Modifications for the Cluster Content Discovery and the Cluster Label Induction Phases of the Lingo Algorithm

Seyfullah Demir, Ebru A. Sezer, and Hayri Sever

Abstract—**Search results clustering techniques help end users to find their related results easier. Both producing correct cluster contents and assigning descriptive, meaningful labels to the clusters are crucial for these techniques. *Lingo* is one of the most popular algorithms which consider both and it is known as a *description-comes-first* algorithm. Lingo has success on assigning descriptive, human-readable cluster labels, but it actually has a minor drawback on assigning documents to the clusters, which cause low recall values. In this paper, we propose two main modifications for the *Cluster Content Discovery* and the *Cluster Label Induction* phases of the Lingo algorithm. The evaluation of the experimental result shows that, although it causes a slight decrease in the precision, our modified Lingo algorithm provides quite higher recall and f-measure values.**

Index Terms—**Information retrieval, search results clustering, cluster content discovery, cluster labeling.**

I. Introduction

Search engines are used in order to get the relevant results for a query. When users run a query on a search engine, a ranked list of the search results are returned with their snippets (partial content). If users run too general or ambiguous queries, it may be difficult to reach what they need, since a large number of results are returned and similar results are not grouped together. Search results clustering techniques are used to overcome this problem.

By employing search results clustering techniques, the search results are returned as labeled groups. A search result clustering algorithm should correctly cluster the results and also should produce descriptive labels for each cluster. Traditional clustering algorithms are not generally good enough at selecting descriptive labels for clusters. To overcome this problem, description-aware and description-centric algorithms [1] were developed. Zamir and Etzioni pioneered the approach of using recurring phrases in the search results clustering processes, within *Suffix Tree Clustering (STC)* algorithm in Grouper system [2]. In STC algorithm, the documents which share the same phrases are grouped together and the phrases that they share are used as cluster labels.

Later on, Osiński introduced the Lingo algorithm in his master thesis [3], and in [4] Osiński and his colleagues presented the Lingo. Lingo is a description-centric algorithm, in which the cluster labels are determined first, and then document assignments to the clusters are made. Osiński and Weiss presented an evaluation of the Lingo algorithm on the Open Directory Project (ODP) data [5]. Although their analysis was mostly non-numerical, they showed that Lingo separates the topics in the search results, better than the STC. Osiński and Weiss then showed Lingo produces significantly purer clusters than STC, by demonstrating a numerical analysis [6]. After that, Osiński investigated the effects of the matrix factorization method used in the search results clustering algorithm. He compared four different methods that are SVD (Singular Value Decomposition), NMF (Non-negative Matrix Factorisation), LNMF (Local Non-negative Matrix Factorisation) and CD (Concept Decomposition) in terms of topic separation, outlier detection and label quality [7]. He showed that NMF methods perform better than other methods including SVD, which was used in the original algorithm. According to the evaluation of his experiments, he also showed that Lingo with NMF-ED (NMF with Euclidean Distance minimisation) is significantly better than STC and TRC (Tolerance Rough Set Clustering) [8], in terms of topic separation and outlier detection.

In 2010, Sameh and Kadray proposed a modification in the *Frequent Phrase Extraction* phase of the Lingo algorithm [9]. They expanded the frequent phrases by including the synonyms which was obtained through the WordNet database [10]. They used the synonyms also when document assignments to the clusters are made. As a result, their algorithm could produce clusters that include the documents which contain the synonyms of the cluster labels, as well.

In Ref. [6], Osiński and Weiss mentioned a future work to enhance the Lingo's document assignment phase, due to low recall values. Since documents were assigned to the clusters using classic VSM (Vector Space Model) [11] approach, some irrelevant documents could be assigned to the clusters, while some semantically relevant documents could not be assigned. As the former could cause lower precision, the latter could lead to low recall. In the current implementation (v3.7.1) of the Lingo algorithm, instead of classic VSM approach, a binary similarity approach is used. Currently, a document is assigned to a cluster if it contains the stems of all words (except stop words) of the cluster label. Considering the results of the current implementation of the Lingo algorithm, we can see that the precision value is satisfactory, but the recall value can still be enhanced.

