User Interfaces with Semi-Formal Representations: a Study of Designing Argumentation Structures

Timothy Chklovski
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292 USA
+1 310 448 8763
timc@isi.edu

Varun Ratnakar
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292 USA
+1 310 822 1511
varunr@isi.edu

Yolanda Gil
Information Sciences Institute
University of Southern California
4676 Admiralty Way
Marina del Rey, CA 90292 USA
+1 310 822 1511
gil@isi.edu

ABSTRACT
When designing mixed-initiative systems, full formalization of all potentially relevant knowledge may not be cost-effective or practical. This paper motivates the need for semi-formal representations that combine machine-processable structures with free text statements, and discusses the need to design them in a way that makes the free text more amenable to automated structuring and processing. Our work is done in the context of argumentation systems, and has explored a range of tradeoffs in combining informal free-text statements with formal connectors. The paper compares alternative argument representations which combine structured argument connectors with free text. We discuss merits of the systems based on a variety of analysis structures that we have collected from Web users to date.

Categories and Subject Descriptors
H.5.2 [**Information Interfaces and Presentation**]: User Interfaces: *Theory and methods, Natural language.* H.1.2 [**Models and Principles**]: User/Machine Systems: *Human information processing*

General Terms
Design, Experimentation, Human Factors, Languages.

Keywords
Semi-formal representations, natural language understanding, decision-making, argumentation, meaning decomposition

1. INTRODUCTION
Mixed-initiative systems present many challenges in terms of user interface design. The system must understand the task at hand, make suggestions in context, and present information in a format useful to the user. The more the system knows, the more helpful it can be. This would argue for systems in which all relevant knowledge about the task is fully formalized and all of it is therefore machine processable. However, formalizing all potentially necessary knowledge is challenging; it takes significant effort that may not have a clear payoff. Ideally, mixed-initiative systems would formally represent and reason with subsets of the problem that can be formalized, while leaving other parts of the problem to the humans and their more thorough understanding of the task. The information that is not formally represented will likely be in natural language, a format that humans are likely to prefer for its familiarity and expressiveness. In this sense, the knowledge about the task and its context will be in what we refer to as *semi-formal representations:* a combination of formal structures together with free text which is not formalized.

IUI'05, January 10–13, 2005, San Diego, California, USA

This paper explores the tradeoffs in designing these semi-formal representations so that the system can provide useful assistance while minimizing the user's effort in formalizing knowledge about the task at hand. Our work investigates these issues in the context of computer-assisted argumentation. There are several well-known systems that use a combination of free text statements anchored within an argument structure. gIBIS [12] focuses on capturing collaborative deliberations about design in the form of graphs containing text at their nodes. SEAS [18] enables users to revise previous analyses in light of new evidence, where an analysis is captured in a tree of relevant issues and sub-issues rated formally through an evidential reasoning system. Work on Belvedere [25], [26] allows users to switch between graph and matrix representations of an argument and focuses on the collaboration facilitated by these representations. ClaiMaker [8], [16] focuses on scientific debate, where scientists can express the positions and contributions in a publication through a combination of free text and structuring constructs. Compendium [23] focuses on facilitating meetings by capturing multiple perspectives on ill-structured problems. Our group has developed Trellis [14], [6], a system for structured argumentation on any topic where users progress from information sources to arguments that intermix free text and structured connectors. A better understanding of the tradeoffs in interplay between free-text and structured argument connectors would be beneficial in designing these and other decision support and mixed-initiative systems.

Ultimately, the more the system can automatically structure the free text portions of an argument or an analysis, the more automated reasoning and assistance will be possible. Alas, processing arbitrary text is still an open research problem, so the challenge is in designing appropriate semi-formal structures that can be further structured by the system automatically. In this sense, a better design of these semi-formal structures is one that 1) is amenable to incremental and automatic formalization through *natural language processing* techniques applied to the free text statements, and 2) supports *machine learning and clustering* algorithms to detect similarities and correspondences among arguments created by different users in different contexts. For example, arguments entered by various users indicating that computer A “is faster than,” “outruns,” “outperforms,” or “runs circles around” computer B could be clustered to indicate that there are four statements to support that “computer A is faster than computer B.”

Given a large collection of arguments and analyses of an issue by several individuals, processing the natural language of these analyses together with graph and matrix alignment techniques could support a new user in analyzing a similar issue. Specifically, the system could suggest additional aspects to consider (based on summing importance of issues across previous relevant arguments). The system could also outline where the opinion or decision of the user differs from those of previous users (possibly requiring further attention or additional justification to be convincing). If additional information on expertise of previous users is available, the system could further leverage natural language processing to identify (despite language variations) issues and aspects frequently missed by novice users but important to more expert users.

