Exploiting Argumentation Content and Structure to Advance Collaboration through Hybrid Recommendations

Spyros Christodoulou and Nikos Karacapilidis
University of Patras
Greece
e-mail: shristod@cti.gr; nikos@mech.upatras.gr

Costas Pappis
University of Piraeus
Greece
e-mail: pappis@unipi.gr

***Abstract*—Contemporary collaborative environments involve a flood of collected and exchanged data and require advanced techniques to enhance data processing, allow data transformation in actionable insights and reduce the subsequent cognitive overhead. In line with these requirements, this paper presents a hybrid recommender engine that builds on the synergy of content-based and collaborative filtering techniques to provide recommendations in argumentative collaboration settings. The proposed engine has been integrated in a web-based collaboration support system and exploits the content and structure of the underlying argumentation. Through a scenario of use, we demonstrate the application of our approach and discuss its usefulness in terms of advancing collaboration and augmenting the quality of decision making.**

Keywords-collaboration; argumentation; decision making; hybrid recommendations.

I. INTRODUCTION

Current data-intensive collaboration and decision making settings require efficient and effective techniques that provide personalized support, enhance the collaboration process and, ultimately, improve the quality and accuracy of the decisions to be made [1]. In this direction, recommender (or recommendation) systems [2], a type of information filtering systems that focus on predicting user responses to options, aim to assist users in processing large amounts of information, by reducing the subsequent cognitive overhead and supporting their decision making tasks [3]. Recommender systems have been proven to be valuable for coping with information overload and have become one of the most powerful and popular tools in diverse areas. Consequently, many applications have integrated recommendation techniques to provide users with helpful suggestions.

A variety of recommendation techniques have been already proposed, each one having certain strengths and weaknesses [4]. Besides, much attention is being lately paid to the exploitation of argumentation towards offering more valid suggestions. Argument-based recommender systems [5], as these tools are usually referred in the literature, are tools aiming to better support users by providing recommendations on the basis of associated arguments. For instance, a prototype of a group argumentation support system that applies frame-based information structure and argumentation to support group decision task generation and identification is presented in [6]; an approach to enhance practical reasoning capabilities of recommender system technology by incorporating argument-based qualitative inference is proposed in [7]; finally, ArgueNet [8] was designed as a recommender system based on a defeasible argumentation framework to classify Web search results according to preference criteria that have been declaratively specified by the user.

In line with the above, this paper presents a hybrid recommender engine that builds on the synergy of content-based and collaborative filtering techniques. The novelty of our approach lies in its meaningful exploitation of the content and structure of an ongoing argumentation in order to provide actionable recommendations. The approach presented in this paper assumes that the collaboration taking place adheres to a classic formal argumentation model, namely *Issue-Based Information System (IBIS)* [9]. Adopting IBIS, an ongoing collaboration is structured as a graph, whose basic elements are *issues* (questions to be answered), each of which are associated with alternative *positions* (possible answers); in turn, these are associated with *arguments* which support or object to a given position or another argument. In any case, the approach described in this paper can be easily adjusted to accommodate alternative argumentation models.

The remainder of this paper is structured as follows: Section II reports on related work in the area of recommender systems. Sections III and IV present in detail the proposed hybrid recommender engine and its integration with an already implemented collaboration support system that adopts the abovementioned model. Through an illustrative example scenario, Section V demonstrates how the recommendations produced by the proposed engine may advance an ongoing collaboration and enhance the quality of collective decision making. Section VI concludes the paper and discusses related remarks.

II. RECOMMENDER SYSTEMS

A recommender system can be viewed as a personalized information agent aiming to assist the natural social process of making choices (suggestions on items a user is likely to be interested in) without sufficient personal experience of the existing alternatives. The development of these systems has been based on diverse techniques, which can be classified in four main categories [10]: (i) *collaborative:* the generated item recommendations for a specific user are based on items rated by other “similar” users; (ii) *content-based:* recommendations for a specific user are generated according to each item’s features and the user’s preferences (i.e., the aggregation of items the user likes or dislikes); (iii) *knowledge-based:* recommendations follow the inferences about one’s needs and preferences, and (iv) *demographic:* the demographic profile of the user is exploited to provide recommendations.

