How Does Research Evolve? Pattern Mining for Research Meme Cycles

Dan He[†], Xingquan Zhu[‡], and D. Stott Parker[†]
[†]Department of Computer Science, UCLA, Los Angeles CA 90095-1596, USA
[‡]QCIS Centre, University of Technology, Sydney, NSW 2007, Australia
danhe@cs.ucla.edu; xqzhu@it.uts.edu.au, stott@cs.ucla.edu

***Abstract*—Recently a great deal of attention has been focused on the problem of tracking news memes over the web, modeling shifts in the ebb and flow of their popularity. One of the most important features of news memes is that they seldom occur repeatedly; instead, they tend to shift to different but similar memes. In this work, we consider patterns in research memes, which differ significantly from news memes and have received very little attention. One significant difference between research memes and news memes is that research memes often have cyclic development, motivating the need for models of research cycles. Furthermore, these cycles may reveal important patterns of evolving research, shedding lights on how research progresses. In this paper, we formulate the modeling of the cycles of research memes, and propose solutions to the problem of identifying cycles and discovering patterns among these cycles. Experiments on two different domain applications indicate that our model does find meaningful patterns and our algorithms for pattern discovery are efficient for large scale data analysis.**

***Keywords*-Research memes, frequent patterns, MeSH hierarchy, shortest paths, topic mining, topic evolution**

I. Introduction

'Memes' refer to cultural units that carry ideas, behavior or style, spreading from person to person. A great deal of work has been done on tracking topics, ideas and memes across the web [9], [10], [11]. Tracking the evolution of memes is an important problem, since it allows us to understand the competition among news and blog items each day, and how certain stories persist while others fade quickly.

The modeling and tracking of memes have been studied in many works, with an emphasis on modeling bursts, tracking trends, and detecting cycles. Much less attention, however, has been aimed at studying memes in scientific and engineering research. News memes and research memes differ in the following aspects:

1) **Evolving speed:** News memes are characterized by sharp, burst-like increases in volume, and rapid decreases. By contrast, the volume of research memes usually increases much more slowly and is more drawn out. As a result, news memes usually spread and fade over time scales on the order of days. Research memes, however, can spread over time scales spanning from months to years, and even many years:
2) **Evolving style:** News memes shift constantly, and seldom shift back — news topics change very fast. A given meme does not usually recur, for clear reasons. Research memes on the contrary do recur, due to how research is conducted. For example, it is argued in http://www.questioning.org/module/cycle.html that many challenge problems require multiple investigations of the same topic prior to consolidation, acquiring enough insight and evidence to move to the next stage.
3) **Evolving driver:** While news focuses on the present, reporting what is happening with emphasis on timely updates to attract public attention, research focuses more on the past, reporting what has been done with emphasis on innovative ideas and significant findings. As a result, research memes are mostly driven by dedicated researchers (or research groups) with a certain degree of continuity; this in turn makes long-term modeling of research memes possible.

In this work, we are specifically interested in discovering patterns that govern shifts of interest, away and then back, in a research meme. We focus on this specific type of pattern for two reasons: (1) Research memes are often more 'cyclical' than news memes, and we observe recurrences of the same meme. Thus the cyclical pattern models a special characteristic of research memes. (2) As shown in the work of [7], the cyclical patterns potentially permit forecasting, and thus may help us to predict future occurrences of memes. Therefore, we believe that understanding the evolution of research memes is an important problem.

Since there can be multiple occurrences of research memes spanning long intervals of time, we build a graph in which each (meme, occurrence)-pair is considered as a node. Thus, since one meme can have multiple occurrences, we may have multiple nodes for the same meme. The two nodes for the meme are naturally considered the start and end of a cycle. We then want to identify a path between the two nodes, charactering shifts or evolution of the memes. Based on the assumption that research interests tend to shift to related topics, we then search for a shortest path between the two nodes in the graph, maximizing the likelihood that the path indeed characterizes true meme shifts of the cycles.

