ON THE EXPLORATION AND EXPLOITATION OF STRUCTURAL SIMILARITIES IN ARGUMENTATIVE DISCOURSES

George Gkotsis and Nikos Karacapilidis
Industrial Management and Information Systems Lab
MEAD, University of Patras, Patras, Greece

Keywords: Argumentation, Argument sequence, Similarity, Mining.

Abstract: Motivated by the fact that contemporary argumentation systems provide low or no support with regards to argument and information processing, this paper presents a generic computational model that is able to identify and assess structural similarities in argumentative discourses. Focusing on the structure of such discourses, we sketch representative scenarios where the proposed model can be applied at a wide range of argumentation systems in order to define, elaborate and mine meaningful argumentation patterns. We argue that the proposed model is of considerable contribution to both theoretical and practical aspects of argumentation.

1 INTRODUCTION

When engaged in argumentative discourses users have to exploit their own cognitive abilities and sentiment. Reality shows that individuals react and understand differently upon the same input, i.e. it is very likely for two users to process the same information in a different manner. For instance, people tend to overlook information that undermines their viewpoints (*confirmation bias* phenomenon (Kuhn, 1991)) and prefer supportive information compared to opposing one (*selective exposure* phenomenon ((Jonas et al., 2001)); some users - newcomers very likely - might prefer to get an abstract representation of the discourse taking place by viewing participants and their corresponding contributions (Rees, 1995) or may want to filter out old contributions, or focus on a specific part of the dialogue; others might prefer to analytically examine every aspect of the dialogue, reconstruct argumentative discourse (Eemeren et al., 1993), identify inconsistencies between peers and attempt defeating standing arguments.

Today's argumentation support systems have to overcome a series of complex technological and social challenges (Shum et al., 2008). From a technological perspective, major advances already taken place concern information exchange (Reed and Rowe, 2004), interoperability among applications and data referring mechanisms (Karacapilidis et al., 2009); however, the corresponding web 2.0 compliant applications do not eliminate the information overload problem. At the same time, even though humans have been extensively engaged in argumentative dialogues, online participation in a discourse is a modern phenomenon. Recent studies reveal facts like social loafing and attrition (e.g. (Johnson, 2001)). Generally speaking, research has a long way to fight the low acceptance of argumentation support systems.

The problem of equipping an argumentation support system with mechanisms that ease information processing has been addressed by various techniques, which can be classified in two complementary categories. On the one hand, we find attempts to identify specific attributes in arguments to be then exploited by inference mechanisms (e.g. in agent-based decision support systems). Classifying arguments according to the above attributes has proven to be a feasible and effective way to compute features, such as *acceptability* (correct or wrong), *ambiguity* (agreement vs. disagreement) and *consistency* (consistent conclusions) (Caminada and Amgoud, 2007). The above can be considered as an attempt to model arguments on a *microstructure* level, since the focus in this case is on the argument *per se* and not on the complete discourse or argumentation structure.

On the other hand, on a *macrostructure* level, one of the key elements that characterize an argumentation support system is the argumentation model adopted. Although each model serves a specific purpose, they all share some common characteristics:

Gkotsis G. and Karacapilidis N.
ON THE EXPLORATION AND EXPLOITATION OF STRUCTURAL SIMILARITIES IN ARGUMENTATIVE DISCOURSES.
DOI: 10.5220/0002845201370143
In *Proceedings of the 6th International Conference on Web Information Systems and Technology (WEBIST 2010)*, page
ISBN: 978-989-674-025-2

they define the *type* of the argumentation objects and the *actions* that can be performed on them[1]. Thus, once the system has defined its model, every discourse is codified and validated against it. Afterward, the interest is mainly focused on supporting well-defined reasoning practices through the definition and modeling of specific activities. From a computational perspective, depending on the formality level adopted, the systems under consideration attempt to ease decision support through reasoning algorithms, assist the identification of logic violations, or simply provide a convenient repository for accessing, amending and publishing knowledge.