Even though a great number of recommender systems belonging to the above categories (referred as “simple”) have evolved since the mid 90’s, all “simple” recommender techniques have certain strengths and weaknesses [11]. For instance, all "learning-based" techniques (i.e., collaborative, content-based and demographic) suffer from the "cold start" problem (i.e., the difficulty in handling new items or new users); the collaborative and content-based techniques suffer from the "portfolio effect" (i.e., an item similar to an item that a particular user has rated before would be never recommended to that user).

Hybrid recommendation approaches try to mitigate the above drawbacks and, at the same time, exploit the advantages of "simple" recommendation techniques by combining two or more of them in a uniform approach. Depending on the particular method applied to combine the "simple" recommendation techniques, hybrid recommender systems may be classified in seven main categories [11]: (i) *weighted:* each item gets a number of partial scores (as many as the number of the "simple" recommendation techniques) reflecting the value of this item with respect to each recommendation technique. The total item score results from the linear combination of the partial scores (weights are used to state the importance of a "simple" recommender technique over another); (ii) *switching:* based on the evaluation of the recommendation situation, the system selects among a number of "simple" recommender techniques to apply. The selection of a reliable criterion to conduct this method is a critical task and remains an open research issue [12]; (iii) *mixed:* the output of two or more recommendation techniques is presented and it is up to the user to select the best items among the different items' lists returned; (iv) *feature combination:* features of one source are injected into an algorithm that was initially designed to perform data processing of a different source; (v) *feature augmentation:* a recommendation technique is applied to extract a number of features, which are then used as input to another recommendation technique; (vi) *cascade:* a "weak" recommendation technique is applied to refine (but not overturn) the decisions made by a "strong" recommendation technique; (vii) *meta-level:* the model resulting from one recommendation technique is used as input to another.

As described in detail in the next section, our approach integrates the collaborative and the content-based filtering techniques by adopting the switching method.

III. A Hybrid Recommendation Engine

A. The need for recommendations in a collaboration setting

In a data-intensive and cognitively-complex argumentative collaboration setting, users often need help in spotting those parts of an ongoing argumentation that can really advance collaboration and augment the quality of decision making. In such settings, a recommender engine could enable users in:

- locating already existing argumentation items that are similar to a new item they have just contributed to an ongoing collaboration; such recommendations may trigger the creation of meaningful interrelations between the new item and the existing ones;
- spotting users with similar profiles, in order to catch up with their argumentation items;
- tracking popular argumentation items, which receive much attention and may influence the evolution of the collaboration;
- gaining insights about the probable outcome of the collaboration.

In the context of an argumentative collaboration support system, an efficient recommender engine should not only rely on the content of the collaboration; it should also exploit the structure of the associated discourse graphs that involve multiple stakeholders. Such a hybrid approach is described in the following, where a content-based recommender exploits features of specific collaboration items, while a collaborative filtering recommender considers the users' rating profiles and the total structure of the argumentative discourses to generate meaningful and helpful recommendations (hereafter, the terms 'collaboration item' and 'argumentation item' are used interchangeably).

B. Content-based recommendations

Generally speaking, content-based recommender systems rely on the users' rating profiles to provide recommendations; items sharing similar features with the items a particular user has liked in the past are recommended to the user [13]. In the context of an argumentative collaboration support system, the proposed procedure of providing users with content-based recommendations breaks up into two distinct tasks: (i) calculating a rating profile for each user, and (ii) spotting similar collaboration items (with compatible features) to each user's rating profile.

With respect to the first task, a user's *Z* rating profile *RP(Z)* is defined as the set of all collaboration items rated by her. Collaboration items that have not been rated by user *Z* are not included in *RP(Z)*. As far as the second task is concerned, spotting similar collaboration items to a user's *Z* rating profile requires comparing each argumentation item of the collaboration space with each argumentation item included in *RP(Z)* to decide about their *degree of similarity*. As a prerequisite, we need a definition of an appropriate *degree of similarity DoS(x,y)* function to reflect how similar two collaboration items *x* and *y* are. As we focus on content-based recommendations in this step, *DoS(x,y)* should be based on items' *x* and *y* contents (i.e., their titles and bodies).

