Descriptive Document Clustering via Discriminant Learning in a Co-embedded Space of Multi-level Similarities

Tingting Mu
School of Electrical Engineering, Electronics and Computer Science
University of Liverpool
Liverpool, UK

John Y. Goulermas
School of Electrical Engineering, Electronics and Computer Science
University of Liverpool
Liverpool, UK

Ioannis Korkontzelos
National Centre for Text Mining
University of Manchester
Manchester, UK

Sophia Ananiadou
National Centre for Text Mining
University of Manchester
Manchester, UK

Abstract

Descriptive document clustering aims at discovering clusters of semantically interrelated documents together with meaningful labels to summarise the content of each document cluster. In this work, we propose a novel descriptive clustering framework, referred to as CEDL. It relies on the formulation and generation of two types of heterogeneous objects, that correspond to documents and candidate phrases, using multi-level similarity information. CEDL is composed of five main processing stages. Firstly, it simultaneously maps the documents and candidate phrases into a common co-embedded space that preserves higher-order neighbour-based proximities between the combined sets of documents and phrases. Then, it discovers an approximate cluster structure of documents in the common space. The third stage extracts promising topic phrases by constructing a discriminant model where documents along with their cluster memberships are used as training instances. Subsequently, the final cluster labels are selected from the topic phrases using a ranking scheme utilising multiple scores based on the extracted co-embedding information and the discriminant output. The final stage polishes the initial clusters to reduce noise and accommodate the multi-topic nature of documents. The effectiveness and competitiveness of CEDL is demonstrated qualitatively and quantitatively with experiments using document databases from different application fields.

Descriptive Document Clustering via Discriminant Learning in a Co-embedded Space of Multi-level Similarities

Introduction

Descriptive clustering is an important task in information retrieval and text mining, and is defined as the “discovering of diverse groups of semantically related documents described with meaningful, comprehensible and compact text labels" (Weiss, 2006). Such a task facilitates the management of the large and ever-growing collection of electronic documents. It is particularly useful in search results clustering for grouping fragments of documents retrieved by a search engine, in order to accommodate the explosive expansion of the information accessible in the internet (Bharambe & Kale, 2011). Traditional document clustering only focuses on one objective, which is to find groups of similar documents without the need to generate descriptive labels for the pinpointed clusters. Differently, the primary goal of descriptive clustering is to extract accurate, comprehensible and succinct descriptive labels for summarising document cluster contents. In certain practical circumstances, the generation of good quality labels is considered even more important than the discovery of accurate document clusters. For example, in the development of a web-based search results clustering systems (Osiński & Weiss, 2005; Weiss, 2006; Stefanowski & Weiss, 2007; Koshman, Spink, & Jansen, 2006), the priority is to help users to understand rapidly the content of a document collection through comprehensible cluster labels.

Relevant descriptive clustering techniques are summarised in various surveys on topic discovery (Jayabharathy, Kanmani, & Parveen, 2011) and search results clustering (Bharambe & Kale, 2011; Carpineto, Osiński, Romano, & Weiss, 2009). These techniques are developed by utilising knowledge and methodologies on information retrieval and machine learning. Frequently used descriptive clustering approaches include phrase-based clustering, e.g., frequent term (FT) based clustering (Beil, Ester, & Xu, 2002; Y. Li, Chung, & Holt, 2008) and suffix tree clustering (STC) and its variations (Zamir, 1999; Janruang & Kreesuradej, 2006; Wang, Mo, Huang, Wen, & He, 2008), various traditional clustering algorithms combined with the procedure of post-assigning cluster labels (Cutting, Karger, Pedersen, & Tukey, 1992; Lagus, Kaski, & Kohonen, 2004; Tseng, 2010), as well as description-comes-first (DCF) clustering approaches, e.g., descriptive k-means (DKM) clustering (Weiss, 2006; Stefanowski & Weiss, 2007) and concept-driven clustering based on matrix factorisation (known as Lingo) (Osiński, Stefanowski, & Weiss, 2004). Discussions of the advantages and disadvantages of these methods, as well as brief descriptions of several representative methods are provided in the subsequent section.

