Published in *Proceedings of ICCE 99—Seventh International Conference on Computers in Education*, Chiba, Japan, Vol. 1, 87–94. Amsterdam: IOS Press.

Using an Intelligent Agent to Improve Argumentation Skills over HyperNews

Ronggang Yu and **Yam San Chee**

School of computing, National University of Singapore
Lower Kent Ridge Road, Singapore 119260
65-8748090
yurongga@comp.nus.edu.sg cheeys@comp.nus.edu.sg

Abstract: We describe an intelligent agent that provides students with suitable online argumentation strategies and rhetorical methods in a computer-supported collaborative argumentation environment. Computer-based arguments contain many regular patterns that occur with high frequency. Further more, these regular patterns are often related with argumentation strategies. Armed with this knowledge, the intelligent agent first retrieves regular patterns from high quality computer-based arguments, then saves them into its corpus of regular patterns. Each pattern is assigned with several suitable argumentation strategies and rhetorical methods. When students write articles either to support or to argue against a viewpoint, the intelligent agent can provide immediate strategies by matching regular patterns in related arguments. If a pattern is matched in arguments, it will retrieve argumentation strategies and rhetorical methods that are associated with that pattern, and then present them to the students. We implemented our intelligent agent on the platform of HyperNews. The regular pattern corpus has been tested with many web-based arguments. Experimental results show that, using our approach, students can readily improve the quality of their arguments with the help of this intelligent agent.

Keywords: Computer-Supported Collaborative Argumentation, Pedagogical Agent, Artificial Intelligence and Networked Learning, Data Mining Technology in Education.

1. Introduction

The Internet is useful because it is a wide area communications network. This global network has seen exponential growth and frenzied development in the last few years, and usage of Computer-Supported Collaborative Argumentation (CSCA) tools such as bulletin board systems, mailing lists, USENET, and HyperNews have likewise increased. Many instructors and researchers are trying to employ these tools for supporting collaborative learning activities. They either embed CSCA tools into their learning and collaborative working environment [2, 6, 10] or build CSCA tools with new features for the convenience of arguers [5, 8, 12]. These trends emphasize on designing new computational support for mediating and structuring argumentation, using either textual or graphical representations.

There are many systems that focus on structuring discourse with the use of graphical interfaces to support computer-based argumentation. Euclid [14] provides a graphical representation language for generic argumentation; gIBIS [3] and JANUS-Argumentation [4] record the process of design in order to support and critique it in accordance with established methods in the design community. Suthers developed Belvedere [13], a prototype for teaching high school students collaborative reasoning and argumentation. Belvedere provides a diagramming tool for constructing representations of the logical and rhetorical relations within a debate. Buckingham Shum’s research focuses on notations and tools to support argumentation as a means of capturing organizational knowledge and design rationale [1]. The general intention of their work has been to help students analyze the relationship of the components of arguments and then organize them into high quality ones.

In this paper, we propose an original approach to designing and implementing an intelligent agent that helps students to improve their web-based argumentation skills; i.e., to improve the quality of their web-based arguments. First, the intelligent agent retrieves regular patterns from high-quality computer-based arguments, and then saves these patterns into a pattern corpus. Each pattern is assigned several suitable argumentation strategies and rhetorical methods by system administrator or privileged users who are proficient in web-based argumentation. When students write articles supporting or arguing against a viewpoint, the intelligent agent can provide immediate argumentation strategies by matching regular patterns in related arguments. If a pattern is matched in arguments, it will retrieve argumentation strategies and rhetorical methods that are associated with this pattern from the corpus, and present them to the students. As HyperNews [7] is one of the most popular CSCA tools over the World Wide Web and the source code is available, we implemented the intelligent agent on this platform. All argumentation strategies and rhetorical methods presented to students are retrieved from the regular pattern corpus. The efficiency and intelligence of the agent are dramatically depended on the quality and quantity of this corpus. Currently the corpus has about 400 regular patterns with their argumentation strategies and rhetorical methods. We tested the frequency and suitability of regular patterns with many web-based arguments. Experimental results show that, using our approach, students can significantly improve the quality of their arguments.