`MoreLikeThisHandler` (from the Apache Solr open source library, `http://lucene.apache.org/solr/`) offers a suitable to our purposes implementation of a *DoS(x,y)* function to compare two documents and decide on their degree of similarity. According to it, *DoS(x,y)* corresponds to an increasing function (i.e., the more similar two documents *x* and *y* are, the larger the *DoS(x,y)* value is) and can be easily applied to perform all the necessary comparisons between pairs of argumentation items.

As our basic target is to spot similar argumentation items to a user's *Z* rating profile *RP(Z)*, we have to compare each argumentation item *x* in the collaboration space with each argumentation item *y* in *RP(Z)* and calculate the respective *DoS(x,y)* values (excluding the items the user *Z* has rated). To decide about how similar an argumentation item *x* is to the set of argumentation items of *RP(Z)*, we define the cumulative degree of similarity *CDoS(x,Z)* of an argumentation item *x* to the rating profile of user *Z* as:

$$CDoS(x,Z) = \sum_{y \in RP(Z)} \mathrm{DoS}(x, y) \qquad (1)$$

Taking into account that *DoS(x,y)* is an increasing function, the larger the *CDoS(x,Z)* is, the more similar an argumentation item *x* is to the rating profile of user *Z*. The calculation of *CDoS(x,Z)* for each item *x* and user *Z* is straightforward (by using Eq. (1)). The argumentation items recommended to user *Z* are the ones with the larger values of *CDoS(x,Z)*.

C. Collaborative filtering based recommendations

As already stated, the central idea of collaborative filtering is to provide a user with recommendations based on the rating history of similar users (i.e., users with similar rating profiles to the active user). In such systems, the recommendation procedure involves two major steps. The first step involves the construction of the utility matrix containing, for each user-item pair, a value that represents what is known about the degree of approval of that user for that item. The respective values reflecting the degree of approval either come from an ordered set or are scalar. Most entries of the utility matrix are usually unknown, i.e., we have no explicit information concerning the users' approval on the full set of items.

In our approach, the utility matrix and the related degrees of approval for each *(item_x, user_Z)* pair are calculated by taking into account two parameters:

- *User's Z rating on the argumentation item x* (denoted as *R(x,Z)*). We assume that a user *Z* may rate each argumentation item using the 1-5 stars rating scale.
- *The argumentation approval score* (denoted as *AAS(x,Z)*) reflecting a user's Z approval of a particular item *x* (as this approval has been expressed through the argumentation process).

We consider that *AAS(x,Z)* is directly related to the number, type (i.e., in favour or against) and structure of arguments that are linked to the specific item *x* (taking into account only the argumentation items put forward by user *Z*). Intuitively, a large number of arguments (created by user *Z*) in favour of a specific argument *x* expresses a larger approval (concerning user *Z*) on item *x* than a small number of arguments in favour of it. What is needed at this point is a method to measure how supportive (or adverse) to a specific item *x* the arguments posed by user *Z* are.

In the direction of assessing user's *Z* collaboration attitude on item *x*, we define a *user's Z argumentation graph for item x* (denoted as *G(x,Z)*) as the aggregation of all paths (denoted as *p(x,Z)*) leading to the item *x*, under the condition that all paths have been created by *Z*:

$$G(x,Z) = \bigcup \mathrm{p}(x, \quad Z)$$

In other words, *G(x,Z)* results from pruning the argumentation graph by removing:

- all relations of the argumentation graph that have been not created by *Z*, and
- all "isolated" items (i.e., items not belonging to any *p(x,Z)*).

