Metro Maps of Science

Dafna Shahaf
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
dshahaf@cs.cmu.edu

Carlos Guestrin
Carnegie Mellon University
5000 Forbes Avenue
Pittsburgh, PA
guestrin@cs.cmu.edu

Eric Horvitz
Microsoft Research
One Microsoft Way
Redmond, WA
horvitz@microsoft.com

ABSTRACT

As the number of scientific publications soars, even the most enthusiastic reader can have trouble staying on top of the evolving literature. It is easy to focus on a narrow aspect of one's field and lose track of the big picture. Information overload is indeed a major challenge for scientists today, and is especially daunting for new investigators attempting to master a discipline and scientists who seek to cross disciplinary borders. In this paper, we propose metrics of influence, coverage, and connectivity for scientific literature. We use these metrics to create structured summaries of information, which we call *metro maps*. Most importantly, metro maps explicitly show the relations between papers in a way which captures developments in the field. Pilot user studies demonstrate that our method can help researchers acquire new knowledge efficiently: map users achieved better precision and recall scores and found more seminal papers while performing fewer searches.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; H.5 [**Information Interfaces and Presentation**]

Keywords

Metro maps, Information, Summarization

1. INTRODUCTION

"Distringit librorum multitudo" (the abundance of books is a distraction), said Lucius Annaeus Seneca; he lived in the first century.

A lot has changed since the first century, but Lucius' problem has only become worse. The surge of the Web brought down the barriers of distribution, and the scientific community finds itself overwhelmed by the increasing numbers of publications; relevant data is often buried in an avalanche of publications, and locating it is difficult.

Search engines have been relied upon in recent years for accessing the scientific literature, and investments have even been made to create special academic search and retrieval tools. However, the search and browsing experience might be best characterized as providing keyhole views onto the literature: while search engines are highly effective in retrieving scientific publications, the task of fitting those publications into a coherent picture remains difficult.

In contrast, we are interested in methods that *explicitly* show the relationships among publications in a way that captures the main developments in the discipline. We believe that such methods can allow a user to explore a new, complex topic and discover hidden connections effectively. We consider as a sample motivation the creation of valuable literature exploration tools that could help people entering a new field, such as new graduate students or experts reaching beyond their traditional disciplinary borders.

Several tools already exist for summarizing and visualizing scientific literature (see [Borner, 2010] for a compendium). However, the output of these systems is often not suitable for a starting researcher. Some systems' level of granularity is too coarse: Boyack et al. [2009] provide a graph-summary of chemistry research, where each node corresponds to a cluster of disciplines ('Biology-Zoology-Ecology'). Bassecoulard and Zitt [1999] produce a hierarchical graph, where nodes correspond to clusters of journals.

We believe that in order to allow researchers to understand how a field is organized, a finer level of granularity is needed. For this reason, we chose *papers* as our unit of analysis. Most current tools that work at this level of granularity provide visualizations of citation (or co-citation) networks, where papers are nodes [Chen, 2004; Dunne et al., 2010]. Importantly, edges between papers are based on *local* computation: the edges are selected because they pass some threshold, or belong to a spanning tree. In such methods, there is no notion of coherent *lines of research*. We believe that the notion of story lines is essential, and facilitates users' knowledge acquisition and comprehension of the frontier and evolutionary history of ideas in a discipline.

Several systems have attempted to create story lines, especially in the news domain [Swan and Jensen, 2000; Yan et al., 2011; Allan et al., 2001]. However, this style of summarization only works for simple stories, which are linear in nature. In contrast, research fields display a very non-linear behaviour: lines of research branch like a tangle of spaghetti with side stories, dead ends, and intertwining narratives. In order to explore these stories, one needs a *map* as a guide through unfamiliar territory.

The metro map metaphor has been used before to display abstract knowledge. For example, Nesbitt's map shows

KDD'12, August 12–16, 2012, Beijing, China.

interconnecting ideas running through his PhD thesis [Nesbitt, 2004]. However, these maps were all manually constructed. In this paper, we adapt the techniques of [Shahaf et al., 2012] (previously applied to news articles) to construct metro maps of scientific literature *automatically*. Our main contributions are as follows:

- Formalizing metrics characterizing good metro maps, taking advantage of the additional structure encoded in the scientific domain:
  - Characterizing the probability that ideas in two papers stem from a common source, then using this notion to define coherence of research lines.
  - Quantifying the impact of one paper on the corpus.
  - Proposing a notion of connectivity that captures how different lines of research can still interact with each other, despite not intersecting.
- Providing efficient methods with theoretical guarantees to compute these metrics and find a diverse set of high-impact, coherent research lines and their interactions.
- Integrating user preferences into our framework by providing an appropriate user-interaction model.
- Performing validation studies with users that highlight the promise of the methodology. Our method outperforms popular competitors.

2. OBJECTIVE

We first review the desired properties of a metro map, following the criteria outlined in [Shahaf et al., 2012]. We shall briefly present these criteria, motivate and formalize them. Later, we present a principled approach to constructing maps that optimizes tradeoffs among these criteria. Before we begin, we formally define metro maps.

Definition 2.1 (Metro Map [Shahaf et al., 2012]). A *metro map* $\mathcal{M}$ is a pair (G, Π), where $G = (V, E)$ is a directed graph and Π is a set of paths in G. We refer to paths as *metro lines*. Each $e \in E$ must belong to at least one line.

The vertices V correspond to scientific papers, and are denoted by $docs(\mathcal{M})$. The lines of Π correspond to aspects of the field. A key requirement is that each line is *coherent*: following the papers along a line should give the user a clear understanding of the evolution of a story.

Coherence is crucial for good maps, but is it sufficient as well? In order to put this matter to a test, we computed maximally coherent lines for the set of papers returned in response to the query 'support vector machines' (using methods detailed below). The results were discouraging. While the lines were indeed coherent, they were not *important*. Many of the lines revolved around narrow topics; many focused on a single research group, never expanding beyond it.

The example suggests that maximizing coherence does not guarantee good maps. Instead, the key challenge is balancing coherence and **coverage**: in addition to being coherent, lines should cover topics that are important to the user.

Finally, a map is more than just a set of lines; there is information in its *structure* as well. Publications offer a rich palette of interaction possibilities: assumption, affirmation, contrast, methodology, related work, and more. Therefore, our last property is **connectivity**. The map's **connectivity** should convey the underlying structure of the field, and how different lines of research interact with each other.

In Sections 3-5, we formalize **coherence**, **coverage** and **connectivity**. In Section 6, we explore tradeoffs among them and combine them into a single objective function to guide the construction of maps.

