The Heterogeneous Cluster Ensemble Method using Hubness for Clustering Text Documents

Jun Hou & Richi Nayak

School of Electrical Engineering and Computer Science,
Queensland University of Technology, Brisbane, Australia

jun.hou@student.qut.edu.au,r.nayak@qut.edu.au

Abstract. We propose a cluster ensemble method to map the corpus documents into the semantic space embedded in Wikipedia and group them using multiple types of feature space. A heterogeneous cluster ensemble is constructed with multiple types of relations i.e. document-term, document-concept and document-category. A final clustering solution is obtained by exploiting associations between document pairs and *hubness* of the documents. Empirical analysis with various real data sets reveals that the proposed method outperforms state-of-the-art text clustering approaches.

Keywords: Text Clustering, Document Representation, Cluster Ensemble

1 Introduction

Grouping of text documents into clusters is an elementary step in many applications such as Indexing, Retrieval and Mining of data on the Web. In traditional Vector space model (VSM), a document is represented as a feature vector which consists of weighted terms. A disadvantage of VSM representation is that it is not able to capture the semantic relations among terms [2]. Researchers have introduced two approaches of enriching document representation: (1) topic modelling, such as pLSI [13] and LDA [14]; and (2) embedding external knowledge into data representation model [1][2][15]. The semantic relations between the terms discovered by these methods are limited in either original document space or concepts represented by Wikipedia articles. They fail to capture other useful semantic knowledge, e.g. Wikipedia category that contains much meaningful information in the form of a hierarchical ontology [11]. In addition, these methods failed to model and cluster documents represented with multiple feature space (or relations).

In this paper, we propose a novel unsupervised cluster ensemble learning method, entitled Cluster Ensemble based Sequential Clustering using Hubness (*CESC-H*), for enriching document representation with multiple feature space and utilising them in document clustering. We propose to enhance data model by using semantic information derived from external knowledge i.e. Wikipedia concepts and categories. We construct a heterogeneous cluster ensemble using different types of feature spaces selected to maximize the diversity of the cluster ensemble. In order to build an accurate cluster ensemble learner, (1) we learn consistent clusters that hold same documents, and (2) utilize the phenomenon of high dimensional data, hubs, to represent cluster center and sequentially join inconsistent documents into consistent clusters to deliver a final clustering solution.

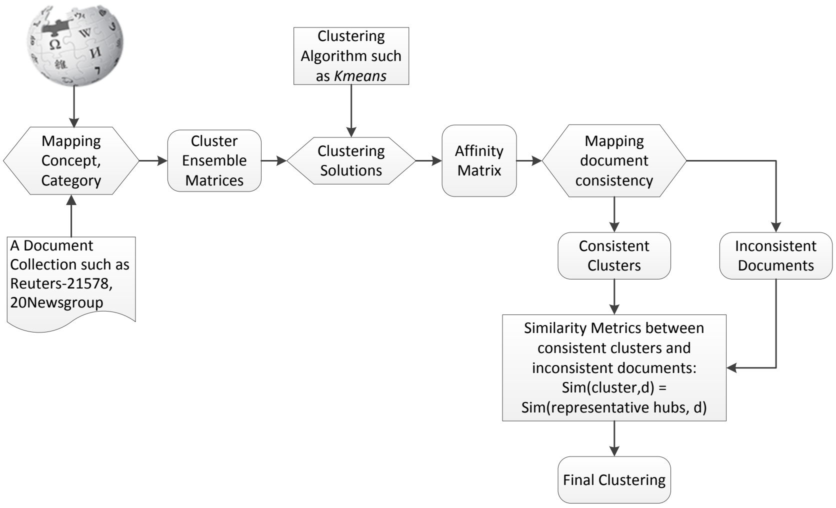
2 Related Work

Our work is related to document representation enrichment techniques that incorporate semantic information from external knowledge, i.e., Wikipedia into Vector space model (VSM) [1][2][15]. [1] maps Wikipedia *concepts* to documents based on the content overlap between each document and Wikipedia articles. [2][15] represented documents as Wikipedia concepts by mapping candidate phrases of each document to anchor text in Wikipedia articles. However, in these works, only Wikipedia articles are considered and our proposed cluster ensemble framework incorporates Wikipedia category and concepts both into document representation, thereby introducing more semantic features into the clustering process.