As results from the above, the value of *AAS(x,Z)* is directly related to the evaluation of the associated argumentation graph. To calculate the value of the argument on the root of the argumentation graph, we follow the "global" approach (tuple-based valuation [14]) stating that the value of an argument is equal to the algebraic summation of the corresponding argumentation scores of argument paths leading to this argument. The argumentation score of an argument path is directly related to the number of *"against"* and *"in favour"* arguments forming the argument path. If e_A is an "against" relation and $|e_A(p(x,Z))|$ is the number of "against" relations along an argument path *p(x,Z)*, then, following the abovementioned "global" approach, the argumentation approval score *AAS(x,Z)* is:

$$AAS(x,Z) = \sum_{\substack{p(x,Z) \in G(x,Z) \\ e_A \in p(x,Z)}} (-1)^{|e_A(p(x,Z))|} \qquad (2)$$

The degree of approval of a user *Z* on a collaboration item *x*, denoted as *DA(x,Z)*, is calculated by combining the two partial scores (user's *Z* rating on item *x*, *R(x,Z,)* and the corresponding argumentation approval score *AAS(x,Z)*). It is:

$$DA(x,Z) = a_1 {*} R(x,Z) + a_2 {*} AAS(x,Z),$$

where a_1 and a_2 are user-defined weights to reflect the relative importance of the two associated scores.

The utility matrix results from the calculation of *DA(x,Z)* for each *(item_x, user_Z)* pair. After calculating the utility matrix, the second step includes feeding a collaborative filtering based recommender with all degrees of approval, so as the implemented algorithm to provide recommendations on demand.

For the collaborative filtering algorithm, a modified version of an *Alternating Least Squares* algorithm to factor matrices has been integrated in our approach. An implementation of this algorithm is offered in the open source `Myrrix` recommender engine [15] (which is currently part of the `Oryx` open source project, see details at: `https://github.com/cloudera/oryx`). According to Myrrix creators, the implemented recommender engine is based on large matrix factorization, tries to learn a small number of features in order to explain users' and items' observed interactions, is nearly immune to the "cold start" problem and can provide quality recommendations for very new users or items.

D. Hybrid recommendations

In the settings under consideration, it would be expected that embedding a collaborative filtering based recommender would be enough to provide effective recommendations. However, especially in the early stages of a collaboration process, the limited users' contribution (in terms of the number of argumentation items added, the number of the relationships created and the ratings of the above items) may not be able to provide accurate recommendations. In such a case, where the scores of the collaborative filtering based recommendations provided are pretty close, content-based recommendations are also exploited to discretize among recommendations of which the value is almost equal.

We follow a cascade hybrid recommender approach to return the appropriate list of recommendations to a user. The proposed hybrid recommender includes the following steps (note that *score(Ri, List1)* is a function returning the score of the *i*-th recommendation on *List1* and *T* is a user-defined parameter):

```
1. Apply the content recommender technique and get
   the top-N1 content recommendations (List1)
2. Apply the collaborative recommender technique and
   get the top-N2 collaborative recommendations
   (List2)
3. Parse List1 and compare each two recommendations
   (Ri, Rj) on List1
4. If (score(Ri, List1) - score(Rj, List1)) < T)
       If (score(Ri, List2)- score(Rj, List2)) > T)
              Interchange(Ri, Rj) on List1
5. Repeat Step 4 until no interchange on List1 has
   taken place.
6. Return List1
```

IV. The Dicode Collaboration Support System

The proposed recommendation engine has been fully integrated in a web-based collaboration and decision making support system, namely *Dicode*, and exploits the content and structure of the underlying argumentation. Dicode follows an IBIS-like argumentation model and aims to augment collaboration in diverse data-intensive and cognitively-complex settings [16][17]. To do so, it builds on prominent high-performance computing paradigms and large scale data processing technologies to meaningfully search, analyse and aggregate data existing in diverse, extremely large, and rapidly evolving sources. The Dicode approach brings together the reasoning capabilities of the machine and the humans and enables the meaningful incorporation and orchestration of a set of interoperable web services to reduce the data-intensiveness and complexity overload in collaborative decision making settings.

