Connecting the Dots Between News Articles

Dafna Shahaf and Carlos Guestrin
Carnegie Mellon University
{dshahaf,guestrin}@cs.cmu.edu

Abstract

The process of extracting useful knowledge from large datasets has become one of the most pressing problems in today's society. The problem spans entire sectors, from scientists to intelligence analysts and web users, all of whom are constantly struggling to keep up with the larger and larger amounts of content published every day. With this much data, it is often easy to miss the big picture.

In this paper, we investigate methods for automatically connecting the dots – providing a structured, easy way to navigate within a new topic and discover hidden connections. We focus on the news domain: given two news articles, our system automatically finds a coherent chain linking them together. For example, it can recover the chain of events leading from the decline of home prices (2007) to the health-care debate (2009).

We formalize the characteristics of a good chain and provide efficient algorithms to connect two fixed endpoints. We incorporate user feedback into our framework, allowing the stories to be refined and personalized. Finally, we evaluate our algorithm over real news data. Our user studies demonstrate the algorithm's effectiveness in helping users understanding the news.

1 Introduction

"Can't Grasp Credit Crisis? Join the Club", stated David Leonhardt's article in the New York Times. Credit crisis had been going on for seven months by that time, and had been extensively covered by every major media outlet throughout the world. Yet many people felt as if they did not understand what it was about.

Paradoxically, the extensive media coverage might have been a part of the problem. This is another instance of the *information overload* problem, long recognized in the computing industry. Users are constantly struggling to keep up with the larger and larger amounts of content that is being published every day; with this much data, it is often easy to miss the big picture.

For this reason, there is an increasing need for techniques to present data in a meaningful and effective manner. In this paper, we investigate methods for automatically *connecting the dots* – providing a structured, easy way to uncover hidden connections between two pieces of information. We believe that the ability to connect dots and form a logical, coherent story lies at the basis of understanding a topic.

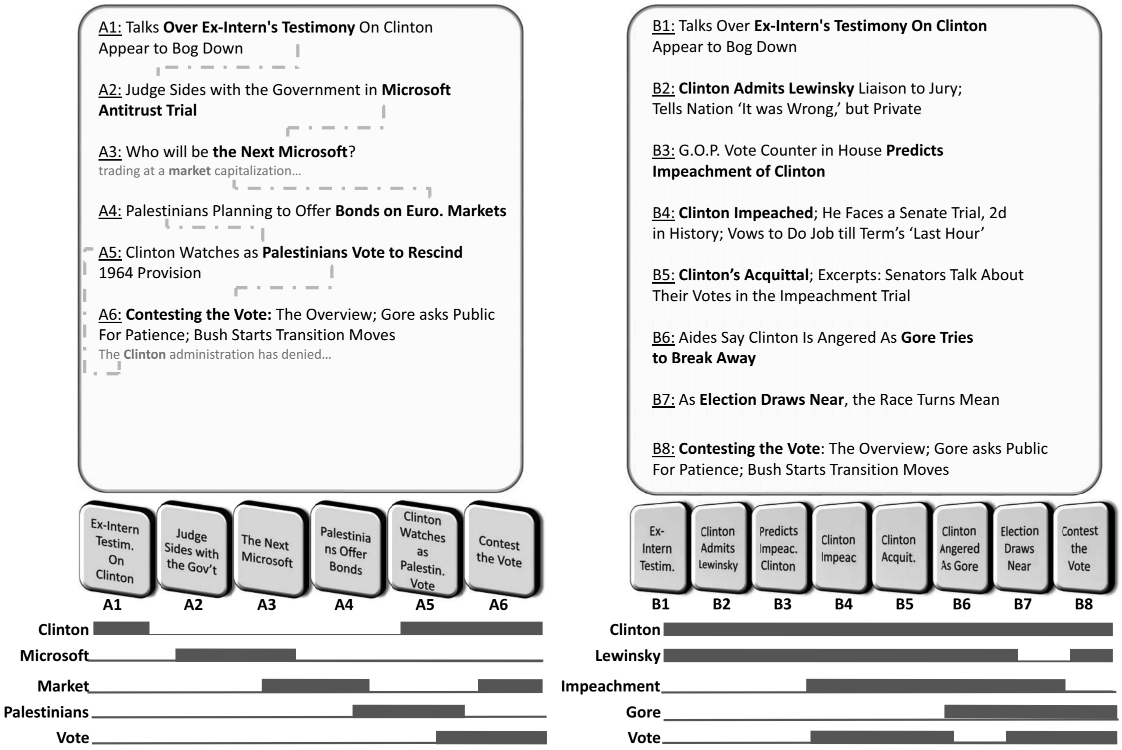
We focus on the news domain: given two news articles, our system automatically finds a coherent *story* (chain of articles) linking them together. For example, imagine a user who is interested in the financial crisis and its effect on the healthcare reform. The user vaguely recalls that the financial crisis is related to the decline of home prices in 2007. The user would then choose representative articles for those two topics and feed them to our system. An output chain may look like this (parenthesized text not part of output):