Ultimately, our goal is to support analysis and decision making on any topic, where the system has no initial formalization of the problem but is able to apply natural language and clustering techniques as suggested above to uncover structure in the arguments. Consider for example a user trying to decide which digital camera to buy, or how to plan a kid-friendly beach vacation. Much of this information is on the Web in free text form, yet users must structure some portions of it to make decisions. A variety of Web sites capture similar kinds of analyses for product reviews, ratings, and comparisons, including Epinions [2], CNet Reviews [1], the Internet Movie Database [3], and Amazon.com. Currently, the analysis at these Web sites is not formally structured. In some cases, the entire analysis is free text; in other, structure of the analysis is hard-coded within the design of the Web pages of that particular site and is specific to the nature of the objects being compared. The contents of the arguments themselves is far from being machine processable. If more structured representations were available, approaches such as [22] for reasoning with multiple opinions and [8], [27] for visualizing decisions and analysis structures could be applied. Our goal is to develop tools that support users in analyzing a topic of interest and making decisions. Possible kinds of support include helping locate relevant sources of data and information, formulate alterative hypotheses or positions and systematically explore the alternatives, select among the identified alternatives, and capture the rationale for given position or decision.

In looking for ways to handle mapping across diverse free text statements found in argument structures, we drew on lessons from our previous work studying canonicalization of more than 100,000 free text statements contributed by volunteers over the Web [9], [10]. In that work, we used natural language processing techniques to detect similarities between statements and enable analogical reasoning over the statements. The analogical reasoning served as an engine for further collection of knowledge on additional topics, in the form of free text statements. We found that we could sufficiently canonicalize knowledge to detect multiple similarities across it and pose well-motivated knowledge acquisition questions. We also observed that this method of collection resulted in free text being used in more uniformly across topics, thus being easier to process.

This paper starts by describing and comparing three approaches to designing semi-formal argument structures and their implementation in three systems: Rich (original) Trellis, Tree Trellis, and Table Trellis. We discuss the rationale of the design of each system, and what we have learned from data we collected from Web users with these systems. We also describe the issues entailed in natural language processing to structure the arguments.

2. TRELLIS

2.1 Rich Trellis

Rich Trellis [14] was originally developed as an interactive tool that helps users annotate the rationale for their decisions, hypotheses, and opinions as they analyze information from various sources. In creating an analysis, Rich Trellis allows a mixture of arbitrary free text with structured argumentation connectors. Examples of connectors are “is elaborated by”, “is supported by”, and “stands though contradicted by.” Figure 1a shows an example of a portion of an argument. In the example, the underlined statements are free text. Each argument organizes the issues considered hierarchically, and each argument is grounded in the sources consulted by the user during the analysis.

Clustering algorithms can be used to uncover regularities across analyses even in these loosely-structured arguments from Rich Trellis. In prior work, we showed how to automatically derive the level of trust that a community of users has in given sources based on the use of the source in the arguments [13], and how to use these trust ratings in suggesting sources for future arguments.

Bringing to bear a range of natural language processing tools to structure the free-text statements helps to sensibly cluster statements and thus helps to better support users. We have worked with techniques including part-of-speech tagging [7], WordNet exceptions database for synonyms and word conjugation [20], morphology and stemming [21], and parsers [24], [17].

Some useful structure can be extracted with these tools, since Rich Trellis statements are generally concise. On the continuum between reliance on full free text and only allowing structured input, Rich Trellis represents a solution where free text is one sentence or sentence fragment, and structure is introduced via connectors and hierarchical arrangement of the argument components.

However, trying to map across arguments created in Rich Trellis turns out to be hard for three reasons.

First, Rich Trellis allows the same argument to be organized in different ways which gives users a lot of flexibility but results in completely different structures that are hard to map automatically. For example, Figures 1a and 1b show two different ways the same portion of an argument can be represented in Rich Trellis.

Second, while Rich Trellis provides a rich vocabulary of connectors to reflect possible semantic relationships between the free text components, Rich Trellis relies on users to utilize structuring connectors consistently, which can be difficult across many contexts in which the connectors are to be used. Third, a variety of considerations could be combined with AND and OR connectors, which at times diffuses the central flow of the argument with more ancillary points. For example, Figure 1b shows a conjunctive statement in which the argument relies more

Macintosh is more usable than Windows

Macintosh is more usable than Windows **is supported by** Macintosh platform has a more stable OS

Macintosh platform has a more stable OS **stands though contradicted by** Windows aims to surpass other platforms in security and stability

Macintosh is more usable than Windows **is supported by** Macintosh, as compared to Windows, has a friendlier UI

Figure 1a. A portion of an argument structure expressed in Rich Trellis. The structuring connectors are shown in bold.