We consider two different applications: (1) memes in computer science research and (2) memes in biomedical research. We propose different models for the two applications according to their properties, including different definitions

of meme occurrence and different distance functions. We also propose an efficient method to identify the shortest path in the graph. Finally we propose an efficient frequent pattern mining algorithm for the two applications. The patterns obtained in our experiments reveal different aspects of meme shifts in each application. For computer science memes, the patterns show that the similarity of memes tends to remain stable during shifts along the cycles. For biomedical memes, the patterns show that the memes tend to shift to more general memes first and then to more specific ones. Then this generality remains stable during the shifts along the cycles.

II. Methods

Given a set of publications, our model is derived in the following steps:

- Identify research meme occurrences
- Identify meme cycles
- Mine frequent patterns for meme shifts in these cycles
- Analyze frequent patterns

A. Research Meme Occurrence

To identify cycles of research memes, we first need to identify occurrences of these memes. We have two different applications, each defining occurrences of topics differently.

1) Occurrence for Computer Science Memes: We consider each named session of a conference as a topic, or meme. Because these sessions occur in a conference annually, we say the meme occurs in a year if the session appears in the conference in that year.

2) Occurrence for Biomedical Memes: Since each MeSH term is considered to be a meme, in this work, we consider the occurrence of MeSH topics as points in time at which the popularity of the MeSH topics shows significant increase. It is well known [2] that topic popularity can be quantified with the frequency of the topic in related literature. Adapting methods like those in [8], we propose to formalize popularity of topics in hierarchies with traditional trend indicators from technical market analysis, such as EMA, MACD and MACD histogram. According to the topic dynamics model, a burst period is defined as the continuous time period over which the MACD histogram is positive. The occurrence time of the topic, or meme, is then defined as the occurrence time of the burst. The readers can refer to [8] for the details of the topic dynamics model.

B. Research Meme Distance

Before we introduce our method to identify cycles of research memes, we need to first define similarity and distance of memes. In this work, we consider two types of meme distance for the two different applications.

1) Distance for Computer Science Memes: Since here we consider each conference session as a meme, the distance of two memes, or two sessions, is naturally determined as one minus the similarity of the two sessions. The similarity of the two sessions is defined as their Jaccard similarity:

$$sim(A,B) = \frac{|T(A) \cap T(B)|}{|T(A)| + |T(B)| - |T(A) \cap T(B)|} \quad (1)$$

where A, B are sessions, $T(A)$ is the set of terms for session A and $|T(A)|$ is the number of terms in A. Therefore, the distance of two memes is defined as $dist(A,B) = 1 - sim(A,B)$, which is within the range $[0,1]$.

2) Distance for Biomedical Memes: Here we consider MeSH terms as memes that are organized into a hierarchy. In this hierarchy, the distance of memes is naturally defined as the minimum topological distance of the two memes in the hierarchy. (The MeSH hierarchy permits multiple supertopics for a given topic, so there can be multiple paths between two memes.)

C. Search for Research Meme Cycles

In order to search for cycles of research memes, we first construct a graph $G = (V, E)$ where V is the set of nodes for each occurrence of each meme. E is the set of edges and $e_{a_i,b_j} = dist(a_i, b_j)$, where a_i is a node in the graph corresponding to the i-th occurrence of meme a and $dist(a_i, b_j)$ is the distance of the two corresponding memes a and b. Notice that since the occurrence of memes is temporal, the graph is directed. A meme occurring at year t can only have outgoing edges to memes occurring at year $t+1$, and it has outgoing edges to all such topics.

Once the graph is built, we can search for research meme cycles in the graph by looking for shortest paths between the two recurring memes. The shortest path then represents a cycle of research memes. This is based on the assumption that attention focused on research topics tends to shift to related topics (rather than to unrelated topics). For example, it is reasonable that research on 'social networks' would shift to 'graph mining' since the two topics are quite related. On the contrary, it is unlikely that research on 'social networks' would shift to 'embedded systems' since they are not clearly related. Therefore, it is reasonable that the shortest path between two occurrences of a meme would maximize the likelihood of the shift.