1.3.07	**Home Prices Fall** Just a Bit
3.4.07	Keeping Borrowers Afloat
	(Increasing delinquent mortgages)
3.5.07	A **Mortgage Crisis** Begins to Spiral, ...
8.10.07	... **Investors Grow Wary** of Bank's Reliance on Debt.
	(Banks' equity diminishes)
9.26.08	Markets **Can't Wait for Congress** to Act
10.4.08	**Bailout Plan** Wins Approval
1.20.09	Obama's Bailout Plan **Moving Forward**
	(... and its effect on health benefits)
9.1.09	Do Bank Bailouts **Hurt Obama on Health?**
	(Bailout handling can undermine health-care reform)
9.22.09	Yes to Health-Care Reform, but Is This the Right Plan?

The chain mentions some of the key events connecting the mortgage crisis to healthcare, including the bailout plan. Most importantly, the chain should be *coherent*: after reading it, the user should gain a better understanding of the progression of the story.

To the best of our knowledge, the problem of connecting the dots is novel. There has been extensive work done on related topics, from narrative generation [Turner, 1994; Niehaus and Young, 2009] to identifying and tracking news events [Nallapati *et al.*, 2004; Mei and Zhai, 2005; Yang *et al.*, 1999; Lewis and Knowles, 1997].

Our work differs from most previous work in two other important aspects – **expressing information needs** and **structured output and interaction**. Often, users know precisely what they want, but it is not easy for them to distill this down into a few keywords. Our system's input method (related articles) might facilitate this task. Our system's output is in-


A1: Talks **Over Ex-Intern's Testimony** On Clinton Appear to Bog Down

A2: Judge Sides with the Government in **Microsoft Antitrust Trial**

A3: Who will be **the Next Microsoft**?
trading at a **market** capitalization…

A4: Palestinians Planning to Offer **Bonds on Euro. Markets**

A5: Clinton Watches as **Palestinians Vote to Rescind** 1964 Provision

A6: **Contesting the Vote:** The Overview; Gore asks Public For Patience; Bush Starts Transition Moves
The **Clinton** administration has denied…

B1: Talks Over **Ex-Intern's Testimony On Clinton** Appear to Bog Down

B2: **Clinton Admits Lewinsky** Liaison to Jury; Tells Nation 'It was Wrong,' but Private

B3: G.O.P. Vote Counter in House **Predicts Impeachment of Clinton**

B4: **Clinton Impeached**; He Faces a Senate Trial, 2d in History; Vows to Do Job till Term's 'Last Hour'

B5: **Clinton's Acquittal**; Excerpts: Senators Talk About Their Votes in the Impeachment Trial

B6: Aides Say Clinton Is Angered As **Gore Tries to Break Away**

B7: As **Election Draws Near**, the Race Turns Mean

B8: **Contesting the Vote**: The Overview; Gore asks Public For Patience; Bush Starts Transition Moves

Figure 1: Two examples of stories connecting the same endpoints. Left: chain created by shortest-path (dashed lines indicate similarities between consecutive articles). Right: a more coherent chain. Activation patterns for each chain are shown at the bottom; the bars indicate appearance of words in the article above them.

teresting, too – instead of the common list of relevant documents, our output is more structured: a chronological chain of articles, and the flow of influences along it. Often, visually exploring a system's results and interacting with it can reveal new and interesting phenomena. Our main contributions are:

- Formalizing characteristics of a good story and the notion of *coherence*.
- Formalizing *influence* with no link structure.
- Providing efficient algorithms for connecting two fixed endpoints while maximizing chain coherence.
- Incorporating feedback and interaction mechanisms into our system, tailoring stories to user preferences.
- Evaluating our algorithm over real news data and demonstrating its utility to news-readers via a user study.