In this study we propose modifications for Lingo's Cluster Content Discovery and Cluster Label Induction phases. In Cluster Content Discovery phase, we propose a modification to overcome the weakness in the assignment process. According to our proposal, if a document contains the stem of

Manuscript received July 22, 2013; revised September 30, 2013.

The authors are with Hacettepe University, Turkey (e-mail: seyfullahdemir@gmail.com).

DOI: 10.7763/IJCTE.2014.V6.842

the word itself or the stem of at least one synonym, for each word in the label**,** then it is assigned to the cluster. Additionally, in Cluster Label Induction phase, we propose a modification to match abstract concepts with cluster label candidates. In Lingo, the term-abstract concept matrix is attained, as a result of reducing the term-document matrix. Moreover, the term-label candidate matrix is built by conforming to the same term space as the term-abstract concept matrix. For the matching process of abstract concepts and labels, column vectors of both are compared via cosine similarity method. As another difference, we find the documents which are related to the abstract concepts and then match the abstract concepts with the label candidates by using the number of their common documents as similarity measure.

II. Method

In this section, firstly we give some brief information about the original and the current Lingo algorithm (v3.7.1), and then present our modification proposals for it.

A. *Original Lingo Algorithm*

A summarized version of the original Lingo algorithm is given below. More detailed algorithms can be seen in [3], [4].

1) Preprocess documents. For each document; do text filtering, identify the language of the document, apply stemming process and mark stop words.
2) Discover frequent terms and phrases as label candidates.
3) Discover abstract concepts by using SVD.
4) Match the abstract concepts with best matching label candidates. Let the matched label candidates become the cluster labels.
5) Prune similar cluster labels.
6) Determine cluster contents for each cluster label by using classic VSM approach.
7) Sort clusters according to calculated cluster scores.

B. *Current Lingo Algorithm*

A brief algorithm of the current Lingo is given below:

1) Preprocess documents
- Extract frequent phrases and single words as cluster label candidates.
- Determine the assigned documents for each label candidate.
- Filter out the label candidates that contain less number of documents than the minimum cluster size threshold.
2) Build the term-document matrix using the stems of the label candidates (except the stop words in the label candidates).
3) Reduce the term-document matrix to the term-abstract concept matrix according to the desired cluster count base threshold.
4) Match the abstract concepts with the cluster label candidates.
5) Select the cluster label candidates that matched with an abstract concept as the labels of the determined clusters.
6) Merge clusters that share higher percentage of documents than the cluster merging threshold.
7) Form the final clusters for presentation.

There are some major differences between the original and the current algorithm. We list the three important changes as follows:

- Default matrix factorization method used in the Cluster Label Induction was changed from SVD to NMF-ED, since the NMF-ED method performs best as showed in [7]
- Documents are not assigned to clusters using classic VSM approach anymore. Instead of the VSM approach, a binary similarity function is used to determine whether a document should be assigned to a cluster or not. A document is assigned to a cluster if the document contains the stems of all words in the cluster’s label.
- Cluster merging phase was included before final clusters are formed. The clusters, which share common documents with a higher percentage than a cluster merging threshold, are merged.

C. *Proposed Modifications for Current Lingo Algorithm*

In order to enhance the low recall value of the Lingo algorithm, we propose two main modifications.

Cluster Content Discovery Phase: We propose a method which aims to provide that the documents, which do not include all of the stems of all words in a cluster’s label whereas those are semantically related to it, could also be assigned to the related cluster. Our method requires a lexical database which can provide synonyms for a given word.

According to the proposed method, for a single word label candidate, the documents that include the stem of the word itself or a stem of at least one of its synonyms are assigned to the cluster label. As for phrase label candidates, the documents that contain the stem of actual word or a stem of at least one of its synonyms, for each word of the label, are assigned to the cluster. We employed the WordNet lexical database as synonym supplier component. Synonym set that is retrieved for a word includes all of the synonyms which can be members of any type and related to any sense.

Cluster Label Induction Phase: In Lingo, the abstract concepts that latently exist in the input document set are discovered by means of SVD [4]. Moreover, the frequent phrases are thought to be potentially capable of describing the abstract concepts [3]. Therefore, each abstract concept vector is matched with a frequent phrase which is a label candidate.