• Europe weights possibility of debt default in Greece	• Europe weights possibility of debt default in Greece
• Why Republicans don't fear a debt default	• Europe commits to action on Greek debt
• Italy; The Pope's leaning toward Republican ideas	• Europe union moves towards a bailout of Greece
• Italian-American groups protest 'Sopranos'	• Greece set to release austerity plan
• Greek workers protest austerity plan	• Greek workers protest austerity plan
Chain A	**Chain B**

3. COHERENCE

How should we measure coherence for a chain of papers? We rely on the notion of coherence developed in Connect-the-Dots (CTD) [Shahaf and Guestrin, 2010]. In the following, we briefly review this approach.

In order to define coherence, a natural first step is to measure similarity between each two consecutive papers along the chain. As a single bad transition can destroy the coherence of an entire chain, we measure the strength of the chain by the strength of its *weakest link*.

However, this simple approach can produce poor chains. Consider, for example, Chain A above. The transitions of Chain A are all reasonable when examined out of context. The first two articles are about debt default; the second and third mention Republicans. Despite these local connections, the overall effect is associative and incoherent.

Now, consider Chain B. This chain has the same endpoints, but it is significantly more coherent. Let us take a closer look at these two chains. Figure 1 shows word appearance along both chains. For example, the word 'Greece' appeared throughout Chain B. It is easy to spot the associative flow of Chain A in Figure 1. Words appear for short stretches; some words appear, then disappear and reappear. Contrast this with Chain B, where stretches are longer and transitions between documents are smoother. This observation motivates our definition of coherence.

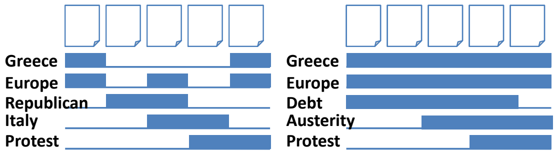

Figure 1: Word patterns in Chain A (left) and Chain B (right). Bars correspond to the appearance of a word in the articles depicted above them.

We represent documents as vectors of concepts (for the sake of presentation, assume concepts $\mathcal{C}$ are words). Given a chain of papers $(p_1, ..., p_n)$, we first score each transition $p_i \to p_{i+1}$ by the number of concepts both articles share:

$$\mathit{Coherence}(p_1, ..., p_n) = \min_{i=1...n-1} \sum_{c \in \mathcal{C}} \mathbb{1}(c \in p_i \cap p_{i+1})$$

However, word appearance is too noisy. Articles must use the exact same words; synonyms are treated as unrelated. Also, all words are treated equally: the word 'Greece' is as important as the word 'today'.

Therefore, we replaced the indicator function $\mathbb{1}(\cdot)$ with a notion of *influence* of concept c in a transition. Intuitively, $\mathit{Influence}(p_i, p_j \mid c)$ is high if (1) both documents are highly connected, and (2) c is important for the connectivity. Note that c does not have to appear in either of the documents.

After the introduction of influence, the objective becomes:

$$\mathit{Coherence}(p_1, ..., p_n) = \min_{i=1...n-1} \sum_c \mathit{Influence}(c \mid p_i, p_{i+1})$$

This objective guarantees good transitions, but associative chains like Chain A can still score well. However, these chains need to use many words in order to achieve high scores, as many of their transitions use a unique set of words. On the other hand, coherent chains (like Chain B) can often be characterized by a *small* set of words, which are important throughout many of the transitions.

Therefore, instead of summing $\mathit{Influence}(c \mid p_i, p_{i+1})$ over all concepts, the problem is transformed into an optimization problem, where the goal is to choose a small set of concepts (called 'active'), and score the chain based only on these concepts. Constraints on possible activations enforce a small number of words and smooth transitions, imitating the behaviour of Figure 1 (right). Formally,

$$\mathit{Coherence}(p_1, ..., p_n) = \max_{\mathit{activations}} \min_{i=1...n-1}$$
$$\sum_c \mathit{Influence}(c \mid p_i, p_{i+1}) \mathbb{1}(c \text{ } \mathit{active\ in}\ p_i, p_{i+1}) \qquad (3.1)$$

Finally, the coherence of a map is defined as the minimal coherence across its lines Π.

3.1 Coherence for Scientific Papers

The coherence notion of [Shahaf and Guestrin, 2010] (Definition 3.1) was developed for the news domain, and relied exclusively on article *content*. It was designed to use very basic features, namely words. However, the simplicity of the representation can sometimes result in incoherent chains. To illustrate the problem, consider the following three papers:

*p*1*:* **Multiagent planning with factored MDPs / Guestrin et al** / *NIPS '01*

*p*2*:* **Timing and power issues in wireless sensor networks / Aakvaag et al** / *ICPP '05*

*p*3*:* **Social network analysis for routing in disconnected delay-tolerant manets / Daly et al** / *MobiHoc '07*

These papers share many words, such as 'network', 'probability' and 'cost', and thus can achieve a good coherence score. However, they clearly do not follow a coherent research line. The problem may be alleviated by higher-level features (e.g., distinguishing between different uses of 'network'); in this section, we choose instead to take advantage of the side information provided by the citation graph, and define a coherence notion more suited for scientific papers.

Upon close examination, our original coherence notion (Definition 3.1) is composed of two main ideas: computing the influence of concepts on transitions, and choosing a small set of active concepts that captures the story well. While the latter idea seems domain-independent, computing influence may benefit from the additional structure of the citation graph.

The citation graph explicitly captures the way papers influence each other: the content of a publication is often affected by cited work, the authors' prior work and novel insights. The influence notion proposed in BKS [El-Arini and Guestrin, 2011] captures exactly this behaviour. In BKS, the authors define a directed, acyclic graph G_c for every concept c in the corpus. Nodes represent papers that contain c and the edges represent citations and common authorship.

To capture the degree of influence, BKS defines a weight $\omega_{u,v}$ for each edge $u \rightarrow v$ in G_c, representing the probability of direct influence from paper u to paper v with respect to concept c. Some probability is assigned to 'novelty', the case that concept c in paper v was novel.

Given a concept-specific weight for each edge in G_c, BKS defines a probabilistic, concept-specific notion of influence between any two papers in the document collection:

Definition 3.1 (Direct Influence [El-Arini and Guestrin, 2011]). Let G_c^r be a random subgraph of G_c, where every edge $u \rightarrow v$ is included in G_c^r with probability $\omega_{u,v}$. The influence between papers p_i and p_j w.r.t. c is the probability there exists a directed path in G_c^r between p_i and p_j.

The BKS notion of influence has many attractive properties: it is simple, and it appears to capture the way ideas travel along the citation graph. However, using it for coherence severely limits the chains we can hope to identify. According to definition 3.1, the only pairs of papers that can have influence between them are ancestor-descendant pairs in G_c. Therefore, chains with high influence are likely to contain only papers that directly build on top of one another, especially papers by the same authors.

