Conceptual Clustering Using Lingo Algorithm: Evaluation on Open Directory Project Data

Stanisław Osiński and Dawid Weiss

Institute of Computing Science, Poznań University of Technology,
ul. Piotrowo 3A, 60–965 Poznań, Poland,
E-mail: dawid.weiss@cs.put.poznan.pl, stanislaw.osinski@man.poznan.pl

Abstract. Search results clustering problem is defined as an automatic, on-line grouping of similar documents in a search hits list, returned from a search engine. In this paper we present the results of an experimental evaluation of a new algorithm named Lingo. We use Open Directory Project as a source of high-quality narrow-topic document references and mix them into several multi-topic test sets for the algorithm. We then compare the clusters acquired from Lingo to the expected set of ODP categories mixed in the input. Finally we discuss observations from the experiment, highlighting the algorithm's strengths and weaknesses and conclude with research directions for the future.

1 Introduction and related work

One of the most typical problems in searching the Internet is about locating documents that match some topic. This task is nowadays quite successfully handled by several major commercial search engines. However, when it comes to *explaining* the search result, to displaying what sort of topics the query matched and what sort of results have been returned, one's options are much more limited. A typical search engine's response to a query is a ranked list of documents along with their partial content (*snippets*). Providing any information about the internal relationships among the documents in the search result is very rare. *Search results clustering*, a search result visualization technique first introduced in the Scatter-Gather [4] system, attempts to provide the user with essential information about the structure of topics in the retrieved set of documents. We believe that users' high demands for such information are best expressed by constantly growing popularity of commercial systems, such as Vivisimo (`http://www.vivisimo.com`), or iBoogie (`http://iboogie.tv/`).

On a scientific side, the problem of search results clustering is an interesting niche of full-document text clustering. Simply porting the well known generic algorithms, such as agglomerative hierarchical clustering [5] or K-means does not work well, because the amount of data for the clustering algorithm is often extremely small and of low-quality (only snippets and document titles returned by the search engine are available).

In a related full-length paper published at the same conference [7], we describe the details of our novel algorithm Lingo, which we believe is able to

capture thematic threads in a search result. In other words: discover groups of related documents and describe the subject of these groups in a way meaningful to a human. Here, we present the results of our most recent experiment with the Lingo algorithm[1], where we selected a number of possibly narrow-topic categories from the Open Directory Project (`http://dmoz.org`). We then mixed documents from these categories to create multi-topic test sets for the algorithm. Finally, we analyzed how Lingo managed to split the test sets back into original categories and compare those results with another search results clustering algorithm—Suffix Tree Clustering [9].

Scientific goals of the experiment can be summarized by the following questions:

1. Is Lingo able to cluster similar documents? If so, what is the algorithm's performance for search results containing unrelated and closely related documents?
2. Is Lingo able to highlight outliers, defined as minor subsets of documents sharing a common topic, but unrelated to majority of the input?
3. Is Lingo able to capture cross-topic relationships (generalizations) among closely related subjects?
4. Are cluster labels meaningful about the topic they supposedly represent?
5. What are key differences between clusters created by Lingo and STC?

2 Related work

Work on search results clustering algorithms began from the Scatter-Gather system [4], where the Fractionation algorithm was used to organize search results into thematic groups. Suffix Tree Clustering, (STC), implemented in the Grouper system [9] pioneered in using recurring phrases as the basis for deriving conclusions about similarity of documents. Algorithms that followed a similar path include MSEEC [3] or SHOC [2]. Refer to [8] for a broader overview. Lingo uses a combination of phrases and Singular Value Decomposition (SVD) techniques. SVD and the related Latent Semantic Indexing were introduced to information retrieval as a way of separating concepts in textual data in [1].

Primary intention of this paper is to evaluate a search results clustering algorithm. Surprisingly, previous work in this area is very scarce. Most authors seem to agree that absolutely objective measures for calculating an algorithm's performance are impossible. There usually exists more than one "good" result of text clustering and even experiments performed with human experts show a surprising lack of consistency [6]. In spite of this, user feedback surveys have been the most favored technique of estimating an algorithm's

[1] The Lingo algorithm is quite complex and we could only afford a short description of it in this paper. We encourage the Reader to read [7] as an introduction to this experiment.

