Using tagFlake for Condensing Navigable Tag Hierarchies from Tag Clouds

Using *tagFlake* for Condensing Navigable Tag Hierarchies from Tag Clouds[*]

Luigi Di Caro
Dip. of Informatica
Univ. of Torino
dicaro@di.unito.it

K. Selçuk Candan
Comp. Sci. and Eng.
Arizona State Univ.
candan@asu.edu

Maria Luisa Sapino
Dip. of Informatica
Univ. of Torino
mlsapino@di.unito.it

ABSTRACT

We present the `tagFlake` system, which supports semantically informed navigation within a tag cloud. `tagFlake` relies on `TMine` for organizing tags extracted from textual content in hierarchical organizations, suitable for navigation, visualization, classification, and tracking. `TMine` extracts the most significant tag/terms from text documents and maps them onto a hierarchy in such a way that descendant terms are contextually dependent on their ancestors within the given corpus of documents. This provides `tagFlake` with a mechanism for enabling navigation within the tag space and for classification of the text documents based on the contextual structure captured by the created hierarchy. `tagFlake` is language neutral, since it does not rely on any natural language processing technique and is unsupervised.

Categories and Subject Descriptors

H.3.1 [**Information Storage and Retrieval**]: Content Analysis and Indexing; H.5 [**Information Interfaces and Presentation**]; I.7 [**Document and Text Processing**]

General Terms

Algorithms, Human Factors

1. MOTIVATION

With the quick growth of content over the web (e.g., the blogosphere), tag-based searches and tag cloud (sets of tags) based visualizations have become popular. Tags, whether provided by the user or extracted from the textual content, annotate online documents (such as blogs and news articles) with popular terms, thus providing an easy way to search and index them.

Most visualizations of tag clouds vary the sizes of the fonts to differentiate most important tags from those that are less important (Figure 1). [2] aims creating *visually pleasant* tag clouds, by presenting tags in the form of seemingly random collections of circles with varying sizes: the size of the circle denotes its frequency. While quickly highlighting the most dominant terms in the corpus, these representations fall short in describing the context in which these terms occur in the collection.

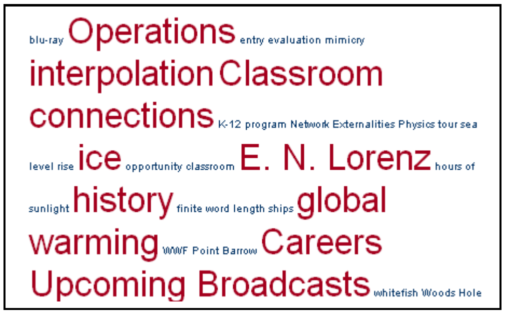

blu-ray Operations entry evaluation mimicry
interpolation Classroom
connections K-12 program Network Externalities Physics tour sea
level rise ice opportunity classroom E. N. Lorenz hours of
sunlight history finite word length ships global
warming WWF Point Barrow Careers
Upcoming Broadcasts whitefish Woods Hole

Figure 1: A tag cloud sample (obtained from *http://expertvoices.nsdl.org/* on 05/26/2008). Such tag clouds fail to represent the contextual relationships between the tags

The need for contextually informed navigation within the blogosphere has been highlighted in the literature. For example, [16] observes that for large blog archives, a simple chronological order is not sufficient and a table of contents (TOC) like navigational hierarchy, depicting the topics development within the blog archive and describing how these topics relate to each other would be more effective.

1.1 Related Work

[14, 15] describe a system for bi-dimensional visualization of tag clouds. Tags are selected on the basis of their frequency of use. Semantic relationships among tags are defined in terms of their similarity, quantified by means of the Jaccard coefficient. K-means clustering is then applied on tag similarity matrix, with an a priori chosen number of clusters and fixed number of selected relevant tags. [15] applies Multidimensional Scaling, using Pearson's correlation as the similarity function, on a tag-to-tag correlation matrix. MDS creates a bi-dimensional space, which is then visualized through a fish-eye system. Research on effective use of 2D spaces for multidimensional data visualization focus on careful selection of the relevant dimensions [17] and organizing data in hierarchical visualization structures, such as TreeMaps, along the relevant dimensions and mapping these two 2D spaces [3].