Our methods are also directly applicable to many other domains. Email, research papers, and military intelligence analysis are but a few of the domains in which it would be immensely useful to automatically connect the dots.

2 Finding a Good Chain

2.1 What makes a story good?

Our goal is to find a good path between two articles, s and t. A natural thing to do would be to construct a graph over the articles and find the shortest s-t path. Since there are no edges between articles, we will have to add them ourselves, e.g., by linking similar articles together.

However, this simple method does not necessarily yield a good chain. Suppose we try to find a coherent chain of events between Clinton's alleged affair and the 2000 election Florida recount. We pick two representative documents,

s: *Talks Over Ex-Intern's Testimony On Clinton Appear to Bog Down (Jan 1998)*

t: *Contesting the Vote: The Overview; Gore asks Public For Patience (Nov 2000)*

and find a shortest path between them. The result is shown on Figure 1 (left). This chain of stories is rather erratic, passing through the Microsoft trial, Palestinians, and European markets before returning to American politics. Note that each transition, when examined out of context, is reasonable: for example, the first and the second articles are court-related. Those correlations are marked by dashed lines in Figure 1.

The problem seems to lie with the locality of shortest-path. Every two consecutive articles are related, but there is no *global*, coherent theme to the chain as a whole. Rather, shortest-path may exhibit *stream-of-consciousness* behaviour, linking s and t by a chain of free associations. The chain of Figure 1 (right) is better: it tells the story of Clinton's impeachment and acquittal, the effect on Al Gore's campaign, and finally the elections and recount. In the following, we identify the properties which make this chain better.

Let us take a closer look at these two chains. Figure 1 (bottom) shows word activation patterns along both chains. Bars correspond to the appearance of a word in the articles depicted above them. For example, the word 'Clinton' appeared throughout the whole right chain, but only at the beginning and the last two articles on the left. It is easy to spot the associative flow of the left chain in Figure 1. Words appear for very short stretches, often only in two neighbouring articles. Some words appear, then disappear for a long period and re-appear. Contrast this with the chain on the right, where the stretches are longer (some words, like Clinton and Lewinsky, appear almost everywhere), and transitions between documents are smoother. This observation motivates our definition of coherence in the next section.

2.2 Formalizing story coherence

Let $\mathcal{D}$ be a set of articles, and $\mathcal{W}$ a set of features (typically words or phrases). Each article is a subset of $\mathcal{W}$. Given a chain $(d_1, ..., d_n)$ of articles from $\mathcal{D}$, we can estimate its coherence from its word activation patterns. One natural definition of coherence is

$$\textit{Coherence}(d_1, ..., d_n) = \min_{i=1...n-1} \sum_w \mathbb{1}(w \in d_i \cap d_{i+1})$$

Every time a word appears in two consecutive articles, we score a point for the transition. Coherence is the minimum transition score. This objective has several attractive properties; it encourages positioning similar documents next to each other, and is very easy to compute. It also takes into consideration the fact that a chain is only as strong as its weakest link. However, this objective suffers from serious drawbacks:

Missing words: Due to our noisy features, some words do not appear in an article, although they should have. For example, if a document contains 'lawyer' and 'court' but not 'prosecution', chances are 'prosecution' is still a highly-relevant word. Considering only words from the article can be misleading in such cases.

Moreover, even if our features were not noisy, an indicator function is not informative enough for our needs.

Importance: Some words are more important than others, both on a corpus level and on a document level. For example, the first two articles of the shortest-path chain shared both 'judge' and 'page'. Clearly, 'judge' is more significant, and should affect the objective more.

