Concept-enhanced Multi-view Co-clustering of Document Data

Valentina Rho* and Ruggero G. Pensa(orcid.org/0000-0001-5145-3438)

Dep. of Computer Science, University of Torino, Italy
{valentina.rho,ruggero.pensa}@unito.it

Abstract. The maturity of structured knowledge bases and semantic resources has contributed to the enhancement of document clustering algorithms, that may take advantage of conceptual representations as an alternative for classic bag-of-words models. However, operating in the semantic space is not always the best choice in those domain where the choice of terms also matters. Moreover, users are usually required to provide a valid number of clusters as input, but this parameter is often hard to guess, due to the exploratory nature of the clustering process. To address these limitations, we propose a multi-view co-clustering approach that processes simultaneously the classic document-term matrix and an enhanced document-concept representation of the same collection of documents. Our algorithm has multiple key-features: it finds an arbitrary number of clusters and provides clusters of terms and concepts as easy-to-interpret summaries. We show the effectiveness of our approach in an extensive experimental study involving several corpora with different levels of complexity.

Keywords: Co-clustering · Semantic enrichment · Multi-view clustering

1 Introduction

Clustering is a widely used tool in text document analysis. Due to its unsupervised nature, it takes part in a wide range of information retrieval applications, including summarization [20], query expansion [12] and recommendation [22], but it is also employed as a first exploratory tool in analyzing new text corpora. The principle of clustering is simple: it aims at grouping together similar documents into groups, called clusters, while keeping dissimilar documents in different clusters. The way similar documents are grouped together strongly depends on the clustering algorithm [1], but the notion of similarity itself is not straightforward. Documents can be viewed as bags of words, thus classic similarity functions (usually, the cosine similarity) can be applied on word vectors; however, they are not sufficient to capture the semantic relationship between two documents, since they do not deal with problems like synonymy (different terms with the same meaning) and polysemy (same term with multiple meanings). Moreover, the document-term matrix (the matrix describing the frequency of terms that occur in a collection of documents) used as input for the clustering algorithm is

usually very sparse and high-dimensional, leading to the well-studied problem of the curse of dimensionality, which in turn results in meaningless cluster structures. To mitigate these problems, semantic approaches can be applied to the document-terms matrix prior to clustering. For instance, Latent Semantic Analysis (LSA) [13] is a dimensionality reduction technique, based on singular-value decomposition (SVD), that provides a set of latent factors related to documents and terms, assuming that words with similar meanings occur in similar portions of text. Then clustering can be executed on a reduced document-factor matrix, rather than on the whole document-term matrix. Although these approaches provide effective solutions to some of the aforementioned problems, they suffer from some limitations weakening their exploitation in many clustering applications. First, polysemy is not handled. Second, the latent factors have no interpretable meaning in natural language, therefore they cannot be used to directly describe clustering results. Yet, cluster interpretation is fundamental in many exploratory applications. Third, the number of latent dimensions is a required parameter of the SVD algorithm performing LSA, and a wrong choice of this parameter may lead to poor clustering results.

With the evolution of structured knowledge bases (e.g., Wikipedia) and semantic resources (e.g., WordNet and BabelNet), in the last decade, new alternative approaches to the semantic enhancement of document clustering algorithms have been proposed. A first class of methods uses semantic resources to create new feature spaces [5, 21]. A second group of algorithms leverages the semantic representation to reduce data dimensionality [19]. Finally, other methods define new similarity measures that take into account the semantic relations between concepts [9, 8, 21]. However, operate solely in the semantic space is not always the best choice for document clustering: even though the same concept can be expressed by different terms, sometimes each term is specific to a particular domain or language register. For instance, the terms *latent class analysis* and *clustering* sometimes refer to the same concept, but the former is used prevalently by statisticians, while the latter is preferred by machine learning experts. In these cases the chosen term is as important as its meaning.