In the next section, we explain what we mean by a regular pattern. In section 3, the agent architecture is presented and components are described. In section 4, we show system services, including the intelligent agent that students can use, during the process of web-based argumentation. In section 5, we present our test results of regular patterns stored in the corpus, and discuss the results. Finally, we state our conclusion and make some comments on the system.

2. Regular pattern

A *word pattern* in arguments refers to a collection of words. A *regular pattern* in arguments is a word pattern that appears with high frequency and reflects the relationship within or among sentences. Regular patterns are the skeleton of web-based arguments, and the intelligent agent can use these regular patterns to provide students with argumentation strategies and rhetorical methods during the process of web-based argumentation. For example, the word pattern "if (1), one would expect (2), yet (3)" can be regarded as a regular pattern for the following two reasons. First, it is a generic word pattern without specific knowledge; thus this pattern can occur in other arguments with high frequency. Second, this pattern can be interpreted or can be used as an argumentation strategy. Consider the following strategy: "Is there any situation where (2) can be reached not because (1) but because of other reasons?". This strategy can be used by the students. They can propose some reasons that conclude (2) to show that the above refutation is weak and ineloquent. However, words such as "if", "yet", etc cannot be regarded as regular patterns because they lack the essential structure of an argument. Some general argumentation strategies can also be associated with regular patterns. If we have a regular pattern, "for example", the strategy for refutation can be "Are examples sufficient in number to support the generalization?".

All regular patterns with their interpretations in a database are referred to as *a regular pattern corpus*. Currently, we have collected about 400 regular patterns in our corpus from several famous web-based argumentation environments on the Internet, such as alt.philosophy.debate, CEDA-L [11], EDEBATE [9]. alt.philosophy.debate is a newsgroup containing arguments on all kinds of topics. CEDA-L is a college debate mailing list archive, and EDEBATE is the successor of CEDA-L. Many of the participants of these web-based argumentation forums are excellent debaters or have a strong interest in web-based argumentation.

In collecting regular patterns, we were very concerned with their quality. There is a tradeoff between the hit rate of regular patterns and their quality. For example, the word pattern "of … of …" occurs in arguments with high frequency, but this word pattern cannot be used as a regular pattern because there are no viable argumentation strategies related to this word pattern. On the other hand, although the word pattern "if … has the property of … one would expect … yet … thus …" has potentially high value as an argumentation strategy, it also cannot practically be regarded as a regular pattern because this word pattern does not occur regularly. Thus, a regular pattern should both be interpretable and occur with high frequency in arguments. Each regular pattern has two attributes; one is frequency and the other is the suitability. frequency is used to evaluate how many times this pattern may be reused in other arguments. suitability is used to evaluate whether it is appropriate or suitable when this pattern is used in other arguments.

We use an example to explain the approach. The following extract is a section of argument made by one student.

> **The case for using animal in research**
>
> The use of animals in medical research has many practical benefits to make or test new drugs. Animal research has enabled researchers to develop treatments for many diseases, such as heart disease and depression. It would not have been possible to develop vaccines for diseases like smallpox and polio without animal research. Every drug anyone takes today was tried first on animals.
>
> Future medical research is dependent on the use of animals. Which is more important: the life of a rat or that of a three-year-old child?
>
> Medical research is also an excellent way of using unwanted animals. Last year, over twelve million animals had to be killed in animal shelters because nobody wanted then as pets.

Another student tries to argue for this proposition. During the supporting procedure, if she has no idea on how to expand her argument, she can get immediate online assistance from an intelligent agent. The intelligent agent analyzes above argument and finds the regular pattern: "(1) has (2) benefits to (3) " in the first sentence of the argument. It retrieves this pattern's argumentation strategy: "you can enumerate more benefits to show that (1) can help to (3)" from the regular pattern corpus. Now the student can get an assistant message from the intelligent agent, "you can enumerate more benefits to show that the use of animals in medical research can help to make or test new drugs". The intelligent agent presents this strategy to the student with related rhetorical methods: using facts, and using example. Now she can begin to think about how to expand her argument with this argumentation strategy and the rhetorical methods provided by the intelligent agent. She can get more advice if there are more patterns were matched in above argument.