Combining **Importance** and **Missing words**, it becomes clear that we need more than a simple word-indicator. Rather, we need to consider the *influence* of d_i on d_{i+1} through the word w. We defer the formal definition of influence to Section 2.3; intuitively, *Influence*$(d_i, d_j \mid w)$ is high if (1) the two documents are highly related, and (2) w is important for the connectivity. w does not have to appear in either of the documents. Refer to Figure 2: the source document d_0 is

d_0 :*Judge Lance Ito lifted his ban on live television coverage of the O.J. Simpson trial*

We calculated word-influence from d_0 to two other documents, using methods explained in Section 2.3. The blue bars (in the back) represent word influence for document

d_1 :*O.J. Simpson's defense lawyers told the judge they would not object to the introduction of DNA evidence*

and the red bars (front) represent word influence for

d_2 :*Winning three consecutive Super Bowls would be a historic accomplishment for San Francisco 49ers*

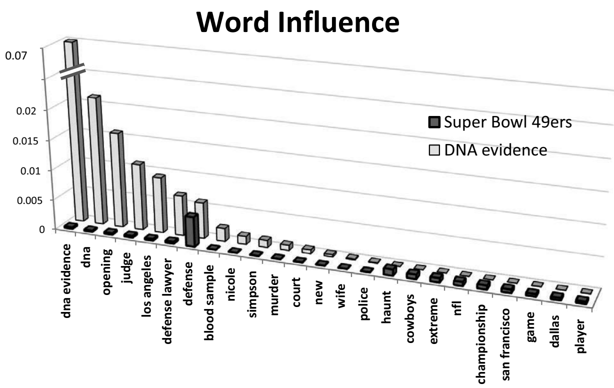


Figure 2: Word influence from an article about the OJ Simpson trial to two other documents (football/ DNA evidence).

First, note that the blue bars are generally higher. This means that d_1 is more relevant to the source article d_0. The influential words for d_1 are mostly court-related, while d_2's are sport-related (interestingly, the word 'Defense' is strong in both documents, for completely different reasons). Many of the influential words do not appear in either of the three articles, thereby solving the **Missing words** problem. With the new *Influence* notion, our objective can be re-defined as

$$\textit{Coherence}(d_1, ..., d_n) = \min_{i=1...n-1} \sum_w \textit{Influence}(d_i, d_{i+1} \mid w)$$

This new objective, while better, still suffers from the problem of **Jitteriness**.

Jitteriness: the objective does not prevent jittery activation patterns, i.e., topics that appear and disappear throughout the chain.

One way to cope with jitteriness is to only consider the longest continuous stretch of each word. This way, going back-and-forth between two topics provides no utility after the first topic switch. Remember, this stretch is not determined by the actual appearance of the word along the chain. Rather, we define an activation pattern arbitrarily for each word, and compute our objective based on it. The coherence is then defined as the score under the best activation pattern:

$$\textit{Coherence}(d_1, ..., d_n) = \max_{\textit{activations}} \min_{i=1...n-1}$$

$$\sum_w \textit{Influence}(d_i, d_{i+1} \mid w) \mathbb{1}(w \textit{ active in } d_i, d_{i+1}) \quad (*)$$

Since influence is non-negative, the optimum activates all words everywhere. In order to emulate the behaviour of the activation patterns in Figure 1, we constrain the patterns we consider: we limit the total number of active words and the number of words that are active per transition. In order to avoid multiple stretches, we allow each word to be activated at most once.

Instead of using binary activations, we propose a softer notion of continuous activations. A word's activation is in the range $[0, 1]$, signifying the degree to which it is active. This leads, quite naturally, to a formalization of the problem as a linear program.

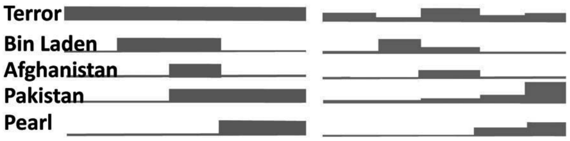


Figure 3: Activation patterns found by our algorithm for a chain connecting 9/11 to Daniel Pearl's murder. Left: activation levels. Right: activation levels weighted by the influence (rescaled). For illustrative purposes, we show the result of the integer program (IP).

Linear Program Formulation

The objective function (*) we defined in the previous section can be readily formalized as a linear program (LP). The LP is specified in [Shahaf and Guestrin, 2010]. As a sanity check, we tried the LP on real chains. Figure 3 (left) shows the best activation pattern found for a chain connecting 9/11 and Daniel Pearl's murder (top five words). This pattern demonstrates some of the desired properties from Section 2: the word 'Terror' is present throughout the whole chain, and there is a noticeable change of focus from Bin Laden to Pakistan and the kidnapped journalist. Figure 3 (right) shows activation × influence (rescaled). Notice that words with the same activation levels can have different levels of influence, and thus different effect on the score.