Descriptive clustering solves a mixture of the document grouping (finding document clusters) and summarisation (finding descriptive labels) problems. Although many methods have been developed, most

of them provide solutions that offer performance improvement to only one of the two problems, e.g., pursuing either higher-quality document clusters or more meaningful descriptive labels (further discussions are included in the subsequent section). However, both tasks of cluster discovery and cluster label generation are actually of equal importance and highly sought by many text mining tools and services, and it is thus important to study how to simultaneously improve each objective without unilaterally focusing on either one. We will show that this new aim can transform the standard homogeneous data analysis problem, e.g., grouping documents or ranking phrase labels respectively on their own, to a more challenging heterogeneous data analysis problem of jointly analysing documents and descriptive phrases by blending various connections between documents, between phrases, and between document and phrases. To effectively solve the introduced heterogeneous problem, we propose a novel and powerful framework that constitutes the principal contribution in this work.

The proposed framework is a multi-stage one driven by three input similarity matrices: a between-document matrix, a between-phrase matrix, and a two-mode one between documents and phrases. It discovers document clusters and induces cluster labels by using clustering, embedding and discriminant learning techniques. It is composed of the following five main processing components: (i) generation of co-embeddings, (ii) exploration of cluster structure, (iii) discovery of topic phrases, (iv) determination of cluster labels, and (v) verification of document clusters. Both the cluster discovery and label generation procedures are performed in a co-embedded space computed from a second-order neighbour-based similarity matrix. This matrix is designed to represent better the syntactic and semantic connections between and within the document sets and candidate phrase sets, and thus leads to more informative co-embeddings. The cluster labels are generated by following a gradual multi-stage search procedure that consults a discriminant model constructed using the documents associated with a set of initial cluster memberships as the training instances, and multiple scores to support ranking. Such a design aims at increasing the internal consistency of the document clusters and complying with the descriptive nature of the language. For evaluation, we use documents from three databases in different application fields to conduct comparative analysis between CEDL and two state-of-the-art descriptive clustering methods. Experimental results demonstrate the effectiveness of our proposed method.

Relevant Techniques

Among existing descriptive clustering algorithms, STC is perhaps the earliest one that discovers and merges clusters based on phrase co-occurrences (Zamir, 1999). Subsequently proposed FT-based clustering approaches (Beil et al., 2002; Y. Li et al., 2008) are designed to further improve STC, by directly manipulating frequently occurred terms with respect to their supporting documents. Both STC and

FT-based approaches operate on sequences of words according to document counts and phrase/term frequencies. However, this may not well imply semantic connections between documents (Janruang & Guha, 2011). Differently, traditional clustering algorithms and statistical topic models (Zeng, Cheung, & Liu, 2013) have better chance of producing good semantic clusters with the use of sophisticated linguistic features (Hatzivassiloglou, Gravano, & Maganti, 2001) or their corresponding embeddings for discovering latent semantic information (Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990; Lam, Tsang, & Wong, 2013), as well as external information resources (Noel, Raghavan, & Chu, 2003; Theodosiou, Darzentas, Angelis, & Ouzounis, 2008; Chang & Chen, 2011). These methods combined with strategies of post-assigning cluster labels constitute an important group of descriptive clustering approaches as described below.

Traditional Clustering with Post-assignment of Labels

This type of approaches typically start with characterising each target document with a set of numerical features, and proceed with grouping the documents into clusters within the feature space. Finally, they generate a text label to summarise the content of each document cluster based on its input text.

To obtain the numerical feature representation for a set of documents, the vector space model (VSM) (Baeza-Yates & Ribeiro-Neto, 1999; Dubin, 2004) is commonly used, and is often combined with a term weighting scheme for computing the feature value (Salton & Buckley, 1987; Lan, Tan, Su, & Lu, 2009). Let $\{\boldsymbol{x}_i\}_{i=1}^{n}$ denote the set of d-dimensional feature vectors extracted from the n targeted documents, where $\boldsymbol{x}_i = [x_{i1}, x_{i2}, \ldots, x_{id}]^T$, and $\mathbf{X} = [x_{ij}]$ denote the corresponding $n \times d$ feature matrix, known as the document-term matrix. A clustering algorithm groups the documents into flat or hierarchical clusters based on the matrix $\mathbf{X}$, for instance, k-means clustering and its variations (Jain, 2010; Cutting et al., 1992; Hearst & Pedersen, 1996; P. Jiang, Zhang, Guo, Niu, & Gao, 2009), matrix decomposition or factorisation based clustering (Drineas, Frieze, Kannan, Vempala, & Vinay, 1999; W. Xu, Liu, & Gong, 2003; Zhang & Dong, 2004; Z. Li, Peng, & Wu, 2008), self-organising maps (Kohonen, 1990; Lagus et al., 2004) and fuzzy clustering (Krishnapuram, Joshi, & Yi, 1999; Z. Jiang, Joshi, K., & Yi, 2000; Matsumoto & Hung, 2010). Other than relying on the feature representation in terms of $\mathbf{X}$, some clustering algorithms directly operate on a similarity matrix or equivalently a relationship graph between documents; these include minimum spanning tree clustering (Rooneya, Pattersona, Galushkaa, & Dobryninb, 2006; Chang & Chen, 2011) and Markov clustering (Dongen, 2000; Theodosiou et al., 2008). For these algorithms, the similarity matrix or graph to be used as the input can either be computed directly from $\mathbf{X}$ (Carmel, Roitman, & Zwerdling, 2009; Hussain, Bisson, & Grimal, 2010), or constructed separately from external