We propose a modification in the abstract concept-label candidate matching process. In the current Lingo algorithm, the matching process is performed by comparing the abstract concept and the label candidate vectors, which lie on the same term space, by using the cosine similarity function. Since the labels generally consist of a few words, their vectors are mostly so sparse. We noticed that comparison of abstract concept vectors with sparse label vectors might not be so successful in selecting correct labels for the abstract concepts. To overcome this, we firstly propose to enrich the label vectors, by including stems of the synonyms of the words in labels, if that stems are included in the term space.

We further propose a new abstract concept-label matching approach for Lingo. In the Lingo algorithm, the term-document matrix is reduced to the term-abstract concept matrix (base matrix) and also the document-abstract concept matrix (coefficient matrix), via a selected factorization method. The latter actually can reveal the similarities between documents and abstract concepts. By using this information, we can determine the documents related to each abstract concept. Since we also have the assigned documents to each label candidate, we can use the number of common documents between the abstract concepts and the label candidates as a similarity measure. For each abstract concept, the top-most similar label candidate is matched. According to this approach, the number of final labels (clusters) could be less than the abstract concept number. For the proposed method, there are two options to determine the abstract concepts that a document should be assigned to:

- 1st Option: Assign a document to the abstract concepts that the document is similar with a higher score than a document-abstract concept similarity threshold.
- 2nd Option: Assign a document to the top-most similar abstract concept only.

For the first option, the coefficient matrix should be column-length-normalized prior to the assignment process.

III. Experiments

An open source implementation of the Lingo algorithm is provided in the Carrot^2 [12], which is an open source search results clustering engine. The Carrot^2 is implemented in Java. The proposed methods were experimented on the Carrot^2 engine. To retrieve the synonyms from the WordNet database, the Java API for WordNet Searching (JAWS) [13] was used.

In our experiments, we used the AMBIENT (AMBIguous ENTries) dataset [14]. It contains 44 ambiguous topics that are selected from the disambiguation pages of Wikipedia. Each topic includes a set of subtopics and 100 ranked documents that were retrieved from a search engine (January 2008). In the dataset, the documents for each topic are matched with the subtopics; whereas some documents are not matched with any subtopics and some subtopics do not contain any documents. In our experiments, we compared our resulting clusters with the given clusters of AMBIENT and we used five different metrics such as contamination, precision, recall, f-measure and normalized mutual information (NMI) to evaluate the experiment results. The Carrot^2 engine provides the calculation of these metrics.

The contamination metric is used to evaluate the purity of the resulted clusters. Its weighted average value for the whole cluster set is calculated. When its value gets closer to zero, it means purer clusters are produced, and vice versa. For the precision, recall and f-measure metrics, the weighted average values are calculated, too. For each true cluster for a query, the cluster which achieves the best f-measure is selected from produced clusters, and then precision, recall and f-measure metrics are calculated. The weighted average values of these metrics are then calculated by using the size of the true clusters and their values. NMI metric is also used to evaluate the quality of the clusters. Its value will be 1, for a perfect clustering algorithm. We use the averages of the weighted average values over all topics in the dataset (Table II, Fig. 1 and Fig. 2).

Our experimental stage consists of 8 sequential steps, which we tagged them as from S1 to S8. The steps are listed in Table I with their definitions. For all steps, we used the default values for the parameters and thresholds, which are defined in the current implementation for Lingo.

In Table II, the values of the experiment results are shown for each step. Fig. 1 shows the f-measure, precision and recall values and Fig. 2 shows the contamination and NMI values.

According to Table II, it can be seen that S2 increases the f-measure and recall values, as it causes a slight decrease in precision, compared to S1. If S2 and S3 are compared, it can be seen that S3 produces purer and more precise clusters. According to these results, it can be said that employing synonyms in document assignment process can provide better clusters.