Indeed, the shortest path is not the only possible shift between the two occurrences of a topic. There might be emerging topics during the process that culminate in a shift to the later occurrence of the topic. The shortest path shift specifically models the process that the research attention shifts away from a topic to related topics and then shift back, for reasons such as that the research on related topics leads to new insights on the original topic. In this work, we focus on modeling this process with several possible patterns. This suggests our first computational problem:

Problem 1: Given a DAG (directed acyclic graph) $G = (V, E)$, a set of f node pairs (n_{11}, n_{12}) (n_{21}, n_{22}),..., (n_{f1}, n_{f2}), find shortest paths between each pair of nodes (n_{i1}, n_{i2}) for $1 \leq i \leq f$.

Searching for shortest path in a dag is a well-studied problem [3] and probably the most famous algorithm is Dijkstra algorithm [4]. The complexity of Dijkstra's algorithm is $O(n^2)$ where n is the number of nodes in the graph. In our work, n is usually a very large number since we consider each occurrence of each meme as a node and one meme can occur multiple times. For example, the MeSH hierarchy contains around 50,000 terms, but the graph built from MeSH terms consists of millions of nodes. Thus the complexity could be very high if we load the whole graph into memory. However, the complexity can be improved significantly since one node has outgoing edges only to the nodes occurring at the next time unit. Thus to search for the shortest path between two nodes, given their occurrence times, we can easily determine the set of nodes and edges that need to be considered (and loaded into memory). The subgraph is usually much smaller than the whole graph, so searching on the subgraph is much faster.

D. Mining Patterns of Research Meme Shifts

To our knowledge there is no previous work on shifts in patterns of research memes, and thus no benchmarks or databases of known patterns to validate our work against. Therefore, we have developed a set of experiments, hoping to discover meaningful patterns.

1) Patterns for Computer Science Memes: Since the memes, or sessions, for conferences contain a set of terms, we want to observe the evolution of the terms involved in the shifting memes. For each session at year i, we consider its relationship with sessions in year i-1. For each year i, we define the following three types of terms, with $T(i)$ indicating the set of terms contained by the meme at year i in the cycle:

1) Disappearing term: $t \in S(i-1)$ and $t \notin S(i)$
2) Emerging term: $t \notin S(i-1)$ and $t \in S(i)$
3) Stable term: $t \in S(i-1)$ and $t \in S(i)$

Notice that the frequency of a term can change during these shifts, if they are involved in either disappearance or emergence.

We want to observe the evolution of each type of term, and see which types of term play the most important roles in the shift of the research memes. Therefore, at each year i, we compute the weight for each type of terms as follows:

$$weight_i = \frac{\sum |T_i|}{\sum_{j=1}^{3} |T_j|} \qquad for \quad 1 \leq i \leq 3$$

Here $weight_i$ is the weight for the i-th type of terms, T_i is the set of terms of the i-th type, and $|T_i|$ is the number of i-th type terms.

Then during the shift, if the weight increases, we mark the meme as 1. If the weight decreases, we mark the meme as -1. If the weight does not change, we mark the meme as 0. Notice that by saying the weight does not change, we indeed mean the change of the weight is very small. In this work, we say if the weight change is less than 0.05, the weight is stable. Given the average term number of a session as 32, a weight change of 0.05 means a difference of one or two terms, which is indeed a very small change. Thus for each shortest path, we obtain three sequences of alphabet size $\{1, -1, 0\}$, one for each type of terms. Then for the sequences of each type of term, we mine frequent patterns among them. Intuitively, we want to tell how the evolution of the terms drives the shifts of research memes in cycles.

In this work, we are interested in patterns that occur at the same position across all the sequences with frequency greater than a threshold (we call them *position-specifc frequent patterns*). For illustration purposes, we show an example in Figure 1.