3. Agent architecture

We implemented our intelligent agent using CGI/PERL programming, W3-mSQL script, and Javascript. The interface was implemented using JavaScript embedded in HTML files. Connection to database was implemented using W3-mSQL script which is also embedded in HTML files. The other components of the agent were implemented by CGI/PERL script. In Figure 1, we present the components of the agent and their relationships.

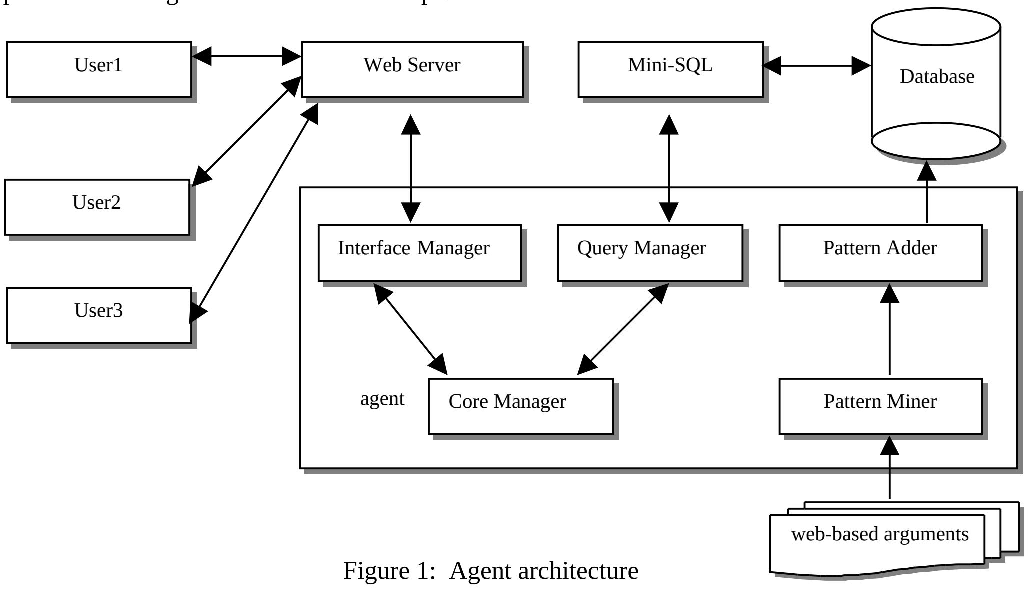


Figure 1: Agent architecture

The *Pattern Miner* is the component that finds regular patterns from the computer-supported collaborative argumentation environment. Web-based arguments are all HTML files. The Pattern Miner will first remove HTML tags from those files. Useless words such as "a", "the", "of" will also be removed, as they contribute nothing to the regular pattern corpus. Each word and word pattern in those arguments will be marked by attributes of file number, sentence number, and word number. The word "expect" may be marked as (expect, 7, 3, 10) if this word appears in the seventh article and in the third sentence of this article, and in the tenth word of this sentence. With the same methods, the word pattern "if … expect … yet" can be marked as (if:expect:yet, 7, 3, 2) if the word "if" appears in the second word of the sentence that was described above. With these attributes, the Pattern Miner can easily mine regular patterns from web-based arguments. First, the Pattern Miner gets two-word patterns by combining all two word combinations in one sentence. Three-word patterns can also be obtained by concatenating two-word patterns. Four-word pattern can be obtained by concatenating two-word patterns and three-word patterns. The Pattern Miner also records the frequency of these patterns as one of their attribute. Finally, It is the system administrator's responsibility to decide whether the word patterns presented are regular patterns that ought to be saved into the corpus.

The *Pattern Adder* is the component that helps the system administrator to add regular patterns into the corpus. It is a password protected HTML/W3-mSQL program which only the system administrator can access. Every regular pattern can have several argumentation strategies, and every argumentation strategy can have several rhetorical methods related. When the system administrator adds a regular pattern, the Pattern Adder will check the syntax of the regular pattern and it's argumentation strategies. If the syntax is correct, the Pattern Adder will save this regular pattern into the corpus. The Pattern Adder also provides many other facilities to the system administrator. The system administrator can delete or update regular patterns in the corpus if the regular patterns saved are obsolete or need to be revised. The Pattern Adder also provides several methods to search for regular patterns in the corpus, such as search by title, search by phrase in pattern, search by phrase in strategy, and search by rhetorical methods. Each pattern has an attribute named suitability to evaluate the quality of this pattern. When adding regular patterns into the corpus, the system administrator can assign a value to this attribute. With the Pattern Adder, the system administrator can easily maintain the quality and quantity of the corpus.

