Conversation Pivots and Double Pivots

Daniel Xiaodan Zhou, Nathan Oostendorp, Michael Hess, Paul Resnick
CommunityLab*
University of Michigan School of Information
Ann Arbor, MI 48109
{mrzhou, oostendo, mlhess, presnick}@umich.edu

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 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 the CPAN archive of Perl modules, and for Drupal.org, the reference site for the Drupal content management system.

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

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

INTRODUCTION

Many websites maintain collections of pages about people, places, and things. These item pages typically include structured data. The sites also frequently include online forums, with an abundant and unstructured repository of user-contributed data about the same items. Drenner et al [1] describe these areas as "item-land" and "forum-land", and describe how the site MovieLens was able to cross-link them. Movie item pages have links to conversation threads mentioning the movies and conversation pages link to the referenced movie pages.

More generally, item-land describes a large variety of online collections:

- Musician profiles on MySpace.com or Last.fm
- Concert or other event listing in a public calendar
- Wikipedia entries for people, places, or things
- Product pages (e.g., on Amazon.com)
- Software module pages on a repository 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 instances 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 pivot 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]

CHI 2008, April 5–10, 2008, Florence, Italy.

* CommunityLab is a collaborative project of the University of Minnesota, University of Michigan, and Carnegie Mellon University. http://www.communitylab.org/

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

		Messages				
		T4	T2	T3	T4a	T4b
Software Module	S1	0	1	0	1	0
	S2	1	0	1	0	0
	S3	1	0	1	0	1

Table 1. A matrix representing the pivot from software modules to conversation messages that reference them. Target messages T4a and T4b are from the same thread.

More formally, we define a pivot as a function that maps from source items to subsets of a set of target items. The pivots can be represented as a matrix P, with one row for each source item and one column for each target item. Each cell in the matrix indicates whether the source and target are related (i.e., share a "common attribute").

In the case of conversation pivots for software modules, the source items are pages describing software modules and the targets are messages. A cell's value encodes whether the message mentions the software module. For example, as shown in Table 1, message T3 refers to the software module that page S1 describes but not the modules described by pages S2 and S3.

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 based on frequency or prominence of the reference, 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 CPAN. 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, will add links to related conversations that occur in the Drupal forums, elsewhere on the site. Each link is to an entire thread, scrolled to the first message in the thread that references the module.

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. Each module gets its own page on CPAN, with information about its history and status and a link to download the actual code. We have focused on an experimental mirror site called AnnoCPAN that displays user annotations on the module pages. The CPAN and AnnoCPAN sites do not host

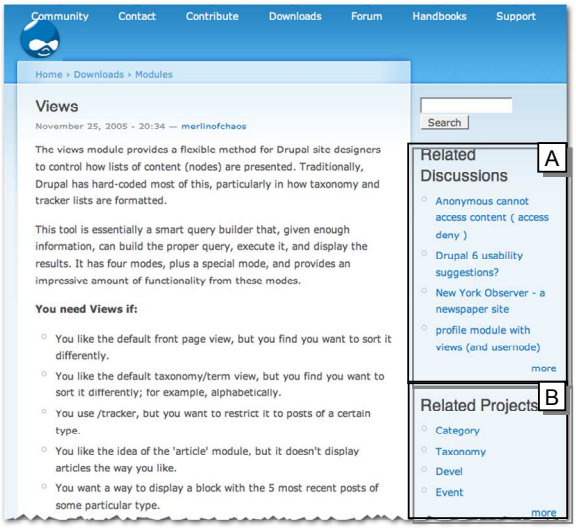

Figure 1. The Drupal.org website with (A) pivot to related conversations and (B) double-pivot to other

conversations about Perl or Perl modules. One popular venue for such conversations 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.

We 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 each link in a pivot block points to 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.

Without the conversation pivot blocks, people could use a search engine to seek forum references to particular software modules. This would require significant user 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 back to subsets of those same 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 identify complementary modules and

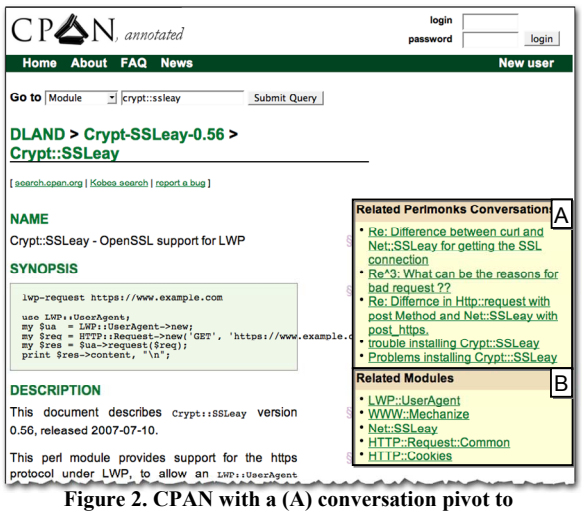

Figure 2. CPAN with a (A) conversation pivot to Perlmonks.org, and a (B) double-pivot to related modules