We view argumentation as an act of social practice, where discussion accommodates the sharing of different opinions, blending of various ideas, and knowledge building. According to (Weinberger and Fischer, 2006), the participation in an argumentative dialogue is divided into several phases: early phases include the externalization and elicitation of opinions, ideas and arguments; intermediate ones include consensus building; last phases are described as integration-oriented and conflict oriented consensus building. Focusing on the intermediate and late phases, we argue that contemporary systems pay little attention and provide low support to them. We believe that argumentation support systems should remedy user disorientation through the provision of mechanisms that allow easy and meaningful evaluation of arguments and argumentation sequences.

An argumentation sequence may be described as set of arguments interrelated in a specific way. Argumentation sequences have proven to be useful abstraction mechanisms that allow systems to elucidate and simplify argumentation dialogues. Until now, the main contribution of argumentation sequences is that they can be used to aggregate small pieces of dialogue to entities of higher meaning and stimulate specific behavioral patterns in a dialogue (Baker, 1999). Following a specific argumentation formalism, and without loss of generality, this paper describes a graph-like model of argumentation dialogues. This representation allows us to quantify node structure similarity in an argumentation context, thus enabling us to consider various aspects of argumentation at both the micro and macro structure levels. The ultimate gain of the proposed model resides in the ability to represent diverse argumentation sequences in a generic, flexible and accurate way. Thus, argumentation sequences can be indexed and handled appropriately. Mining of unnoticed sequences can also be achieved.

In the following sections, we describe the proposed model, sketch representative scenarios of its usage, and conclude by discussing related work and the contribution of our approach.

[1]Note that the pioneer argumentation model by Toulmin focuses on describing the discourse in terms of claims, warrants and grounds; participants merely constitute an interesting entity (Toulmin, 1958).

2 PROPOSED APPROACH

We first model an argumentative discussion as a discussion graph. For this graph, we assume the following properties:

- The graph is a *connected weighted undirected tree*;
- The *issue* or topic of the discussion is handled as the *root* of the graph;
- Any *alternative* is linked to the root of the graph through a neutral type edge. Neutral relationships have weight equal to 0.
- Any *argument* is a node[2] connected to another argument or alternative. Any argument participating in a relationship expresses exclusively either agreement or disagreement. Agreement has value of 1, where disagreement has value -1.

Vertex refinement query, introduced in (Hay et al., 2008), is a mathematical model that allows the identification of similarities in undirected, unweighted graphs and has been applied in social networks. In this paper, we are going to modify the vertex refinement query process in order to apply it to a graph with the above characteristics (undirected, weighted discussion graphs).

2.1 Node Value Definition

For any node x - except the root of the graph - we define a property called *value*: V_x where $V_{min} \leq V_x \leq V_{max}$. V_{min} and V_{max} are the minimum and maximum values of this property correspondingly. This value may be computed by one or more argument attributes. For example, this may be the average rating for a system that supports item rating, the expertise of its author or any other scalable attribute any system might introduce. Selecting an appropriate attribute is a crucial step. In the simplest scenario, where no specific attribute is chosen, the value for each node may be 1.

[2]For the rest of this section, the terms *node* and *vertex* will be treated interchangeably.

2.2 Node Similarity for Degree Value 0

We define a node similarity function between arguments x and y for degree value of 0[3], noted as $Sim(x_0, y_0)$, which is the complement of the normalized to the scale between 0 and 1 Euclidean difference, as:

$$Sim(x_0, y_0) = 1 - \frac{|V_x - V_y|}{V_{max} - V_{min}}$$

2.3 Vertex Refinement Query

Vertex refinement query for degree i and node x, noted as $H_i(x)$, allows us to identify vertices structural similarity by exploring the nearby vertices and is expressed as an iterative recursive query:

$$H_i(x) = \{\pm H_{i-1}(z_1), \pm H_{i-1}(z_2), \ldots, \pm H_{i-1}(z_m)\}$$

where $H_0(x) = \{\pm V_x\}$, $z_1, z_2, \ldots, z_m$ are nodes adjacent to x, and i expresses the degree of the query. The sign $\pm$ is inserted in front of every vertex refinement query and is either positive, if the adjacent node is supporting node x, or negative if it opposes node x. If the adjacent node is the root of the tree (discussion issue), the node is ignored.