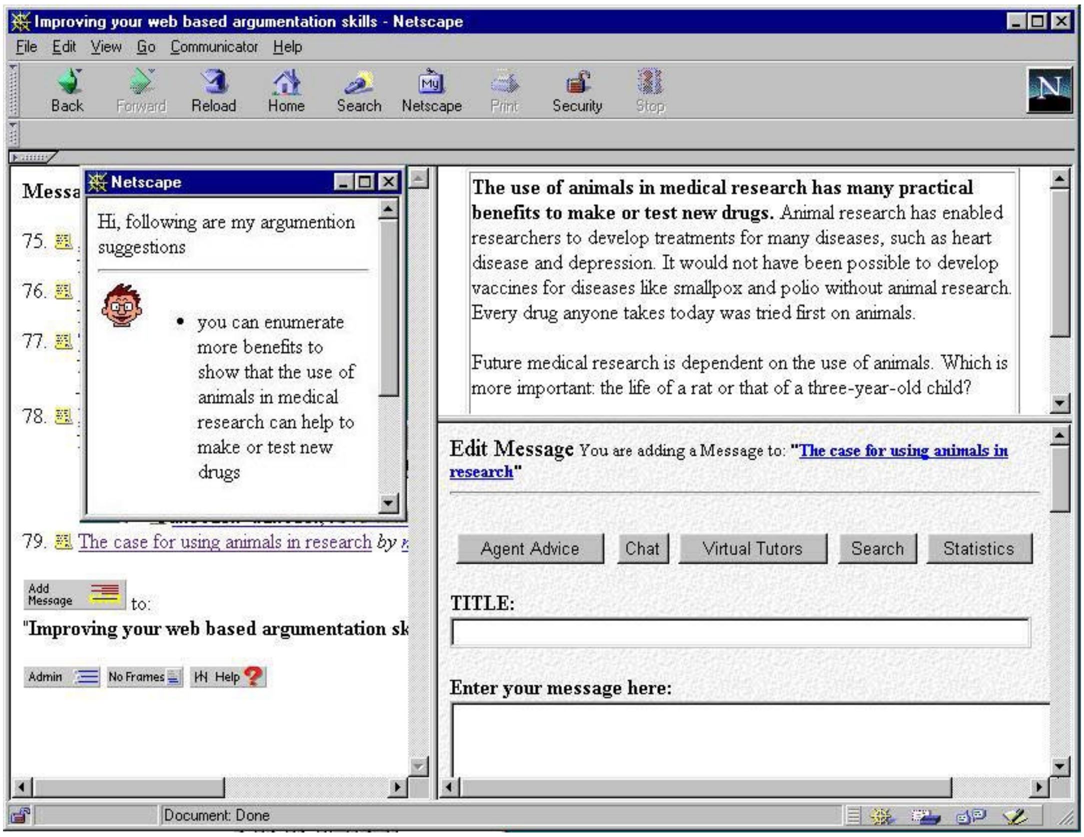
The *Interface Manager* is the component that interacts with students. First, the Interface Manager receives students' arguments and requests. It also obtains environment variables and sends them to the Core Manager described below. Once the Core Manager handles the request and sends back the handling results, the Interface Manager will make the sentence that the regular pattern was matched with bold, and a small window will be opened to show the related argumentation strategies and rhetorical methods.

The *Query Manager* is the component that handles mSQL queries. The Query Manager is connected to a mSQL server Once it receives query scripts from the Core Manager, it will process the query and search whether the word patterns are regular patterns that were saved in the corpus. Query results will be sent by the Query Manager to the Core Manager. A query script includes the word pattern and the sentence that has this word pattern. If the word pattern is found to be a regular pattern, the query results include this pattern and it's argumentation strategies and rhetorical methods.