TABLE I: Definitions of Sequential Steps

Step	Definition
S1	The current implementation of the Lingo algorithm as of release 3.7.1
S2	The modification in cluster content discovery phase, for both single word and phrase label candidates, is employed
S3	The modification in cluster content discovery phase, for only phrase label candidates, is employed
S4	The modification used in S2 and the modification of enriching label vectors, in cluster label induction phase, are employed
S5	The modification in cluster label induction phase by using the first option with the similarity threshold value of 0.75 is employed
S6	The modification in cluster label induction phase by using the second option is employed
S7	The modifications used in both S3 and S5 are employed
S8	The modifications used in both S3 and S6 are employed

In Table II, it is also shown that S4 cannot make any valuable improvement over S2. This shows that enriching label vectors does not enhance abstract concept-label candidates matching.

Fig. 1 shows that our proposed abstract concept-label candidates matching method is successful, since f-measure and recall values for S5, S6, S7 and S8 are dramatically higher than S1 and Fig. 1 also shows that using the second option, for the proposed matching method, leads slightly better results than using the first option. Moreover, it can be seen that including the modification in cluster content discovery phase provides an improvement over S5 and S6, in S7 and S8. As Fig. 2 shows, for S5, S6, S7 and S8, higher NMI and lower contamination values are achieved, compared to the S2 and S4.

The best improvement was achieved by applying the modifications for abstract concept-label candidate matching method by using second option and employing synonyms in document assignment process for the phrase label candidates (S8). Due to the size limitation, we further compare the more detailed results only for S1 and S8. Therefore Table III, Table V–Table VIII and Fig. 3 show the related results for only S1 and S8. Moreover, the Beagle topic is selected randomly for the comparisons.

In Table IV, the partitions (true clusters) for the Beagle topic are shown with their sizes and definitions. Table

V–Table VIII show the values of f-measure, precision and recall and the labels of the best matching clusters, for each true partition. In addition, Table III shows the weighted average precision, recall and f-measure values for the topic.

From Table V, Table VII and Table VIII, it can be seen that S8 significantly outperforms S1. Only for P2, S1 seems better in the results, but P2 can be seen as an outlier subtopic, since it contains only 2 of the total 86 documents. The improvements can also be seen in the Table III, which demonstrates the weighted average values for the Beagle topic.

TABLE II: AVERAGE VALUES OVER ALL TOPICS

Steps	Contamination	F-Measure	Precision	Recall	NMI
S1	0.260	0.630	0.908	0.554	0.651
S2	0.338	0.654	0.891	0.590	0.646
S3	0.281	0.638	0.906	0.566	0.646
S4	0.335	0.643	0.886	0.578	0.643
S5	0.291	0.730	0.884	0.688	0.691
S6	0.296	0.734	0.880	0.694	0.693
S7	0.298	0.741	0.880	0.703	0.692
S8	0.294	0.743	0.878	0.707	0.691

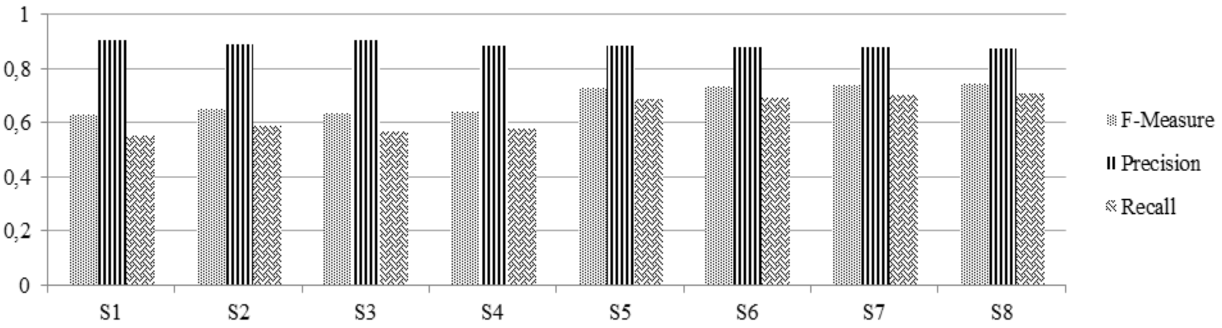


Fig. 1. Average F-measure, precision and recall values over all topics.