2.4 Computing Sequence Similarity

In order to compute the sequence similarity between nodes for a given degree, a function is introduced that takes as input the result of the vertex refinement queries. More specifically, the sequence similarity function between 2 nodes takes as input the result of the 2 corresponding vertex refinement queries (in fact, the result of a vertex refinement query is a set of signed nodes) and returns a real number between 0 and 1 that expresses the sequence similarity. More precisely, sequence similarity is calculated through the matching of nodes between the first and the second set. Through this matching, the sequence similarity is expressed as the sum of the similarities between pairs of nodes and such that this sum is maximized.

Even though the thorough analysis of the complexity of this problem exceeds the purpose of this paper, we are going to present a solution for it, by describing how this problem can be reduced to a well known problem cited in graph theory, as follows:

We create a bipartite graph G(A, B, E) where A and B are the vertices of first and second set of the signed nodes, respectively. Let E be the set of edges connecting *every* vertex from A to B. The weight of every edge is computed following definition in 2.2 and expresses the node similarity for degree value of 0, including its $\pm$ sign. This bipartite graph is complete (every vertex from set A is connected to every vertex from set B, since we can always compute the similarity for every pair of nodes). For this bipartite graph, we seek to find a matching $M \subseteq E$ among vertices A, B such that the sum of the weight for edges $\in M$ is maximized. This is an old problem and is known as ”maximum weight matching in complete bipartite graphs”. Hopcroft-Karp algorithm (Hopcroft and Karp, 1973) runs in $O(\sqrt{n}W) = O(n^{5/2})$, where n is the number of vertices and W is the total number of edges of graph G (in complete bipartite graphs $W = n^2$).

We define sequence similarity for degree $i > 0$ between arguments x, y as:

$$Sim(x_i, y_i) = \frac{\sum_{i=1}^{i=|M|} weight(e)}{\max(|A|, |B|)} \text{for every } e \in M,$$

where A, B are the vertex refinement queries of x, y for the selected degree i and M is the maximum weight matching described above. Furthermore, the result is divided by the max set size in order to be normalized to the scale between 0 and 1.

2.5 Vertex Equivalence

We define two nodes x and y as *equivalent*, with respect to degree of value i, denoted as $x \equiv_i y$, iff:

$Sim\big(H_{i(x)}, H_i(y)\big) > threshold_i$, where $threshold_i$ is a constant (user-defined) real number between 0 and 1.

2.6 Vertex Identity

We define two nodes x and y as *identical* with respect to degree of value i, iff:

$Sim\big(H_{i(x)}, H_i(y)\big) > threshold_t$ for every $t \in [0, i]$, where $threshold_t$ is also a constant user-defined real number between 0 and 1 (which does not necessarily have the same value with equivalence threshold).

Note that equivalence relationship is weaker than identity, since there might be several inequalities for lower values of i.

2.7 Example

Figure 1 illustrates a simple argumentation graph. P_1, P_2, P_3, P_4 are arguments, A_1, A_2 are alternatives and I is the issue of the discussion. Dashed line depicts opposition, while solid line depicts agreement. To demonstrate our model, we are going to search whether repetitions of argumentation sequences appear, i.e. we will search for structural similarity of

[3] Degree value of 0 actually reflects the case where we are only interested about the two nodes under consideration, without paying attention to their adjacent nodes. In the general case, as the degree value raises, so does the scope of the similarity function.

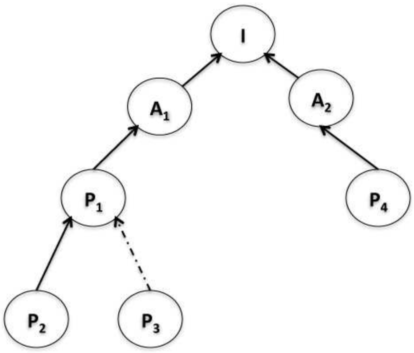

Figure 1: A simple argumentation graph.

each argument of the discussion against all other arguments.