Finding a Chain as a Search Problem

In the previous sections we discussed a method to score a fixed chain. However, we are still left with the problem of *finding* a chain. In [Shahaf and Guestrin, 2010], we formulate this problem as another LP and propose a rounding technique with proved guarantees. This technique has produced good chains, but generating the chains was a slow process. In addition, this approach only provided an *approximate* solution.

In [Under Review, 2011], we have explored a different technique, based on the *general best-first* search strategy [Dechter and Pearl, 1985]. We keep a priority queue of selected chains; at each iteration, we expand the chain which features the highest (heuristic) merit, generating all of its valid extensions. The algorithm is guaranteed to find the optimum, although it can take exponential time in the worst case. In practice, however, it is much faster than [Shahaf and Guestrin, 2010].

2.3 Measuring influence without links

Our objective function required evaluating *influence*$(d_i, d_j \mid w)$ – the influence of d_i on d_j w.r.t. word w. Several methods for measuring influence have been proposed [Kleinberg, 1999; Kempe *et al.*, ; Brin and Page, 1998]. The vast majority focus on directed weighted graphs (e.g., the web, social networks, citations), and take advantage of edge structure. However, in our setting no edges are present. In this section, we explore a different notion of influence; despite the fact that it is based on random walks, it requires no edges.

First, we construct a bipartite directed graph. Vertices correspond to documents and words. We add edges (w, d) and (d, w) if word w appears in document d. Figure 4 shows a simple graph with four (square) documents and four (circle) words. Edge weights represent the strength of the relation between a document and a word (e.g., TF-IDF weights). Since we interpret weights as random walk probabilities, we normalize them over all words in the document. E.g., the rightmost article is mostly (.7) about Al Gore, and somewhat about 'Judge' (.2) and 'Clinton' (.1). The word-to-document weights are computed using the same numbers, but normalized over the documents. The word 'Gore' can only be reached by a single document, so the edge weight is $\frac{.7}{.7} = 1$. We use this weighted graph to define influence.

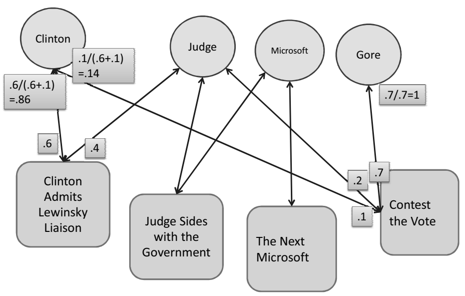


Figure 4: A bipartite graph used to calculate influence.

As mentioned before, *Influence*$(d_i, d_j \mid w)$ should be high if the two documents are highly connected, and w plays an important role in this connection. Intuitively, if the two documents are related, a short random walk starting from d_i should reach d_j frequently. We compute the stationary distribution of random walks starting from d_i. The stationary distribution is the fraction of the time the walker spends at each node:

$$\Pi_i(v) = \epsilon \cdot \mathbb{1}(v = d_i) + (1 - \epsilon)\Sigma_{(u,v) \in E}\Pi_i(u)P(v \mid u)$$

where $P(v \mid u)$ is the probability of reaching v from u, and random restart probability ϵ controls the expected length.

We now need to factor in the effect of w on these walks. We turn w into a sink node: let $P^w(v \mid u)$ be the same probability distribution as $P(v \mid u)$, except there is no way out of node w. Let $\Pi_i^w(v)$ be the stationary distribution for this new graph. If w was influencial, the stationary distribution of d_j would decrease a lot: in Figure 4, without the word 'Judge' article 1 is no longer reachable from article 2.

The influence on d_j w.r.t. w it defined as the difference between these two distributions, $\Pi_i(d_j) - \Pi_i^w(d_j)$. Figure 2 shows an example of word-influence results calculated by this method. Refer to Section 2.2 for a detailed explanation.