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. Such a process would be tedious, however, for users to perform manually. 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], where two items are related if they are highly rated (or purchased) by the same people. It is also analogous to co-citation analysis in bibliometrics, where two papers are related if they are cited in the same other paper [4].

In computing conversation double pivots, we first collapse messages into threads. Column T4a in Table 1 is a message that might have said something like, "Module S1 doesn't quite do what I need. Anyone have suggestions?" T4b is a reply suggesting module S3 as an alternative. Other users, when visiting the page for module S1, might then benefit from a double pivot link to S3, even though no single message mentions both S1 and S3. Thus, we treat two modules as related if they are referenced in the same thread.

Table 2 illustrates the double pivot matrix resulting from the original matrix from Table 1. It indicates that S1 and S2 are never referenced in the same thread, but S2 and S3 are both referenced in two different threads, and S1 and S3 in one thread. Thus, the double pivot block for the page about S2 would include a link to S3 but not to S1.

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.

		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

DETECTING CONVERSATION SUBJECTS

The pivot matrix is constructed by examining the potential target threads to see which source items they reference. One possibility would be to rely on message authors to explicitly link to source items rather than using natural language to identify them. Tools such as auto-complete could simplify this task for message authors. Even so, it seems unwise to depend on authors to do this, since the primary beneficiaries would not be the authors or even the readers of the messages, but unknown and unseen future visitors to module pages.

Instead, a software program can automatically infer references to items as they are naturally expressed in messages, albeit with some error. The errors will be reduced to the extent that authors refer to items using distinctive canonical identifiers, such as ISBN numbers. For example, the PHOAKS project mined Usenet for references to web pages, where the natural way to refer to a site in conversation was to put in a complete URL [5].

Two factors make it practical to automatically mine conversations for software module references. The first is a high tolerance for errors of omission: detecting even a small fraction of the actual references to software modules may be sufficient to generate useful pivot and double-pivot blocks for navigation. The second factor is that there are regularities in module names and in patterns of reference to them that can be exploited when creating a reference index.

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 mining Perlmonks conversations, we used the exact matching on module titles 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 module, so we discarded those messages. Even after discarding 35 outliers that referenced more than ten modules each, 41% of the 4,548 total CPAN modules were cited on Perlmonks, and those had a median of 5 references.

Based on those detected module references, excluding 35 outliers, we computed the double pivot matrix. Of the 1,884 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.

String matching on titles can be extended to include searching for distinctive aliases. For example, most authors refer to the Drupal module titled "Content Construction Kit (CCK)" as "CCK" rather than using the full title. Even if it is unrealistic to expect all message 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. The authors of those pages, typically the people responsible for maintaining the software modules, have both the knowledge of commonly used aliases and the incentive to enter them in to the system, in order to help users find what others are saying about the modules.

String matching on titles can also be narrowed 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 adjacent 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 292,139 messages, 47,794 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 6 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.

CONCLUSIONS AND FUTURE RESEARCH

The results from our implementations with Drupal.org and AnnoCPAN/Perlmonks lead us to believe that conversation 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 users with a shortcut through time-intensive manual search.

The vision of the semantic web [6] 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 to infer it imperfectly, but well enough to enable particular kinds of processing. We have developed techniques for detecting references to software modules in online 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 types of items.

ACKNOWLEDGEMENTS

This material is based upon work supported by the National Science Foundation under Grant Nos. 0308006 and 0325837. We wish to thank the administrators of Drupal.org and Perlmonks.org for allowing us access to their data sets and to Ivan Tubert-Brohman for allowing us to add the pivot feature to the AnnoCPAN site.

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. Small, H. Co-citation in the scientific literature: A new measure of the relationship between two documents. *Journal of the American Society for Information Science*, vol. 24, 4 (1973), 265 – 269.
5. Terveen, L., Hill, W., Amento, B., McDonald, D., and Creter, J. PHOAKS: a system for sharing recommendations. *Communications of the ACM*. 40, 3 (1997), 59 – 62.
6. Berners-Lee, T., Hendler, J., and Lassila,O.,The semantic web. *Scientific American.* 284 (2001), 5, 34.