In order to keep the example simple, we assign to each node a value of 1. It is clear that for our example: $Sim(x_0, y_0) = 1$, for every x, y.

Furthermore, since we decided to assign the same value for each node, we raised the threshold for both equivalence and identity operations to 1. Nevertheless, if we had used real data - where the value for every node rarely should be expected to have value of 1-, we should experiment with lower threshold values.

We obtain the following (see Table 1):

There is no node that appears structurally identical to another if the degree is higher than 1. However, for degree value of 1, nodes P_2 and P_4 are identical since they are the only nodes with their adjacent nodes connected in the same way.

Table 1: Vertex refinement queries for the graph of Figure 1.

	H_1	H_2
P_1	$\{A_1, P_2, -P3\}$	$\{P_1, P_1, -P1\}$
P_2	$\{P_1\}$	$\{P_2, -P_3, A_2\}$
P_3	$\{-P_1\}$	$\{-P_2, P_3, -A_1\}$
P_4	$\{A_2\}$	$\{P_4\}$

3 ARGUMENTATION SEQUENCE IDENTIFICATION

3.1 Case 1: Finding Specific Argumentation Sequences

In this case, we assume that a user wants to search in one or more discussion graphs for arguments interrelated in some meaningful way. This includes cases where a set of arguments connected to each other constitutes a structure of higher meaning, such as *defeated arguments*, *well supported arguments*, *undercutters* or *ill supported arguments* . Thus, a structure of higher meaning refers to an aggregation of arguments interrelated in such a way that these arguments can be regarded as a meaningful concept from an argumentation point of view. In that way, these structures can assist the user analyze the outcome of a discussion in a more convenient way, so that ultimately he will be able to identify expressed behaviors, like consensus, disputes and refutes.

We assume a system supporting community rating for every argument. Without loss of generality, let this rating be a real number between 0 and 1.

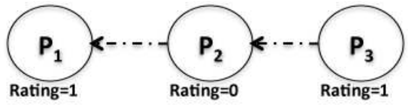

Figure 2: A defeated argument.

In the argumentation sequence shown in Figure 2, dashed lines depict disagreement and rating is annotated for every argument. This argumentation sequence describes that a user tried to defeat argument P_1 through argument P_2, P_2 received low rating, while one more argument (P_3) defeated argument P_2 and received high rating. In summary, we can claim that this argumentation sequence is a sequence where argument P_2 is *defeated*.

We are going to use the above argumentation sequence as a training sequence. A real-world dialogue may contain approximate values, like the argumentation sequence shown in Figure 3.

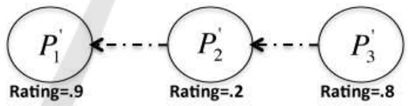

Figure 3: A real-world argumentation sequence.

Following our model, we are going to calculate the similarity for degree values 0 and 1 of arguments P_1', P_2', and P_3' against argument P_2. It is:

- For argument P_1' :

$$Sim(P_1', P_2)_{degree=0} = 1 - \frac{0.9 - 0}{1 - 0} = 0.1$$

$$\begin{aligned} Sim(P_1', P_2)_{degree=1} &= sim\big(\{-P_2'\}, \{-P_1, -P_3\}\big) \\ &= sim\big(\{-0.2\}, \{-1, -1\}\big) \\ &= 0.1 \end{aligned}$$

- For argument P_2' :

$$Sim(P_2', P_2)_{degree=0} = 1 - \frac{0.2 - 0}{1 - 0} = 0.8$$

$$
\begin{aligned}
Sim(P'_2,P_2)_{degree=1} &= \\
&= sim\big(\{-P'_1,-P'_3\},\{-P_1,-P_3\}\big) \\
&= sim\big(\{-0.9,-0.8\},\{-1,-1\}\big) \\
&= 0.85
\end{aligned}
$$

- For argument P'_3 :

$$Sim(P'_3,P_2)_{degree=0} = 1 - \frac{0.8-0}{1-0} = 0.2$$

$$
\begin{aligned}
Sim(P'_3,P_2)_{degree=1} &= sim\big(\{-P'_2\},\{-P_1,-P_3\}\big) \\
&= sim\big(\{-0.2\},\{-1,-1\}\big) \\
&= 0.1
\end{aligned}
$$

The above reveal that P'_2 is identical to P_2, given that our threshold is equal to or higher than 0.8, while arguments P'_1 and P'_3 have value lower than 0.2 for every degree.