Macintosh is more usable than Windows

Macintosh is more usable than Windows **is supported by** Macintosh platform has a more stable OS **AND** Macintosh, as compared to Windows, has a friendlier UI

Macintosh platform has a more stable OS **stands though contradicted by** Windows aims to surpass other platforms in security and stability

Figure 1b. Another way to express the same analysis in Rich Trellis.

Macintosh is more usable than Windows

pro: Macintosh platform has a more stable OS

con: Windows aims to surpass other platforms in security and stability

pro: Macintosh, as compared to Windows, has a friendlier UI

Figure 2. A sample of an argument structure as it would be expressed in Tree Trellis, using the example shown in Figures 1a and 1b. Only "pro" and "con" connectors are allowed.

heavily on the first conjunct. To address these obstacles to using Rich Trellis as a representation over which assistance with analysis and argumentation is rendered, we have developed Tree Trellis, described in the next section.

2.2 Tree Trellis

The key differentiating features of Tree Trellis are its simplified argumentation structure which supports only the most general structuring connectors. First, conjuncts such as the AND in Figure 1b are no longer allowed, addressing the issue of it being too easy to structure the same argument in different ways. This change comes at the expense of some expressiveness. This change also

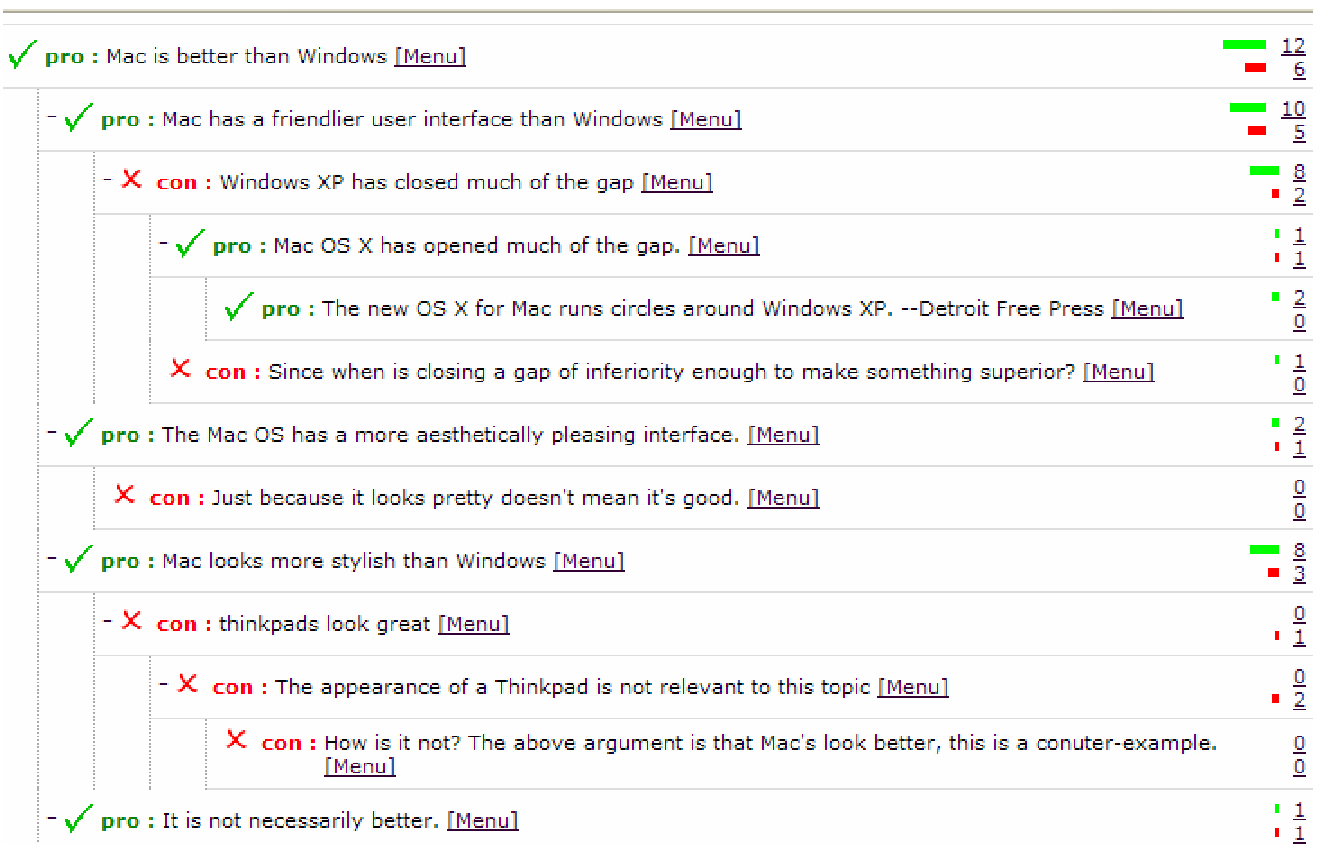

Figure 3. An excerpt of an analysis in Tree Trellis. The system leverages the simplicity of the argument connectors. The hyperlinked numbers on the right allow other users to specify agreement/disagreement in fully structured form.

simplifies the overall structure, improving ability to further elaborate the argument without restructuring it. Second, the rich vocabulary of connectors of Rich Trellis is reduced to two connectors: “pro” or “con.” The simplicity of these connectors helps consistency of use across users and arguments. Third, eliminating clauses which are joined by AND or OR addresses the issue of ancillary points being mixed with more central ones. Tree Trellis makes it clear which points engender more support and analysis. Finally, both entry and presentation are streamlined, by eliminating repetition of statements at lower levels (as “Macintosh is more usable than Windows” is repeated within each of the argument portions shown in Figures 1a and 1b). Figure 2 shows an excerpt of an argument rendered in Tree Trellis. The system leverages the simplicity of the argument connectors to encode them using (color) icons as well as text. Pro or con justifications can be added inline by elaborating on any level in the tree. Figure 3 provides a screenshot of Tree Trellis.

