Conversation Pivots and Double Pivots

First Author Name (Blank if Blind Review)
Affiliation (Blank if Blind Review)
Address (Blank if Blind Review)
e-mail address (Blank if Blind Review)

Second Author Name (Blank if Blind Review)
Affiliation (Blank if Blind Review)
Address (Blank if Blind Review)
e-mail address (Blank if Blind Review)

ABSTRACT
Many sites on the web offer collaborative databases that catalog items such as bands, events, products, or software modules. *Conversation pivots* allow readers to navigate from pages about these items to conversations about them that may occur on the same site or elsewhere on the Internet. *Double pivots* allow readers to navigate from item pages to pages about other items mentioned in the same conversations. Using text mining techniques specific to the collection it is possible to find references to collected items in online conversations. We implemented conversation pivots for Drupal.org, a development site for the Drupal content management system and for the CPAN archive of perl modules.

Author Keywords
Online Discussion, Conversation, Recommender, Pivot, Drupal, Perlmonks

ACM Classification Keywords
H5.4 Hypertext/Hypermedia Navigation. H5.2. User Interfaces

INTRODUCTION
There are many websites dedicated to being a definitive online resource for a specific genre. Typically records on these sites are in a rigidly structured data format and moderated to ensure accuracy. These sites also frequently include free-form, more open online forums, creating an abundant and unstructured repository of user-contributed data.

Drenner et al [1] describes these areas as "item-land" and "forum-land", and describes a process where the site MovieLens was able to create references between these two discrete sections based on mining related keywords. But this idea of item-land can be used to describe a large variety of resources:

- Musician profiles on MySpace.com or Last.fm
- An entry in a public event calendar such as a concert
- A Wikipedia entry for a person, place, or thing
- Vendor pages for products ie it's page on Amazon.com
- A repository for software modules such as PEAR or CPAN

It is easy to imagine how bridges between "item-land" and "forum-land" could be very useful in these contexts, especially when "forum-land" might be separate from the item collection (i.e., on a separate website.)

While previous work has explored the connections in the case of movie items, we explore them in the case of item pages that describe software modules. We explain how the unique features of software modules and conversation threads can be used in inferring links between them. We also show how these links can be used not only to help navigate from software modules to related conversation but also from software modules to related modules.

CONVERSATION PIVOTS
Links between items and forums are instance of a more general class of navigation aids that we call *pivots*. A pivot enables navigation from an object to a set of other objects that share some attribute in common. Perhaps the most familiar instantiation of the pivots concept is the ability, at sites such as del.icio.us and Flickr and many individual blogs, to click on a "tag" in order to move from one page or photo to a set of others that have been classified with the same tag. Many other pivots are possible. For example, a pivot can allow navigation from a message to other messages by the same author, or from an event announcement to other events at the same venue, or other events at the same time, or other events featuring the same speaker.[1]

More formally, a pivot function P maps from source items S to subsets of a set of target items T. Commonly, the pivot function can be represented as a matrix P, with one row for each source item and one column for each target item, as illustrated in Table 1. Each cell in the matrix indicates

[1] We borrow this usage of the term *pivot* from spreadsheet pivot tables and the description of features of the Ning platform for building social applications. [2]

whether the source and target are related (i.e., share a "common attribute").

In the case of conversation pivots from software modules, the source items are the pages describing the software modules, and the target items are conversation threads. A cell's value encodes whether it mentions the software module. For example, thread T3 refers to the same software module that page S1 describes but not the modules described by pages S2 and S3.

		Conversation Thread			
		T1	T2	T3	T4
Software Module	S1	1	0	1	0
	S2	0	1	0	1
	S3	1	1	0	1

Table 1. A matrix of conversation threads and software module references.

Depending on the application, it may be useful to automatically display on a source page a *pivot block* with links to a few targets. In other applications, users may need to explicitly request that related targets be displayed. This is common, for example, in tagging interfaces, where a user has to click on a tag before the set of related items is displayed. In either case, if too many target items are related to a single source, it is helpful to order them based on which are likely to be most useful when navigating from that source item. To accommodate that, a cell in the pivot matrix can contain a similarity or relevance score, rather than just a binary indicator of a reference to the source item.