Identifier, collection size	ODP category path	Description
	Movies	
BRunner, 77	`Arts/Movies/Titles/B/Blade_Runner`	Information about the *Blade Runner* movie.
LRings, 92	`Arts/Movies/Titles/L /Lord_of_the_Rings_Series`	Information about the *Lord of the Rings* movie.
	Health care	
Ortho, 77	`Business/Healthcare /Products and Services/Orthopedic`	Orthopedic equipment and manufacturers
	Photography	
Infra, 15	`Arts/Photography /Techniques_and_Styles/Infrared`	Infrared photography references
	Computer science (databases)	
DWare, 27	`Computers/Software/Databases /Data_Warehousing/Articles`	Articles about data warehouses (integrator databases)
MySQL, 42	`Computers/Software/Databases/MySQL`	MySQL database
XMLDB, 15	`Computers/Software/Databases /XML/Proprietary`	Native XML databases
Postgr, 38	`Computers/Software/Databases /PostgreSQL`	PostgreSQL database
	Computer science (miscellaneous)	
JavaTut, 39	`Computers/Programming/Languages /Java/FAQs_Help_and Tutorials /Tutorials`	Java programming language tutorials and guides
Vi, 37	`Computers/Software/Editors/Vi`	Vi text editor

Fig. 1. Open Directory Project categories selected for the experiment

usefulness. An implicit, web server-log based, clustered interface efficiency test has been proposed in [9], but it requires an alternative 'regular' search interface to compare the results to. In [8,5], authors manually cluster the input data set and then compare the results acquired from an algorithm to this "ground truth" set of clusters. Our evaluation technique of mixing ODP categories is similar to this approach, but instead of manually clustering some search result, we assume that ODP categories represent single-topic (or at least narrow-topic) clusters already and create an artificial search result that we provide to the algorithm.

3 Description of the Lingo algorithm

Our novel algorithm, Lingo [7], is particularly suited to solving the problem of search result clustering. Unlike most other algorithms, it first attempts to discover descriptive names for future clusters and only then proceeds to assigning each cluster with matching documents. This reversed process, compared to other search results clustering algorithms, allows Lingo to partially avoid the trap of verbally unexplainable clusters.

Lingo consists of five phases. In phase one, input snippets (document fragments) are preprocessed—the text is separated into tokens (terms), an

attempt is made to identify each document's language and apply appropriate stemming and stop-word marking procedure (see [8] for definition of stemming and stop-words). In phase two, frequent terms and phrases are discovered in a combined set of all documents by smart utilization of suffix-arrays. Phase three is the actual group label induction. SVD is used to extract orthogonal vectors of the term-document matrix, believed to represent distinct topic in the input data [1]. A description of each orthogonal vector is assembled from previously extracted common phrases by using a Vector Space Model and calculating score of each phrase against the SVD-decomposed matrix. After label pruning, the algorithm enters phase four, which consists of cluster content discovery. Vector Space Model is used again to apply group labels as artificial queries against the input document set. Highest scoring documents for each cluster are assigned as that cluster's content. The last phase is applying a score function to all clusters to sort them for display.

This paper is effectively an extension and verification of ideas presented in [7]—published at the same conference and available in the same volume of conference proceedings. For this reason we allowed ourselves to only briefly indicate the key concepts of Lingo. We encourage the Reader to refer to [7], where a more detailed description of the algorithm can be found.

4 The experiment

4.1 Input data preparation

Open Directory Project is a tree-like, human-collected thematic directory of resources in the Internet. Each branch of this tree, called a category, represents a topic and contains links to resources in the Internet that relate to this topic. Every link added to the ODP must be accompanied by a short description (25–30 words) of a resource it points to. We assumed these short descriptions would serve as a substitute for snippets returned in a search engine's response to user query.

To answer the questions stated in Sect. 1, we prepared several artificial data sets for Lingo. We selected 10 categories out of approximately 575,000 present in the ODP database. The exact choice of categories was random, given that each selected category contained at least 10 documents and had a meaningful English description of each document inside. To check how Lingo clusters similar subjects, a subset of related categories was drawn from one parent branch of ODP. The final choice of categories was connected to four abstract subjects: *movies*, *health care*, *photography* and *computer science*. The last group contained a set of related sub-categories about various database systems. Categories and their topics are presented in Fig. 1.