The *Core Manager* is the most important component of the agent architecture. It receives arguments from the Interface Manager. It then calls the Query Manager to check whether the word patterns are regular patterns in the corpus. First, the Core Manager splits arguments into sentences. In a sentence, the Core Manager tries to combine all the possible word patterns, and then sends these word patterns to the Query Manager to check whether they are regular patterns in the corpus. If they are, the Core Manager will sends back these patterns and their argumentation strategies and rhetorical methods, with the match location to the Interface Manager. If they are not, the Core Manager will try the next sentence, until all sentences are analysed.

4. System service

A number of functions to facilitate the web-based argumentation including the intelligent agent have been built on top of the HyperNews system developed by Daniel LaLiberti. These system services are shown in Figure 2.



Hi, following are my argumention suggestions

- you can enumerate more benefits to show that the use of animals in medical research can help to make or test new drugs

79. The case for using animals in research by

"Improving your web based argumentation sk

The use of animals in medical research has many practical benefits to make or test new drugs. Animal research has enabled researchers to develop treatments for many diseases, such as heart disease and depression. It would not have been possible to develop vaccines for diseases like smallpox and polio without animal research. Every drug anyone takes today was tried first on animals.

Future medical research is dependent on the use of animals. Which is more important: the life of a rat or that of a three-year-old child?

Edit Message You are adding a Message to: "**The case for using animals in research**"

Figure 2. Agent providing advice in web-based argumentation

Agent Advice can be used by students to get immediate online argumentation advice. If students make propositions either supporting or arguing against a viewpoint, but have no idea on how to expand their arguments, they can press the button named “Agent Advice” for assistance. The intelligent agent will provide argumentation strategies and rhetorical methods suitable for proposition or refutation to students. Meanwhile, the sentence that contains a regular pattern will be bolded. Students can continue to get advice by clicking this button until no more regular patterns are found.

Chat is another facility that students can use to communicate with each other. Students can enter a chat room and conduct real-time conversations with other people. The Chat facility provides a discussion environment that simulates face-to-face talking.

Virtual Tutor is the function provided for students to find virtual tutors in a computer-supported collaborative argumentation environment. Virtual tutors can assess the arguments that were written by their virtual students by commenting on arguments. Virtual students improve their web-based argumentation skills by reflecting on these comments, while virtual tutors can also improve their argumentation skills by scaffolding students in their argumentation process [15]. One tutor can have several virtual students, and one student can also have several tutors, but the virtual tutor-student relationship can only be established if each recognizes the other.

Search is the function that students can use to retrieve appropriate arguments that they are interested in. As students may be confused by arguments and have difficulty to find articles that they are interested in in a large scale web-based argumentation environment, this is a necessary and useful tool for students to derive greater benefit from web-based argumentation. There are four kinds of search methods: by title, by author, by date, and by phrase in arguments. If students are interested in all arguments by an author, they can use the method of *search by author*. If students know the title of an argument, they can quickly find the article using the method of *search by title* without having to scroll over the entire screen. If students are interested in articles that were posted during a specific period, they can use the method of *search by date*. We also support the method of *search by phrase* in arguments. If students are interested in articles talking about a specific topic, they can input the topic phrase or name and quickly retrieve all arguments that have this topic phrase or name.

Statistics is the function that students can use to see how many arguments they have posted. When students use HyperNews, they can press the icon named "Statistics" to calculate the number of articles that they have posted and the current article's sentence number and word number. This function helps students to be aware of the status of their contributions to web-based argumentation in a community of learners.

5. Empirical evaluation

In this paper, we proposed a new method of providing students with immediate argumentation strategies and rhetorical methods through an intelligent agent, during the process of web-based argumentation. The intelligent agent first retrieves regular patterns from web-based arguments, then saves them into the corpus. When students participate in the web-based argumentation and click the button named "Agent Advice", it will try to match regular patterns in referred arguments. If a regular pattern is matched, the related argumentation strategies and rhetorical methods will be retrieved from the corpus, then presented to students. The efficiency and intelligence of the agent depends critically on the quality and quantity of the corpus.

Currently, we have collected about 400 regular patterns in the corpus. We tested this corpus from two perspectives: one is the frequency of regular patterns and the other is the suitability of the argumentation strategies.