information resources (Noel et al., 2003; Theodosiou et al., 2008; Chang & Chen, 2011).

To summarise the content of an acquired document cluster with a description label, a candidate label pool is first constructed, which includes multiple potential descriptive labels. There are several ways for collecting such candidates. For example, they can be short lists of single words contained within the documents (Cutting et al., 1992; Lagus & Kaski, 1999; Lagus et al., 2004), phrases directly extracted from the input text (Zamir, 1999; Weiss, 2006; Jones & Paynter, 2002), key terms obtained from external resources, e.g., the electronic lexical database WordNet (Tseng, 2010), MeSH terms from PubMed (Theodosiou et al., 2008) and the free on-line encyclopedia Wikipedia (Syed, Finin, & Joshi, 2008; Carmel et al., 2009), as well as phrases contained by a predefined ontology to guarantee comprehensibility (Weiss, 2006). Subsequently, for each document cluster, a set of descriptive scores is computed for these candidates and the one possessing the highest score is selected as the corresponding cluster label. Usually, the scores can be formulated with phrase lengths, term and/or document frequencies. Different scoring schemes that can be used for such purpose are presented in (Lagus & Kaski, 1999; Treeratpituk & Callan, 2006; Geraci, Pellegrini, Maggini, & Sebastiani, 2006; Tseng, 2010).

Although good semantic clusters of documents can be discovered due to the use of powerful features or (dis)similarity representations of documents, the generated cluster labels may not always be satisfactory (Weiss, 2006). This is because most strategies of post-assigning cluster labels are solely based on occurrence frequencies of the candidate phrases, and thus, they can be semantically shallow. To improve this, strategies that focus more on generating comprehensible, concise and transparent cluster labels than document groupings are proposed, known as DCF with emphasis on cluster description. In the following, we briefly introduce the two commonly used DCF approaches of DKM and Lingo.

Descriptive k-Means Algorithm

The basic idea of DCF is to determine the cluster labels first, and then discover the document clusters based on the induced labels. DKM (Weiss, 2006; Stefanowski & Weiss, 2007) uses frequent phrases and noun phrases extracted from the input text of the targeted documents via suffix trees and a pre-trained statistical chunker as the candidate label pool.

To obtain the cluster labels, DKM first employs k-means clustering to group the targeted documents into a pre-defined number of clusters. The used features for characterising these documents are obtained by combining VSM and a term weighting scheme with each feature dimension corresponding to a uni-gram word. For each cluster centroid feature vector, its top b features that possess the highest feature values indicate the top b words that are the most relevant to the topic of that cluster. DKM determines each cluster label based on the corresponding top b relevant words. Specifically, DKM builds a simple

VSM-based information retrieval system with the top words forming a weighted boolean query. A descriptive score measuring the relevance of each candidate label to the query is computed based on this retrieval system, where the users are allowed to adjust the score by penalising it with a desired length (number of words) for the label. For each cluster centroid, the candidate label possessing the highest descriptive score is selected as the descriptive label.

Other than adopting the clustering results of k-means as the final document clusters, DKM implements an extra document reallocation procedure to produce overlapping between document clusters. For each cluster label, any document that includes the exact copy of this label, its possibly distorted version or any synonymous phrases, it is assigned as its member. Cluster labels that do not collect enough documents are removed.