The extracted categories were then merged into test sets, constructed to answer specific questions given in Sect. 1. Details about the test sets and rationale for each of them is presented in Fig. 2.

Identifier	Merged categories	Test set rationale
G1	*LRings*, *MySQL*	Separation of two unrelated categories.
G2	*LRings*, *MySQL*, *Ortho*	Separation of three unrelated categories.
G3	*LRings*, *MySQL*, *Ortho*, *Infra*	Separation of four unrelated categories, highligting small topics (*Infra*).
G4	*MySQL*, *XMLDB*, *DWare*, *Postgr*	Separation of four conceptually close categories, all connected to database.
G5	*MySQL*, *XMLDB*, *DWare*, *Postgr*, *JavaTut*, *Vi*	Four conceptually very close categories (database) plus two distinct, but within the same abstract topic (computer science).
G6	*MySQL*, *XMLDB*, *DWare*, *Postgr*, *Ortho*	Outlier highlight test – four dominating conceptually close categories (databases) and one outlier (*Ortho*)
G7	All categories	All categories mixed together. Cross-topic cluster detection test (movies, databases).

Fig. 2. Merged categories – test sets for the experiment

4.2 Algorithm's implementation and thresholds

We used Lingo's implementation available as part of the Carrot^2 system (`www.cs.put.poznan.pl/dweiss/carrot`). Lingo component uses Porter stemming algorithm for documents it recognizes as being in English and an appropriate stop-word list. We initially kept all control thresholds of the algorithm (see [7] for explanation) at their default values (*cluster assignment threshold*—0.15, *candidate cluster threshold*—0.775). Later we repeated the experiment for other threshold levels and observed no noticeable change that would affect our conclusions from the experiment. The experiment was performed on a live on-line demo of Carrot^2, using a set of automatic scripts written in BeanShell and XSLT.

4.3 Criteria for evaluation of results

As mentioned in the introduction, numerical evaluation of search results clustering is always problematic. Fuzzy definitions of cluster's value, such as: "good", "meaningfull" or "concise" are hard to define numerically. Besides, even human evaluation performed by experts can vary a great deal between individuals, as shown in in [6]. In our previous work [8] we made an attempt to employ a statistical approach to comparing a clustered set of search results to an "ideal, ground truth" set of clusters acquired from human experts. Unfortunately, any differences between the result of automated clustering and the "ideal" structure would count as a mistake of the algorithm, even if in fact the algorithm just made a different, equally justified decision—for instance splitting a larger subject into its sub hierarchy. We did not want to penalize Lingo like this, because it was not the objective of the experiment. Instead, we decided to manually investigate the structure of created clusters, represented by classes-in-cluster distribution (see Fig. 3), and try to answer the experiment's questions based on such analysis.

4.4 The results

We will discuss the results in the order of questions stated in Sect. 1.

For each test set, Lingo created between 24 and 36 clusters. In Fig. 3, we present selected results showing how documents from ODP categories contributed to the overall cluster size. Ideally, clusters representing unrelated topics should be of solid color (originate from a single category). Also, all original categories mixed into a test set should be present in the result, preferably in topmost clusters. For unrelated topics (G1–G3, Fig. 3), this expectation has been met. In G1, *LRings* category has a much more complex internal structure and spreads into more clusters. *MySQL* clusters are however present and not mixed with the other category. The same applies to G3, where four input categories are represented in the top 5 clusters. Even though category *Infra* was much smaller, it is still highlighted as a separate subject. Topics were successfully separated even for conceptually similar categories, in test G4. Note however, that cluster "MySQL Server" contains only 50% of documents from *MySQL* category. This indicates a misassignment problem in Lingo's cluster content discovery phase, which is also present in cluster "Information on infrared", in G7. The source of this problem lies in the fact that Lingo does not operate on full phrases once it has discovered group labels. So "Information on infrared" cluster matches documents with strong influence of "Information" and "infrared", but not necessarily both of them—for example, "Information and Images" document matched, even though it originated from *LRings* category.