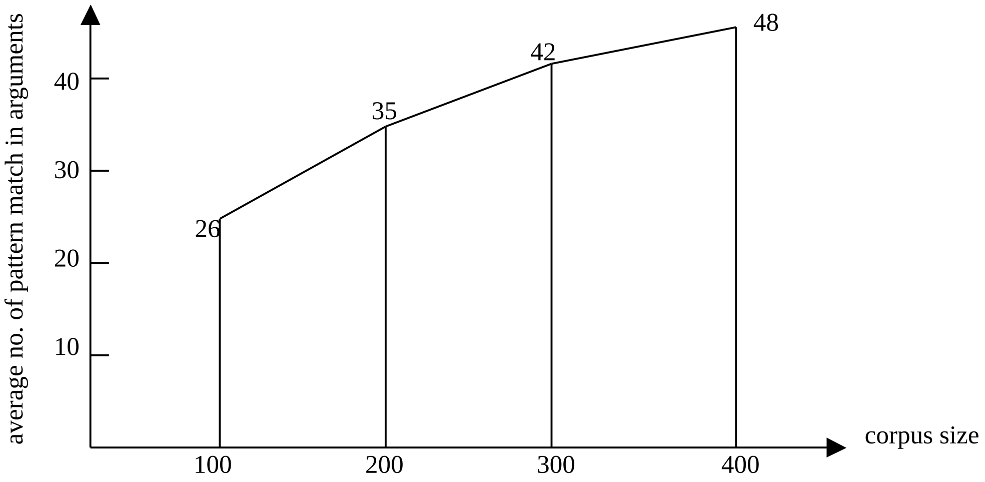


Figure 3. Test result of frequency of regular patterns

Figure 3 shows the test result of the frequency of regular patterns in the corpus on 30 arguments from CEDA-L Jan, 1996. Each argumentation strategy has two attributes. one is the frequency of its regular pattern, which was given by the Regular Pattern Adder, the other is the suitability of this strategy, which was given by the system administrator. We combine these two attributes as the value of this strategy, and order the first, the second, the third, and the fourth hundred patterns by this value from highest to lowest. In this experiment, we select them only by the attribute of suitability without considering the attribute of frequency, as the work of ranking argumentation strateties has not yet fully finished. When corpus size was 100, i.e. the first hundred regular patterns in the corpus were selected, there were 26 regular patterns matched in every argument. When the size of corpus increased, the average no. of matches on average argument also increased. From Figure 3, we see that the larger the corpus size, the slower the increase rate in matching. This means that newly added regular patterns have a smaller matching frequency than previous patterns.