Another related work is cluster ensemble learning. Cluster ensemble learning is a process of determining robust and accurate clustering solution from an ensemble of weak clustering results. Researchers have approached this problem by finding the most optimizing partition, for instance, hypergraph cutting-based optimization [5] and probabilistic model with finite mixture of multinomial distributions [6]. Finding a consensus clustering solution has also been approached by learning ensemble information from the co-association matrix of documents such as fixed cutting threshold [7], agglomerative clustering [8] and weighted co-association matrix [10]. Our proposed cluster ensemble learning not only models documents with multiple feature spaces but also provides accurate clustering result by differentiating consistent clusters and inconsistent documents and utilizing *hubness* of document for grouping.

3 The Proposed Cluster Ensemble based Sequential Clustering using Hubness (CESC-H) Method

Figure 1 illustrates the process of clustering text documents. Firstly, each document is mapped to Wikipedia articles (concepts) and categories, and cluster ensemble matrices are formed. A heterogeneous cluster ensemble construct is obtained by applying the (same) clustering algorithm on each matrix separately. The Affinity Matrix is proposed for identifying consistent clusters and inconsistent documents based on documents consistency in the cluster ensemble. Using the concept of representative hubs, the final clustering solution is delivered by placing all inconsistent documents to the most similar consistent cluster.



Clustering Algorithm such as *Kmeans*
Mapping Concept, Category
Cluster Ensemble Matrices
Clustering Solutions
Affinity Matrix
Mapping document consistency
A Document Collection such as Reuters-21578, 20Newsgroup
Consistent Clusters
Inconsistent Documents
Similarity Metrics between consistent clusters and inconsistent documents: Sim(cluster,d) = Sim(representative hubs, d)
Final Clustering

Fig. 1. – The Cluster Ensemble based Sequential Clustering using Hubness Method

3.1 Cluster Ensemble Matrices

In this section, we discuss how to construct a heterogeneous cluster ensemble with different cluster ensemble matrices.

Document-Term (D-T) Matrix. A document d_j is represented as a term vector $T_{d_j} = \{t_1, t_2, \dots t_{|T|}\}$ with term set T. Each value of the vector represents the equivalent weight (TF*IDF) value of the term,

$$W_{(d_j,t_i)} = W_{TF*IDF} = tf(d_j, t_i) \times log \frac{N}{df(t_i)} \quad (1)$$

where $tf(d_j, t_i)$ is term frequency and $df(t_i)$ denotes inverse document frequency for N documents.

Document-Concept (D-C) Matrix. In the D-C matrix, a document d_j is represented by a concept vector $C_{d_j} = \{c_1, c_2, \dots c_{|C|}\}$ where C is the total number of concepts, and each value of C_{d_j} is the concept salience $SAL(d_j, c_i)$, calculated as in equation 3. If the anchor text of a Wikipedia article appears in a document, the document is mapped to the Wikipedia article. However, an anchor text, for example “tree”, can appear in many Wikipedia articles, “tree (the plant)” or “tree (data structure)”. Therefore, we find the most related Wikipedia article for an ambiguous anchor text by calculating the sum of the relatedness score between unambiguous Wikipedia articles and candidate Wikipedia articles. The relatedness of each pair of Wikipedia concepts (c_i, c_j), is measured by computing the overlap of sets of hyperlinks in them [9]:

$$Rel(c_i, c_j) = 1 - \frac{max(log|c_i|, log|c_j|) - log\ |c_i \cap c_j|}{W - min(log|c_i|, log|c_j|)} \quad (2)$$

where W is the total number of Wikipedia articles. In order to punish irrelevant concepts and highlight document topic related concepts, the concept salience [18] is applied as the weight of a concept c_i integrating local syntactic weight and semantic relatedness with other concepts c_l in document d_j:

$$SAL(d_j, c_i) = W_{(t_i, d_j)} * Rel(t_i, c_i|d_j) \quad (3)$$

where $W_{(t_i, d_j)}$ is the syntactic weight in equation (1) of the corresponding term t_i. $Rel(t_i, c_i|d_j)$ is the sum of relatedness of concept c_i with other concepts that the rest of terms in document $d_j$map to:

$$Rel(t_i, c_i|d_j) = \sum_{t_l \epsilon\ d_j\ \&\ t_l \neq t_i\ \&\ t_l \to c_l} Rel(c_i, c_l) \quad (4)$$

where $Rel(c_i, c_l)$ is obtained using equation (2). If a concept is mapped to a *n-gram* (n>=2) phrase, the syntactic weight in (3) is the sum of the weight of each uni-gram term.