We note that taxonomies, hierarchical representations of terms that are important in a given application domain, are very effectively used in diverse knowledge-rich applications, including clustering, browsing/navigation support interfaces, and recommendation systems. The main difference between a taxonomy and a tag cloud is that tags in a taxonomy are *contextually organized*: a node in a given taxonomy

[*] This work is partially supported by NSF Award #0735014,"MAISON: Middleware for Accessible Information Spaces on NSDL".

KDD'08, August 24–27, 2008, Las Vegas, Nevada, USA.
ACM 978-1-60558-193-4/08/08.

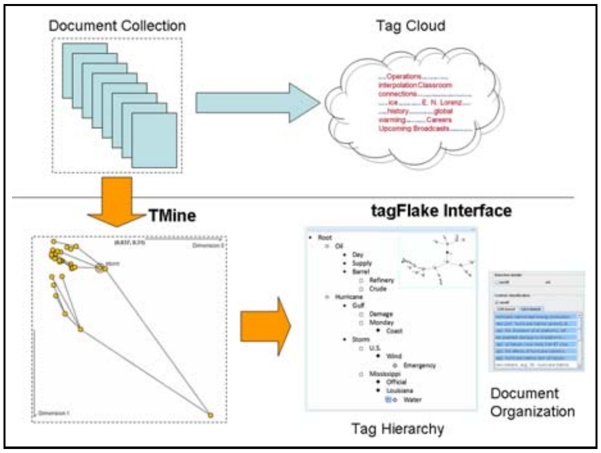


Figure 2: Unlike traditional tag clouds, `tagFlake` analyzes the given document collection for contextual relationships between the tags and creates tag hierarchies that can be used to navigate within the tag space and classify and organize documents

clusters all its descendant nodes and essentially acts as a context for the descendant nodes; similarly, descendants of a given node also act as a context for the node, differentiating the ancestor from others that are similarly labeled.

[8] describes a system to visualize the semantic information contained in a set of textual documents, relying on Latent Semantic Analysis. The proposed system is purely a visualization system. [9, 10] present a supervised method to mine ontological relations on concepts extracted from text corpora, by furthering their use of Latent Semantic Analysis techniques. [4] proposes a Natural Language Processing based method applying Formal Concept Analysis (FCA) to automatically acquire taxonomies. [18] presents a completely unsupervised, language neutral, method uncovering latent topics in text documents. To identify the concepts, it relies on the Markov Chain Monte Carlo process of Gibbs sampling, following the Latent Dirichelet Allocation model, while we use SVD transform.

Taxaminer [13] automatically constructs a taxonomy from a large corpus of documents, by means of a suite of clustering and NLP techniques. OntoMiner [5, 6] is an unsupervised system which leverages the HTML regularities in the Web documents to detect key-concepts and bootstrapping and populating specialized domain ontologies by organizing and mining taxonomy-directed Web sites.

1.2 Overview of `tagFlake`

`tagFlake` analyzes tag clouds and places tags extracted from a collection of documents (e.g. blogs and news articles) into a hierarchy which relates the tags in terms of the contexts in which they appear in the documents. As our running example, we use a corpus of 750 short news articles about "Katrina" published between August 25 2005 and March 4 2006 [1].

The reason why we chose this data set is that the event has a multitude of, now well understood, facets, including geographic, humanitarian, economic (e.g., employment and energy related), politic (local, regional, and federal), that spanned and evolved over the chosen period of time.

Unlike the prior works, `tagFlake` aims to visualize the tags in a latent semantic space without having to rely on high-dimensional visualization mechanisms or having to compress the available number of dimensions to only 2. Instead, `tagFlake` helps the user navigate within the multi-dimensional space using the tag hierarchy extracted by `TMine` algorithm as a blueprint (Figure 3) and by adaptively picking the best pair of dimensions for projecting the tags and the documents, based on the user's navigation context.