An outlier test G6 has been handled correctly in our experiment. *Ortho* category was not obscured by database-related documents and was highlighted at top positions of the group ranking. The same held for smaller categories, for example *Infra* in test G7, or *XMLDB* in test G5. Interestingly, category *XMLDB* vanished from results of test G6 and G7, with some of its documents assigned to other database-related clusters. We suppose Lingo did not separate *XMLDB* because it was too close to other database-related categories and at the same time too small to create a separate conceptual group during label discovery.

Lingo captured some of the cross-topic relationships; for example, in G7, cluster "Movie review" nicely combined documents from *LRings* and *BRunner* categories, similarly in G4, clusters "SQL", or "Tables" combined documents spanning different categories. However, cross-topic spanning capabilities of Lingo was not evident and perhaps requires a different test.

Cluster labels were perceptively meaningful and human-readable. Exceptions included mostly single term labels, which were ambiguous ("free"), or too broad ("News"). We noticed an interesting effect of group re-labeling, depending on the granularity (size) of the discovered groups. For example, cluster "MySQL" in G1, becomes "MySQL Database" in G2, to go back to "MySQL" in G3.

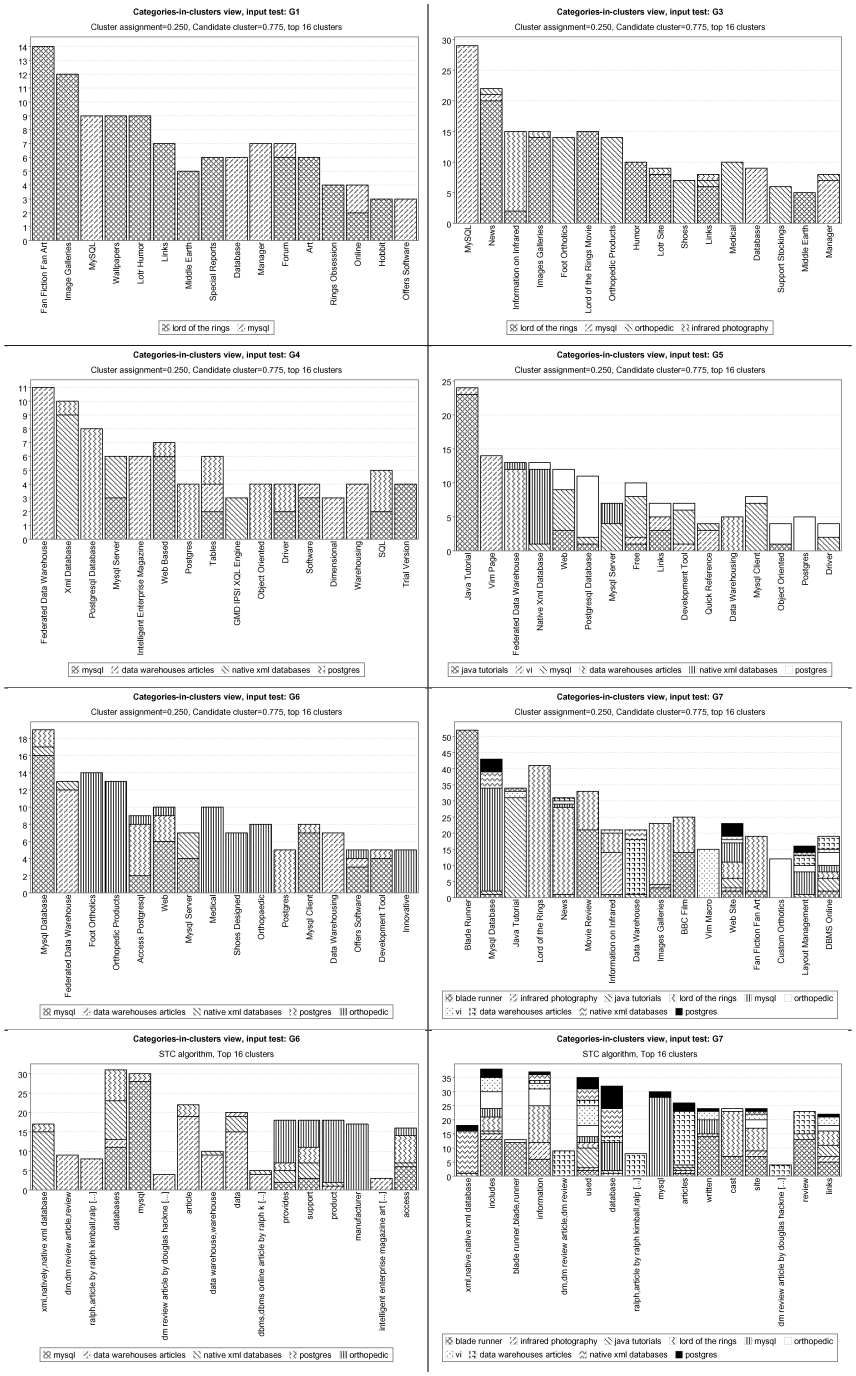

Fig. 3. Structure of clusters for test sets in case of Lingo and STC

In spite of minor noise groups, Lingo's clusters seemed better (meaningful) than STC's. We think that selection of group labels in STC, based solely on common phrase frequency, is inferior to Lingo's SVD-based method. For example, in test G6, the outlier category *Ortho* was predominated by database-related clusters. STC also failed to clearly separate topics in test G7, choosing common terms for group labels and mixing all categories based on frequent, but meaningless words ("used", "site").

5 Conclusions and future work

We have presented the results of an interesting approach to evaluating a search results clustering algorithm Lingo. It is clear that the results from our mostly non-numerical analysis are not to be taken as an ultimate proof of Lingo's correctness or superiority over other algorithms, but we believe they to some degree support our claims that Lingo does a decent job at separating topics present in search results and labels them