Compared to the label post-assignment procedures combined with the traditional clustering approaches, the prior label generation procedure used by DKM is more sophisticated. It has higher chance of producing more meaningful cluster labels due to its two-stage label selection procedure from the top b words to the final label phrase, as well as the carefully designed descriptive score for ranking the candidate phrases. However, the later document reallocation may affect the quality of document clusters when it replaces the original cluster structure discovered by k-means clustering with ones obtained purely based on phrase occurrence rate. During this procedure, semantic connections between documents may be lost.

Concept-driven Clustering based on Matrix Factorisation

Lingo is another popular DCF approach known as concept-driven clustering based on matrix factorisation (Osiński et al., 2004; Osiński & Weiss, 2005; Weiss, 2006; Osiński, 2006). Its candidate label pool includes both uni-gram words and a set of m frequent phrases extracted from the input text of the targeted documents based on suffix trees. By applying VSM with a tf-idf (term frequency - inverse document frequency) weighting scheme to both documents and candidate labels, Lingo operates on two input matrices. One is the $n \times d$ feature matrix $\mathbf{X}$ of documents, obtained by converting each targeted document to tf-idf values of a total of d uni-gram words. The other is the $(m + d) \times d$ feature matrix $\mathbf{X}_c = [\mathbf{X}_p^T, \mathbf{I}_{d\times d}]^T$ of candidate labels. The $m \times d$ matrix $\mathbf{X}_p$ converts each phrase candidate to tf-idf values of the d uni-gram words. The identity matrix $\mathbf{I}_{d\times d}$ represents the feature matrix of the word candidates in terms of uni-gram words themselves. The row vectors of $\mathbf{X}$ and $\mathbf{X}_p$ are normalised to have unit length.

To discover latent semantic topics from the targeted documents, the singular value decomposition (SVD) of $\mathbf{X}$ is computed. Letting $\mathbf{U}_k$ denote the $n \times k$ left singular vector matrix, $\mathbf{V}_k$ the $d \times k$ right singular vector matrix, and $\mathbf{S}_k$ the $k \times k$ singular value matrix, the quantity $\mathbf{U}_k\mathbf{S}_k\mathbf{V}_k^T$ is well-known to provide the k-rank approximation of $\mathbf{X}$. The number of k is controlled by Lingo through a percentage

threshold $0 < q \leq 1$. By treating the $k \times d$ matrix $\mathbf{V}_k^T$ as the feature matrix of k latent topic vectors of dimensionality d, Lingo computes a similarity matrix between the k topics and the $(m + d)$ candidate labels as $\mathbf{S}_{tc} = \mathbf{V}_k^T \mathbf{X}_c^T$. Then, the candidate that is the most similar to each semantic topic is selected as its descriptive label. Such a procedure discovers a total of k cluster labels corresponding to k potential document topics.

To discover document clusters based on the obtained cluster labels, Lingo computes an $n \times k$ membership matrix $\mathbf{M} = [m_{ij}]$ via $\mathbf{M} = \mathbf{X}\mathbf{X}_l^T$, where $\mathbf{X}_l$ denotes the $k \times d$ feature matrix of the cluster labels in terms of uni-gram words. Each element m_{ij} indicates the relevance between the ith document and the jth cluster label. The ith document is added to the jth cluster if m_{ij} exceeds a snippet assignment threshold. Documents that are not assigned to any cluster constitute a new cluster of "others".

Lingo generates cluster labels via manipulating latent variables produced by SVD. Its performance heavily relies on the validity of the similarity assumption formulated by $\mathbf{S}_{tc} = \mathbf{V}_k^T \mathbf{X}_c^T$, which unfortunately can be unreliable when processing lengthy documents, and can thus, lead to unsatisfactory cluster quality.

Overall, existing methods, e.g., STC and FT-based clustering, that partition documents by individually looking into their shared and/or frequent phrases and sequences of terms are not able to capture well the hidden semantic information in the text. By utilising more sophisticated clustering models and DCF models, improvement can be achieved in terms of either document clusters or descriptive labels, but not both. To further improve the development of descriptive clustering techniques, we design a novel descriptive clustering framework, aiming at simultaneously producing good quality of both document clusters and descriptive labels.