To address all these limitations, we propose a multi-view clustering approach that processes simultaneously two representations of the same collection of documents: a classic document-term matrix and an enhanced document-concept representation. In our work, concepts are *abstract representations* of terms and are extracted from the document collection by means of a conceptualization approach that combines entity linking and word sense disambiguation, two natural language processing (NLP) techniques aiming at recognizing all concepts mentioned in a text. The two views are processed with a multi-view co-clustering approach that has multiple key-features: *(i)* it takes into account the peculiarity of the statistical distribution in each view, by implementing an iterative star-structure optimization approach; *(ii)* it provides an arbitrary number of clusters thanks to the adoption of an association function whose optimization does not depends on the number of clusters; *(iii)* it provides clusters of terms and concepts that can be used as easy-to-interpret summaries of the document clusters

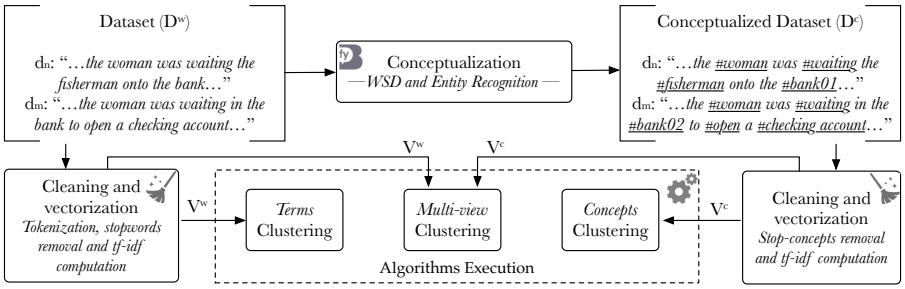

Fig. 1. A graphic overview of the overall *CVCC* clustering approach.

in both representation spaces. Our approach, then, also transforms texts into their direct conceptual representation, but, differently from the aforementioned methods, we embed this new representation into a 2-view setting in which both terms and concepts contribute to the clusters generation process. We show the effectiveness of our approach in an experimental study involving several corpora with different levels of complexity.

The remainder of the paper is organized as follows: we present the theoretical details of our approach in Section 2; in Section 3 we report some experimental results and discuss them; finally, we end up with some concluding remarks and ideas for future work in Section 4.

2 Combining Words and Concepts

We present a clustering approach that combines the expressive power of both terms and concepts to provide meaningful clusters of documents and an associated collection of clusters of features. First, we introduce a sketch of the overall clustering approach; then, we describe a possible way to extract a collection of concepts from a given text corpus. Finally, we provide more details on the multi-view co-clustering algorithm that we use to address our clustering problem.

2.1 Overall Clustering Approach

This section aims at describing the overall clustering approach, shown in Figure 1, called *CVCC* (Concept-enhanced multi-View Co-Clustering). For a given collection of documents, represented by both a term and a conceptual view, *CVCC* partitions documents into an arbitrary number of clusters by also providing two related partitions of terms and concepts. Before entering the details of the approach, we first introduce some useful notation.

The input of *CVCC* is a dataset D, defined as a set of raw textual documents $\{d_1, ..., d_n\}$, each one represented by the sequence of words $< w_1, w_2, ... >$ that occur in it. The first step is to apply a *conceptualization* process on D, to

obtain the *conceptualized* dataset D^c. Each document $d^c \in D^c$ is the conceptual representation of the corresponding document d in D and is defined as the sequence of concept identifiers $< c_1, c_2, ... >$ that occur in d. Then, we define three cleaning sets for each dataset D in order to ignore information that are considered not relevant to our purposes: $S = \{w_1^s, ..., w_p^s\}$ where each w_i^s is a word that is considered very common in the considered language (stopwords); $F = \{w_1^f, ..., w_q^f\}$, where each w_i^f is a word that occurs in more than t_f documents in D; $U = \{w_1^u, ..., w_r^u\}$, where each w_i^u is a word that occurs in less than t_u documents in D; t_f and t_u are two threshold values given in input to the pipeline. The last two sets are also computed on D^c, obtaining respectively F^c (too frequent concepts) and U^c (too rare concepts).