Consider papers $p2$ and $p3$ from above. Their notion of 'network' is similar, but there is no direct path from $p2$ to $p3$ in the corresponding graph. To mitigate this problem, we introduce a different notion of influence. Rather than requiring that p_i influence p_j, we are only interested in whether concept c in p_i and concept c in p_j refer to the same idea. To capture this property, we modify the notion of influence:

Definition 3.2 (Ancestral Influence). The influence between papers p_i and p_j with respect to concept c is the probability p_i and p_j have a common ancestor in G_c^r.

See Figure 2 for an illustration of the difference between direct influence (left) and ancestral influence (right). In order for p_i to have direct influence on p_j, there has to be a path from p_i to p_j. In order for p_i to have ancestral influence on p_j, it is sufficient that they have a common ancestor in the graph. The ancestor can also be p_i itself.

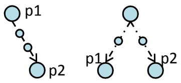

Figure 2: Direct (left) vs. ancestral influence (right).

As for $p2$ and $p3$: with no direct path among them, their direct influence is zero. However, as both cite Perkins' 1999 networks paper, their ancestral influence is non-zero.

4. COVERAGE

In addition to coherence, we need to ensure that the map has high coverage. Before defining coverage of a map, we need to understand which elements we wish to cover.

4.1 What to cover?

In [Shahaf et al., 2012], we only had the articles' content to rely upon, and thus the covered elements were *concepts*. We denoted the amount an article p covered a concept c by $\mathit{cover}_p(c)$, and looked for a set of articles that, when combined, achieved high coverage for many important concepts.

However, when we applied the same technique to scientific papers, we encountered a problem: papers with similar content may appear exchangeable w.r.t. their coverage, but they will not necessarily be equivalent in the user's eyes. For example, the user may notice that the papers aim at different communities, or that one paper is more seminal than the other. Consider the following two papers:

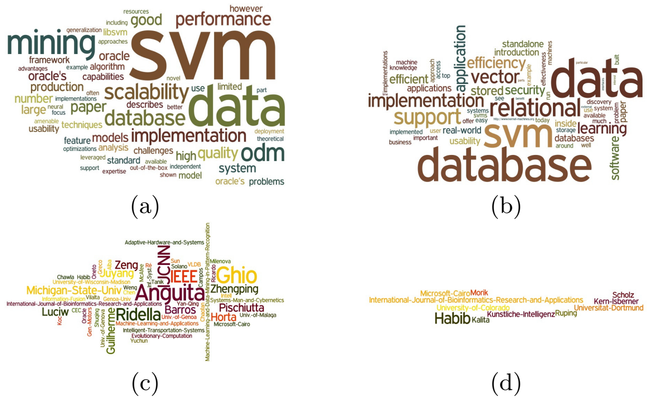

Figure 3: Tag clouds for $p1$ and $p2$. The size of a word is proportional to its frequency. (a-b) $p1$ and $p2$'s content, respectively. (c-d) Venues and authors of papers affected by $p1$ and $p2$, respectively. Note that (a) and (b) are very similar, but (c) and (d) are not.

p1: **SVM in Oracle database 10g: Removing the barriers to widespread adoption of support vector machines / Milenova et al**
VLDB '05 Proceedings of the 31st International Conference on Very Large Data Bases

p2: **Support Vector Machines in Relational Databases / Rüping**
SVM '02 Proceedings of the First International Workshop on Pattern Recognition with Support Vector Machines

The content of $p1$ and $p2$ is similar. Figures 3(a)-(b) display the papers as tag clouds: both papers share many of their important words ('data','database', 'svm', 'implementation'). Numerous other words have a closely related match ('performance'/ 'efficiency', 'Oracle'/ 'relational database').

One way to distinguish between the aforementioned papers is to examine their impact. Figures 3(c)-(d) show tag clouds of authors and venues for papers *citing* $p1$ and $p2$. Figure 3(c) has more words than 3(d), implying that $p1$ has affected more unique authors and venues than $p2$. Interestingly, despite the similar content of the papers, there is almost no intersection between the papers citing them; only a single paper cites both (Mona Habib from Microsoft Cairo).

Based on this intuition, we propose to use the papers themselves as elements of coverage. A paper p should cover itself and the papers it has had impact on. By this definition, a high-coverage set of papers consists of papers that, when combined, had impact on a large portion of the corpus.

The idea that a paper covers its descendants (and not its ancestors) may seem counterintuitive at first. After all, how can a paper cover future contributions? Nevertheless, we believe that examining a paper's ancestors merely helps understanding the context in which the paper was written, while its descendants truly reveal the gist of its contribution.

4.2 Coverage of a single paper

4.2.1 Desiderata

We would like papers to cover their descendants. Instead of a hard, binary notion of coverage, we prefer a softer notion, allowing us to express that descendants are covered to various degrees (depicted as a gradient in Figure 4a).

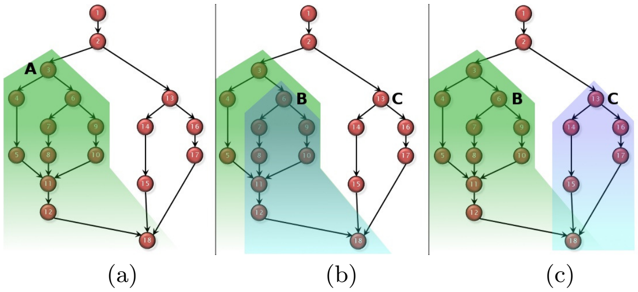

Figure 4: A simple citation graph. Edges traverse in the direction of impact, from cited to citing paper. (a) Coverage of document A. Gradient indicates different degrees of coverage. (b-c) The effect of adding papers B and C (respectively) to paper A. Since B's descendants are already covered to some extent by A, we prefer C.

Let us concentrate on the degree to which paper p covers its descendant q, $cover_p(q)$. In order to evaluate the impact that p had on q, we examine the way q is connected to p in the citation graph. Intuitively, if q can be reached from p by many paths, p had a high impact on q. Since impact is diluted with each step, shorter paths are more important than longer ones.

Before we devise a coverage formulation based on paths between p and q, we consider another point: impact is not necessarily transitive. Consider, for example, Figure 5. The figure outlines a (small) fraction of the descendants of Nicolo Cesa-Bianchi's paper, 'How to Use Expert Advice'. As before, edges indicate citation. A snippet from the citation text appears by each edge.

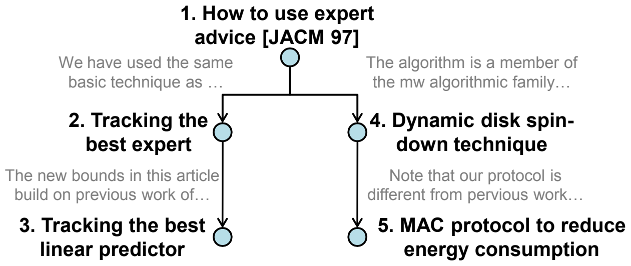
1. How to use expert advice [JACM 97]
We have used the same basic technique as …
The algorithm is a member of the mw algorithmic family…
2. Tracking the best expert
4. Dynamic disk spin-down technique
The new bounds in this article build on previous work of…
Note that our protocol is different from pervious work…
3. Tracking the best linear predictor
5. MAC protocol to reduce energy consumption

Figure 5: Two branches in the citation graph. The left branch is coherent; the right one is not.