The Proposed Method

Descriptive clustering is a unique heterogeneous data analysis task, which requires to manipulate two sets of heterogeneous objects, a set of n documents $\{\mathfrak{D}_\mathfrak{i}\}_{i=1}^n$ to be clustered and a set of m phrases $\{\mathfrak{P}_\mathfrak{i}\}_{i=1}^m$ to be used as candidate labels, according to links between them. The considered link information, such as similarities between documents, or connections between phrases and document contents, dominates the procedure of what document clusters to discover and which phrase labels to choose. For example, STC and FT-based approaches utilise frequent phrases occurring in documents as the link information to find descriptive labels and also partition documents. Some traditional approaches combined with post-assignment of labels utilise similar frequency-based link information to choose labels, but different types of between-document similarities as the link information to discover document clusters. While DKM and Lingo utilise frequency-based link information to partition documents, they rely on different types of post-computed link information between phrases and latent topics to find descriptive labels. Limited by

not considering all links between objects or not quantifying these links in sophisticated ways, their clustering performance and label quality can be suboptimal.

Considering these issues, we aim at building a powerful framework that is able to effectively blend all possible links between documents, between phrases, and between document and phrases, so that appropriate document clusters and descriptive labels can be optimally discovered. Also, the constructed framework should be as generic as possible, so that it becomes capable of encoding link information quantified by different measures or computed by different means. To achieve this, we build a descriptive clustering framework through the use of clustering (C), embedding (E) and discriminant learning (DL) techniques, which we will refer to as CEDL. In the subsequent subsections, we first explain how to process the input documents to extract useful information for document clustering and label determination. We then detail the five processing stages of the proposed algorithm that include (1) generation of co-embeddings, (2) exploration of cluster structure, (3) discovery of topic phrases, (4) determination of cluster labels, and (5) verification of document clusters. The overall operations of CEDL and its main flow of the data structures through its various stages are overviewed in Figure 1. The notation of its variables and quantities used to formulate the method are summarised in Table 1. Within the algorithm, Stage 1 constructs a common platform for simultaneous manipulation of documents and phrases, Stage 2 performs document clustering, while Stages 3 and 4 determine the cluster label based on a gradual multi-stage search process. A running example for this search process is shown in Figure 2. Finally, Stage 5 further verifies and trims the obtained clusters and labels.

Input Preparation

The CEDL framework handles two object sets of documents $\{\mathfrak{D}_\mathfrak{i}\}_{i=1}^{n}$ and candidate phrases $\{\mathfrak{P}_\mathfrak{i}\}_{i=1}^{m}$. It operates on three input similarity matrices of $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$, where the $n \times n$ matrix $\mathbf{S}_{dd}$ represents similarities between the n documents, the $m \times m$ matrix $\mathbf{S}_{pp}$ represents connections between the m candidate phrases, and the $n \times m$ matrix $\mathbf{S}_{dp}$ represents relevance values between the n documents and the m phrases. These matrices are capable of storing fundamental links between any two objects from the combined $\{\mathfrak{D}_\mathfrak{i}\}_{i=1}^{n}$ and $\{\mathfrak{P}_\mathfrak{i}\}_{i=1}^{m}$. Furthermore, as shown in the following subsection, these matrices are generic enough to accommodate different measures or ways for quantifying or computing these links. Below, we first explain how to extract the candidate phrases and then describe how to compute the three matrices.

Candidate Phrase Extraction. The candidate phrases $\{\mathfrak{P}_\mathfrak{i}\}_{i=1}^{m}$ can either be extracted from the input text of the n documents using established techniques for phrase extraction (Weiss, 2006; Zamir, 1999; Jones & Paynter, 2002) or collected from various external resources (Tseng, 2010; Theodosiou et al., 2008; Syed et al., 2008; Carmel et al., 2009; Weiss, 2006). In this work, we follow the former way due to the data

availability, and extract candidate phrases from the input documents using the term extractor Termine[1]. Since it is more natural to use noun phrases as short descriptions to summarise the topic of a document cluster, Termine collects from the documents all the multi-word terms that are sequences of adjectives and nouns, contain at least two unigram words, and end with nouns. Adjectives and nouns are recognised using the GENIA part of speech tagger[2] (Tsuruoka et al., 2005). A score of importance, referred to as C-value, is computed for each term according to its frequency, length and nestedness, given as (Frantzi, Ananiadou, & Mima, 2000)

$$C = \log_2 |\mathfrak{P}| \left(f(\mathfrak{P}) - \frac{1}{|T_{\mathfrak{P}}|} \sum_{\mathfrak{P}_i \in T_{\mathfrak{P}}} f(\mathfrak{P}_i) \right). \tag{1}$$

Here, $f(\cdot)$ denotes the frequency of a term appearing in the documents. For a given term $\mathfrak{P}$, the set $T_{\mathfrak{P}}$ includes $\mathfrak{P}$ itself and all the longer terms that contain $\mathfrak{P}$. The notation $|\cdot|$ computes the number of words contained by an input term, or the number of terms contained by an input set. This score encourages the extraction of longer, more frequently appearing but less nested terms. The top m terms with the highest scores are used as the candidate phrases.