As compared to Rich Trellis, Tree Trellis gave away ground in terms of machine-processable structure. When users were allowed to enter arbitrary free text to convey a whole statement, and were not given additional constraints about the system, the users sometimes resorted to very implicit argumentation acts. For example, in analyzing the proposition “Quebec should separate from the rest of Canada”, one user has stated

> “Quebec needs more international presence to promote its culture and values.”

The argument is that one alternative, separation would achieve a certain result, “increased international presence” for Quebec, and that this result is desirable, because it would lead to another desirable effect, namely letting Quebec “promote its culture and values.” Elaborate natural language processing is required to fully understand the implied thrust of this argument.

A salient point about arguments collected in Tree Trellis is how inherent comparisons were to the analyses and arguments collected. We have examined the types of comparisons which came up in Tree Trellis arguments. We summarize the resulting observations to underscore the pervasiveness of comparisons in analyses and the diversity of ways in which comparisons can be phrased in free text [11].

Comparisons can be classified by whether they are comparing objects or events. When comparing objects, the comparison may be on an explicitly stated criterion. For example, the statement “laptops of company X are sturdier than laptops of company Y” compares laptops by two makers along the dimension of sturdiness, with company X’s laptops being better along that dimension. The free text is syntactically different if two objects are compared via some action (“Centrino laptops *run longer without recharging than* non-Centrino laptops”). Even within a given comparison type, the stronger entity (in some dimension) can be indicated in a variety of ways. For example, the above statement could also have been phrased as “non-Centrino laptops *do not last as long without recharging as* Centrino laptops”. In analyses about selection of music, we encountered statements such as “The music group X is the greatest rock band of the 80’s”, which are also (implicitly) comparing the music group X to the set of other rock bands of the 80’s. Also, the criterion of comparison is implicit in this case (“greatest”). Further argumentation may drill down to state that music group Y had the greatest drummer, or that music group Z had sold the most records. Whether the criterion is explicitly mentioned affects the exact phrasing of the free text. The situation with comparing events is similar, although the syntactic structure of the free text is different (e.g., “to book tickets in advance *can cost less than* booking them at the last minute”).