TABLE III: WEIGHTED AVERAGE RESULTS OF THE SELECTED STEPS FOR THE BEAGLE TOPIC

Step	F-Measure	Precision	Recall
S1	0.351	0.988	0.221
S8	0.622	0.960	0.477

TABLE IV: TRUE PARTITIONS FOR THE BEAGLE TOPIC

Partition	Definition	Document Count
P1	Beagle is a dog breed	55
P2	HMS Beagle, the ship in which Charles Darwin undertook the travels during which he made many observations which became important for his formulation of his theory of evolution	2
P3	Beagle 2, a failed British Mars lander named after HMS Beagle. It crashed on 25 December 2003	11
P4	Beagle (software), a desktop search service for GNU/Linux users.	18

TABLE V: RESULTS FOR THE P1 PARTITION OF THE BEAGLE TOPIC

	S1	S8
Label	Adopting	Breed
F-Measure	0.281	0.571
Precision	1.000	1.000
Recall	0.164	0.400

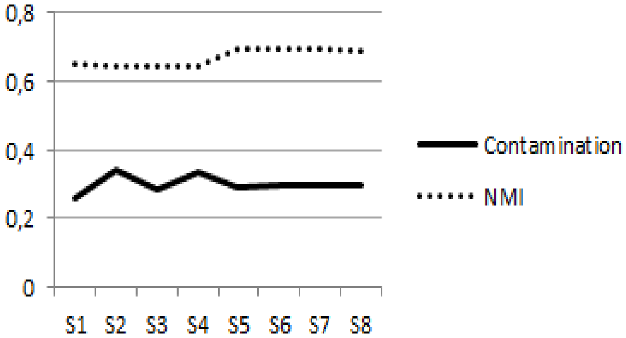


Fig. 2. Average contamination and NMI values over all topics.

TABLE VI: RESULTS FOR THE P2 PARTITION OF THE BEAGLE TOPIC

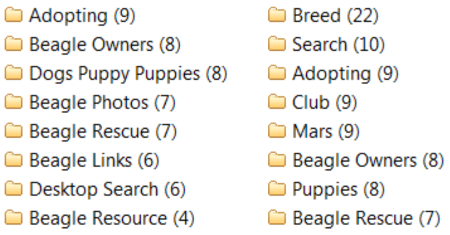
	S1	S8
Label	Project	Beagle Resource
F-Measure	0.500	0.286
Precision	0.500	0.200
Recall	0.500	0.500

TABLE VII: RESULTS FOR THE P3 PARTITION OF THE BEAGLE TOPIC

	S1	S8
Label	ESA’s Mars Express	Mars
F-Measure	0.429	0.900
Precision	1.000	1.000
Recall	0.273	0.820

TABLE VIII: RESULTS FOR THE P4 PARTITION OF THE BEAGLE TOPIC

	S1	S8
Label	Desktop Search	Search
F-Measure	0.500	0.643
Precision	1.000	0.900
Recall	0.333	0.500



- Adopting (9)
- Beagle Owners (8)
- Dogs Puppy Puppies (8)
- Beagle Photos (7)
- Beagle Rescue (7)
- Beagle Links (6)
- Desktop Search (6)
- Beagle Resource (4)

- Breed (22)
- Search (10)
- Adopting (9)
- Club (9)
- Mars (9)
- Beagle Owners (8)
- Puppies (8)
- Beagle Rescue (7)

(a) (b)

Fig. 3. The labels of the clusters for the beagle topic; (a) Generated by S1 and (b) Generated by S8.

In Fig. 3, the lists of the top-8 produced clusters for the Beagle topic for S1 and S8 are demonstrated. According to it, while S1 lists two clusters, which match a true cluster, in 1st and 7th positions, S8 list three clusters in 1st, 2nd and 5th positions. These results show that the proposed methods outperformed the current Lingo algorithm.

IV. Conclusion

We propose two main modifications for the Lingo algorithm, in order to eliminate the disadvantages due to the low recall values. First of them is to benefit from the synonyms for document assignments to the cluster labels. The other is to use the number of common documents between the abstract concepts and the label candidates as a new similarity measure to match the abstract concepts with the cluster label candidates. We experiment two alternatives to determine which documents should be assigned to which abstract concepts. One of them lets the documents be assigned only to one single abstract concept, while the other considers more than one abstract concept.