This argumentation sequence can be subject of further exploitation: one may choose to filter out every argumentation sequence marked as defeated argument; more generally speaking argumentation sequences can be highlighted as indicators for further processing. A similar rule may be applied for ”well-supported arguments”. Furthermore, due to the fact that the proposed model matches similar and not identical structures, it is clear that training the model with an additional small set of sequences can cover even the most complex structures.

It is worth noting that our example was based on the definition of the rating attribute. Further attributes (or even a set of relevant attributes) can be used to identify similar sequences. The only prerequisite is to define the node similarity function for degree of value 0. Representative attributes that can be also taken into account are:

- *Authorship.* Our model keeps track of argument creators. The function for calculating similarity for degree of value 0 returns 0 for same names and 1 for different names. A more sophisticated way to find relevant sequences given that our system supports user profiling and clustering could be to define the similarity function so that it returns the similarity value between user profiles. Using authorship enables to allow searching for argumentation sequences where ”user x defeats user y while user z defeats user x”, or where ”people with opposite/similar profiles dispute in this community”.
- *Creation Date.* In this case, the similarity function returns the time distance between two different arguments. Using creation date, we can find sequences where ”an argument has been asserted after time instance x and was defeated after a period of time y”.
- *Element Type.* Even though most argumentation support systems are based on the IBIS-like model, it is very common that their model can use different terminology. Our reasoning model can be configured to take into account the associated argument types. Thus, the similarity function for degree value 0 will return 1 if the element type is the desired one or 0 if it is not. For example, it is possible to define an argumentation sequence as inconsistency if an element of type ”idea” is opposed to an argument, or detect arguments that defy first-order logic. Exploiting the element type in similarity measurements can be a very useful technique to identify dialogue inconsistencies or setting ad-hoc new rules.

3.2 Case 2: Extracting Unnoticed Sequences

In this case, we assume that many discussions have already taken place. Similarly to case-based reasoning (more specifically, analogy-based reasoning (Aamodt and Plaza, 1994)), where the system tries to re-use existing similar argumentation situations, our model will be used to *mine* unnoticed argumentation sequences. In order to achieve the above, the model is going to search for structural similarity of *every* argument in any dialogue against *any other* argument. From a technological point of view, this is a time-consuming process, which can be handled by modern computer technology (e.g. cloud computing, where splitting the process into tasks can be parallelized and run periodically).

The attempt to identify structural similarities of every argument against another argument in a big corpus of argumentative dialogues is expected to return a large number of similarities. In order to present valuable information to the user, a ranking of these results is needed. The *criteria* for ranking similar argumentation sequences are the degree of the similarity, the number of occurrences and some user preferences. More specifically, argumentation sequences with higher degree and/or more occurrences will have higher ranking and will therefore be promoted, since it is rational to expect from them to carry more valuable information. User preferences may include one or more attributes (see previous case) that the user wants to take into account while mining for argumentation sequences.

Mining argumentation sequences in an argumentation corpus is a procedure that can be especially valuable for newcomers and users with low participation activity. Users with low experience in an online community have difficulties in getting socially attached to already existing users. One of the primary reasons for this is the fact that once they enter the community, they are called to catch up with discourses that have already occurred and analyze user behavior. This cultivates a sense of being left behind, since the online community already carries a lot of collective experience, knowledge and social interaction. Our model assists users identify relationships that have come up already in this community, since it highlights sequences that have appeared elsewhere.