In particular, the aim of *Dicode's collaboration and decision making services* is twofold: (i) to exploit the reasoning abilities of humans through the creation, management and use of innovative workspaces that augment synchronous and asynchronous collaboration, and (ii) to intelligently support stakeholders in decision making activities by enabling the use and exploitation of appropriate reasoning mechanisms. These services build on an appropriate formalization of the collaboration and exploit a series of reasoning mechanisms to support stakeholders in their daily decision making processes. Dicode implements alternative visualizations of the collaboration space (called *"views"*), each one offering a varying degree of formality.

In the context of this paper, we will focus on the *"mind-map view"* of the collaboration workspace. In this view (Figure 1), the collaboration workspace is displayed as a mind map, where users can upload and interrelate diverse types of items. This view uses a spatial metaphor to organize items, allowing users to select and freely move around any item. Item types supported include *ideas*, *notes* and *comments*. Ideas stand for items that deserve further exploitation; they may correspond to an alternative solution to the issue under consideration. Notes are generally considered as items expressing one's knowledge about the overall issue. Finally, comments are items that usually express less strong statements and may include some explanatory text or potentially useful information. Users can customize the set of available item types by creating additional ones, thus better annotating a particular collaboration workspace.

Two collaboration items can be explicitly connected using directed edges (relations). Visual cues are used to express semantics: for instance, a user may appropriately choose the width and colour of an edge to express a specific semantic relationship between two collaboration items (e.g., a red edge denotes an "against" relation, a green one stands for an "in favour" relation). Additional functionalities offered include the creation of adornments (a grouping mechanism to aggregate items related to a particular alternative), a "like/dislike" mechanism to express a user's acceptance/rejection concerning a collaboration item, rating of collaboration items, calculation of workspace statistics, and a replay mechanism that helps a user review the evolution of a workspace over time. The mind-map view builds on the reasoning capabilities of humans to support ease-of-use and expressiveness, as well as individual and group sense-making, by supporting stakeholders in locating, retrieving and meaningfully interacting with relevant information; moreover, in monitoring and comprehending the evolution of collaboration.

V. Scenario Of Use

To better illustrate the proposed approach (and its integration in the Dicode system), this section presents an illustrative real-world scenario from the area of prostate cancer research. A physician (George), an urologist (John) and a biomedical researcher (Jane) aim to investigate which is the best alternative treatment for the prostate cancer. Initially, they set up a Dicode collaboration workspace and start using it in the mind-map view (Figure 1).

John suggests that one of the best and most popular treatments for the prostate cancer (Figure 1(a)) is the "active surveillance". He adds an alternative to make his statement (Figure 1(a)). Jane is not in favour of this option, because it requires close monitoring (regular digital rectal exams, PSA tests, and prostate biopsy) to monitor for signs of progression, so she adds her "against" position on the collaboration workspace (Figure 1(b)). Contrary to Jane, George supports John's opinion ("in favour" position supporting the alternative suggested by John (Figure 1(c)), in the sense that active surveillance avoids site effects from radiation therapy or prostatectomy. On the other hand, he is skeptical as with Active Surveillance there is no post-treatment staging information ("against" position - Figure 1(d)).

Jane argues that "Brachytherapy" has been also used to treat tumors in many body sites and this could be one option (Figure 1(e)). One of its major advantages is that this procedure does not need hospitalization ("in favor" position, Figure 1(f)) and, furthermore, there are no surgical risks involved. John is not convinced by her arguments as Brachytherapy requires close monitoring ("against" position, (Figure 1(g)), which may even include hospital visits. To support his consideration against the Brachytherapy, John denotes that there is no post-treatment staging information which is also an important factor ("against" position, (Figure 1(h)).