The left branch of Figure 5 revolves around Online Learning Theory. The papers in this branch (#2 and #3) build on top of each other. Intuitively, the root paper had impact on both of them. In contrast, the right branch is more difficult to follow. Both descendants deal with extending the battery life of devices, but while paper #4 is a direct application of the root paper, paper #5 is not. In fact, when #5 cites #4, the citation reads 'Note that our protocol is different from previous work'. In other words, paper #5 is no longer relevant to the root node, and should not be covered by it.

The difference between the two branches can be captured by the coherence notion of Section 3: The left branch is much more coherent than the right one. Based on that intuition, we only want a paper to cover the descendants that can be reached by a *coherent* path. Unlike Section 3, we are only interested in direct-influence coherent chains (Definition 3.1), as they model the true impact of a paper.

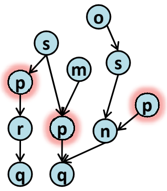

Figure 6: Coherence graph. Nodes represent papers (names appear inside). Paths represent coherent chains. Each paper may have multiple corresponding vertices: the highlighted vertices are all copies of paper p.

4.2.2 Formulation

In the previous section, we provided desiderata for $cover_p(q)$: coverage is high if there are many short and coherent paths between p and q. In order to formalize this idea, we employ the technique of random walks.

Let q be a paper. Consider a walk from q to its ancestors, taking only coherent paths into account. At each step, the walker either terminates (with probability α), or chooses an ancestor uniformly at random among the coherent paths that extend the current walk. If there are many short, coherent paths between p and q, there is a high probability that the walk reaches p before termination. We denote this probability by $cover_p(q)$.

Let us formalize this intuition now. Since we only consider coherent paths, it is more convenient to formulate coverage in terms of walks performed directly on a *coherence graph* G. A coherence graph is a graph representing all coherent chains in the domain (See Figure 6 for an example. In Section 7.1 we explain how to encode the graph compactly). Each vertex v of G corresponds to a single paper, which we denote $paper(v)$; each paper p may have multiple corresponding vertices in G, which we denote $copies(p)$. In Figure 6, $copies(p)$ are highlighted.

Let G be a coherence graph. For each paper q, we construct the graph G_q by reversing the direction of all edges in G and adding an additional vertex, v_q. v_q is the starting vertex of our walk. We connect v_q to each vertex of G which corresponds to paper q, $copies(q)$. This way, a walk from v_q will always proceed to a copy of q, and then to its ancestors in the coherence graph G. Since the graph is a DAG, the probability that a walk reaches vertex v is easy to compute. We first compute a topological ordering on G_q, and compute the probabilities in this order:

$$cover_v(q) = \begin{cases} P(v_q \to v), & v \in copies(q) \\ (1-\alpha)\cdot(\sum_{u:u\to v} P(u \to v)\cdot cover_u(q)), & \text{o/w} \end{cases}$$

where $P(u \to v)$ is the probability the walker chose to go from vertex u to vertex v. We want the walker to choose uniformly among the coherent paths that extend the current walk; in other words, we want to bias the walker towards ancestors that participate in many coherent paths. Therefore, we compute for each vertex v the number of coherent paths that end in v, $\#Path(v)$. For example, the number of paths that end in the vertex marked 'n' in Figure 6 is two (o,s,n and p,n). Since G_q is a DAG, computing the number of paths takes polynomial time. The probability that the walker chooses to go from vertex v to vertex u is proportional to $\#Path$:

$$P(u \to v) = \frac{\#Path(v)}{\sum_{w:u\to w} \#Path(w)}$$

We now have a coverage notion for vertices of $\mathbb{G}$. However, we are interested in a coverage notion for *papers*. In order to compute the coverage of paper p, we need to sum up the scores of all vertices in $copies(p)$:

$$cover_p(q) = \sum_{v\in copies(p)} cover_v(q)$$

This score corresponds to the probability of reaching p before termination. In particular, since p can never appear more than once along a path in G, this score always less than 1.

4.3 Map Coverage

Now that we have defined coverage of a single document, let us define coverage of a map. In order to encourage diversity, we view set coverage as a sampling procedure: each paper p_i in the map tries to cover document q with probability $cover_{p_i}(q)$. The coverage of q is the probability at least one of the documents succeeded.

$$cover_{\mathcal{M}}(q) = 1 - \prod_{p_i \in docs(\mathcal{M})} (1 - cover_{p_i}(q))$$

Thus, if the map already includes papers which cover q well, $cover_{\mathcal{M}}(q)$ is close to 1, and adding another paper which covers q well provides very little extra coverage of q. This encourages us to pick papers which cover new areas of the graph, promoting **diversity**.

Figures 4b and 4c illustrate this idea. Suppose we already have paper A in our map, and we need to choose between papers B and C, whose content is similar. Figures 4b and 4c show the effect of choosing B and C, respectively. Since B's descendants have already been covered by A, we would prefer to choose C. (Note that since our coverage is soft, choosing B will still provide gains in coverage.)

We now have a way to measure how well a map covers a single paper. Finally, we want to measure how well a map covers the entire *corpus*. Remember, our goal is to ensure that the map touches upon **important** aspects of the corpus. Therefore, we first assign weights λ_q to each paper q, signifying the importance of the paper. We model the amount $\mathcal{M}$ covers the corpus as the weighted sum of the amount it covers each paper:

$$Cover(\mathcal{M}) = \sum_q \lambda_q cover_{\mathcal{M}}(q)$$

The weights cause *Cover* to prefer maps which cover important papers. They offer a natural mechanism for personalization: With no prior knowledge about the user's preferences, we set all of the weights to 1. This is equivalent to asking for a map which covers as much of the corpus as possible. In Section 10 we discuss learning weights from user feedback, resulting in a personalized notion of coverage.

5. CONNECTIVITY

A map is more than just a set of lines; there is information in its *structure* as well. The map's **connectivity** should convey the underlying structure of the story, and how different aspects of the story interact with each other.