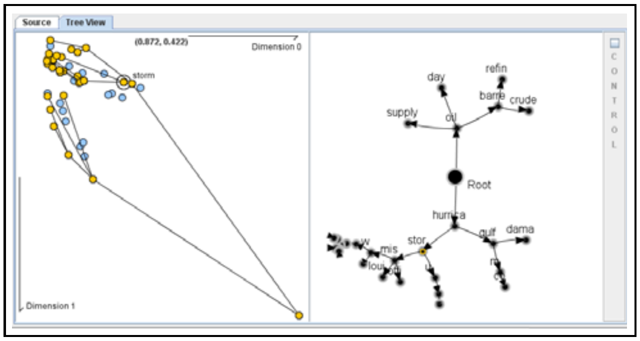
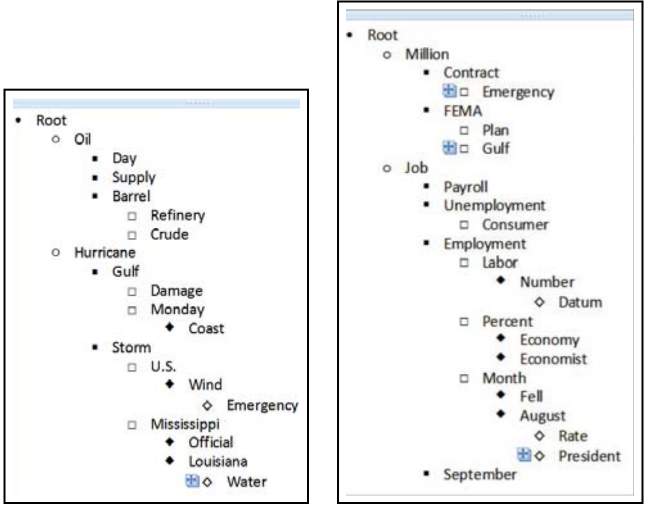


Figure 3: The `tagFlake` helps the user navigate within the multi-dimensional space using the tag hierarchy extracted by `TMine` algorithm as a blueprint: `tagFlake` visualizes the extracted tag hierarchy (hyperbolically, in this example) on the right hand side of the interface. It also visualizes the tags (and the documents from which the tags are extracted) in a latent semantic space (on the left hand side). The tags in the tags space are organized along the hierarchy extracted by `TMine`. The two dimensions used for visualizing the tag/document space are adaptively selected by `tagFlake` based on the current navigation context (the dimensions are chosen for the tag "storm" in this example)



- Root
  - Oil
    - Day
    - Supply
    - Barrel
      - Refinery
      - Crude
  - Hurricane
    - Gulf
      - Damage
      - Monday
        - Coast
    - Storm
      - U.S.
        - Wind
          - Emergency
      - Mississippi
        - Official
        - Louisiana
          - Water

(a)Aug 30-Sept 1 2005

- Root
  - Million
    - Contract
      - Emergency
    - FEMA
      - Plan
      - Gulf
  - Job
    - Payroll
    - Unemployment
      - Consumer
    - Employment
      - Labor
        - Number
          - Datum
      - Percent
        - Economy
        - Economist
      - Month
        - Fell
        - August
          - Rate
          - President
    - September

(b) February 10-13 2006

Figure 4: Tag hierarchies extracted from Katrina related news articles published at different times

Figure 4 presents an alternative (more end-user centric) visualization of the tag hierarchies, extracted by `TMine` from Katrina related news published at different times. The tag hierarchies not only show the important keywords in the articles, but also make the contexts in which these keywords are used explicit. We call this process *condensing the tag cloud* into *tag flakes*. For example, early during the disaster, the *hurricane* and its affects and *damage* on various parts of

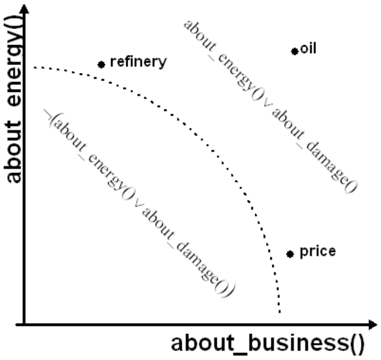


Figure 5: The extended Boolean interpretation of the tag space: according to this interpretation, in this example, the tag "*oil*" is more general than the tags "*refinery*" and "*price*"

the *Mississippi* and *Louisiana* and the impact of the disaster on the *oil* supply are the major topics. One month later, on the other hand, the cost of *contracts* and *FEMA*'s plans for reconstructing the *Gulf* and the impact of the damage on the *job and employment* numbers over the past *month* are the main topics of discussion. Each of the tag hierarchies presented here are extracted using only 15 articles randomly picked among those published on the stated dates.