George argues that the best alternative, in his opinion, is "radical prostatectomy" (Figure 1(i)) as it is quite common with very good results. John is in favour of this option ("in favour" position) as this solution is proven to reduce prostate cancer death rates (Figure 1(j)). Moreover, the removed tissue allows accurate stating (Figure 1(k) – "in favour" position),

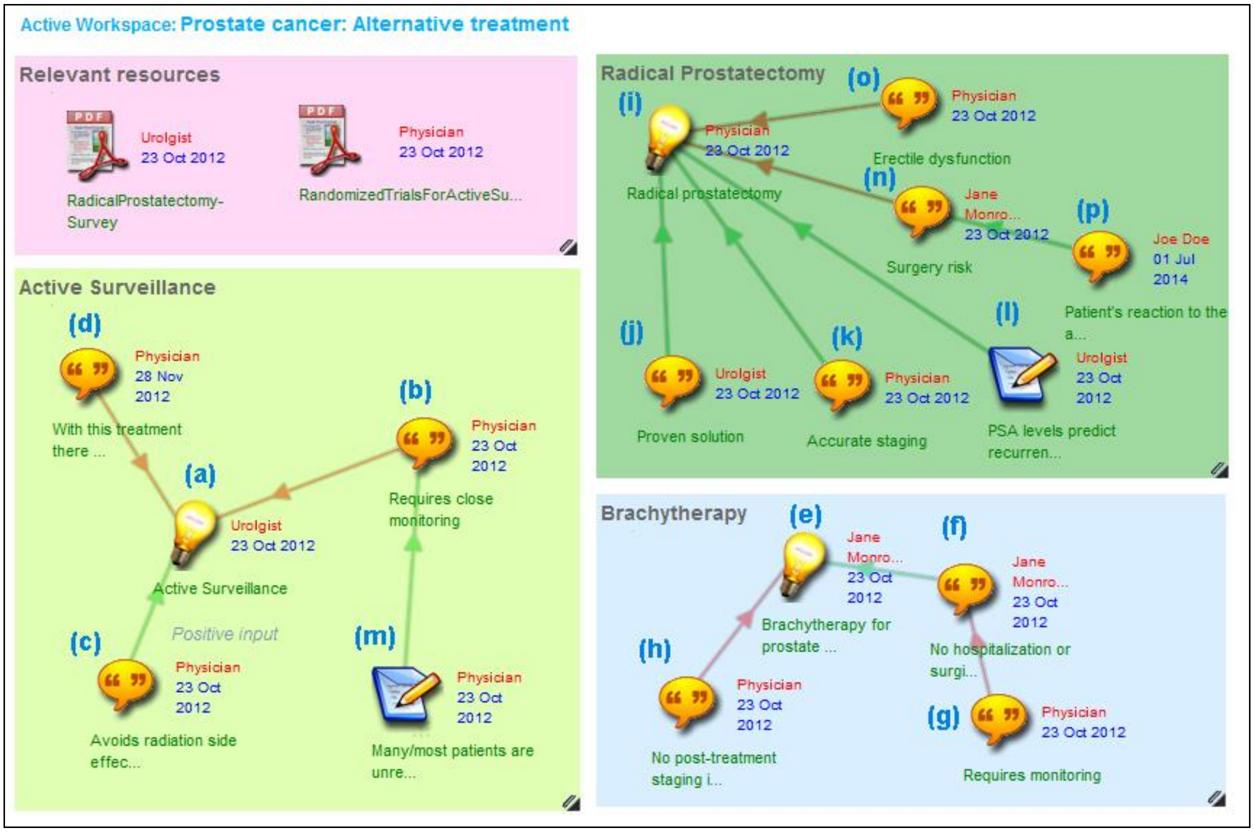

Figure 1. An instance of a real-world scenario concerning collaboration in the area of prostate cancer research.

which is very important and the PSA levels may reliably predict the recurrence (Figure 1(l) – “in favour” position).

Based on the collaboration on the mind-map view so far and his expertise on the field, John is convinced that he is able to contribute on the ongoing collaboration process; however, he is not absolutely certain about the most appropriate collaboration item he should react on (e.g., by creating an argument in favor or against it). He decides to invoke the hybrid recommender to get some insights. As a result, a list of recommended items (Figure 2) is returned. It is noted that these items are compatible to his rating profile and the rating profiles of similar to him Dicode users.

Collaborative recommended collaboration items:

Ranking	docTitle	score
1	With this treatment there .	15.84
2	Requires monitoring	15.78
3	No post-treatment staging	15.67
4	Proven solution	6.40
5	RadicalProstatectomy-Sur	6.40
6	RandomizedTrialsForActiv	6.40
7	No hospitalization or surgi	3.20

Page 1 of 1 10 View

Figure 2. The output of the collaborative recommender for John.