A preprocessing step applied to both D and D^c allow us to generate respectively a bag-of-words dataset V^w and a bag-of-concepts dataset V^c that will be the input of our clustering algorithm. In the former case, V^w, is represented as a $|D| \times |W|$ matrix, where W is defined as $\{w_j \mid \exists d_i \in D \wedge w_j \in d_i \wedge w_j \notin \{S \cup F \cup U\}\}$; each element v_{ij}^w of V^w is a numerical value representing the relevance of the term w_j in the document d_i. This numerical value could be computed with the well-known *tf-idf* (term frequency-inverse document frequency) function. In a similar way, we can define the preprocessed bag-of-concepts dataset V^c as a $|D| \times |C|$ matrix. In this case C is defined as $\{c_j \mid \exists d_i^c \in D^c \wedge c_j \in d_i^c \wedge w_j^c \notin S \wedge c \notin \{F^c \cup U^c\}\}$, where w_j^c is the term associated to the concept c in the corresponding document in D. Notice that the two sets are created independently. In fact, too rare (resp. too frequent) words may refer to more frequent (resp. unfrequent) concepts and vice versa, due to synonymy and polysemy. V^w and V^c are the two representations that feed the clustering algorithm used to compute the partitions on D, W and C.

2.2 Conceptualization Process

Many interpretations of what a *concept* is have been proposed during years, the most generic defining it as a high level representation of a set of items that share common characteristics. Here we embrace the commonly accepted definition of concept as an *abstract representation* of something in one's mind.

In document analysis, there are several advantages in using the conceptual representation with respect to the standard bag of words one. For example, concepts allow: to distinguish different meanings of the same word, by taking advantage of the context (polysemy, e.g. *bank* as *financial institution* or as a *land alongside water*); to aggregate different words with the same meaning (synonymy, e.g. *film* and *movie*); to identify named entities (e.g. *pink* as a color or *Pink* as the singer); to automatically consider n-grams instead of single terms (e.g. *United States*). In addition, another key point to consider when dealing with concepts is that, as they are *abstract*, they are language-insensitive: the same abstract concept labeled *#dog01* represents words *dog*, *cane*, *chien*, *hund* and so on, allowing us to work with multi-language text corpora.

In order to transform a generic document represented as a sequence of terms, into its conceptual representation we have to face the nontrivial issues of *entity*

linking (assigning each word to the correct concept) and word-sense disambiguation (deciding which is the correct sense of each word, depending on its context). To address these issues, we make use of *Babelfy* [16], a multi-lingual semantic resource that aims at performing both entity linking and word-sense disambiguation on generic sentences. Babelfy is grounded on BabelNet, a multilingual encyclopedic resource created by the automatic integration of other well-known resources, e.g. WordNet and Wikipedia [17]. In practice, in our approach, concepts are intended as BabelNet identifiers. The conceptualization process transforms a sentence, represented as a sequence of words, into a list of concepts. We let the reader refer to [16] for more details about Babelfy.

2.3 Clustering Algorithm

We define our clustering approach as a 2-view co-clustering problem on the two matrices V^w and V^c. The goal of the 2-view co-clustering approach is to compute a set of n document clusters $X = \{x_1, \ldots, x_n\}$ on D, a set of l word clusters $Y^w = \{y_1^w, \ldots, y_l^w\}$ on W and a set of m concept clusters $Y^c = \{y_1^c, \ldots, y_m^c\}$ on C. X is such that $\bigcap_{k=1}^{n} x_k = \varnothing$ and $\bigcup_{k=1}^{n} x_k = D$. Y^w and Y^c are subject to similar constraints. Differently from most document clustering problems, n, m and l are not provided as input, i.e., our clustering approach is able to identify partitions with an arbitrary non predefined number of clusters. To achieve this goal, similarly to [11], we adopt an optimization function that is independent on the number of clusters: the Goodman and Kruskal's $\tau_{X_1|X_2}$ association measure [6]. It estimates the association between two categorical variables X_1 and X_2 by the proportional reduction of the error in predicting X_1 knowing or not the variable X_2. This measure requires that partitions X, Y^w and Y^c are defined as discrete random variables. Variable Y^w has l categories $y_1^w, \ldots, y_l^w$, corresponding to the l word clusters, with probabilities $q_1^w, \ldots, q_l^w$. Variable Y^c is defined similarly, while variable X has n categories $x_1, \ldots, x_n$ corresponding to n document clusters. However, for each view, the n categories of X have different probabilities $p_1^w, \cdots, p_n^w$, and $p_1^c, \cdots, p_n^c$. Moreover, the joint probabilities between X and Y^w (resp. Y^c) are denoted by r_{st}^w (resp r_{st}^c). All probabilities are computed directly from matrices V^w and V^c. As an example, the joint probabilities r_{st}^w between X and Y^w are computed as follows:

$$r_{st}^w = \frac{\sum_{d_i \in x_s} \sum_{w_j \in y_t^w} \overline{v}_{ij}^w}{\sum_i \sum_j \overline{v}_{ij}^w}$$

where $x_s \in X$, $y_t^w \in Y^w$, and $\overline{v}_{ij}^w$ is the value of v_{ij}^w normalized by sum of all elements in V^w.