3 Evaluation

Evaluating the performance of information retrieval tasks often focuses on canonical labeled datasets (e.g., TREC competitions) amenable to the standard metrics of precision, recall and variants thereof. The standard methods do not seem to apply here, as they require labeled data, and we are not aware of any labeled dataset suitable for our task. As a result, we evaluated our methods by conducting user studies to capture the utility of our algorithms as they would be used in practice.

We evaluate our algorithm on real news data from the New York Times and Reuters datasets (1995-2003). We preprocessed more than half a million articles, covering a diverse set of topics. We considered major news stories: the OJ Simpson trial, Clinton's impeachment, the Enron scandal, September 11th and the Afghanistan war. For each story, we selected a

Google News Timeline: Osama bin Laden is denounced by his family // Osama Family's Suspicious Site (Web designer from LA buys a bizarre piece of Internet history) // Are you ready to dance on Osama's grave? (How should one react to the death of an enemy?) // Al-Qaeda behind Karachi blast // LIVE FROM AFGHANISTAN: Deadline of Death Delayed for American Journalist // Killed on Job But Spared 'Hero' Label (About Daniel Pearl)

Connect the Dots: Dispatches From a Day of Terror and Shock // Two Networks Get No Reply To Questions For bin Laden (Coverage of September 11th) // Opponents of the War Are Scarce on Television (Coverage of the war in Iraq and Afghanistan) // 'Afghan Arabs' Said to Lead Taliban's Fight // Pakistan Ended Aid to Taliban Only Hesitantly // Pakistan Officials Arrest a Key Suspect in Pearl Kidnapping (Pearl abducted in Paksitan while investigating links to terror) // The Tragic Story of Daniel Pearl

Figure 5: Example output chains for Connect-Dots and Google News Timeline. Users were given access to the full articles. The GNT chain is a lot less coherent, and includes several insignificant articles.

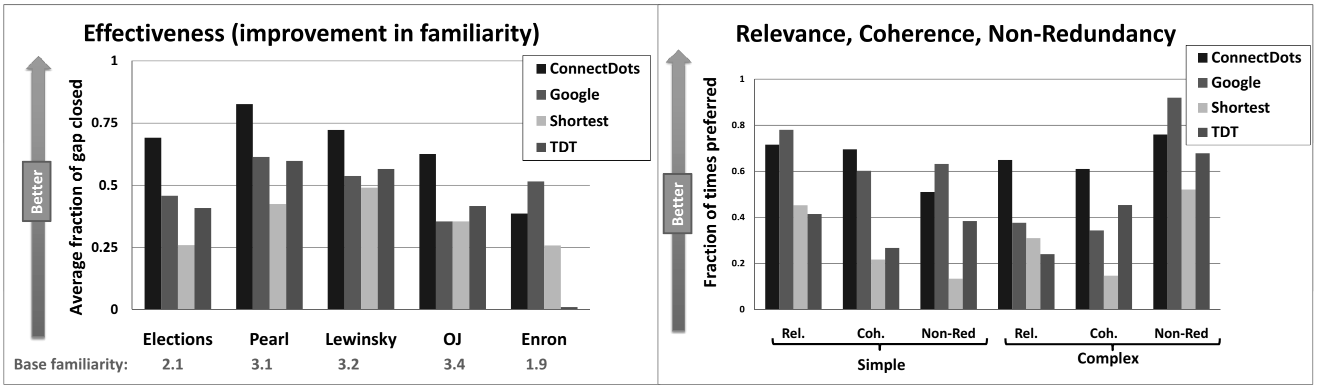


Figure 6: Left: evaulating effectiveness. For each story and each technique, we average over users the fraction of familiarity gap which closed after reading a chain. The number under the story indicates the average familiarity (on a scale of 1 to 5) before reading any chain. Right: Relevance, coherence, and non-redundancy (broken down by simple vs. complex stories). The y axis is the fraction of times each method was preferred, compared to another chain. Users could say both chains are equally good, and therefore the numbers do not sum to 1. Our algorithm outperformed the competitors almost everywhere, especially for complex stories.

set of $500 - 10,000$ candidate articles, based on keyword-search.