Document-Category(D-Ca) Matrix. A document d_j is represented by a category vector $E_{d_j} = \left\{E_{c_1}, E_{c_2} \dots E_{c_{|C|}}\right\}$ where $E_{c_i}$is a vector $\{e_1, e_2 \dots e_k\}$ that contains parent categories assigned for the concept c_i. The weight of a category is the weight of the corresponding concept. If a category is assigned to more than one concept, the sum of the weight of these concepts is the category's weight.

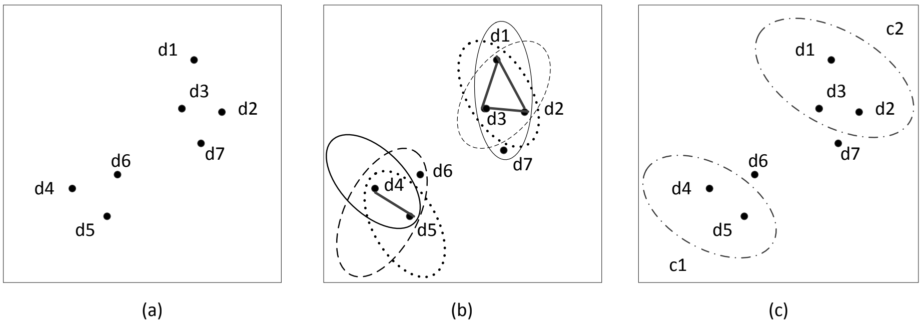


Fig. 2. – The Process of Identifying Consistent Clusters and Inconsistent Documents. (a) A document collection (b) Three Component clustering results (ovals with solid line, dotted line and dashed line) and documents which have larger value than threshold θ (connected with bold line) (c) Consistent clusters (c1 and c2) and inconsistent documents ($d_6$and d_7).

3.2 Affinity Matrix

In this section, we discuss how to construct an ensemble learner based on Affinity Matrix to identify consistent clusters. Figure 2 illustrates the process in simpler manner. Affinity Matrix (AM) is constructed by identifying document pairs which co-locate in the same partition of all component solutions (steps 1-3 in Figure 3). Let Cluster ensemble $\Pi = \{\Pi_1 \Pi_2, \dots, \Pi_H\}$ contains all component clustering solution. A component clustering solution $\Pi_i = \{\pi_1, \pi_2, \dots, \pi_K\}$ contains partitions for the specific feature space. For a document d_k, function $f_i(\cdot)$ searches through each clustering partition space π_m in each Π_i to identify which partition d_k belonging to:

$$f_i(d_k) = \begin{cases} m, & if\ d_k \in \pi_m \\ 0, & otherwise \end{cases}, 0 < m < K + 1 \qquad (5)$$

where K is the number of partitions and i is the identifier of the component clustering result. The consensus function $con(\cdot)$ then accumulates the total co-occurrence of a document pair (d_x, d_y) in all component clustering solutions using $f_i(\cdot)$ in (5):

$$con\left(d_x, d_y\right) = \sum_{i=1}^{H} \delta\left(f_i(d_x), f_i\left(d_y\right)\right), x \neq y, \delta(a, b) = \begin{cases} 1, & if\ a = b \\ 0, & if\ a \neq b \end{cases} \quad (6)$$

Affinity Matrix ($N \times N$ symmetric matrix where N is the number of documents) contains consistency degree ($con\left(d_x, d_y\right)$) for each pair of documents $\left(d_x, d_y\right)$ in the corpus. The proposed method differentiates consistent clusters and inconsistent documents by setting a threshold (θ) on the values in AM and, consequently, is able to obtain reliable consistent clusters with setting a high threshold (steps 4-15 in Figure 3). Documents whose consistency degree is above θ are combined to form consistent clusters (c1 and c2 in Figure 2). Documents with lower consistency degree are dropped to a waiting list Z as inconsistent documents (d_6 and d_7 in Figure 2). The next section shows the process of joining inconsistent documents to consistent clusters.