In [Shahaf et al., 2012] we simply define connectivity as the number of lines of Π that intersect:

$$Conn(M) = \sum_{i<j} \mathbb{1}(\pi_i \cap \pi_j \neq \emptyset)$$

Unfortunately, this simple objective does not suffice in the scientific domain. Consider the two chains in Figure 7: the top chain describes the progress of margin classifiers – from perceptrons, through linear SVMs, to kernel machines. The bottom chain describes the progress of face-recognition

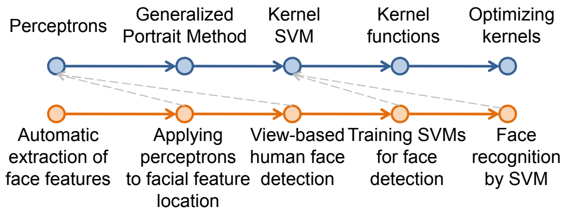
Perceptrons
Generalized Portrait Method
Kernel SVM
Kernel functions
Optimizing kernels
Automatic extraction of face features
Applying perceptrons to facial feature location
View-based human face detection
Training SVMs for face detection
Face recognition by SVM

Figure 7: Two coherent chains (theory of SVMs, application of SVM to vision). The chains do not intersect, yet are related: the application chain uses tools from the theory chain. Dashed gray lines indicate impact.

challenge problems in vision: from facial feature location, through face detection, to face recognition. Both chains are clearly related; the vision papers use techniques from the theory chain. However, there is no way to find an article that would belong to both chains, unless we sacrifice coherence considerably. As a result, maps that optimize the aforementioned connectivity notion are often disconnected.

Finding papers that would belong to both chains may be difficult, but we can easily find theory papers that have had a big impact on vision papers. For example, some of the vision papers in Figure 7 directly cite papers from the theory chain. These citations are depicted as dashed lines.

Figure 7 motivates us to prefer a softer notion of intersection. Rather than requesting that the lines intersect, we also accept lines which are related to each other:

$$Conn(M) = \sum_{i<j} \mathbb{1}(\pi_i \cap \pi_j \neq \emptyset) \ + \gamma \cdot cover(\pi_i, \pi_j)$$

where $cover(\pi_i, \pi_j)$ is the maximal $cover_p(q)$ for $p \in \pi_i, q \in \pi_j$, or vice versa. We choose to use the maximum (instead of sum) in order to encourage connections between as many pairs of lines as possible. Scoring all the connections between π_i and π_j may lead to maps where only a few lines are very well-connected, and the rest are disconnected. The parameter γ is chosen empirically.

This softer notion of intersection is especially suited to scientific literature. Publications offer a rich palette of interaction possibilities, such as affirmation, criticism, contrast, methodology, and related work. Exposing the relationships between two lines of research can prove extremely valuable to researchers.

6. JOINT OBJECTIVE FUNCTION

Now that we have formally defined our three properties, we can combine them into one objective function. We need to consider tradeoffs among these properties: for example, maximizing coherence often results in repetitive, low-coverage chains. Maximizing connectivity encourages choosing similar chains, resulting in low coverage as well. Maximizing coverage leads to low connectivity, since there is no reason to re-use an article for more than one line.

The objective of [Shahaf et al., 2012] applies to the scientific domain as well. We include it here for completeness. For a full discussion, please refer to the paper.

Problem 6.1. Given a set of candidate documents $\mathcal{D}$, find a map $\mathcal{M} = (G, \Pi)$ over $\mathcal{D}$ which maximizes $Conn(\mathcal{M})$ s.t. $Coherence(\mathcal{M}) \geq \tau$ and $Cover(\mathcal{M}) \geq (1-\epsilon)\kappa$, where κ is the maximal coverage across maps with coherence $\geq \tau$ and ϵ is given.

There are several ways to restrict the size of $\mathcal{M}$; we chose to restrict $\mathcal{M}$ to K lines of length at most l. Alternatively, since some stories are more complex than others, one may prefer to add lines until coverage gains fall below a threshold.

7. ALGORITHM

In this section, we outline our approach for solving Problem 6.1. We adapt the algorithm of [Shahaf et al., 2012] to solve the problem. In the following we review the algorithm, highlighting the main differences.

We start by addressing the **coherence** constraint: In Section 7.1 we represent all coherent chains as a graph. In Section 7.2 we use this graph to find a set of K chains that maximize coverage; in Section 7.3, we increase connectivity without sacrificing coverage.

7.1 Representing all coherent chains

In order to pick good chains, we first wish to list all possible candidates. However, representing all chains whose coherence is at least τ is a non-trivial task. The number of possible chains may be exponential, and therefore it is infeasible to enumerate them all, let alone evaluate them.

The algorithm of [Shahaf et al., 2012] employs a divide-and-conquer approach to the problem, constructing long chains from shorter ones. This allows us to compactly encode many candidate chains in a graph structure which we call a *coherence graph*. $\mathbb{G}$ is a compact representation of the graph displayed in Figure 6. Vertices of $\mathbb{G}$ correspond to short coherent chains, and there is a directed edge between each pair of vertices which can be conjoined and maintain coherence. Importantly, this property is transitive: every path in $\mathbb{G}$, no matter how long, represents a coherent chain.

The only change in the algorithm lies in the computation of influence. Direct influence and ancestor influence are instances of the k-terminal reliability problem [Ball, 1986], which is $\#P$-complete, so we cannot hope for a polynomial-time solution. Instead, we apply approximations.

In BKS, the authors presented a deterministic, linear-time dynamic programming heuristic for calculating direct influence. This heuristic is based on the assumption that the paths between two nodes are independent of each other. Unfortunately, this assumption is too strong for ancestor influence. The paths between $p1, p2$, and possible ancestors are often dependent, and treating them as independent results in significant errors. Instead, we employ a simple Monte Carlo sampling method with theoretical guarantees (BKS also proposed a similar sampling approach).

In order to calculate m values with (ϵ, δ)-approximation guarantees (where ϵ and δ denote the upper bound of relative error and failure probability), we need $O(\frac{1}{\delta^2} \log(\frac{m}{\delta}))$ samples. m is the number of document-pairs with a common ancestor in the graph. In the worst case, m is quadratic in the number of papers (in practice, it is often much smaller). Therefore, the number of samples needed is logarithmic in the number of papers. Also note that influences can be precomputed once and stored for future use.

7.2 Finding a high-coverage map

After representing all coherent chains as a graph $\mathbb{G}$, we wish to find a set of chains which maximize coverage, subject to map size constraints.

Problem 7.1. Given a coherence graph $\mathbb{G}$, find paths $\pi_1, ..., \pi_K$ in $\mathbb{G}$, $|docs(\pi_i)| \leq l$ that maximize $Cover(docs(\bigcup_i \pi_i))$.