Based on the above observations about ubiquity and diversity of

Best Computer (-) (*) (x)

Type >> (x)	<< Design >> (x) (T)	<< Costs >> (x) (T)	<< Troubleshooting >> (x) (T)	<< Business >> (x) (T)	<< Performance >> (x) (T)	<< User Experience >> (x) (T)	<< Security (x) (T)	(|)
Apple	excellent	moderate	excellent	excellent	excellent	excellent	excellent	(x)
PC	medium	moderate	moderate	moderate	excellent	poor	moderate	(x)
								(+)

Business (-) (*) (x)

Computer Type >> (x)	<< Company Profitability >> (x)	<< Compatibility >> (x) (T)	<< Software Quality >> (x)	<< Market Share (x)	(|)
Mac	Link , Link: always pushes industry forward	excellent because apple controls hardware	high, only best titles come to mac	low, stable, not relevant: Link	(x)
PC	intense price competition, many competitors	not very good because of numerous hardware vendors	medium, lots of crap	majority, volatile between manufacturers	(x)
					(+)

Software Compatibility (-) (*) (x)

Costs (-) (*) (x)

Company >> (x)	<< Productivity >> (x)	<< Maintenance >> (x)	<< Software >> (x)	<< Hardware (x)	(|)
Apple	high, due to high quality of user experience	good - fewer necessary software upgrades, little maintenance due to reliability	same prices as PC for commercial, many quality apps "free" from apple	desktops and displays are more costly than average, laptops have best price performance in market	(x)
Dell	moderate due to reliability and user experience	moderate due to low reliability and frequent software updates	less "free" quality apps	comparable to Apple, not the cheapest PC	(x)
Other	moderate due to reliability and user experience	moderate due to low reliability and frequent software updates	some have software bundles, but generally expensive -- software can cost more than hardware	can be extremely cheap	(x)
					(+)

Figure 4. Excerpt from the “Macintosh vs. Windows” argument created in Table Trellis. Table Trellis allows representation of nested features by linking a given feature to a sub-table. The text “link” in the second table is a hyperlink to a source of evidence on the Web. The “software compatibility” table has been collapsed to show more of the high level argument.

comparisons, we decided to investigate centering arguments not around sentence long free text statements and connectors, but around the frequent in arguments comparison features (dimensions of comparison).

2.3 Table Trellis

In structuring argumentation around features and their values, it became clear that a tabular (matrix) representation may be well suited to capture such a structure [13], [5], [15] (ch. 8), where rows are the alternatives being compared, the columns are the comparison features, and the feature values are stated in the table cells. To investigate this alternative and feature based approach in greater depth, we have developed a system called Table Trellis. In this system, users express their argument in the form of a set of *nested alternative vs. feature matrices,* as shown in Figure 4. By presenting users with the tabular structure of alternatives vs. features, Table Trellis encourages framing the argument in terms of a number of feature/value pairs, arranged in a matrix. Moreover, Table Trellis allows ad-hoc structuring of the issue at hand, providing a way to state the features (dimensions of comparison) important in the analysis, without obscuring them by embedding them in longer natural language statements.

Further processing of this structure, (e.g. by relating it to an ontology in an approach similar to [6]), as well as comparison of features across arguments could treat these tables as machine-processable entities, and potentially allow the computer to render the following help: 1) suggest additional features, 2) populate values of table cells, and 3) suggest additional alternatives (rows) which may also be worth considering (with similarity judged on the table name and name of first column).

Table Trellis encourages structuring of analysis in terms of clearly identifiable features and their values, with each row representing an alternative to which the feature/value pair applies.

As shown in Figure 4, Table Trellis also supports nesting features via linking additional sub-tables to a given column header (feature). Something as simple as the price of a computer may require further elaboration (e.g., differences between vendors, hidden shipping costs, extended warranties, etc.). Nesting in Table Trellis allows both a summary view of the argument and ability to drill-down into the details.

Tables provide an interesting avenue for entering information in a semi-structured way. Table Trellis is aimed at supporting structured analysis and decision making on topics which are novel to the decision maker, with the decision affected by factors which may be subjective or difficult to identify. When analyzing a novel topic, which may range from a political debate about secession of Quebec to a purchasing decision, a user may benefit from decomposing the issue into clearly identifiable features [25]. Interestingly, today's comparison-shopping Web sites often resort to tables to present alternatives, presumably because tables concisely summarize alternatives and are easily understood by users. Unlike in Table Trellis, current comparison tables used by comparison-shopping sites use tables rigidly pre-configured to the task domain, showing such information as vendor, price, and quality rating.

Table Trellis emphasizes deciding on an issue by assessing tradeoffs among the (preferences indicated by) given features. By contrast, most argumentation systems emphasize evaluation of individual claims (features). Also, graph or tree based approaches typically lend themselves most naturally to expressing pairwise comparisons. Table Trellis may be particularly well suited to comparison of numerous alternatives, where a pairwise approach would prove unwieldy.