Our goal was to construct chains representing the stories, and have users evaluate them. For each story, we chose several pairs of articles. We then tried finding stories linking each pair using **Connecting-Dots**, **Shortest-path**, **Google News Timeline** [New,] and **Event threading (TDT)**[Nallapati *et al.*, 2004]. The exact techniques used to construct the chains appear in [Shahaf and Guestrin, 2010].

We presented 18 users with a pair of source and target articles. We gauged their **familiarity** with those articles, asking whether they believe they knew a coherent story linking them together (on a scale of 1 to 5). We showed the users pairs of chains connecting the two articles, generated by the above methods in a double-blind fashion, and asked them to indicate which chain is more **Coherent**, **Redundant**, and **Relevant** (better captures the events connecting the two articles).

In addition, we measured the **effectiveness** of the chains. We asked users to estimate how their answer to the **familiarity** question changed after reading each chain. **Effectiveness** is the fraction of the familiarity gap closed. For example, if the new familiarity is 5, this fraction is 1 (gap completely closed). If the familiarity did not change, the fraction is 0. This way, we test whether users feel that the chain helped them gain better understanding of the big picture.

Example output chains are shown in Figure 5. Figure 6 shows the results of our user-study. After analyzing the results, we identify two types of stories: *simple* and *complex*. Simple stories tend to focus around the same event, person or institution (Simpson trial, Enron), and can usually be summarized by a single query string. In complex stories the source and target article are indirectly connected through one or more events (Lewinsky-impeachment-elections, September 11th-Afghanistan war-Daniel Pearl).

The left plot shows the **effectiveness** (closing the **familiarity** gap) for each of the methods. Underneath each story we display the average familiarity score before reading any chain (e.g., the Enron story is not well-known). Our algorithm does better than the competitors on all stories but Enron. The difference is especially pronounced for complex stories. In simple stories, such as Enron, it seems that the simple method of picking documents from GNT was sufficient for most people. However, when the story could not be represented as a single query, the effectiveness of GNT decreased.

Figure 6(right) shows the percentage of times each method was preferred for relevance, coherence and non-redundancy. As users could prefer one chain, both or neither, numbers do not sum to 100%. Our algorithm is amongst the best in all measures at a statistically significant level. Most importantly, it achieves the best coherence scores (especially in the complex case). This indicates that the notion of coherence devised in this paper matches what the actual users perceive.

As expected, for all methods, relevance is good for simple stories but achieving low redundancy is harder. There is a tradeoff – redundancy is easy to avoid by picking random, possibly irrelevant articles. Relevance is easy to achieve by picking articles similar to s or t, but then redundancy would be high. We discuss some other interesting findings in [Sha-

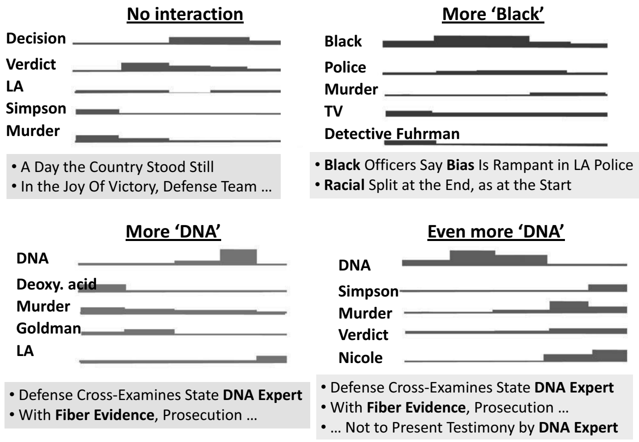


Figure 7: Our interactive component demonstrated. Top left: the original chain. The rest are derived from it by requesting the words 'black', 'DNA', and 'DNA'×2. For each chain we show activation levels for the most important words, and a few selected articles.

haf and Guestrin, 2010].

4 Interaction Models

Thus far, we have defined a way to find chains connecting two endpoints. However, the user may not find the resulting chain satisfactory. In information retrieval systems, the solution is often to let the users revise their queries. In this section, we propose to take advantage of the structured nature of the chains, and explore more expressive forms of interaction:

Refinement: We provide the user with a mechanism to indicate areas in the chain which should be further refined; a refinement may consist of adding a new article, or replacing an article which seems out of place.