Pos:	1	2	3	4	5	6
Seq 1:	1	-1	1	1	0	1
Seq 2:	1	-1	0	0	0	-1
Seq 3:	0	-1	1	-1	0	-1

Figure 1. Example for frequent patterns.

Unlike the traditional frequent pattern mining problem, here the sequence of patterns is temporal. Therefore, occurrences of a pattern at different positions are considered as different occurrences and thus not accumulated for the pattern. For example, in Figure 1, the pattern $(1, -1)$ occurs at position 1 twice in all three sequences, and at position 3 once in all sequences. If the threshold is two, we say the pattern $(1, -1)$ is frequent only at position 1 but not at position 3. Our goal is to identify all such frequent patterns. We next propose our second computational problem:

Problem 2: Given a set of n sequences $(s_1, s_2, \ldots, s_n)$, find all *position-specific frequent patterns* p's such that $\frac{support(p,i)}{n} \geq t$, where $support(p, i)$ indicates the support of pattern p at position i, and t is the frequency threshold.

Many methods exist to mine frequent patterns [6]. In a given sequence, frequent pattern is defined as patterns with frequency greater than a pre-defined threshold, where frequency is the ratio of support of the pattern (number of occurrences in the sequence) and the total number of possible patterns. It's also quite often that people just use support directly instead of frequency.

Direct application of Apriori algorithm [1], which may be the most well-known frequent pattern mining algorithm, is not efficient for the problem since the sequences needs to be re-scanned for each newly generated pattern. To identify all frequent patterns efficiently, we build a suffix tree for all the sequences of each type of terms. Since we care about the occurrences of the suffix at different positions, we attach an index bit at the beginning of each suffix that indicates the starting position of the suffix. Then we build

Input: A set of sequences $s_1, s_2, \ldots, s_m$ and threshold t
Output: All position-specific frequent patterns with frequency no less than t
1. suffixTree ← *buildSuffixTree*$(s_1, s_2, \ldots, s_m)$
2. nextSet ← $\{-1, 1, 0\}$
3. frequentPattern ← null
4. start breadth-first search from the root of suffixTree
5. while (nextSet is not empty) {
6. newNextSet ← null
7. for each pattern $p_i \in$ nextSet {
8. frequency ← computeSupport$(p_i, suffixTree)/m$
9. if (frequency $\geq t$) {
10. frequentPattern ← frequentPattern + $\{p_i\}$
11. newNextSet ← newNextSet +
12. extend$(p_i, suffixTree)$
13. }
14. nextSet ← newNextSet
15. }
16. output frequentPattern

Figure 2. The algorithm to generate all temporal frequent patterns for a set of sequences.

the suffix tree on top of the new suffix. The advantage is that by traversing from the root, we can immediately identify the starting position of a pattern. Thus to identify all frequent patterns, we can traverse from the root with breadth first search, where the depth of the search indicates the length of the pattern. The support of the pattern is the sum of the support from all leaf nodes in the subtree of the pattern. During the traversal, if the frequency is less than the threshold, the subtree can be pruned from any further traversal. Therefore, the algorithm only scans the sequences once. Since constructing suffix trees for a length n string is $O(n)$ [5], we can construct a suffix tree for all sequences with time complexity $O(n \times m)$, where n is the length of the sequences and m is the number of sequences.

The procedure of the algorithm is shown in Figure 2. On line 1, the function *buildSuffixTree*$(s_1, s_2, \ldots, s_m)$ applies the classical linear algorithm [5] to build suffix tree for the set of sequences $s_1, s_2, \ldots, s_m$. The only difference is that we attach the position of a suffix as an extra bit at the beginning of the suffix for all the suffixes while building the suffix tree. On line 8, $computeSupport(p_i, suffixTree)$ computes the support of the pattern p_i in suffixTree, by summing the support of all leaf nodes in the subtree. On line 12, $extend(p_i, suffixTree)$ returns the set of patterns by extending the pattern p_i by one bit in suffixTree, following all possible branches.