We use the coverage-maximization algorithm of [Shahaf et al., 2012] to find a high-coverage map. The proof relies on formulating the problem in terms of *orienteering*. Orienteering problems are motivated by maximizing a function

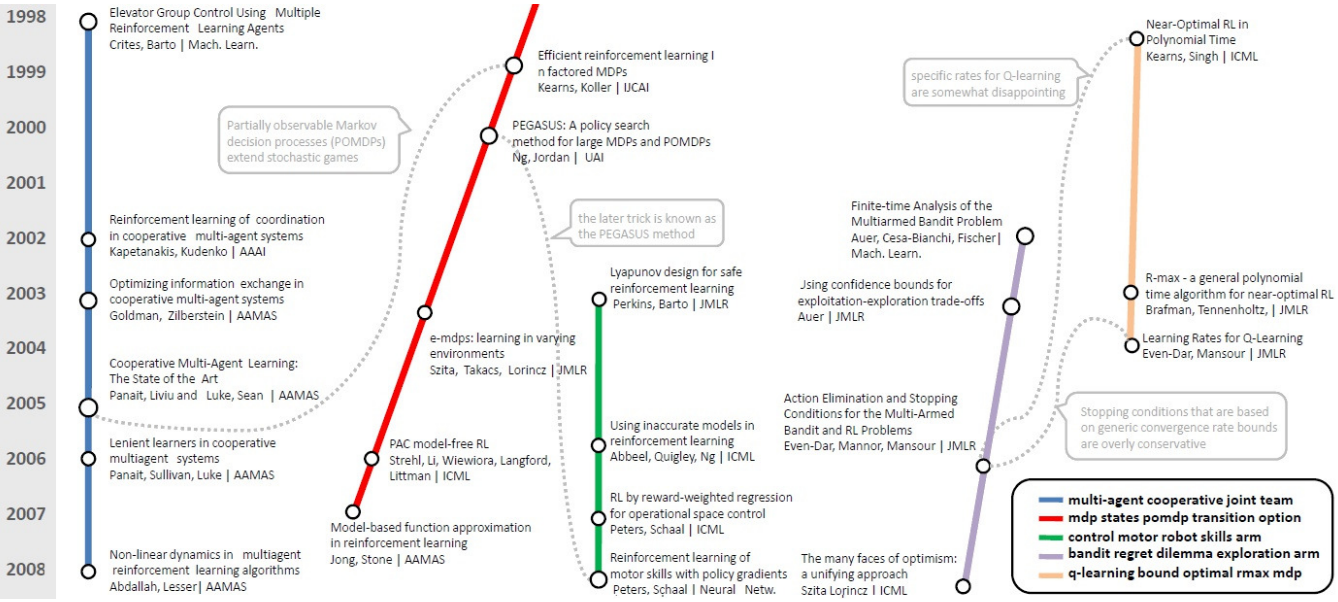
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
Elevator Group Control Using Multiple Reinforcement Learning Agents Crites, Barto | Mach. Learn.
Reinforcement learning of coordination in cooperative multi-agent systems Kapetanakis, Kudenko | AAAI
Optimizing information exchange in cooperative multi-agent systems Goldman, Zilberstein | AAMAS
Cooperative Multi-Agent Learning: The State of the Art Panait, Liviu and Luke, Sean | AAMAS
Lenient learners in cooperative multiagent systems Panait, Sullivan, Luke | AAMAS
Non-linear dynamics in multiagent reinforcement learning algorithms Abdallah, Lesser| AAMAS
Partially observable Markov decision processes (POMDPs) extend stochastic games
Efficient reinforcement learning I n factored MDPs Kearns, Koller | IJCAI
PEGASUS: A policy search method for large MDPs and POMDPs Ng, Jordan | UAI
e-mdps: learning in varying environments Szita, Takacs, Lorincz | JMLR
PAC model-free RL Strehl, Li, Wiewiora, Langford, Littman | ICML
Model-based function approximation in reinforcement learning Jong, Stone | AAMAS
the later trick is known as the PEGASUS method
Lyapunov design for safe reinforcement learning Perkins, Barto | JMLR
Using inaccurate models in reinforcement learning Abbeel, Quigley, Ng | ICML
RL by reward-weighted regression for operational space control Peters, Schaal | ICML
Reinforcement learning of motor skills with policy gradients Peters, Sçhaal | Neural Netw.
Finite-time Analysis of the Multiarmed Bandit Problem Auer, Cesa-Bianchi, Fischer| Mach. Learn.
Jsing confidence bounds for exploitation-exploration trade-offs Auer | JMLR
Action Elimination and Stopping Conditions for the Multi-Armed Bandit and RL Problems Even-Dar, Mannor, Mansour | JMLR
The many faces of optimism: a unifying approach Szita Lorincz I ICML
specific rates for Q-learning are somewhat disappointing
Near-Optimal RL in Polynomial Time Kearns, Singh | ICML
R-max - a general polynomial time algorithm for near-optimal RL Brafman, Tennenholtz, | JMLR
Learning Rates for Q-Learning Even-Dar, Mansour | JMLR
Stopping conditions that are based on generic convergence rate bounds are overly conservative
🟦 multi-agent cooperative joint team
🟥 mdp states pomdp transition option
🟩 control motor robot skills arm
🟪 bandit regret dilemma exploration arm
🟧 q-learning bound optimal rmax mdp

Figure 8: Part of the map computed for the query 'Reinforcement Learning'. The map depicts multiple lines of research (see legend at the bottom). Interactions between the lines are depicted as dashed gray lines, and relevant citation text appears near them.

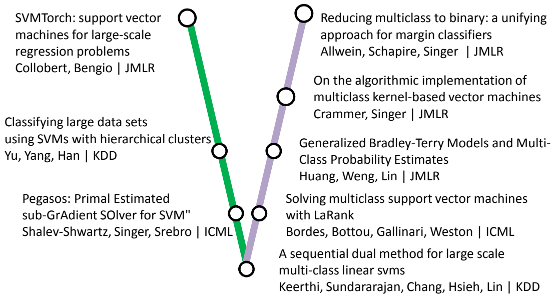
SVMTorch: support vector machines for large-scale regression problems Collobert, Bengio | JMLR
Classifying large data sets using SVMs with hierarchical clusters Yu, Yang, Han | KDD
Pegasos: Primal Estimated sub-GrAdient SOlver for SVM" Shalev-Shwartz, Singer, Srebro | ICML
Reducing multiclass to binary: a unifying approach for margin classifiers Allwein, Schapire, Singer | JMLR
On the algorithmic implementation of multiclass kernel-based vector machines Crammer, Singer | JMLR
Generalized Bradley-Terry Models and Multi-Class Probability Estimates Huang, Weng, Lin | JMLR
Solving multiclass support vector machines with LaRank Bordes, Bottou, Gallinari, Weston | ICML
A sequential dual method for large scale multi-class linear svms Keerthi, Sundararajan, Chang, Hsieh, Lin | KDD

Figure 9: A segment of a map computed for the query SVM/ Support vector machine, showing the intersection of two lines: multi-class SVMs and large-scale SVM. In the interest of space, we condensed the timeline.

of nodes visited during a tour, subject to a budget on tour length. The [Shahaf et al., 2012] coverage notion is submodular, so we applied the Submodular orienteering algorithms of [Chekuri and Pal, 2005] to the problem.