3. DISCUSSION OF PROSPECTS FOR AUTOMATED PROCESSING

In this section, we discuss the feasibility of processing free text components of Tree Trellis and Table Trellis, the two systems with the most data available. The observations made are based on small data samples and should be considered preliminary.

3.1 Tree Trellis

In Tree Trellis, we collected 517 statements about 83 arguments spanning a variety of topics from more than 60 registered and a number of anonymous contributors over the Web.

Some arguments centered around explicit comparisons, and in others the dimension of comparison was only implied. We discuss the explicit comparisons first and the implicit second. The explicit included arguments whether Windows or Macintosh is a better platform, whether Mozilla is better than Internet Explorer, whether cats are better pets than dogs, whether baseball is the greatest sport ever, and whether the Funk Brothers were one of the greatest bands ever.

When rendered in natural language, comparisons exhibit much syntactic variation. We investigated what lexico-syntactic templates could be use to map these comparisons stated in natural language to comparison criteria. While this analysis is not definitive, it provides the flavor of the challenges an automated approach would encounter.

Of 20 templates derived (this was done manually), no two were identical, or even very similar. Some of the simpler sample templates which probably could be reused in a slightly larger sample size are "N1 has better NP than N2" for "The Mac OS has better compatibility than Windows" and "N1 V much more NP than N2" for "Basketball requires much more athleticism than Baseball" (here N1 and N2 denote the objects being compared).

The analysis pointed to the following challenges. In some cases, separately matched terms would need to be combined to form the final description of the dimension (feature) on which the two objects are being compared. For example, to map to the feature "effort to install" or "installation effort" from "Installing the Mac OS takes much less effort than installing Windows," the term "installing" would have to be matched separately from "effort" and then the two would need to be combined.

In other cases, as in "dogs care about persons, cats only care about places," the dimension of comparison is really only a phrase fragment ("care about") and a system mapping this statement to its dimension of comparison would need to carry out complex processing to separate "persons" and "places".

Another observed phenomenon, presence of idiomatic expressions, limits coverage of lexico-syntactic patterns. For example, the use of the expression *takes advantage of* in "Macintosh takes advantage of the Internet more and better than Windows" gives the statement a different syntactic structure than it would have if *uses*, *leverages*, or *interoperates with* were used instead. Additional challenges include use of anaphora, such as "its OS X operating system…" and "Its OS X was…", where it the knowledge that the referent is Macintosh is assumed.

Also note the length of entries, which is indicative of syntactic complexity. The longest entry of those examined had 36 words and two sentences, the second longest had 30 words in a single sentence. Some of these problems can potentially be addressed with instructions to the user, and with warnings about issuing overly long, syntactically elaborate, or unparsable text.

The Windows vs. Macintosh argument also had a pair of practically redundant top-level points (namely, "On similar hardware, doing similar tasks, the Mac OS runs faster than Windows a majority of the time" and "The latest released Mac OS operating system runs faster overall than the latest released Windows operating system"), suggesting that variations of natural language in this case also obscured the semantics from the user.

3.1.1 *Interpreting non-comparison arguments*

There are arguments in which the comparison of alternatives is only implied rather than syntactically manifested. The body of such statements is typically dedicated to providing the reason why the (implicit) comparison is an important one. Recall the previously mentioned example of the argument "Quebec should separate from the rest of Canada," and the statement that "Quebec needs more international presence to promote its culture and values." The assertion is that one alternative, "separation," would achieve a certain result, "increased international presence" for Quebec. This result, the assertion continues, is desirable, because it would lead to another desirable effect, namely letting Quebec "promote its culture and values."

When alternatives are actions, indirect statements can argue that taking an action would be effective or ineffective, action will lead to a good or bad side effect, and so on. The specific argument is tied up with the semantics of what is being argued. For example, arguments may be based on the presence or lack of popular support for the separation, drawing on knowledge that separation of a Canadian province can be determined by a vote of its population.

Because implicit arguments are so varied and connected to the semantics of the specific point being argued, extracting the comparison criterion from them or automatically mapping across such points on anything but a keyword level seems extremely difficult. Providing only keyword, bag-of-words based assistance in such cases is also not likely to perform well, as it will miss the semantic gist of the argument.

A more tenable alternative may be to shift the task to the users, by asking them to carry out their analysis in a representation (such as Table Trellis) in which all comparisons are made explicit. While potentially challenging to the users, this approach actually would be promising in managing otherwise very challenging indirect arguments.