informatively, at least compared to the STC algorithm. There were some clear indicators where Lingo could be improved, which count as another valuable outcome of the experiment. Some work will be required in cluster content discovery phase. Some advanced methods of either pruning single-term labels, or organizing them into hierarchical super-sub topic structures would increase the legibility of the clusters' structure.

Acknowledgment

The authors would like to thank an anonymous reviewer for helpful suggestions. This research has been supported by funds from an internal university grant.

References

1. Michael W. Berry, Susan T. Dumais, and Gavin W. O'Brien. Using linear algebra for intelligent information retrieval. Technical Report UT-CS-94-270, 1994.
2. Zhang Dong. *Towards Web Information Clustering*. PhD thesis, Southeast University, Nanjing, China, 2002.
3. Peter Hannappel, Reinhold Klapsing, and Gustaf Neumann. MSEEC - a multi search engine with multiple clustering. In *Proceedings of the 99 Information Resources Management Association International Conference*, Hershey, Pennsylvania, May 1999.
4. Marti A. Hearst and Jan O. Pedersen. Reexamining the cluster hypothesis: Scatter/gather on retrieval results. In *Proceedings of SIGIR-96, 19th ACM International Conference on Research and Development in Information Retrieval*, pages 76–84, Zürich, CH, 1996.

5. Yoëlle S. Maarek, Ronald Fagin, Israel Z. Ben-Shaul, and Dan Pelleg. Ephemeral document clustering for web applications. Technical Report RJ 10186, IBM Research, 2000.
6. Sofus A. Macskassy, Arunava Banerjee, Brian D. Davison, and Haym Hirsh. Human performance on clustering web pages: A preliminary study. In *Knowledge Discovery and Data Mining*, pages 264–268, 1998.
7. Stanisław Osiński, Jerzy Stefanowski, and Dawid Weiss. Lingo: Search results clustering algorithm based on Singular Value Decomposition. Submitted to Intelligent Information Systems Conference 2004, Zakopane, Poland, 2003.
8. Jerzy Stefanowski and Dawid Weiss. Carrot^2 and language properties in web search results clustering. In *Proceedings of AWIC-2003, First International Atlantic Web Intelligence Conference*, volume 2663 of *Lecture Notes in Computer Science*, pages 240–249, Madrid, Spain, 2003. Springer.
9. Oren Zamir and Oren Etzioni. Grouper: a dynamic clustering interface to Web search results. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(11–16):1361–1374, 1999.