In order to adapt the algorithm to the scientific domain, we changed only the way coverage is computed (see Section 4). Note that the new coverage notion is submodular. Figure 4 provides the intuition for that: adding a paper to a smaller set of papers helps more than adding it to a larger set (diminishing returns). Therefore, we can use the same submodular orienteering algorithm with the new coverage notion, and achieve the same guarantees.

7.3 Increasing connectivity

We now know how to find a high-coverage, coherent map. Our final step is to increase connectivity without sacrificing (more than an ϵ-fraction of) coverage.

In order to increase connectivity, we apply a local-search technique. It starts from map $\mathcal{M}_0$, and takes steps in the search space by applying local moves. Each local move replaces a single line in Π. At iteration i, we consider each path $p \in \Pi_{i-1}$. We hold the rest of the map fixed, and try to replace p by p' that increases connectivity (Section 5) and does not decrease coverage. At the end of the iteration, we pick the best move and apply it, resulting in $\mathcal{M}_i$. The full details of the algorithm are in [Shahaf et al., 2012].

8. EXAMPLE MAPS

Figure 8 shows a part of a map computed for the query 'Reinforcement Learning'. As can be seen, the map depicts multiple lines of research: MDPs, robotics and control, multi-agent cooperation, bounds and analysis, and exploration-exploitation tradeoffs. The map shows how the MDP line affects the multi-agent and robotics lines, and how the exploration-exploitation line interacts with the analysis line. Those relations are depicted as gray dashed paths. Note that the map does not capture all the interactions; for example, connections between MDPs and the analysis line are not captured.

As mentioned in Section 5, intersection is rare for broad queries. Figure 9 shows one such intersection between two lines in the SVM map. One line is about large-scale SVMs, the other is about multi-class SVMs. The lines intersect at Keerthi's paper about large scale multi-class linear SVMs.

9. USER STUDY

In our user study, we evaluated the effectiveness of metro maps in aiding users navigate, consume, and integrate different aspects of a specific, multi-faceted information need.

Evaluating metro maps in the scientific domain poses some significant challenges. Since the metro-map output is unique, we cannot conduct a double-blind comparison study, as subjects inevitably differentiate between the different systems. Therefore, we cannot have a within-subject study, but are instead forced to choose a between-subject design. This design, in itself, causes a new problem: since we need a different group of participants for each condition tested (metro-map or competitor), we cannot tailor the query to users.

Rather, we have to find a single domain such that all of our participants will (1) be able to read scientific publications in that domain and (2) not know the domain well in advance.

We recruited 30 participants from our university. All participants were graduate students with background in Machine Learning or related fields. The domain we chose was Reinforcement Learning. The machine learning background of the participants was enough to make them comfortable with the subject, but none of them had conducted research in the field or studied it extensively.

We asked participants to imagine themselves as first-year graduate students embarking on a research project in Reinforcement Learning. The participants were asked to conduct a quick literature survey. In particular, they were asked to update a survey paper from 1996: identify up to five research directions that should be included in the updated survey, and list a few relevant papers for each direction. We recorded participants' browsing histories, and took a snapshot of their progress every minute. We limited their time to 40 minutes to simulate a quick first pass on papers.

We used the ACM dataset to compute a map for the query 'Reinforcement learning'. The dataset contains more than 35,000 papers from ACM conferences and journals. As the number of papers is relatively small, scalability was not an issue. We extracted features as described in [El-Arini and Guestrin, 2011]. We had two conditions, GS and MP+GS: In GS, participants were allowed to use Google Scholar [1], a search engine that indexes scholarly literature. In the second condition (MP+GS), participants were given the pre-computed metro map, and asked to pretend that they stumbled upon it; they were not instructed how to use the map. In addition to the map, the participants could access Google Scholar.

We also included two simulated conditions in the study, MP and WK: In MP, we pretended our map was the user's output, and listed all of its papers. In WK, we used references from the Wikipedia article about reinforcement learning.

We decided to compare against Wikipedia and Google Scholar since they represent two of the most popular starting points for research queries today. Other systems we considered including in the comparative analysis were either unavailable for download, or very restricted in the span of the scientific domain represented.

Before grading, we discarded data from four participants. One did not understand the task, and wrote a (nice) essay about reinforcement learning. The others, despite visiting many web pages, listed less than 5 papers when time ran out.

We had an expert judge evaluate the results of the rest of the participants. We combined all of the papers that users had entered into one list. Each entry includes the paper's information and URL. In addition, we listed the labels that the users supplied for each paper. The judge did not know the method used to find the papers.

Our expert judge scored the papers on a 3-point scale:
0 – Irrelevant, 1 – 1: Relevant, 2 – Seminal. Each label was given a 0-1 score, based on whether it was a good match to the paper. The results are summarized below.

9.1 Results and Discussion

9.1.1 Information collection patterns

Avg:	Pages visited	Papers listed	Visited/Listed
GS	46	12.2	4.51
MP+GS	36.3	9.75	3.79

The table shows the average number of web pages visited throughout the session, the average number of papers listed by the user, and the average ratio of pages visited to papers listed. GS users visited more pages and listed more papers on average. However, when looking at the average ratio, only one out of 4.5 pages visited by GS users was added to their list, while MP+GS added one out of 3.8. In other words, the map users were more focused: they may have visited less pages, but they found these pages satisfactory.

9.1.2 Precision

Users' satisfaction level is important, but the real test is the expert's opinion. The next table shows the average normalized scores given by the judge: For each user, we calculate the average paper score and average label score. Then, we average over the users in each condition:

Avg:	Normalized Score	Normalized Label Score
GS	74.2%	71.6%
MP+GS	84.5%	80.2%

Both the paper and label scores of MP+GS users are higher than the scores of GS users (the median scores exhibit similar behaviour). In addition, the average number of seminal papers discovered by GS users was 1.2 , while MP+GS users have discovered on average 1.62 seminal papers.

The simulated Wikipedia user WK did not do well: out of 15 references, only four qualified for the study (papers published after 1996), and only two were deemed relevant. In Wikipedia's defense, the other references included seminal books, which could have been useful for our hypothetical first-year student.

Finally, let us examine the map (MP) user performance. Comparing the map directly to user output is challenging as the map contained 45 papers, many more than the average user. Out of these papers, seven were deemed seminal, and 21 were deemed relevant. Interestingly, many of the papers that were deemed irrelevant were used as bridges between relevant (or seminal) papers in the map.

The finding that many of the map users did not identify the seminal papers in the map is somewhat concerning. A possible explanation may be that the users were instructed to focus on at most five lines of research, while the seminal papers were spread among more lines. Note that despite this fact, the average normalized score of MP+GS users is still higher than the score for the map. In any case, this phenomenon highlights the need for more targeted research on locating and visualizing important nodes in the map.