The 2-view co-clustering problem can be defined as a multi-objective optimization problem defined over the following Goodman and Kruskal's τ coefficients, depending on which variable is considered as independent:

$$\tau_{X|Y^w,Y^c} = \frac{e_X - E[e_{X|Y^w,Y^c}]}{e_X},\ \tau_{Y^w|X} = \frac{e_{Y^w} - E[e_{Y^w|X}]}{e_{Y^w}},\ \tau_{Y^c|X} = \frac{e_{Y^c} - E[e_{Y^c|X}]}{e_{Y^c}} \quad (1)$$

where e_X (resp., e_{Y^w}, e_{Y^c}) is the sum of the errors over the independent variables Y^w and Y^c (resp. X). $E[e_{X|Y^w,Y^c}]$ (resp. $E[e_{Y^w|X}]$, $E[e_{Y^c|X}]$) is the expectation of the conditional error taken with respect to the distributions of Y^w and Y^c (resp. X). To optimize the objective functions we use the star-structure multi-objective optimization approach proposed in [11] which iteratively optimizes the three partitions X, Y^w and Y^c based on Goodman-Kruskal's τ measure using Equations 1. The reader may refer to [11] for further algorithmic details.

3 Experiments

In this section we report the results of the experiments that we conducted to evaluate the performances of our document clustering approach. We first describe the datasets adopted and how we processed them. Then we introduce the algorithms involved in our comparative analysis and provide the details of the experimental protocol. Finally, we present the results and discuss them.

3.1 Datasets

The experiments are conducted on two well-known document corpora: Reuters-21578[1] and 20-Newsgroups[2]. For both datasets, categories are given that describe the content of each document. However, while 20-Newsgroups contains equally distributed disjoint categories, in Reuters-21578 corpus categories are not equally distributed and often cover very similar topics. Moreover, documents may belong to more than one category. For these reasons, we manually aggregated some of the original Reuters categories to create more homogeneous and semantically correlated groups (see Table 1 for the result of this process). Categories *earn* and *acq* are used as is. For both datasets, we prepared three reduced datasets, consisting of four categories each, as shown in Table 2. These three datasets are created to represent different complexity levels for the document clustering perspective: *level 1* (easy) datasets contain well-separated categories, *level 2* (medium) datasets contain two semantically similar categories and two different ones, *level 3* (hard) datasets are composed by two pairs of similar categories. Table 2 shows a detailed description of each considered dataset.

Table 1. Reuters-21578 aggregated categories. In **bold** the name of the resulting category, followed by the names of the Reuters categories that compose it.

economic-indices ipi, wpi, jobs, trade, gnp, bop, cpi, income	**money** yen, money-fx, interest, dlr	**energy** crude, gas, fuel, propane, ship, nat-gas, naphtha, pet-chem, heat
cereals oat, sorghum, oilseed, coconut-oil, sun-oil, rye, grain, sunseed, corn, wheat, palm-oil, barley, soybean, rice, cotton-oil, cotton, rapeseed, rape-oil, veg-oil, soy-oil		

[1] http://www.nltk.org/book/ch02.html#reuters-corpus
[2] http://scikit-learn.org/stable/datasets/twenty_newsgroups.html

Table 2. Datasets composition and statistics, in terms of no. of documents, features and density. T and C columns refers to term and concept matrices, respectively. For included categories the number of elements of each category is reported in parentheses; pairs of semantically similar categories within each dataset are highlighted in *italic*.