Input: Cluster ensemble $\Pi = \{\Pi_1\Pi_2, \dots, \Pi_H\}$ for document collection D with N documents on H types of feature space; consistency degree threshold θ; number of nearest-neighbour k and representative hub threshold η.
Output: Final document partition $C = \{C_1, C_2, \cdots\ C_K\}$ and K is the number of clusters.
Initialization: Set the affinity matrix AM as a null $N \times N$ matrix, and set C and Z (inconsistent document set) as empty.

```
 1:  for $i = 1$ to $H$ do
 2:    $AM(d_x, d_y) \leftarrow \Pi_i$ using equation (5) and (6)
 3:  end for
 4:  for $x = 1{:}\,|D|$ do
 5:    for $y = 1{:}\,|D|$ do
 6:        if $AM(d_x, d_y) > \theta$ and $d_y \neq d_x$
 7:            if $d_x \in C_i$ && $d_y \in C_i$ && $C_i \in C$
 8:                $C_i \leftarrow (d_x, d_y)$
 9:            else
10:                $C_{i+1} \leftarrow (d_x, d_y)$; $C \leftarrow C_{i+1}$
11:        else
12:            $Z \leftarrow d_x$
13:        end if
14:    end for
15:  end for
16:  for $z = 1\ to\ |Z|$ do
17:    $(C_m \in C) \leftarrow d_z$ using equation (8)
18:    $Re(d_z)$ and $Up(H_m)$ using equation (7)
19:  end for
20:  Return final partition $C$
```

Fig. 3. – The Unsupervised Cluster Ensemble Learning Algorithm.

3.3 Hubness of Document

The traditional centroid based partitional methods usually fail to distinguish clusters due to the curse of high dimensionality [4]. In this paper, in order to join inconsistent documents to the consistent clusters, we propose to use hubs as representation of the cluster center instead of centroids (step 16-19 in Figure 3). Let d_x be a document in a consistent cluster and d_y be a documents in the document collection D. Let $D_k(d_x, d_y)$ denote the set of documents, where document d_x is among the k–nearest-neighbour list of document d_y and $d_y \neq d_x$. The hubness score of d_x, $N_k(d_x)$, is defined as:

$$N_k(d_x) = \left|D_k\left(d_x, d_y\right)\right| \tag{7}$$

The hubness score of d_x depends on the $distance(d_x, d_y)$ and the $k-$ nearest-neighbour at data point d_y. We make use of top-η (top η proportion of) documents ranked by hubness score as hubs. Let H_m be the set of representative hubs for cluster $C_m(C_m \in C)$. For an inconsistent document d_z, we find the most similar consistent cluster C_m whose representative hub set H_m is most close to $d_z$where K is the number of clusters:

$$(C_m \in C) = \underset{K}{argmin}(\|H_K - d_z\|) \tag{8}$$

4 Experiments and Evaluation

4.1 Experimental Setup

Data Set	Description	#Classes	#Documents	#Terms	#Wikipedia Concepts	#Wikipedia Categories
D1	Multi5	5	500	2000	1667	4528
D2	Multi10	10	500	2000	1658	4519
D3	R-Min20Max200	25	1413	2904	2450	5676
D4	R-Top10	10	8023	5146	4109	9045

Table 1. Data set summary

As shown in table 1, two subsets from the 20Newsgroups data set are extracted in the same line as [2]: Multi5 and Multi10. Other two subsets were created from the Reuters-21578 data set as [2]: R-Min20-Max200 and R-Top10. For each data set, Wikipedia concepts and categories are mapped to the terms of documents via methods discussed in the section 3.1.

The proposed approach (*CESC-H*) is benchmarked with following approaches: ***Single Feature Space.*** This is a vector-space-model-based Bisecting K-means approach [3] based on each feature space: term (D-T), concept (D-C), caetegory (D-Ca) which are represented as a, b and c in Tables 2.