9.1.3 Recall

In addition to measuring precision (the fraction of retrieved papers that are relevant), we also tested user's *recall* (the fraction of relevant papers retrieved). It is not enough for the users to find good papers; rather, it is also important that they do not overlook important research areas.

In order to measure recall, we have composed a list of the top-10 subareas of reinforcement learning by going over conference and workshop tracks and picking the most frequent topics. Each user had to list up to five research directions; for each user, we computed the fraction of these directions that appeared in our top-10 list. GS users received an average score of 46.4%, while MP+GS users outperformed them with an average score of 73.1%.

Finally, further analysis of the snapshots taken throughout the study provides anecdotal evidence of the utility of the map. Several MP+GS users started by composing a short

[1] http://scholar.google.com

list of research directions; throughout the session, these users have progressively added papers to each direction. GS users, in contrast, did not exhibit this 'big picture' behaviour.

9.2 User Comments

After the study, we asked the map users to tell us about their experience. Below are some of their comments:

Positive: "Helpful... gave me keywords to search for" / "I noticed directions I didn't know about... Haven't heard of predictive state representations before" / "Useful way to get a basic idea of what science is up to" / "That was a great starting point" / "Easy to identify research groups... in this context, this guy is good" / "Timeline is very useful"

Negative: "Takes a while to grasp" / "For a beginner, some papers are too specific... may be more useful after I read some more" / "Legend is confusing if you do not know the topic in advance" / "Didn't necessarily understand the logic behind edges... why don't you draw words on edges?" / "It is hard to get an idea from paper title alone"

Most importantly, many participants found the map useful in making sense of the field. Some of the participants had trouble interpreting elements of the map, or felt like the map was more suited for researchers with deeper background knowledge. We found that many of the negative comments could be addressed by improvements in the design of the user interface.

10. A NOTE ON PERSONALIZATION

When we defined our coverage notion (Secion 4), the weight of each paper was set to 1. In other words, the objective was to cover as much of the corpus as possible. However, some parts of the corpus may be more important to the user than others. In order to be useful, the model must be capable of representing the user's interests.

In this section, we rely on user feedback in order to learn their preferences and adjust the maps accordingly. We use the interaction algorithm of [Shahaf et al., 2012]. This algorithm lets the user provide *feature-based feedback*. Feature-based feedback is a very natural way for specifying preferences. We show the user a tag cloud describing the papers of the map. Clicking on a word lets the user adjust its importance. For example, importance of 0.9 implies that 90% of the documents in which the word appears are interesting to the user. The relative transparency of the model allows users to make sense of feature weights.

Feature-based feedback is especially useful in the research domain, as users can employ it to indicate which authors and venues they trust. In addition, since our coverage notion is biased against newer papers (new papers did not have enough time to make a big impact), the users can indicate their preferences for new, state-of-the-art papers.

When we increase the weight of the years 2005-2008 in the reinforcement learning map, the resulting map contains chains about more recent topics, such as hierarchical reinforcement learning. When biasing for AAMAS (a conference on autonomous agents and multiagent systems), the resulting map includes a new chain about robot soccer.

In the future, it may be interesting to formulate a notion of baseline personalization, where default weights are set based on authors' *reputations*: if an author has written many high-impact papers, his new paper is likely to be important. We may also explore other notions of personalized coverage, such as [El-Arini and Guestrin, 2011; Yue and Guestrin, 2011].

11. CONCLUSIONS AND FUTURE WORK

In this paper, we have devised a method for constructing metro maps of science. Given a query, our algorithm generates a metro map: a concise structured set of research lines which maximizes coverage of salient pieces of information. Most importantly, metro maps explicitly show the relations between the research lines.

We conducted promising pilot user studies, comparing our system to two systems that dominate today's research-related queries. The results indicate that our method can help users acquire knowledge efficiently.

In the future, we plan to experiment with richer forms of input, output, and interaction models. Promising directions include edge-annotation based on citation function, notions of coverage that combine structure and content, paper-based and line-based feedback mechanisms, and the integration of higher-level semantic features. We have also created a website that allows interactive visualization of metro maps, which we hope to launch soon. We believe that metro maps hold the potential to become effective tools to help researchers cope with information overload.

Acknowledgments: This work was partially supported by ONR PECASE N000141010672, ARO MURI W911NF0810242, ONR MURI N000141010934 and NSF Career IIS-0644225. Dafna Shahaf was supported in part by Microsoft Research Graduate Fellowship. An initial version of metro maps for science was created during an internship at Microsoft Research.

References

Allan, J., Gupta, R., and Khandelwal, V. (2001). Temporal summaries of new topics. In *SIGIR '01*.

Ball, M. O. (1986). Computational complexity of network reliability analysis: An overview. *IEEE Transactions on Reliability*, 35(3):230 –239.

Bassecoulard, E. and Zitt, M. (1999). Indicators in a research institute: A multi-level classification of scientific journals. *Scientometrics*, 44(3):323–345.

Borner, K. (2010). *Atlas of Science: Visualizing What We Know*. MIT Press.

Boyack, K., Börner, K., and Klavans, R. (2009). Mapping the structure and evolution of chemistry research. *Scientometrics*, 79:45–60.

Chekuri, C. and Pal, M. (2005). A recursive greedy algorithm for walks in directed graphs. In *FOCS '05*.

Chen, C. (2004). Searching for intellectual turning points: Progressive knowledge domain visualization. *PNAS*, 101(Suppl 1):5303–5310.

Dunne, C., Shneiderman, B., Dorr, B., and Klavans, J. (2010). iopener workbench: tools for rapid understanding of scientific literature. In *Proc. 27th Annual Human-Computer Interaction Lab Symposium*.

El-Arini, K. and Guestrin, C. (2011). Beyond keyword search: Discovering relevant scientific literature. In *KDD' 11*.

Nesbitt, K. (2004). Getting to more abstract places using the metro map metaphor. In *Information Visualisation '04*.

Shahaf, D. and Guestrin, C. (2010). Connecting the dots between news articles. In *KDD '10*.

Shahaf, D., Guestrin, C., and Horvitz, E. (2012). Trains of thought: Generating information maps. In *WWW '12*.

Swan, R. and Jensen, D. (2000). TimeMines: Constructing Timelines with Statistical Models of Word Usage. In *KDD' 00*.

Yan, R., Wan, X., Otterbacher, J., Kong, L., Li, X., and Zhang, Y. (2011). Evolutionary timeline summarization: a balanced optimization framework via iterative substitution. In *SIGIR' 11*.

Yue, Y. and Guestrin, C. (2011). Linear submodular bandits and their application to diversified retrieval. In *NIPS' 11*.