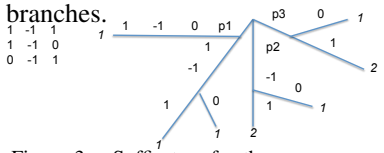

Figure 3. Suffix tree for three sequences.

For illustration, we show an example in Figure 3. The support of the leaf nodes are in italian. Let's assume the support threshold is $\frac{2}{3}$. As we can see, we can traverse from the root with breadth-first search. For length-one frequent patterns, we find $(p3, 1)$, $(p2, -1)$ and $(p1, 1)$ having support 2, 3, 2, respectively. Therefore we identify the frequent pattern (1) at positions 1 and 3, and (-1) at position 2. Continuing the breadth-first search, for length-two frequent patterns we find $(p2, -1, 1)$ and $(p1, 1, -1)$. Notice the support of $(p1, 1, -1)$ is the sum of the support of the two leaf nodes in the subtree and therefore is 2. Thus we identify two frequent patterns of length-two $(-1, 1)$ at position 2 and $(1, -1)$ at position 1.

pos	l = 1	l = 2	l = 3	l = 4
1	1 (0.56)	1, -1 (0.16)	-1, 1, -1 (0.086)	-1,1,-1,1 (0.075)
2	1 (0.28)	1, -1 (0.13)	1, -1, 1 (0.086)	N/A
3	-1 (0.18)	-1, 1 (0.107)	-1, 1, -1 (0.065)	-1,1,-1,1 (0.054)

Table I
THE PATTERN FOR DISAPPEARING TERMS, WHERE '1' INDICATES 'DECREASE', '-1' INDICATES 'INCREASE' AND '0' INDICATES 'NO CHANGE'. THE NUMBER IN () IS FREQUENCY.

2) Patterns for MeSH topics: For each node on each shortest path, we identify the depth of the corresponding topic in the hierarchy. Since the depth of the topics in the hierarchy indicates their generality, we are able to then observe shifts in generality of research memes. We want to observe frequent patterns for these shifts. We mark topics that shift to increased generality (topics with lower depth) as 1, and those that shift to lower generality (topics with higher depth) as -1, and topics with no change in generality as 0. Then we obtain a sequence of alphabet $\{1, -1, 0\}$ for each shortest path. We then search for frequent patterns in all these sequences. Through this application, we can tell how generality drives the shifts of research memes.

III. EXPERIMENTAL RESULTS

A. Application for Memes in Computer Science Research

1) Experiment Settings: For the second application, namely memes in computer science research, we download the conference sessions as well as all the paper titles the sessions contain for six conferences: 'SIGMOD', 'VLDB', 'KDD', 'ICDM', 'CIKM', 'SIGIR', from year 1975 to year 2010. We remove all stop-words from the paper titles and leave only the terms. The sessions with the same name are merged. There are totally 64,396 terms and 1,996 unique sessions. Thus on average each session contains 32 terms.

2) Patterns on Terms: We first check the patterns for all the three types of terms — disappearing, emerging and stable terms. Instead of comparing the absolute numbers of each type of terms, we compute the weight of each type by normalizing the number of each type with the total number of terms in the session. Since we have no knowledge about the patterns, it's hard for us to set a threshold to determine if a pattern is frequent or not. Thus we set a relatively low threshold and list the patterns of different length at each position with the highest frequency (that is no less than the threshold). The threshold we used is 0.05.