Linear Combination of Feature Space. This approach clusters do-cuments based on linearly combined syntactic and semantic feature space: term and concept (D-(T+C)), term and category (D-(T+Ca)) and term, concept and category (D-(T+C+Ca)) which are d, e and f, respectively in Tables 2.

HOCO [3]. This is a High-Order Co-Clustering method using the consistency information theory to simultaneously cluster documents, terms and concepts.

Cluster Ensemble Based Methods. The *CSPA, HGPA* and *MCLA* are hypergraph-based methods [5] whereas *EAC* uses evidence accumulation [8].

CESC. Variation of the proposed method CESC-H but computing similarity between consistent clusters and inconsistent documents using the cluster centroid instead of hubs.

4.2 Experimental Results

Table 2 presents the clustering performance in *FScore* and *NMI* on each data set and method respectively. *CSPA* was not able to work on data sets D3 and D4 due to computation complexity.

Data Set	*D1*		*D2*		*D3*		*D4*		*Ave*	
	F	*NMI*	*F*	*NMI*	*F*	*NMI*	*F*	*NMI*	*F*	*NMI*
a	0.771	0.664	0.580	0.484	0.624	0.692	0.593	0.548	0.642	0.597
b	0.752	0.659	0.61	0.54	0.622	0.698	0.576	0.54	0.64	0.609
c	0.734	0.636	0.557	0.481	0.62	0.69	0.567	0.532	0.633	0.584
d	0.771	0.662	0.601	0.506	0.609	0.695	0.564	0.543	0.636	0.601
e	0.766	0.648	0.597	0.501	0.615	0.702	0.581	0.544	0.639	0.598
f	0.774	0.665	0.613	0.548	0.632	0.708	0.595	0.555	0.653	0.619
HOCO	0.972	0.917	0.705	0.603	0.692	0.779	0.601	0.569	0.742	0.717
CSPA	0.950	0.855	0.321	0.313	-	-	-	-	0.635	0.584
HGPA	0.797	0.601	0.466	0.61	0.648	0.702	0.509	0.179	0.605	0.523
MCLA	0.822	0.692	0.286	0.583	0.691	0.747	0.73	0.467	0.632	0.622
EAC	0.323	0.319	0.714	0.625	0.722	0.746	0.806	0.546	0.641	0.559
CESC	**0.982**	**0.922**	0.791	0.68	0.758	0.782	0.814	0.587	0.836	0.743
CESC-H	**0.982**	0.921	**0.799**	**0.691**	**0.771**	**0.797**	**0.822**	**0.591**	**0.844**	**0.75**

Table 2. – *Fscore (F)* and *NMI* for Each Data Set and Method.

We can see that the proposed methods (*CESC-H* and *CESC)* and *HOCO* get significantly better performance than clustering with linear combination of feature space. More importantly*, CESC-H* (and *CESC*) outperforms *HOCO* as it uses heterogeneous cluster ensemble with additional external knowledge (i.e. Wikipedia category). The *CESC* and *CESC-H* approach consistently outperforms other cluster ensemble methods, *CSPA*, *HGPA, MCLA* and *EAC*. Different from our proposed method, these ensemble methods do not differentiate consistent clusters and inconsistent documents. Moreover, *CESC-H* performs better than *CESC* on each data set, as hubness scores of clusters can better represent the cluster than the cluster centriod, thereby improving the accuracy of joining inconsistent documents to consistent clusters.

4.3 Sensitivity Analysis

As shown in Figure 4, when θ increases, performance of *CESC-H* is improved. The larger θ value compels a document pair to be grouped in the same partition by more cluster ensemble components. Similarly with the higher value of η (η = 0.85), *CESC-H* achieves the best result. When the neighbourhood size is large enough (k = 15), representative hubs are stable and accurate clustering solution is obtained.