2. BACKGROUND: TMINE

In this section, we provide a brief overview of `TMine`.

2.1 Tag Space Construction

Given a collection of text documents, the first step of TMine involves the use of Latent Semantic Analysis [7] based process to create the corresponding tag space. Each article is represented as a term frequency vector (normalized with respect to the article's length), after stop word elimination and lemmatization pre-processing. *LSA* is based on the *SVD* matrix factorization technique [11]: the input data to the SVD module is the article-term occurrence matrix, $A(m, n)$, which is decomposed in three matrices, $A = U\Sigma V^T$, where

- U is a real, column-orthonormal $m \times r$ matrix, such that $UU^T = I$,
- Σ is an $r \times r$ positive valued diagonal matrix, where $r \leq min(m, n)$ is the rank of the matrix, A,and
- V^T is the transpose of a real, column-orthonormal $r \times n$ matrix, such that $VV^T = I$

The r column vectors of V form an r dimensional basis. These r dimensions are referred to as the *latent semantics* of the given corpus of documents.

2.2 Organizing Tags into a Hierarchy

The position of a tag in a hierarchy needs to reflect two important aspects of the underlying semantics: the degree of generalization/specialization of tag-nodes, captured by their levels in the tree, and the degree of contextual-similarity between tags. Each branch of a taxonomy contains tags which are more similar to the other tags in the same branch than to the tags in the other branches.

`TMine` uses both generality and contextual relationship between the tags to construct a tag hierarchy. For this purpose, it applies the Extended boolean model, which associates r predicates to the r extracted concepts, $A_1, \ldots, A_r$, and interprets the origin of the r-dimensional space $(0, \ldots, 0)$ as a (hypothetical) tag entry, $c_{(0,\ldots,0)}$, which does not satisfy any of these A_i predicates. It holds $\neg A_1(c_{(0,\ldots 0)}) \wedge \ldots \wedge \neg A_r(c_{(0,\ldots,0)}) = \neg(A_1(c_{(0,\ldots 0)}) \vee \ldots, \vee A_r(c_{(0,\ldots 0)}))$. Thus, under the extended Boolean model interpretation of the tag space, a tag, c, further away from $(0, \ldots, 0)$ better satisfies the logical formula, $A_1(c)\vee, \ldots, \vee A_r(c)$, than any other tag entry closer to $(0, \ldots, 0)$. Under this interpretation, the degree of *generality* of a tag with respect to semantic predicates, $A_1, \ldots, A_r$, can be quantified in terms of the *length* of the corresponding tag vector: the shorter a vector in the tag space is (i.e., the closer to the origin $c_{(0,\ldots,0)}$), the more specific its corresponding tag. This induces a total (generality) order on the tags, in the extracted latent concept space. The hierarchy construction process is then guided by the inferred generality order. To enforce the requirement that tags which are more similar to each other are clustered in the same subtree, we consider the dissimilarity graph $G(V, E)$, where V is the set containing all the extracted tags plus the dummy tag "All" representing the most general concept $(1, \ldots, 1)$ in the considered concept space, and E contains edges between any pairs of nodes, from the more general to the less general. Edge weights reflect the cosine dissimilarity between the tags associated to the connected vertices: the lower the weight, the higher the similarity. Thus, *minimizing* the overall cost of a subgraph corresponds to *maximizing* the similarity associated to the connected nodes. This justifies our choice of constructing the tag hierarchy by computing the *minimum spanning tree* of the dissimilarity graph. Figure 6 depicts a tag space before and after tag hierarchy construction.