Since evaluating a single chain is quick, the refinment process is very efficient. We try all possible $O(\mathcal{D})$ replacement/insertion actions, evaluate each chain (Section 2), and return the best one. In a pilot user study, users who performed a refinement operation preferred our local-search chains to greedily-refined chains 72% of the time.

Incorporate user interests: There can be many coherent ways to connect s and t. We provide a mechanism for the user to focus the chains around words they find important. In order to take user's feedback into account, we augment our objective with importance weight π_w for each word w:

$$\sum_{w} \pi_{\mathbf{w}} \mathit{Influence}(d_i, d_{i+1} \mid w) \mathbb{1}(w \textit{ active in } d_i, d_{i+1})$$

π_w are updated based on user feedback, and a new chain is computed, optimizing the personalized objective.

Figure 7 shows actual system output. The top-left chain (before any interaction took place) focuses on the verdict. The other chains are derived from it by increasing the weight of 'Black' (top right) or 'DNA' (bottom left), or 'DNA' twice (bottom-right). Increasing the weight of a word causes the chain to change its focus accordingly.

In a pilot user study, we have asked users to 'reverse engineer' our system, and identify words whose were used in order to obtain one chain from another. Users identified at least one word 63.3% of the times.

5 Conclusions and Future Work

In this paper we describe the problem of connecting the dots. Our goal is to help people fight information overload by providing a structured, easy way to navigate between topics. We explored different desired properties of a good story, formalized it as a linear program, and provided efficient algorithms to connect two articles. We evaluate our algorithm over real news data via a user study, and demonstrate its effectiveness compared to other methods, such as Google News Timeline.

Our system is unique in terms of input and output, and incorporating feedback into it allows users to fully exploit its capabilities. In the future, we plan to explore richer forms of input and output, allowing for more complex tasks, e.g., creating a roadmap – a set of intersecting chains that covers a topic from several aspects.

We believe that the system proposed in this paper may be a promising step in the battle against information overload. The ability to connect two pieces of information and form a logical, coherent story has applications in many areas. Perhaps most importantly, significant scientific discoveries can come from forming connections between different fields. We plan to extend our methods to scientific papers; we believe that tools to automatically connect the dots can be a great vehicle to enable new discoveries.

Acknowledgements: This work was partially supported by ONR YIP N00014-08-1-0752, PECASE N000141010672, ARO MURI W911NF0810242, and NSF Career IIS-0644225. Dafna Shahaf was supported in part by Microsoft Research Graduate Fellowship.

References

[Brin and Page, 1998] S Brin and L Page. The anatomy of a large-scale hypertextual web search engine. In *Computer Networks and ISDN Systems*, 1998.

[Dechter and Pearl, 1985] Rina Dechter and Judea Pearl. Generalized best-first search strategies and the optimality of a*. *J. ACM*, 32:505–536, July 1985.

[Kempe *et al.*,] D Kempe, J Kleinberg, and É Tardos. Maximizing the spread of influence through a social network. In *KDD '03*.

[Kleinberg, 1999] Jon Kleinberg. Authoritative sources in a hyperlinked environment, 1999.

[Lewis and Knowles, 1997] D. D. Lewis and K. A. Knowles. Threading electronic mail: A preliminary study. *Information Processing and Management*, 33, 1997.

[Mei and Zhai, 2005] Q Mei and C Zhai. Discovering evolutionary theme patterns from text: an exploration of temporal text mining. In *KDD '05*, 2005.

[Nallapati *et al.*, 2004] R Nallapati, A Feng, F Peng, and J Allan. Event threading within news topics. In *CIKM '04*, 2004.

[New,] Google news timeline, http://newstimeline.googlelabs.com/.

[Niehaus and Young, 2009] J Niehaus and R M Young. A computational model of inferencing in narrative. In *AAAI Spring Symposium '09*, 2009.

[Shahaf and Guestrin, 2010] Dafna Shahaf and Carlos Guestrin. Connecting the dots between news articles. In *KDD'10*, 2010.

[Turner, 1994] S R Turner. The creative process: A computer model of storytelling and creativity, 1994.

[Yang *et al.*, 1999] Y Yang, J Carbonell, R Brown, T Pierce, B Archibald, and X Liu. Learning approaches for detecting and tracking news events. *IEEE Intelligent Systems*, 14(4), 1999.