Having elaborated the output of the collaborative recommender, John selects the second recommendation as the one closest to his knowledge profile and contributes to Jane’s comment ((Figure 1(b))). He is contradicting to her point of view because, according to his experience, most patients are unreliable as many (most) of them neglect to visit doctors (Figure 1(m)).

Content recommended collaboration items:

Ranking	docTitle	score
1	Requires close monitoring	37.66
2	Erectile dysfunction	16.74
3	No hospitalization or surgi	12.79
4	No post-treatment staging	9.97
5	Active Surveillance	8.14
6	Accurate staging	8.14
7	PSA levels predict recurrer	2.63
8	Proven solution	1.86

Page 1 of 1 10 View 1 -

Figure 3. The output of the content recommender for user “John”.

 ISBN: 978-1-61208-436-7

Jane does not share the enthusiasm for the radical prostatectomy alternative as, due to surgery, a certain amount of risk is involved (Figure 1(n) – "against" position). Apart from this, an erectile dysfunction is expected at the level of 30-50% in 5 years (Figure 1(o)). Joe, who has just joined the collaboration, adds a new collaboration item to support Jane's opinion on the surgery risk involved stating that the danger of a patient's reaction to the anesthesia drugs should be taken into account (Figure 1(p)).

As he is new to the collaboration process and his rating profile is relatively poor, using the hybrid recommendation mechanism to get recommendations invokes the content-based recommender algorithm (due to the collaborative recommender's failure to provide accurate results), which returns a list of collaborative items with similar content to the collaborative item he has just added (Figure 3). Exploiting the recommendations of the content-based recommender, he is now in a better position to contribute to the ongoing collaboration process.

VI. DISCUSSION AND CONCLUSION

The proposed approach builds on the content and structure of an evolving argumentative collaboration, as well as on the rating profiles of similar users, to provide hybrid recommendations in platforms following the IBIS model of argumentation. Concerning the collaborative filtering based recommender, the major benefit of the proposed approach lies in the fact that, in order to provide accurate collaborative recommendations, it exploits the structure of the associated argument trees to estimate the value of the user's inferences on each argumentation item. The application of the proposed hybrid recommender has been demonstrated in the case of Dicode, a collaboration and decision making support platform. Following a similar method of integration, the proposed approach may be easily integrated to any IBIS-like system.

Dicode collaboration support services (including the recommendation support engine presented in this paper) have been thoroughly evaluated in three real-life contexts (clinico-genomic research, medical decision making, and opinion mining from Web 2.0 data). Generally speaking, the feedback received was positive, which clearly points out that the overall approach is promising (a comprehensive description of the evaluation process appears in [18]). Evaluators indicated that our approach reduces the data-intensiveness and overall complexity of real-life collaboration and decision making settings to a manageable level, thus permitting stakeholders to be more productive and concentrate on creative activities [19].

Future work directions include the application of the proposed hybrid recommender in diverse real-life collaborative settings. Through such efforts, we first plan to fine-tune our approach as far as the various parameters of the open-source libraries exploited are concerned. In addition, since there is a number of alternatives to integrate the results of the proposed recommenders (for instance, by using an appropriate switching criterion, the effective selection of which remains an open research issue), more tests have to be conducted in order to decide about the most appropriate method (per collaborative setting) to combine the outputs of the content-based and the collaborative filtering based recommenders.

REFERENCES

[1] M.J. Eppler and J. Mengis, "The Concept of Information Overload: A Review of Literature from Organization Science, Accounting, Marketing, MIS, and Related Disciplines". The Information Society, Vol. 20, pp. 325–344, 2004.

[2] F. Ricci, L. Rokach, and B. Shapira, "Introduction to Recommender Systems Handbook". In: Ricci, F., Rokach, L., Shapira, B. and Kantor, P. (Eds), Recommender Systems Handbook, Springer, pp. 1-35, 2011.