4 RELATED WORK

On the broad field of collaboration support systems, several models that quantify participation and interaction have been already proposed. For instance, OCAF (Avouris et al., 2002) follows a generic diagrammatic collaborative model and introduces terms like density and degree of participation as metrics that quantify group participaton. CAF (Fesakis et al., 2003) is a model that can be applied in synchronous communication, providing teachers with a mechanism for tracking information about the collaboration. Kaleidoscope (Dimitrakopoulou et al., 2006) attempts to quantify several aspects of learning activities, through the exploitation of social networking analysis techniques like the measurement of activity level, network density and centrality in order to provide social awareness about actors. Even though the above systems are inspired by common ideas with our model, they differentiate in the fact that their approach does not take advantage of the *structural dimension* in information flow, and more specifically, of the argumentation sequences.

On the field of argumentation, Belvedere (Suthers et al., 1995) adopts a diagrammatic visual representation with special notation, to assist students identify the overall structure of arguments. Araucaria introduces the notion of argumentation scheme, to refer to ”stereotypical patterns of nondeductive reasoning” (Reed and Rowe, 2004) that allow the description of argumentation components. Moreover, both Compendium (Selvin et al., 2001) and CoPe_It! (Karacapilidis et al., 2009) allow users to organize sets of arguments in higher structures (maps and adornments, respectively). In summary, several argumentation support systems have attended the need of providing abstractions and mechanisms for an argumentative discourse; nevertheless, none has presented mechanisms that allow the computational processing of argument aggregations.

5 DISCUSSION

We have presented a generic computational model, that is able to identify and assess structure similarities in argumentation-based discourses. Through the deduction of discourses as graphs, the model establishes a way to represent diverse and meaningful argumentation sequences in a generic, flexible and quantified way. In this section, we discuss the benefits of our approach, focusing on how the proposed model may affect the system performance from a user perspective.

As stated above, argumentation sequence is an abstraction that can aggregate discourse parts and be exploited in several ways. First, a set of argumentation sequences can be defined to represent meaningful (and well-known) reasoning patterns. The above set of sequences can act as a training dataset that will assist users during the analysis of a discourse and provide them with hints about parts of the discourse. This set of sequences can grow by users through the addition of new sequences found in a discourse or can be personalized according to specific user needs. In such a way, discourses can be represented in a more abstract way and their analysis is facilitated. Moreover, notification mechanisms can be integrated to help users keep track of changes in discourses. For example, an argumentation support system that adopts the proposed model can inform a user that his argument has been defeated, accepted, or even that a new alternative appears as a leading candidate. Finally, argumentation sequences can be exploited in queries such as ”show me sequences where users A, B, C attempt and fail to defeat user D’s contributions”.

We have also described how our model can be used to extract unnoticed argumentation sequences. Similarly to data mining, this functionality can result to the acquisition of useful information that in most cases would have passed unnoticed. In that way, states of passiveness and quiescence that are commonly met in online communities can be significantly eliminated. Moreover, our model can help discussion moderators or community leaders identify meaningful patterns of communication. For example, it can detect that one or more users have constant disputes with another group of users or that certain dialogues tend to develop in never-ending conflicts. In this case, community leaders can choose to get notified for such behaviors, so that they can take appropriate actions against them. Thus, they will be able to act preemptively in favor of the community.

6 CONCLUSIONS

Engaging in online argumentative discourses is a complex and challenging task. Amongst other requirements, users have to surpass social and technological barriers in order to process and evaluate information provided by their peers. We argue that contemporary argumentation support systems pay little attention to the above. Motivated by the fact that argumentation sequences can aggregate arguments into entities of higher meaning, we have presented a generic but flexible model that is capable of discovering and assessing similar argumentation sequences. We argue that this model is of considerable contribution to both theoretical and practical aspects of argumentation.

As a a note for further study it is worth investigating whether mining argumentation sequences may improve the performance of relevant features in an argumentation support system, like user profiling, rating, social network analysis and decision support algorithms.