We show the results in Table III-A2, III-A2, III-A2, where '1' indicates 'decrease', '-1' indicates 'increase' and '0' indicates 'no change'. We show only the frequent patterns at position 1, 2, 3 since after position 3, the length of the frequent patterns becomes very short. As we can see, the most frequent patterns for disappearing terms and emerging

pos	l = 1	l = 2	l = 3
1	-1 (0.47)	1, -1 (0.18)	1, -1, 1 (0.086)
2	-1 (0.28)	-1, 1 (0.12)	-1, 1, -1 (0.054)
3	1 (0.18)	1, -1 (0.11)	1, -1, 1 (0.065)

Table II
THE PATTERN FOR EMERGING TERMS, WHERE '1' INDICATES 'DECREASE', '-1' INDICATES 'INCREASE' AND '0' INDICATES 'NO CHANGE'. THE NUMBER IN () IS FREQUENCY.

pos	l = 1	l = 2	l = 3
1	0 (0.74)	0, 0 (0.24)	0, 0, 0 (0.13)
2	0 (0.37)	0, 0 (0.17)	0, 0, 1 (0.065)
3	0 (0.22)	0, 1 (0.08)	0, 1, 0 (0.054)

Table III
THE PATTERN FOR STABLE TERMS, WHERE '1' INDICATES 'DECREASE', '-1' INDICATES 'INCREASE' AND '0' INDICATES 'NO CHANGE'. THE NUMBER IN () IS FREQUENCY.

terms always have an alternating trend for adjacent shifts (namely for one shift, the weight increases/decreases, then for the next shift, the weight decreases/increases). We actually observe similar patterns at positions beyond 3. However, since the length of the cycles are usually short, these patterns have relatively low frequency. This indicates that the weights of disappearing terms and emerging terms usually do not keep on increasing or decreasing. Instead, when a relatively large number of terms disappear or emerge during one shift, only a relatively small number of terms disappear or emerge in the next shift. One explanation is that if more terms disappear or emerge during the shift, the memes become further and further dissimilar to the initial meme. However, in order to yield a cycle of shifts, the memes cannot keep on shifting away from the initial meme. They need to shift back at some stage. The alternate pattern of the disappearing and emerging terms is consistent with this assumption. Furthermore, the stable terms remain quite stable, having weights with almost no changes during the shift.

3) Patterns on Distance: We next compute the distance of a meme in the cycle of shifts to the initial meme. We want to validate our assumption obtained from the previous set of experiments, namely the memes in the cycle of the shifts tend to shift away and shift back to the initial meme alternatively.

Our assumption is since the research attention shifts alternatively between more similar memes and less similar memes, we should expect the distance of the memes in the cycle to the initial meme to be relatively stable. We say the distance remains stable if the change of the distance from the previous shift is within certain threshold. We tried threshold from 0.01 to 0.05 and found that when the threshold is above 0.03, the distance shows very stable patterns. We show the patterns in Table III-A3. The distance between the memes on the cycle of the shift to the initial meme remains very stable, indicating research attention shifts that alternate between more- and less-related memes, while still maintaining high similarity to the initial meme. This also indicates that it's usually unlikely that the memes shift very far away from the initial meme before heading back. In most of the cases, once the memes shift away a little bit, they shift back immediately.

L	pos = 1	pos = 2	pos = 3
1	1 (0.45)	0 (0.27)	0 (0.16)
2	0, 0 (0.18)	0,0 (0.22)	0,0 (0.11)
3	0,0,0 (0.15)	0,0,0 (0.14)	0,0,0 (0.11)
4	0,0,0,0 (0.09)	0,0,0,0 (0.1)	0,0,0,0 (0.08)
5	0,0,0,0,0 (0.08)	0,0,0,0,0 (0.06)	N/A

Table IV
DISTANCES OF MEMES TO THE INITIAL MEME, WHERE '1' INDICATES 'DECREASE', '-1' INDICATES 'INCREASE' AND '0' INDICATES 'NO CHANGE'. THE NUMBER IN () IS FREQUENCY.