The experiment results demonstrate that our proposals for the Lingo algorithm lead quite better clusters, compared to the current algorithm. Despite the slight decrease in the precision values, our proposals make the recall and f-measure values increase dramatically.

As a future work, the merging process of the Lingo algorithm can be modified so that the recall values can be increased even more, without decreasing the precision and f-measure values.

Acknowledgment

Seyfullah Demir thanks to Stanislaw Osiński and Dawid Weiss who were always helpful and kind whenever he has questions or need advices.

References

[1] C. Carpineto, S. Osiński, G. Romano, and D. Weiss. "A survey of Web clustering engines.," *ACM Computing Surveys (CSUR)*, vol. 41, no. 3, pp. 1-38, July 2009

[2] O. Zamir and O. Etzioni, "Grouper: a dynamic clustering interface to Web search results," *Computer Networks: The International Journal of Computer and Telecommunications Networking*, vol. 31, no. 11-16, pp. 1361-1374, May 1999

[3] S. Osiński, "An Algorithm for Clustering of Web Search Results," M.S. thesis, Poznan University of Technology, Poland, 2003.

[4] S. Osiński, J. Stefanowski, and D. Weiss, "Lingo: Search results clustering algorithm based on singular value decomposition.," *Intelligent Information Processing and Web Mining*, vol. 25, pp. 359-368, 2004

[5] S. Osiński and D. Weiss, "Conceptual clustering using lingo algorithm: evaluation on open directory project data," *Intelligent Information Processing and Web Mining*, vol. 25, pp. 369-377, 2004.

[6] S. Osiński, D. Weiss, "A concept-driven algorithm for clustering search results," *IEEE Intelligent Systems*, vol. 20, no. 3, pp. 48-54, May-June 2005.

[7] S. Osiński, "Improving quality of search results clustering with approximate matrix factorisations," *Advances in Information Retrieval*, vol. 3936, pp. 167-178, 2006.

[8] N. C. Lang, "A tolerance rough set approach to clustering web search results," M.S. thesis, Faculty of Mathematics, Informatics and Mechanics, Warsaw University, 2004.

[9] A. Sameh and A. Kadray, "Semantic Web Search Results Clustering Using Lingo and WordNet," *International Journal of Research and Reviews in Computer Science (IJRRCS)*, vol. 1, no. 2, pp. 7-76, 2010.

[10] Princeton University "About WordNet" WordNet. Princeton University. (2010). [Online]. Available: http://wordnet.princeton.edu

[11] G. Salton, "Automatic text processing: the transformation, analysis, and retrieval of information by computer," Addison-Wesley Longman Publishing Co., Inc., Boston, MA, 1989

[12] S. Osiński and D. Weiss, "Carrot2: Design of a Flexible and Efficient Web Information Retrieval Framework," *Advances in Web Intelligence*, vol. 3528, pp. 439-444, 2005.

[13] Java API for WordNet Searching (JAWS). [Online]. Available: http://lyle.smu.edu/~tspell/jaws/

[14] G. Romano and C. Carpineto. (2008). Ambient dataset. [Online]. Available: http://credo.fub.it/ambient/

Seyfullah Demir was born in Konya, Turkey. He received the bachelor degree in the Department of Computer Engineering in Hacettepe University, Turkey. Now, he is studying for master of Science in the same department. He has been work for The Scientific and Technological Research Council of Turkey since 2010. His research interests include information retrieval and software engineering.

Ebru A. Sezer received the bachelor degree in the Department of Computer Engineering in Hacettepe University, Turkey in 1996. She gets the M.Sc. and Ph.D. degrees at the same department in 2000 and 2006. Her active research topics include information retrieval, data and web mining, fuzzy-logic systems and geographical information systems.

Hayri Sever is the head of the Department of Computer Engineering in Hacettepe University, Turkey. He is involved in several domestic and international research projects as a leader or consultant. He also has several international publications. His active research topics include information retrieval (and filter) models, vertical search engines, data and web mining, geographical information systems, and business process managements systems. His area of expertise includes database and information retrieval systems, software engineering, artificial intelligence, and internet computing.