We have implemented, but not yet publicly released conversation pivot blocks for two popular software platforms, Drupal and Perl. On the Drupal.org website, there is a page for each of the hundreds of available add-on modules. The Related Discussion block, on the top right in Figure 1, adds links to related conversations that occur in the Drupal forums, elsewhere on the site.

For the programming language Perl, there are thousands of software libraries, also referred to as modules, that programmers can download from a site called CPAN, or from one its mirror sites. Each module gets its own page on CPAN, with information about its history and status and a link to download the actual code. There is also an experimental mirror site called AnnoCPAN that displays user annotations on the module pages. The CPAN and AnnoCPAN sites do not host conversations about Perl or Perl modules. One popular venue for such conversation is a website called PerlMonks. We have implemented a conversation pivot block for AnnoCPAN that shows links to related conversation threads on the PerlMonks website, as shown in Figure 2.

Without the conversation pivot blocks, people could use a search engine to seek forum references to particular software modules. This would require significant user

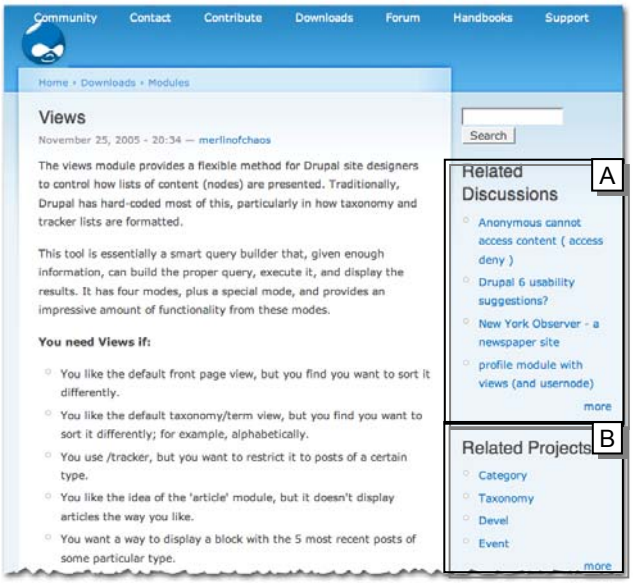

Figure 1. The Drupal.org website with (A) pivot to related conversations and (B) double-pivot to other modules.

effort. Moreover, we are able to tune our search algorithm to take advantage of the structure of software module pages and conversation threads, in a way that would be difficult for users to simulate if they had to construct their own search queries.

Double Pivots

Any pivot matrix relating source items to subsets of target items can be used to generate another pivot matrix relating the source items to subsets of the source items. For example, from the page for a software module, we can display a pivot block of other related software modules, as shown in the "Related Modules" blocks in Figures 1 and 2. This can help users to identify complementary modules and especially substitutes that may be preferable to the current module.

Conceptually, the double pivot block automatically follows the pivot to related conversation threads and then pivots again to a set of modules that are also referenced by those threads. By automatically aggregating the other module references from the set of messages that reference the current module, the body of contributing content is much larger than just the few messages that can be displayed to a user on a page.

The double-pivot technique is closely related to the technique of item-item collaborative filtering [3]. There, two items are related if they are highly rated (or purchased) by the same people. Here, two items are related if they are referenced in the same conversations.

The simplest way to implement the double pivot is with matrix multiplication. If P is matrix whose rows are the source items and columns are the conversation threads, then the transpose matrix, P', has the source items as columns. PP' is a symmetric matrix with one row and one column for