B. Application for Memes in Biomedical Research

1) Experiment Settings: We are interested in topics of MeSH terms (Medical Subject Headings; we use 'topic' and 'term' interchangeably), a hierarchy of topics in biomedical research. Each article in PubMed/MEDLINE is annotated with descriptive MeSH terms. We use the same data set as He and Parker [8] used, which is a collection of articles for the years 1950 through 2008. For each term in the MeSH ontology, we counted its frequency of occurrence in each year, and accumulateed these frequencies through the ontology (so that the frequency of a node is the sum of the frequencies of all descendents of the node as well as of itself). We then compute the occurrence time of these memes using the toptic dynamic model [8].

2) Memory Efficiency: To first illustrate the effectiveness of the memory efficient algorithm, we compare the performance of the algorithm with the naive shortest path search algorithm where the entire graph is loaded into memory first. The entire graph contains 103,168 nodes and 206,040,375 edges, which takes around 3.5GB memory while the subgraphs loaded into memory contains on average 35,559 nodes 36,321,073 edges. Therefore, the complete graph contains 3 times of nodes and 6 times of edges of the subgraphs on average. Loading the entire graph thus may be prohibitive for computers with limited memory.

3) Patterns on Generality: We next show the patterns of generality for memes in biomedical research, using the MeSH hierarchy. We annotate the generality of the memes along cycles, and apply the frequent pattern mining algorithm to obtain a set of patterns. These patterns are shown in Table III-B3. As we can see, almost 80% of the memes shift to more general memes at the very beginning. Then 40% of the memes shift to more specific memes. Again 17% of the memes then shift to more general memes. Starting from position 3, then, the most frequent pattern is the one in which the memes show little or no change of generality during the shift. This indicates that generality of the memes tends to shift up and down a bit at the beginning of the cycle, but subsequently remains relatively stable.

Following the above demonstration of memes evolving at the beginning of cycles, we study memes evolving at the end. Since the cycles are usually of different lengths, we can reverse the cycles and re-conduct the pattern mining process

L	pos = 1	pos = 2	pos = 3	pos = 4
1	1 (0.77)	-1 (0.45)	-1 (0.36)	0 (0.36)
2	1,-1 (0.39)	-1,1 (0.19)	0,0 (0.17)	0,0 (0.2)
3	1,-1,1 (0.17)	-1,1,0 (0.1)	0,0,0 (0.1)	0,0,0 (0.12)
4	1,-1,1,0 (0.1)	-1,1,0,0 (0.06)	0,0,0,0 (0.06)	0,0,0,0 (0.08)
5	1,-1,1,0,0 (0.06)	N/A	N/A	0,0,0,0,0 (0.05)

Table V
THE PATTERN FOR DISTANCE OF THE MEMES TO THE INITIAL MEME, WHERE '1' INDICATES 'BECOMES MORE GENERAL', '-1' INDICATES 'BECOMES MORE SPECIFIC' AND '0' INDICATES 'NO CHANGE OF GENERALITY'. THE NUMBER IN () IS FREQUENCY.

L	pos = -1	pos = -2	pos = -3	pos = -4
1	1 (0.52)	1 (0.45)	1 (0.39)	1 (0.38)
2	1,-1 (0.27)	-1,1 (0.21)	-1, 1 (0.2)	-1,1 (0.18)
3	1,-1,1 (0.16)	1,0,-1 (0.15)	0,-1,1 (0.1)	-1,1,1 (0.08)
4	1,-1,1,1 (0.076)	1,0,-1,1 (0.1)	-1,1,1,-1 (0.07)	N/A
5	N/A	1,0,-1,1,1 (0.06)	N/A	N/A

Table VI
THE PATTERN FOR DISTANCE OF THE MEMES TO THE INITIAL MEME, WHERE '1' INDICATES 'BECOMES MORE GENERAL', '-1' INDICATES 'BECOMES MORE SPECIFIC' AND '0' INDICATES 'NO CHANGE OF GENERALITY'. THE NUMBER IN () IS FREQUENCY.

from their ends. We show the resulting frequent patterns in Table III-B3. As we can see, unlike at the beginning, generality of memes at the end of the cycles is not stable; it continues to change without following any clear patterns.