Dataset		Doc.	Features		Density		Included categories
			T	C	T	C	
20-newsg.[3]	L1	2025	9523	9767	0.45%	0.44%	hardware (590), autos (594), religion (377), politics (546)
	L2	2058	10128	9478	0.37%	0.39%	*windows* (591), religion (377), autos (594), *hardware* (590)
	L3	2293	11099	10521	0.39%	0.41%	*windows* (591), crypt (595), *hardware* (590), electronics (591)
reuters-21k	L1	1495	5925	6206	0.77%	0.76%	cereals (400), energy (400), money (400), earn (400)
	L2	1501	6523	6874	0.79%	0.77%	cereals (400), energy (400), *money* (400), *acq* (400)
	L3	1555	5453	6001	0.83%	0.78%	earn (400), economic-indices (400), *acq* (400), *money* (400)

3.2 Experimental Settings

To evaluate *CVCC*, we compared its performances with those of three well-known algorithms: Non-negative Matrix Factorization (*NMF*), *K-Means* and *EBC*. *NMF* [3, 14] is a dimensionality reduction algorithm that has been proved to be useful in different tasks, included document clustering. *K-Means* [15] is a popular clustering algorithm; in our setting we preprocess the data using Latent Semantic Analysis (LSA), in order to reduce their sparsity and improve clustering performances. The last competitor, *EBC* [18], is a very recent improvement of the well-known Information-Theoretic Co-clustering algorithm [4] and it is proven to perform well with large sparse data matrices [18].

The experiments were conducted as follows. First of all, for each dataset described in Section 3.1 we compute matrices V^w and V^c, as shown in Section 2.1. We run the selected algorithms in three different configurations: *(i)* using only the **terms** matrix, in order to assess the capabilities of *CVCC* with respect to the competitors in a standard setting; *(ii)* using only the **concepts** matrix, to evaluate the performances of all algorithms when moving from a lexical perspective to a more semantically enhanced interpretation of documents; *(iii)* using **both** terms and concepts (hereafter *both* configuration), to test *CVCC* multi-view approach dealing with two representations of the same documents; in this last case, for single-view algorithms, we consider the hybrid matrix $[V^w, V^c]$ as the concatenation of the two original terms and concepts matrices, while for *NMF* we execute a recent co-regularized version (*CoNMF*) [7] that extends *NMF*

[3] 20-Newsgroups categories have been renamed for the sake of readability, as follows: comp.sys.ibm.pc.hardware as *hardware*, comp.os.ms-windows.misc as *windows*, soc.religion.christian as *religion*, rec.autos as *autos*, talk.politics.guns as *politics*, sci.crypt as *crypt*, sci.electronics as *electronics*

for multi-view clustering. Additionally, when using *NMF*, we apply Non-negative Double Singular Value Decomposition (NNDSVD) [2] to preprocess the sparse input matrix. Since all competitors require the number of clusters to find as input parameter, we set this value to four, that is the "correct" number of embedded clusters in our datasets; we let *CVCC* algorithm adapt this value autonomously. The number of iterations of *CVCC* has been configured, for each dataset, to $20 \times (n\_documents + n\_features)$, rounded to the nearest thousand[4].

To measure the performance of each algorithm, we adopt the Adjusted Rand Index (ARI) [10]. It measures the agreement of two different partitions of the same set, but, differently from other common statistics like Purity or Rand Index, it is not sensitive to group imbalance and allows the comparison of partitions with different number of clusters. Here, we use it to compare the cluster assignments proposed by each algorithm with the original assignment provided by the given *true* categories. As all algorithms are nondeterministic we perform 30 executions for each considered configuration and compute the ARI mean and standard deviation. All algorithms are written in Python and executed on a server with 16 3.30GHz Xeon cores, 128GB RAM, running Linux.

3.3 Results and Discussion

The results of the experiments are shown in Table 3 and two different aspects of our approach are highlighted. First, considering each algorithm independently, the best representation of each dataset is formatted in *italics*. Then, the best algorithm for each dataset representation is highlighted in **bold**.

As a general observation, the configurations that consider either the conceps view or the combination of terms and concepts often lead to the best results with very few exceptions, independently from the considered clustering algorithm. This result confirms that, in most contexts, the classical terms based approaches do not capture the embedded cluster structure sufficiently. Moreover, regarding the second evaluated aspect, *CVCC* almost always performs the best with 20-Newsgroups and exhibits significant differences with the other algorithms regardless of the complexity level of the dataset.