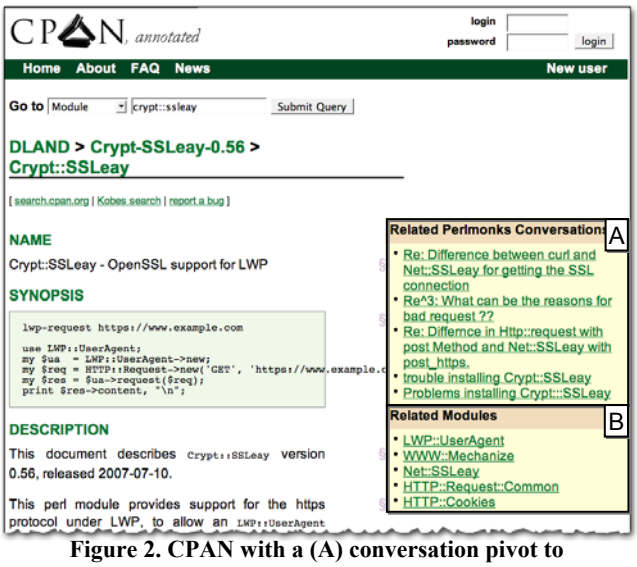

Figure 2. CPAN with a (A) conversation pivot to Perlmonks.org, and a (B) double-pivot to related modules

each source item. When P is a binary matrix with a 1 when the source and target are related, the cell $(PP')_{ij}$ just gives a count of how many threads referenced both module i and module j. Table 2 illustrates the resulting product matrix PP' for the original matrix from Table 1. This indicates that S1 and S2 are never referenced in the same thread, but S2 and S3 are both referenced in two different threads. Thus, the double pivot block for the page about S2 would include a link to S3 but not to S1.

		Software Module		
		S1	S2	S3
Software Module	S1	2	0	1
	S2	0	2	2
	S3	1	2	3

Table 2. Module by Module Associations from a double-pivot

More sophisticated implementations of the double pivot would account for the overall popularity of source items. For example, when two modules that are rarely referenced are referenced in the same thread, it is a stronger indicator that the two are related than when the same outcome occurs for a pair of frequently referenced modules.

DETECTING CONVERSATION SUBJECTS

The pivot matrix is constructed by examining the potential target threads to see which source items they reference. In the MovieLens system[1], message authors explicitly link their messages to movie pages, and a linking tool with auto-completion simplifies this task. The system also suggests movie references based on searching the text of messages for movie titles, but relies on message authors to review the suggestions and accept or reject them. More generally, this technique is analogous to authors providing unique identifiers to items in conversations. With explicit IDs, it is possible to easily harvest the references with a simple parser. For example, ISBN numbers or Amazon ASIN or URLs may serve as target references that can be easily detected.

It may not be practical to expect authors of conversational messages to explicitly tag each reference to a software module. Instead, references must be inferred. We have implemented algorithms that leverage the special features of software modules as source items and the special features of conversation messages as targets. Given the application, where the purpose is to help users navigate from software module pages to related conversation, recall is far less important than precision in a matching algorithm.

Some software systems employ a naming system that allows exact string matching on module titles to perform well. Perl modules follow a hierarchical naming convention with "::" as a separator found rarely in normal conversation. If the text string "Time::ParseDate" appears in a message, the author almost certainly meant to refer to the Perl module of that name, so that exact matching on module titles will have high precision. Moreover, a social norm has emerged so that an author who wants to refer to that module will typically use that text string to refer to it, rather than a shorter alias such as ParseDate. Thus, exact text matching will retrieve the correct references with high recall as well, although it will occasionally miss matches due to misspellings.

In the Perlmonks data set, we used the exact match on module title technique. Of 550,679 total comments that had been posted on the Perlmonks site, 16% contained module references. Some messages contained long lists of modules; it seemed unlikely that a user examining any particular module would find it useful to navigate to a message containing a long list that happened to contain the source list. Even after discarding those outliers, 41% of the 4,548 total CPAN modules were cited on Perlmonks, and those had a median of 5 references. The most referenced module, "Data::Dumper" had 6,399 references.

Based on those detected module references, excluding 35 outliers, we computed the double pivot matrix. Of the 4,548 modules that were cited at all, 1,702 had at least one other module co-cited in the same conversation. Of these, the median number of "related modules" was 6. "Data::Dumper" had the most related modules, 709.

String matching on titles can be extended to handle situations where simple application would miss too many references. Some module titles are so long that messages refer to them using shorter aliases. For example, most authors refer to the module titled "Content Construction Kit" as "CCK" rather than using the full title. Module authors can be asked to provide a list of popular aliases for the modules that are commonly used in conversation. Even if it is unrealistic to expect all conversation authors to tag all their references to modules, it may be quite reasonable to expect the authors of the module pages to identify aliases that are frequently used in conversation. There are relatively few module pages, and the authors of those

pages, typically the people responsible for maintaining the software modules, have an incentive to help users find relevant conversation and related modules.