[3] M. Aljukhadar, S. Senecal, and C.E. Daoust, "Information Overload and Usage of Recommendations", Proceedings of the ACM RecSys 2010 Workshop on User-Centric Evaluation of Recommender Systems and Their Interfaces (UCERSTI), Barcelona, Spain, pp. 26-33, Sep 30, 2010.

[4] D. Almazro, G. Shahatah, L. Albdulkarim, M. Kherees, R. Martinez, and W. Nzoukou, "A survey paper on recommender systems", arXiv:1006.5278, 2010.

[5] C. Chesnevar, A.G. Maguitman, and M.P. Gonzalez, "Empowering Recommendation Technologies Through Argumentation", In: I. Rahwan and G. Simari (Eds.), Argumentation in AI, Springer, pp. 403-422, 2009.

[6] P. Zhang, J. Sun, and H. Chen, "Frame-based argumentation for group decision task generation and identification". Decision Support Systems, vol. 39, pp.643-659, 2005.

[7] C.I. Chesñevar, A.G. Maguitman, and G. Simari, "A first Approach to Argument-Based Recommender Systems based on Defeasible Logic Programing", Proc. of the 10th Intl. Workshop on Non Monotonic Reasoning (NMR 2004), Whistler, Canada, pp. 109-117, June 2004.

[8] C.I. Chesñevar and A.G. Maguitman, "Arguenet: An argument-based recommender system for solving web search queries". In: Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference. vol. 1, pp. 282-287, 2004.

[9] W. Kunz and H. Rittel, "Issues as Elements of Information Systems", Working paper No. 131, Studiengruppe für Systemforschung, Heidelberg, Germany, July 1970.

[10] G. Adomavicius and A. Tuzhilin, "Toward the next generation of recommender systems: a survey of the state-of-the-art and possible extensions", IEEE Transactions on Knowledge and Data Engineering, vol.17, no.6, pp. 734-749, June 2005.

[11] R. Burke, "Hybrid web recommender systems". In: P. Brusilovsky, A. Kobsa, and W. Nejdl, (Eds.), The adaptive web, Lecture Notes In Computer Science, Vol. 4321. Springer-Verlag, Berlin, Heidelberg, pp. 377-408, 2007.

[12] S. Gündüz-Öğüdücü, Web Page Recommendation Models: Theory and Algorithms, Morgan & Claypool, 2010.

[13] P. Lops, M. de Gemmis, and G. Semeraro, "Content-based recommender systems: State of the art and trends", Recommender Systems Handbook, pp. 73-105, Springer, 2011.

[14] C. Cayrol and M.C. Lagasquie-Schiex, "Graduality in Argumentation", Journal of Artificial Intelligence Research, vol. 23, pp. 245-297, 2005.

[15] Y. Hu, Y. Koren, and C. Volinsky, "Collaborative Filtering for Implicit Feedback Datasets", Proc. of ICDM '08 - The 8th IEEE International Conference on Data Mining, pp. 263-272, Dec. 2008.

[16] N. Karacapilidis, M. Tzagarakis, N. Karousos, G. Gkotsis, V. Kallistros, S. Christodoulou, C. Mettouris, and D. Nousia, "Tackling cognitively-complex collaboration with CoPe_it!", International Journal of Web-Based Learning and Teaching Technologies, vol. 4, no 3, pp. 22-38, 2009.

[17] N. Karacapilidis (Ed.), Mastering Data-Intensive Collaboration and Decision Making: Cutting-edge research and practical applications in the Dicode project, Studies in Big Data Series, Vol. 5, Springer, 2014.

[18] G. Tsiliki, N. Karacapilidis, S. Christodoulou, and M. Tzagarakis, "Collaborative Mining and Interpretation of Large-Scale Data for Biomedical Research Insights", PLoS ONE 9(9): e108600, 2014, doi: 10.1371/journal.pone.0108600.

[19] N. Karacapilidis, M. Tzagarakis, and S. Christodoulou, "On a meaningful exploitation of machine and human reasoning to tackle data-intensive decision making". Intelligent Decision Technologies Journal, vol. 7, no 3, pp. 225-236, 2013.