3.2 Table Trellis

As a way to get feedback and some preliminary data about Table Trellis and how its argument structures compare to Tree Trellis, we collected and analyzed arguments from several users on two specific topics. One topic was selecting one of three popular books about the Java programming language, and justifying the decision by using information at an ecommerce site (Amazon.com), including top two reviews about each book. The other topic is "Macintosh versus Windows," highlighting the differences in the argument structures used in Table Trellis and Tree Trellis on the same topic. This topic prompted the most detailed and structured analysis in both Tree Trellis and Table Trellis.

In Table Trellis, we emphasize processing of the feature names rather than feature values. While some feature values still contain up to 15 words, the feature names, with very few exceptions, are short phrases free of natural language issues such as anaphora, ambiguous or complex syntactic structure, and so on.

We briefly overview the syntactic forms that feature names in Table Trellis tend to assume. We state the strengths of Table Trellis and its potential to sidestep or address some challenges of processing natural language when processing Table Trellis features rather than Tree Trellis statements.

Consider the analysis about purchasing a Java book. Top-level features identified by one user include "contents and coverage" and "size". Another user mentioned "book title", "ease of use", "reviews", "publisher", and "examples" (which referred to the presence and quality of programming examples). In a few cases, subjects resorted to longer verb phrases to describe the columns, such as "covers graphics, databases, and networking". Occasionally, columns were designed with full questions such as "how well does book cover topic?" A more straightforward way to state the name for this feature would have been "quality of topic coverage."

Consider the analysis of the "Mac vs. Windows" topic in Table Trellis, an excerpt of which is shown in Figure 4. The argument was captured in 9 tables, with the top-level table summarizing eight additional nested tables. Tables not shown were of similar size and quality.

Of 29 feature names entered, 16 consisted of a single word (e.g., *performance*, *upgradeability*, *graphics*), 10 contained exactly two words (e.g., *customer support*, *operating system*, *online help*), only 2 contained three word (*ease of setup*, *ease of use*), and only one contained more than three words (comparing on *3rd party internet help*).

The alternatives column in Table Trellis contains the expressions which name the alternatives. The alternatives, which reference specific objects or actions also can be useful in retrieving related analyses.

To supplement Table Trellis data with data on a different subject, we located several camera reviews on the Web, identifying the subheadings. The subheadings represented the phrase that reviewers have used in evaluating a product, a task similar to comparing products in Table Trellis. The subheadings given in one review were: *battery life*, *usage and handling*, *image quality*. Another review listed *ease of use*, *durability*, *battery life*, *photo quality*, and *shutter lag*. In this example, *battery life* aligns verbatim, another aligns using a simple WordNet lookup (*photo* is a kind of *image* in WordNet), hence *photo quality* can be aligned with *image quality*. Finally, *ease of use* can be weakly matched with *usage* if *use* and *usage* can be aligned using morphological processing.

In all, Table Trellis largely sidesteps some of the discussed challenges in processing the free text of Tree Trellis because in Table Trellis, features are mostly noun or adjectival phrases of one or two words rather than syntactically more complex pieces of free text. Specifically, Table Trellis largely sidesteps issues of anaphora, idiomatic expressions, and implicit arguments.

In Table Trellis, mapping across arguments can further cope with possible noise in individual mappings by searching for alignment of multiple features. Multi-level structuring of features in Table Trellis may also be used to boost scores of multi-level matches.

4. CONCLUSIONS

This paper motivates the need for semi-formal representations that combine structuring connectors with free text to provide an interface which both allows users to express what they mean and is sufficiently structured to allow machine processing. We have explored tradeoffs in combining informal free-text statements with formal connectors in the Trellis family of systems. Preliminary evaluation suggests that the Table argument format offers the fewest complications to natural language processing.

Based on processing the natural language in the Trellis systems, we aim to extend Trellis to automatically assist in constructing an argument by identifying relevant prior arguments, and synthesizing from them the most relevant suggestions for the current one. In the Table Trellis format, the automated assistance would include using earlier analyses to suggest additional relevant evaluation factors (issue features), suggest additional alternatives to consider, and provide factor values supplied in previous analyses.

5. REFERENCES

[1] CNet Reviews – http://reviews.cnet.com/

[2] Epinons – http://www.epinions.com

[3] Internet Movie Database – http://www.imdb.com

[4] Slashdot – http://slashdot.org

[5] WebGrid III – http://tiger.cpsc.ucalgary.ca:1500/WebGrid/

[6] Blythe, J. and Gil, Y. Incremental Formalization of Document Annotations through Ontology-based Paraphrasing. *In Proceedings of the International World Wide Web Conference (WWW-2004)*

[7] Brill, E. Transformation-Based Error-Driven Learning and Natural Language Processing: A Case Study in Part of Speech Tagging. *Computational Linguistics*. December, 1995.