With the low complexity instance (L1) of Reuters-21578 corpus, instead, the differences among the four algorithms are less marked, with our algorithm providing always the second best results. This behavior is confirmed with L2, but in this case, the LSA-enhanced version of *K-means* performs significantly better than any other competitor. This is probably due to a minor contribution of polysemy in these two version of the dataset, which also explains the exceptional outperformances of the conceptual representation with *CVCC*. However, with the L3 instance of Reuters data, *CVCC* outperforms all other competitors by far, thus confirming that in more complex scenarios our multi-view co-clustering approach shows its effectiveness compared to other approaches.

Finally, it is worth noting that, in general, the adoption of a two-view schema has two positive effects on the number of discovered clusters: not only does it

[4] An iteration in *CVCC* corresponds to a single object movement [11].

Table 3. Mean and standard deviation of Adjusted Rand Index. The best ARI value for each experimental setting is highlighted in **bold**, while the best representation for each algorithm is formatted in *italic*.

Dataset	View	No. clusters	Adjusted Rand Index (ARI)			
			CVCC	EBC	LSA-KM	(Co)NMF
20ng-l1	terms	8.3 (3.01)	**0.53 (0.11)**	0.28 (0.10)	0.23 (0.03)	0.28 (0.00)
	conc.	10.1 (4.21)	**0.44 (0.09)**	0.25 (0.10)	0.21 (0.02)	0.41 (0.00)
	both	5.8 (2.26)	***0.54 (0.06)***	*0.32 (0.08)*	*0.24 (0.01)*	*0.48 (0.07)*
20ng-l2	terms	10.5 (2.7)	**0.46 (0.04)**	0.21 (0.08)	0.19 (0.03)	0.23 (0.00)
	conc.	9.9 (2.36)	**0.42 (0.04)**	0.17 (0.07)	0.14 (0.02)	0.31 (0.00)
	both	8.3 (1.32)	***0.47 (0.03)***	*0.25 (0.06)*	*0.20 (0.02)*	*0.36 (0.06)*
20ng-l3	terms	17.9 (7.99)	***0.30 (0.05)***	*0.23 (0.06)*	*0.21 (0.01)*	0.26 (0.00)
	conc.	13 (6.01)	**0.25 (0.03)**	0.21 (0.05)	0.19 (0.01)	0.21 (0.00)
	both	9.5 (3.46)	0.28 (0.04)	*0.23 (0.07)*	*0.21 (0.01)*	***0.31 (0.03)***
reut-l1	terms	7.2 (2.41)	0.43 (0.11)	0.41 (0.09)	**0.45 (0.02)**	0.18 (0.00)
	conc.	9.9 (2.43)	*0.54 (0.12)*	0.39 (0.11)	***0.55 (0.07)***	*0.45 (0.00)*
	both	7.6 (2.14)	**0.52 (0.17)**	*0.42 (0.12)*	0.45 (0.03)	0.25 (0.13)
reut-l2	terms	13.4 (1.6)	0.54 (0.06)	0.34 (0.11)	**0.57 (0.15)**	0.51 (0.00)
	conc.	12.2 (1.1)	*0.64 (0.07)*	0.36 (0.10)	***0.71 (0.05)***	*0.52 (0.00)*
	both	11.9 (1.12)	0.61 (0.06)	*0.40 (0.11)*	***0.71 (0.10)***	0.49 (0.09)
reut-l3	terms	2.4 (0.95)	**0.43 (0.01)**	*0.37 (0.10)*	0.39 (0.11)	0.18 (0.00)
	conc.	7.4 (2.54)	**0.48 (0.06)**	0.36 (0.09)	0.43 (0.07)	*0.21 (0.00)*
	both	3.1 (0.89)	***0.51 (0.06)***	*0.37 (0.10)*	*0.44 (0.10)*	0.20 (0.03)

better approach the correct number of categories with respect to single-view representations, but it also becomes more stable.

4 Conclusions

We presented a novel multi-view approach to semantically enhanced document co-clustering. Our algorithm can simultaneously process multiple representations of the same document. In particular, in the current setting we consider two views: document-term and document-concept. In the majority of cases, the results showed a clear advantage in using this strategy, compared to other well-known methods for document clustering. As future work, we plan to expand the conceptual representation with the inclusion of semantically related information in order to take advantage of relations between concepts. Finally, we will inspect the performances of our approach on different domains, e.g., image data or geographically annotated data, in which elements can be represented by additional views, e.g., SIFT and georeferred features.

