in various domains. *Knowledge-Based Systems*. 2011a; 24(2).
- [20] Galitsky B. Exhaustive simulation of consecutive mental states of human agents. *Knowledge-Based Systems*. 2013. <http://dx.doi.org/10.1016/j.knsys.2012.11.001>
- [21] Galitsky B, *et al.* Assessing plausibility of explanation and meta-explanation in inter-human conflict. *Engineering Application of AI*. 2011b; 24(8): 1472-86.
- [22] Galitsky B, González MP, Chesñevar CI. A novel approach for classifying customer complaints through graphs similarities in argumentative dialogue. *Decision Support Systems*. 2009; 46(3): 717-29. <http://dx.doi.org/10.1016/j.dss.2008.11.015>
- [23] Antoniou G, Bikakis A, Wagner G. A Defeasible Logic Programming System for the Web. 16th IEEE International Conference on Tools with Artificial Intelligence (ICTAI'04) pp.756-7. Boca Raton, Florida.
- [24] Antoniou GD, Governatori BG, Maher M. Representation results for defeasible logic. *ACM Transactions on Computational Logic*. 2001; 2(2): 255-87. <http://dx.doi.org/10.1145/371316.371517>
- [25] Ferretti E, Errecalde M, García AJ, *et al.* An Application of Defeasible Logic Programming to Decision Making in a Robotic Environment. *LPNMR*. 2007: 297-302.
- [26] Gomez SA, Chesñevar CI, Simari GR. Reasoning with Inconsistent Ontologies through Argumentation. *Applied Artificial Intelligence*. 2010; 24(1&2): 102-48.