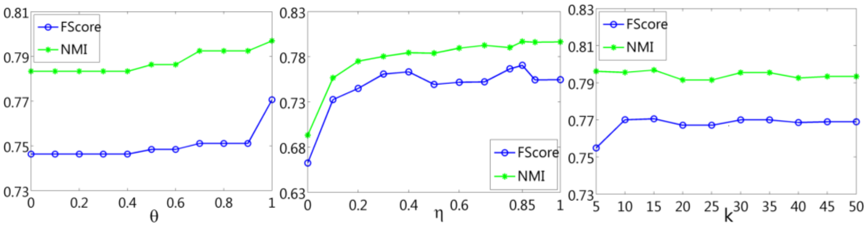


Fig. 4. – FScore/NMI as function of different trade-off parameters: consistency degree threshold θ; representative hub threshold η; and number of nearest-neighbour k

5 Conclusions and Future Work

The proposed novel Cluster Ensemble based Sequential Clustering using Hubness (*CESC-H*) method, integrating unsupervised cluster ensemble learning and hubness of documents, has the capability of clustering documents represented with multiple feature space (or relations). *CESC-H* is able to introduce and model more external knowledge for document representation and maximize the diversity of cluster ensemble. With the support of *hubness* of documents, *CESC-H* learns accurate final clustering solution by joining inconsistent documents into consistent clusters. Experiments on four data sets demonstrate that the proposed approach outperforms many start-of-the-art clustering methods. In future, we will investigate other cluster ensemble learning approaches with more features.

References

1. Hu, X., Zhang, X., Lu, C., Park, E., Zhou, X.: Exploiting wikipedia as external knowledge for document clustering. In: Proc. of the 15th ACM SIGKDD, pp. 389–396 (2009)
2. Jing, L. P., Jiali Yun, Jian Yu and Joshua Huang: High-Order Co-clustering Text Data on Semantics-Based Representation Mode. In Proc. of the PAKDD, pp. 171-180 (2011)
3. Steinbach, S., Karypis, G., Kumar, V.: A comparison of document clustering techniques. In: Proc. of the Workshop on Text Mining at ACM SIGKDD (2000)
4. R. Agrawal, J. Gehrke, D. Gunopulos, and P. Raghavan: Automatic subspace clustering of high dimensional data for data mining applications. In Proc. of the 1998 ACM SIGMOD, pp. 94-105 (1998)
5. Strehl A, Ghosh J.: “Cluster Ensembles – A Knowledge Reuse Framework for Combining Multiple Partitions.” Journal of Machine Learning Research, 3, pp. 583–617 (2003)
6. A. Topchy, A. Jain, W. Punch: “A mixture model for clustering ensembles”, In Proceedings of the SIAM International Conference on Data Mining, pp. 331–338, (2004)
7. Fred, A.: Finding consistent clusters in data partitions. In: Kittler, J., Roli, F. (eds.) MCS 2001. LNCS, vol. 2096, pp. 309–318. Springer, Heidelberg (2001)
8. Fred, A. N., A. K. Jain: “Combining multiple clusterings using evidence accumulation”, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 27 (6), pp. 835-850 (2005)
9. Medelyan, O., Witten, I., Milne, D.: Topic indexing with wikipedia. In: Proc. Of AAAI (2008)
10. Sandro Vega-Pons, José Ruiz-Shulcloper, Alejandro Guerra-Gandón: Weighted association based methods for the combination of heterogeneous partitions. Pattern Recognition Letters 32:16, 2163-2170 (2011)
11. Köhncke, B., Balke, W.-T.: Using Wikipedia Categories for Compact Representations of Chemical Documents. In: Proc. of the ACM CIKM (2010)
12. Tomasev N, Radovanovic M, Mladenic D, Ivanovic M: The role of hubness in clustering high-dimen-sional data. In: Proceedings of PAKDD, pp 183–195 (2011)
13. Deerwester, S. C., S.T. Dumais,T.K. Landauer, G.W.: Furnas,and R.A.harshman. Indexing bylatent semantic analysis. Journal of the American Society of Information Science, 41(6):391–407 (1990)
14. D. Blei, A. Ng, and M. Jordan: Latent Dirichlet allocation. Journal of Machine Learning Research, 3:993–1022 (2003)
15. A. Huang, D. Milne, E. Frank, I.H.: Witten, Clustering documents using a Wikipedia-based concept representation, in: 13th PAKDD, Bangkok, Thailand, pp. 628–636 (2009)