REFERENCES

Aamodt, A. and Plaza, E. (1994). Case-based reasoning. *Proc. MLnet Summer School on Machine Learning and Knowledge Acquisition*, pages 1–58.

Avouris, N., Dimtracopoulou, A., Komis, V., and Fidas, C. (2002). OCAF: An object-oriented model of analysis of collaborative problem solving. *Computer Support for Collaboratie Learning: Foundations for A Cscl Community (cscl 2002 Proceedings)*, page 92.

Baker, M. (1999). Argumentation and constructive interaction. *Foundations of argumentative text processing*, pages 179–202.

Caminada, M. and Amgoud, L. (2007). On the evaluation of argumentation formalisms. *Artificial Intelligence*, 171(5-6):286–310.

Dimitrakopoulou, A., Petrou, A., Martinez, A., Marcos, J., Kollias, V., Jermann, P., Harrer, A., Dimitriadis, Y., and Bollen, L. (2006). State of the art of interaction analysis for Metacognitive Support & Diagnosis. *Interaction Analaysis (IA) JEIRP Deliverable D.31.1.1*.

Eemeren, F. H. V., Grootendorst, R., Jackson, S., and Jacobs, S. (1993). *Reconstructing Argumentative Discourse (Studies in Rhetoric and Communication)*. University of Alabama Press.

Fesakis, G., Petrou, A., and Dimitracopoulou, A. (2003). Collaboration activity function: an interaction analysis tool for computer supported collaborative learning activities. In *4th IEEE International Conference on Advanced Learning Technologies (ICALT 2004)*, pages 196–200.

Hay, M., Miklau, G., Jensen, D., Towsley, D., and Weis, P. (2008). Resisting structural re-identification in anonymized social networks. *Proceedings of the VLDB Endowment archive*, 1(1):102–114.

Hopcroft, J. and Karp, R. (1973). An $n^{5/2}$ Algorithm for Maximum Matchings in Bipartite Graphs. *SIAM Journal on Computing*, 2:225.

Johnson, C. (2001). A survey of current research on online communities of practice. *The internet and higher education*, 4(1):45–60.

Jonas, E., Schulz-Hardt, S., Frey, D., and Thelen, N. (2001). Confirmation bias in sequential information search after preliminary decisions: An expansion of dissonance theoretical research on selective exposure to information. *Journal of Personality and Social Psychology*, 80(4):557–571.

Karacapilidis, N., Tzagarakis, M., Karousos, N., Gkotsis, G., Kallistros, V., Christodoulou, S., Mettouris, C., and Nousia, N. (2009). Tackling cognitively-complex collaboration with cope_it! *International Journal of Web-Based Learning and Teaching Technologies*, 4(3):22–38.

Kuhn, D. (1991). *The skills of argument*. Cambridge Univ Pr.

Reed, C. and Rowe, G. (2004). Araucaria: Software for argument analysis, diagramming and representation. *International Journal of AI Tools*, 14(3-4):961–980.

Rees, M. (1995). Analysing and evaluating problem-solving discussions. *Argumentation*, 9(2):343–362.

Selvin, A., Buckingham Shum, S., Sierhuis, M., Conklin, J., Zimmermann, B., Palus, C., Drath, W., Horth, D., Domingue, J., Motta, E., et al. (2001). Compendium: Making meetings into knowledge events. In *Knowledge Technologies*, volume 2001. Citeseer.

Shum, B. et al. (2008). Cohere: Towards Web 2.0 Argumentation. In *Proceeding of the 2008 conference on Computational Models of Argument: Proceedings of COMMA 2008*, pages 97–108. IOS Press.

Suthers, D., Weiner, A., Connelly, J., and Paolucci, M. (1995). Belvedere: Engaging students in critical discussion of science and public policy issues. In *Proceedings of AI-Ed*, volume 95, pages 266–273. Citeseer.

Toulmin, S. (1958). *The uses of argument*. Cambridge University Press.

Weinberger, A. and Fischer, F. (2006). A framework to analyze argumentative knowledge construction in computer-supported collaborative learning. *Computers & Education*, 46(1):71–95.