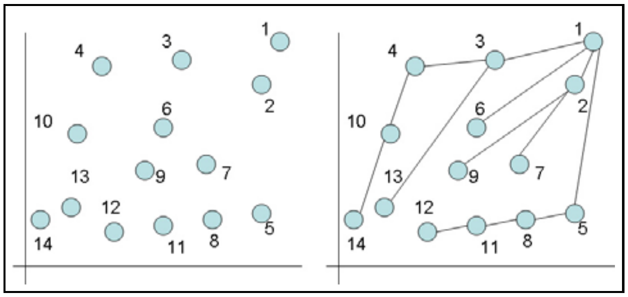


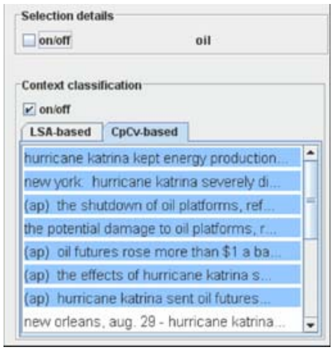
Figure 6: Outline of the hierarchy construction process: Node labels reflect the order in which the tags are inserted in the tree

3. USE FLOW AND DEMO PLAN

The system evolves along 3 major steps. First, the subsystem `TMine` extracts the relevant tags from the document corpus and organizes them in a semantic hierarchy. The hierarchy is then used for various complementary purposes. The tag hierarchy extracted by `TMine` is used by `tagFlake` as a context-informed navigation tool within the tag and document spaces. Text entries are associated to their more representative tags, and the hierarchy is used as a navigation tool. In addition, hierarchies extracted from articles published in different time frames are compared, and their differences are quantified and visualized to describe the evolution of the topics over time.

3.1 Context Informed Navigation in Tag and Document Spaces

In Section 1.2, we have provided an overview of the tag hierarchy visualization and navigation options provided by `tagFlake`. `tagFlake` associates to the tags in the hierar-



hurricane katrina kept energy production...
new york: hurricane katrina severely di...
(ap) the shutdown of oil platforms, ref...
the potential damage to oil platforms, r...
(ap) oil futures rose more than $1 a ba...
(ap) the effects of hurricane katrina s...
(ap) hurricane katrina sent oil futures...
new orleans, aug. 29 - hurricane katrina...

Figure 7: Interface of the classification module. All articles classified under "oil" (based on this tag's use context in the corresponding hierarchy) are listed.

chy the set of documents that the tags (when considered in the corresponding hierarchical context) best represent. The degree of matching between a tag and a document is captured by means of structural similarity, for which `tagFlake` relies on the *Concept Vectors in Concept Space (CP/CV)* approach [12]. This method associates to each tag a unique tag-vector which captures its structural relationships with other tags.

By measuring the similarity between the tag-vector (corresponding to the given tag) and the documents in the corpus, `tagFlake` computes the degree of match between the tag and the articles. Figure 7 shows an example.

3.2 Tracking Topic Evolution

The usage of the tag may evolve over time. For example, the term "oil" may initially be used within the context of damages to the refineries during a disaster, while its uses may later switch to discussion about the impact of resulting rises in the oil price and its impact on businesses. Thus, tag hierarchies also enable `tagFlake` to help users track the topic development patterns in the temporally evolving documents, such as news articles.

`tagFlake` provides a *topic development tracking module*, which can visualize the topical differences between two sets of documents in terms of the evolution of their tag hierarchies. Figure 8 shows the evolution tracking interface. `tagFlake` also provides interfaces which can track and visualize how the usage of a single tag evolves over time. The interface visualizes not only the changes in the popularity of a tag, but also the contexts in which the tag has been used over a given period of time.

3.3 Demo Scenario and Conclusion

In the demo we will use various news article and blog collections. We will demonstrate how the text documents of interest (chosen at demonstration time by user) are loaded into `tagFlake`, how the entries on which the user might want to concentrate her analysis are interactively picked, and how the relevant tags are extracted and organized in a navigation hierarchy by `TMine`. We then demonstrate the navigation, classification, and tracking features of `tagFlake`.