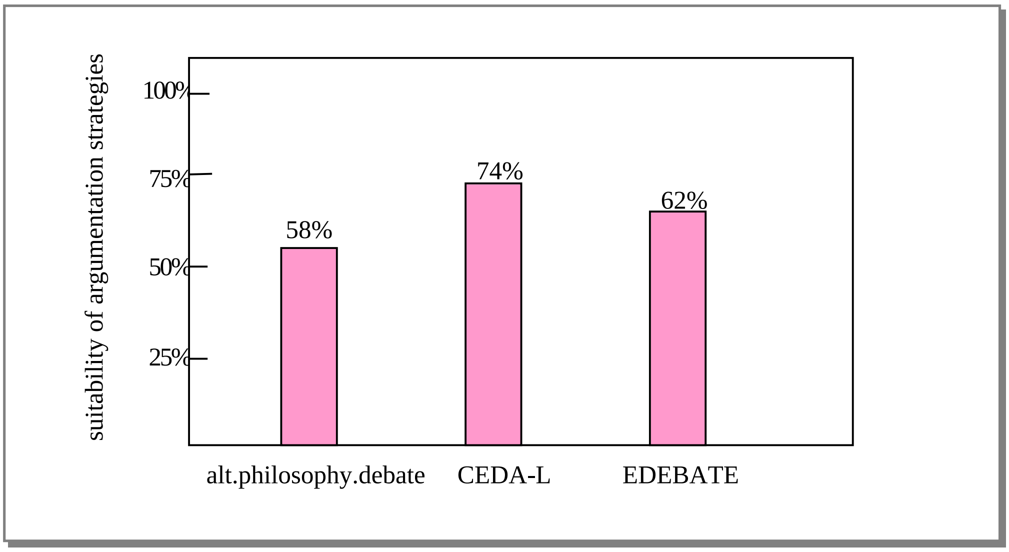


Figure 4. Test result of suitability of argumentation strategies

Figure 4 shows the test result of suitability of the argumentation strategies that were attached with regular patterns. We tested the suitability on arguments from alt.philosophy.debate, CEDA-L, and EDEBATE respectively. We reposted the arguments on HyperNews, and used the intelligent agent to obtain advice for these arguments. From Figure 4, we see that about 65% of argumentation strategies on average are suitable or can be used by students to improve the quality of their arguments. Other strategies are irrelevant, and students can discard them. The suitability of argumentation strategies on CEDA-L is a little higher than the other two sources. This result is from the fact that we obtain most regular patterns from CEDA-L.

5. Conclusion and Future Work

In this paper, we reported a method to automatically provide students with argumentation strategies and rhetorical methods during the web-based argumentation. A regular pattern corpus has been set up and the interface with W3-mSQL database and Web Server has been built. The intelligence of the agent is achieved by finding the regular patterns in arguments, then retrieving the strategies and rhetorical methods from the corpus, and providing immediate online assistance to students. From the results of our experiment, we find that the intelligent agent can effectively scaffold when they are trying to improve their web-based argumentation skills.

6. References

[1] Buckingham Shum, S. & Hammond, H. (1994). Argumentation-based design rationale: What use at what cost? *International Journal of Human-Computer Studies*, 40(4), 603-652.

[2] Buckingham Shum, S., Tamara, S. and Diana, L. (1996). On the future of journals: Digital publishing and argumentation. *In Proceedings of HCI'96.* http://kmi.open.ac.uk/sbs.

[3] Conklin, J. & Begeman, M. L. (1987). gIBIS: A hypertext tool for team design deliberation. *In proceedings of Hypertext'87*, pp. 247-252.

[4] Fischer, G., McCall, R., and Morch, A. (1989). JANUS: Integrating hypertext with a knowledge-based design environment. *In Proceedings of Hypertext'89*, pp.105-117.

[5] Gaspar, R. F. & Thompson, T. D. (1995). Current trends in distance education. *Journal of Interactive Instruction Development,* pp. 21-27.

[6] Hurwitz, R. & Mallery, J. C. (1995). The open meeting: A web-based system for conferencing and collaboration. *In Proceedings of the 4th International Conference on the World Wide Web*. http://www.ai.mit.edu/people/rhhu/rhhu.html.

[7] LaLiberte, D. & Wooley, D. (1997). Presentation features of text-based conferencing systems. *WWW Computer-Mediated Communication Magazine, May 1997,* http://www.hypernews.org.

[8] Ou, K. L., Chang, C. K., and Chen, G. D. (1998). Web-based asynchronous discussion system. *In Proceedings of the 8th International Conference on Computer in Education*, Vol 1, 108-117.

[9] Ryder, J. (1997). Team topic debating in America, http://list.uvm.edu/archives/edebate.html.

[10] Schlageter, G., Buhrmann, P., Laskowski, F. and Mittrach, S. (1997). Virtual university: a new generation of net-based educational systems. *In Proceedings of the 7th International Conference on Computers in Education.* http://www.vsnet.ch/stubrig.

[11] Stanton, J. (1997). CEDA mailing list archive, http://www.cs.jhu.edu/~jonathan/debate/ceda-l/archive.

[12] Suthers, D., Toth, E. E. and Weiner, A. (1997). An integrated approach to implementing collaborative inquiry in the classroom. *In Proceedings of Computer Supported Collaborative Learning'97.* http://lilt.ics.hawaii.edu/suthers/.

[13] 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 the 7th World Conference of Artificial Intelligence in Education*. pp. 266-273.

[14] Smolensky, P., Fox, B., King. R., and Lewis, C., (1987). Computer-aided reasoned discourse, or how to argue with a computer. In R. Guindon (Ed.), *Cognitive Science and Its Applications for Human-ComputerIinteraction*, pp. 109-162.

[15] Chan, T-W. (1996). Learning companion systems, social learning systems, and the global social learning club. *Journal of Artificial Intelligence in Education*, Vol. 7, No.2, 125-129.