[8] Buckingham Shum, S. Uren, V., Li, G., Domingue, J. and Motta, E. Visualizing Internetworked Argumentation. In: P. Kirschner, S. Buckingham Shum, C. Carr (Eds), Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making, Springer, 2003.

[9] Chklovski, T. *Using Analogy to Acquire Commonsense Knowledge from Human Contributors*, MIT Artificial Intelligence Laboratory technical report AITR-2003-002, February 2003.

[10] Chklovski, T. LEARNER: A System for Acquiring Commonsense Knowledge by Analogy. In *Proceedings of Second International Conference on Knowledge Capture (K-CAP 2003)*. October 2003

[11] Chklovski, T., Gil, Y., Ratnakar, V. and Lee, J. TRELLIS: Supporting Decision Making via Argumentation in the Semantic Web. In *Proceedings of the 2nd International Semantic Web Conference (ISWC2003)*, short paper track.

[12] Conklin, J. and Begeman, M. gIBIS: A hypertext tool for exploratory policy discussion. ACM Transactions on Office Information Systems, 6(4):303--331, October 1988.

[13] Gaines B. and Shaw, M. Knowledge acquisition tools based on personal construct psychology. The Knowledge Engineering Review, 8(1), 1993.

[14] Gil, Y. and Ratnakar, V. TRELLIS: An Interactive Tool for Capturing Information Analysis and Decision Making. *In Proceedings of the 13th International Conference on Knowledge Engineering and Knowledge Management, Siguenza, Spain, October 1-4, 2002.*

[15] Heuer Jr, R. *Psychology of Intelligence Analysis*. Government Printing Office, 1999.

[16] Li, G., Uren, V., Motta, E., Buckingham Shum, S. and Domingue, J. ClaiMaker: Weaving a Semantic Web of Research Papers. In *Proceedings of the First International Semantic Web Conference on The Semantic Web* (ISWC 2002). 2002.

[17] Lin, D., Dependency-Based Evaluation of MINIPAR. In: *Proceedings of the Workshop on the Evaluation of Parsing Systems, First International Conference on Language Resources and Evaluation*. Granada, Spain. 1998.

[18] Lowrance, J., Harrison, I., and Rodriguez, A. Capturing Analytic Thought. *Proceeding of the First International Conference on Knowledge Capture (K-CAP 2001)*, 2001.

[19] Maybury, M. (Ed.), *Intelligent Multimedia Interfaces*, MIT Press, Cambrudge, MA, 1993.

[20] Miller, G. WordNet: An On-line Lexical Database. In *International Journal of Lexicography*, Vol.3, No.4, 1990.

[21] Porter, M. An Algorithm for Suffix Stripping Program. In J. Karen, and P. Willet, (Eds), Readings in Information Retrieval, San Francisco, Morgan Kaufmann, 1997.

[22] Richardson M. and Domingos, P. Building large knowledge bases by mass collaboration. In *Proceedings of Second International Conference on Knowledge Capture (K-CAP 2003)*. October 2003.

[23] Selvin, A., Buckingham Shum, S., Sierhuis, M., Conklin, J., Zimmermann, B., Palus, C., Drath, W., Horth, D., Domingue, J., Motta, E. and Li, G. Compendium: Making Meetings into Knowledge Events. In *Knowledge Technologies 2001*, March 4-7, Austin, TX, 2001

[24] Sleator, D. and Temperley, D. Parsing English with a Link Grammar. *Third International Workshop on Parsing Technologies*. 1993.

[25] Suthers, D., Connelly, J., Lesgold, A., Paolucci, M., Toth, E., Toth, J., & Weiner, A. Representational and Advisory Guidance for Students Learning Scientific Inquiry. In Forbus, K., & Feltovich, P. (Eds.). Smart machines in education: The coming revolution in educational technology. AAAI/MIT Press, 2001. pp. 7-35.

[26] Suthers, D., Representational Guidance for Collaborative Inquiry. In Andiressen, J.,Baker, M., and Suthers, D. (Eds.), *Confronting Cognitions: Arguing to Learn*, Kluwer, 2003.

[27] van Gelder, T. Enhancing Deliberation Through Computer Supported Argument Visualization. In: P. Kirschner, S. Buckingham Shum, C. Carr (Eds), *Visualizing Argumentation: Software Tools for Collaborative and Educational Sense-Making*, Springer, 2004