Therefore, the overall pattern of cycles we observe is that memes shift up and down a bit at the beginning, quickly go into a stable stage, then shift up and down for quite a while before the end. We propose two explanations about this observation:

1) The shorter the cycle, the more reliable the cycle and the more likely the cycle reflects meaningful evolution of the memes. This is because the shorter the cycles are, the stronger and more clear the patterns are. The longer the cycles, the more random the patterns.
2) If the patterns we obtained truly reflect the evolution of memes, a common research flow revealed is this: once research on a topic is triggered, people tend to explore general topics, and eventually narrow down to a set of specific topics of similar generality. At that point there is hesitation about which topic merits further investigation, and a period of exploration follows, in which people randomly study extensions of current topics. Ultimately this exploration returns back to the initial topic.

IV. CONCLUSIONS

Research is the driving force for advancement of science and technology in our society, where it is common to see that some research memes persist while others fade quickly. Despite the radical differences between different research fields, there are many general patterns for research memes that can possibly help us understand how research memes evolve or help us predict the future research trends.

In this paper, we reported our research endeavors in unfolding evolving patterns of research memes. More specifically, we focus on a specific type of evolving process: from an initial meme, research attention shifts to related topics and then shifts back, in a cycle of shifts. We modeled this shifting process of the cycles with shortest paths in a graph constructed of nodes representing memes. We then proposed an efficient algorithm for mining position-specific frequent patterns using all shortest paths. Our experiments on two different applications – computer science research memes and biomedical research memes – revealed shift patterns from different perspectives.

We believe that further techniques for mining research meme evolution of this kind are important positive contributions that the data mining community can make; with continued work in this area we can develop new mining methods that speed innovation in research as well as shed light on possible recurring of some research topics.

ACKNOWLEDGMENT

This research is sponsored by Australian Research Council (ARC) Future Fellowship under Grant No.FT100100971.

REFERENCES

[1] R. Agrawal, R. Srikant, et al. Fast algorithms for mining association rules. In *Proc. 20th Int. Conf. Very Large Data Bases, VLDB*, volume 1215, pages 487–499. Citeseer, 1994.

[2] J. Andelin and N. C. Naismith. *Research Funding as an Investment: Can We Measure the Returns?* U.S. Government Printing Office, Washington, DC, 1986.

[3] B.V. Cherkassky, A.V. Goldberg, and T. Radzik. Shortest paths algorithms: theory and experimental evaluation. *Mathematical programming*, 73(2):129–174, 1996.

[4] E.W. Dijkstra. A note on two problems in connexion with graphs. *Numerische mathematik*, 1(1):269–271, 1959.

[5] D. Gusfield. *Algorithms on strings, trees, and sequences: computer science and computational biology*. Cambridge Univ Pr, 1997.

[6] J. Han, H. Cheng, D. Xin, and X. Yan. Frequent pattern mining: current status and future directions. *Data Mining and Knowledge Discovery*, 15(1):55–86, 2007.

[7] D. He and D. Parker. Learning the funding momentum of research projects. *Advances in Knowledge Discovery and Data Mining*, pages 532–543, 2011.

[8] D. He and Douglas S. Parker. Topic Dynamics: an alternative model of 'Bursts' in Streams of Topics. In *The 16th ACM SIGKDD Conference*, July 25-28, 2010.

[9] J. Kleinberg J. Leskovec, L. Backstrom. Meme-tracking and the dynamics of the news cycle. In *Proc. of the 15th ACM SIGKDD Conference, July, France*, 2009.

[10] Jon M. Kleinberg. Bursty and hierarchical structure in streams. *Data Min. Knowl. Discov.*, 7(4):373–397, 2003.

[11] Yunyue Zhu and Dennis Shasha. Efficient elastic burst detection in data streams. In *Proc. of the 9th ACM SIGKDD Conference, Washington, DC, USA*, pages 336–345, 2003.