4. REFERENCES

[1] http://www.usatoday.com, http://www.abcnews.go.com, http://www.forbes.com, http://www.nytimes.com, http://www.smh.com.au, http://www.guardian.co.uk, http://news.bbc.co.uk, http://www.cbsnews.com, http://www.chron.com, http://www.washingtonpost.com, http://news.com.com, http://www.boston.com, http://www.iht.com.

[2] http://phasetwo.org/post/a-better-tag-cloud.html, 05/31/08

[3] G.Chintalapani, C. Plaisant, B.Shneiderman: Extending the Utility of Treemaps with Flexible Hierarchy. IV'04, 2004.

[4] P. Cimiano, S. Staab, J. Tane. Automatic Acquisition of Taxonomies from Text: FCA meets NLP. ECML/PKDD. Work. on Adaptive Text Extraction and Mining, 2003.

[5] H. Davulcu, S. Vadrevu, S. Nagarajan OntoMiner: Bootstrapping and Populating Ontologies From Domain Specific Web Sites. Semantic Web and Databases Work., 2003

[6] H. Davulcu, S. Vadrevu, and S. Nagarajan. OntoMiner: Bootstrapping Ontologies From Overlapping Domain Specific Web Sites. 13th Int. WWW Conference, 2004.

[7] S. Deerwester, S. Dumais, G.Furnas, R. Harshman, T. Landauer, K. Lochbaum and L. Streeter. Computer Information Retrieval using Latent Semantic Structure, US Patent, 1989.

[8] B. Fortuna, D. Mladenic, M. Grobelnik. Visualization of text document corpus. *Informatica J.*, 29 (2005).

[9] B. Fortuna, M. Grobelnik, D. Mladenic System for semi-automatic ontology construction. ESWC, 2006.

[10] B. Fortuna, M. Grobelnik and D. Mladenic. Semi-automatic data-driven ontology construction system SiKDD, 2006.

[11] Eckart, C., Young, G. The approximation of one matrix by another of lower rank. Psychometrika, 1936.

[12] J. W. Kim, K. S. Candan CP-CV: Concept Similarity Mining without Frequency Information from Domain Describing Taxonomies, CIKM, 2006.

[13] V. Kashyap, C. Ramakrishnan, C. Thomas, D. Bassu, T. C. Rindflesch, A. Sheth TaxaMiner: An Experimentation Framework for Automated Taxonomy Bootstrapping. TR Univ. Georgia, 2004.

[14] Y. Hassan-Montero, V. Herrero-Solana. Improving Tag-Clouds as Visual Information Retrieval Interfaces. InSciT2006.

[15] Y. Hassan-Montero, V. Herrero-Solana. Interfaz visual para recuperación de información basada en análisis de metadatos, escalamiento multidimensional y efecto ojo de pez El Profesional de la Informacion 15(4).

[16] Y. Qi and K. S. Candan. CUTS: CUrvature-Based Development Pattern Analysis and Segmentation for Blogs and other Text Streams. Hypertext06.

[17] J.Seo and B.Shneiderman. A Rank-by-Feature Framework for Unsupervised Multidimensional Data Exploration Using Low Dimensional Projections. Infovis'04 2004.

[18] E. Zavitsanos, G. Paliouras, G. A. Vouros, and S. Petridis. Discovering Subsumption Hierarchies of Ontology Concepts from Text Corpora. Int. Conf. on Web Intelligence, 2007

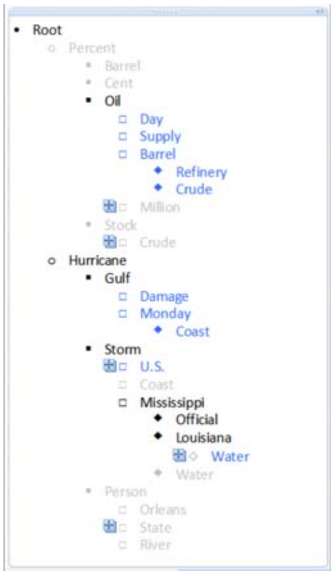


Figure 8: Interface of the topic development tracking module. Gray subtrees denote *depreciated* parts of the old hierarchy, while blue are *recent* additions