String matching on titles can also be extended to handle situations where it would yield too many false positives. Some software systems use module titles that are potentially ambiguous. For example, Drupal modules generally use common words such as "Event" or "Upload" for titles. Simple text matching on these titles would yield many false positives, conversation messages that use these words but not in reference to the software modules. Real references to the Upload module, however, frequently contain the word "module" in close proximity to the word "Upload". Thus, a matching algorithm that searches for the title in close proximity to the magic word "module" will likely generate many fewer false positives, though possibly missing more correct references.

We used this method to find module references in the Drupal.org conversation forums. Of 71,742 messages, 19,546 contained at least one module reference. After discarding 21 outliers that each referred to more than ten modules, 915 of the 1,590 modules were cited in at least one message and those had a median of 5 references. The most referenced module, "Image", had 1,713 references.

Based on those detected module references, we computed the double pivot matrix. Of the 915 modules that were cited at all, 747 had at least one other module co-cited in the same conversation. Of these, the median number of "related modules" was 6. "File" had the most related modules, 300.

If the source page about a software module includes text describing what the module does, it is plausible that messages about a module may contain vocabulary found on the module page, but rarely elsewhere. Thus, it may be useful to compute a full-text similarity score between a conversation thread and the text of the software module page. We have implemented a cosine scoring function [4] using the open-source package Lucene, although we have opted to use the exact matching algorithm in our initial employments.

We can also exploit the structure and chronology of conversation threads to help determine which messages are most important. Among messages that reference a software module, conversations that are longer and more recent are more likely to be useful to display in a pivot block. Although the target of the link in a pivot block will be a particular message in a thread, we compute the message's importance based on features of the entire thread. Our initial implementation simply computes a timestamp based on the time of the most recent message posted in the thread. Among messages that reference the source item, those whose threads have more recent activity are shown first.

CONCLUSIONS AND FUTURE RESEARCH

The results from our implementations with the Drupal.org and AnnoCPAN/Perlmonks lead us to believe that conversational pivots and double pivots hold a great deal of promise for making online collections more useful for users. By automatically mining forum data for item references and generating recommendations of other modules we provide the users with a shortcut through time-intensive manual search.

In the future, we plan to test our matching and ranking algorithms and assess their value to our target communities. We will test the precision of our matching algorithms by having raters assess whether the identified module references are correct. We hope to evaluate whether the pivot and double pivot blocks provide value, and compare among different ranking algorithms, using click-through rates as a measure of utility. We could also use user feedback to improve results by either asking users to flag particularly relevant or irrelevant matches, or by using implicit metrics such as click-through rates.

The vision of the semantic web [5] is that information for human consumption will also be tagged in a way that computers can process in useful ways. When semantic markup is not available, however, it may be possible infer it imperfectly, but well enough to enable particular kinds of processing. We have developed techniques for detecting references to software modules in on-line conversations that are sufficient to enable the creation of navigation aids in the form of conversation pivots and double pivots. The techniques may be extensible to creating conversation pivots for other kinds of items.

ACKNOWLEDGEMENTS

We wish to thank the administrators of Drupal.org and Perlmonks.org for allowing us access to their data sets.

REFERENCES

1. Drenner, S., Harper, M., Frankowski, D., Riedl, J., and Terveen, L. Insert movie reference here: a system to bridge conversation and item-oriented web sites. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems* CHI '06 (2006)
2. Ning Platform Pivots Feature http://blog.ning.com/2005/11/new_on_the_ning_pivot.html
3. Linden, G., Smith, B., York, J. Amazon.com recommendations: item-to-item collaborative filtering, *Internet Computing, IEEE*, 7, 1 (2003), 76-80.
4. Salton, G., Wong A., and Yang C.S., A Vector Space Model for Automatic Indexing, *Communications of the ACM*, vol. 18 (1975), nr. 11, pages 613–620.
5. Berners-Lee, T., Fischetti, M., (1999). Weaving the Web. *Harper, San Francisco*, ISBN 9780062515872