Acknowledgments. The work is supported by Compagnia di San Paolo foundation (grant number Torino_call2014_L2_157).

References

1. Aggarwal, C.C., Zhai, C.: A survey of text clustering algorithms. In: Mining Text Data, pp. 77–128. Springer (2012)
2. Boutsidis, C., Gallopoulos, E.: SVD based initialization: A head start for nonnegative matrix factorization. Pattern Recognit. 41(4), 1350–1362 (2008)
3. Cichocki, A., Phan, A.H.: Fast local algorithms for large scale nonnegative matrix and tensor factorizations. IEICE Trans. 92-A(3), 708–721 (2009)
4. Dhillon, I.S., Mallela, S., Modha, D.S.: Information-theoretic co-clustering. In: Proc. ACM SIGKDD 2003. pp. 89–98. ACM (2003)
5. Gabrilovich, E., Markovitch, S.: Feature generation for text categorization using world knowledge. In: Proc. IJCAI 2005. pp. 1048–1053 (2005)
6. Goodman, L.A., Kruskal, W.H.: Measures of association for cross classification. J. Am. Stat. Assoc. 49, 732–764 (1954)
7. He, X., Kan, M., Xie, P., Chen, X.: Comment-based multi-view clustering of web 2.0 items. In: Proc. WWW 2014. pp. 771–782 (2014)
8. Hu, J., Fang, L., Cao, Y., Zeng, H., Li, H., Yang, Q., Chen, Z.: Enhancing text clustering by leveraging wikipedia semantics. In: Proc. SIGIR 2008. pp. 179–186. ACM (2008)
9. Huang, A., Milne, D.N., Frank, E., Witten, I.H.: Clustering documents using a wikipedia-based concept representation. In: Proc. PAKDD 2009. vol. 5476, pp. 628–636. Springer (2009)
10. Hubert, L., Arabie, P.: Comparing partitions. J. of Classif. 2(1), 193–218 (1985)
11. Ienco, D., Robardet, C., Pensa, R.G., Meo, R.: Parameter-less co-clustering for star-structured heterogeneous data. Data Min. Knowl. Discov. 26(2), 217–254 (2013)
12. Kalmanovich, I.G., Kurland, O.: Cluster-based query expansion. In: Proc. ACM SIGIR 2009. pp. 646–647. ACM (2009)
13. Landauer, T.K., Foltz, P.W., Laham, D.: An Introduction to Latent Semantic Analysis. Discourse Process. 25(2-3), 259–284 (1998)
14. Lin, C.: Projected gradient methods for nonnegative matrix factorization. Neural Comput. 19(10), 2756–2779 (2007)
15. Lloyd, S.P.: Least squares quantization in PCM. IEEE Trans. Inf. Theory 28(2), 129–136 (1982)
16. Moro, A., Raganato, A., Navigli, R.: Entity linking meets word sense disambiguation: a unified approach. Trans. of the ACL 2, 231–244 (2014)
17. Navigli, R., Ponzetto, S.P.: Babelnet: The automatic construction, evaluation and application of a wide-coverage multilingual semantic network. Artif. Intell. 193, 217–250 (2012)
18. Percha, B., Altman, R.B.: Learning the structure of biomedical relationships from unstructured text. PLoS Comput. Biol. 11(7) (2015)
19. Recupero, D.R.: A new unsupervised method for document clustering by using wordnet lexical and conceptual relations. Inf. Retr. J. 10(6), 563–579 (2007)
20. Shen, C., Li, T., Ding, C.H.Q.: Integrating clustering and multi-document summarization by bi-mixture probabilistic latent semantic analysis (PLSA) with sentence bases. In: Proc. AAAI 2011. pp. 914–920. AAAI Press (2011)
21. Wei, T., Lu, Y., Chang, H., Zhou, Q., Bao, X.: A semantic approach for text clustering using wordnet and lexical chains. Expert Syst. Appl. 42(4), 2264–2275 (2015)
22. West, J.D., Wesley-Smith, I., Bergstrom, C.T.: A recommendation system based on hierarchical clustering of an article-level citation network. IEEE Trans. Big Data 2(2), 113–123 (2016)