Similarity Matrix Construction. Based on the text content of the extracted candidate phrases and the target documents, a general procedure for obtaining $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$ is proposed. Initially, three intermediate matrices of tf-idf values are generated: the $n \times d$ feature matrix $\mathbf{X}_{dw}$ of documents, which converts each document to tf-idf values of a total of d uni-gram words; the $m \times d$ feature matrix $\mathbf{X}_{pw}$ of phrases, which converts each candidate phrase to tf-idf values of the same set of d uni-gram words; and another $n \times m$ feature matrix $\mathbf{X}_{dp}$ of documents, which converts each document to tf-idf values of the m candidate phrases.

To obtain $\mathbf{X}_{dp}$, we treat both the possibly distorted and synonymous versions of a candidate phrase the same as the exact copy of itself, in order to take into account better semantic connections between phrases and documents. Using these three tf-idf matrices $\mathbf{X}_{dw}$, $\mathbf{X}_{pw}$ and $\mathbf{X}_{dp}$, we then compute the sought ones $\mathbf{S}_{dp}$, $\mathbf{S}_{dd}$ and $\mathbf{S}_{pp}$ as follows:

- The similarity matrix between documents and candidate phrases is computed as

$$\mathbf{S}_{dp} = \frac{\mathbf{X}_{dp}}{\max(\mathbf{X}_{dp})} + \cos(\mathbf{X}_{dw}, \mathbf{X}_{pw}). \tag{2}$$

The function $\max(\cdot)$ returns the maximum element of the input matrix. The function $\cos(\cdot,\cdot)$ returns cosine similarities between two sets of instances represented by its two input matrices. Specifically, letting $\mathbf{A}$ denote an $n \times d$ and $\mathbf{B}$ an $m \times d$ input matrix, the $n \times m$ output matrix $\cos(\mathbf{A}, \mathbf{B})$ is computed by $\mathbf{D}_A^{-\frac{1}{2}} \mathbf{A}\mathbf{B}^T \mathbf{D}_B^{-\frac{1}{2}}$, where $\mathbf{D}_A = \text{diag}\left(\mathbf{A}\mathbf{A}^T\right)$, $\mathbf{D}_B = \text{diag}\left(\mathbf{B}\mathbf{B}^T\right)$. The function $\text{diag}(\cdot)$ returns a diagonal

[1] http://www.nactem.ac.uk/software/termine

[2] http://www.nactem.ac.uk/tsujii/GENIA/tagger

matrix formulated with the diagonal vector of its input matrix, or the corresponding diagonal matrix if its input is a vector. The first term of Eq. (2) is a scaled version of $\mathbf{X}_{dp}$ that is based on the normalised frequencies of candidate phrases appearing in documents to measure the relevance between them. The second term evaluates the similarity between a document and a candidate phrase through their word co-occurrences. The combined term attempts to capture deeper semantic connections between documents and candidate phrases than each individual term.

- The similarity matrix between documents is given by

$$\mathbf{S}_{dd} = \cos\left([\mathbf{X}_{dw}, \mathbf{X}_{dp}], [\mathbf{X}_{dw}, \mathbf{X}_{dp}]\right). \tag{3}$$

In the above equation, we measure the cosine-based similarity between documents using the aggregate $n \times (d + m)$ feature matrix $[\mathbf{X}_{dw}, \mathbf{X}_{dp}]$. This matrix captures the semantic closeness between documents, based on word and (synonymous) phrase co-occurrences.

- The similarity matrix between candidate phrases is given by

$$\mathbf{S}_{pp} = \cos(\mathbf{X}_{pw}, \mathbf{X}_{pw}) + \cos(\mathbf{F}_{pd}, \mathbf{F}_{pd}), \tag{4}$$

where the ijth element of the binary $m \times n$ matrix $\mathbf{F}_{pd}$ is 1 when the ith phrase and its distorted or synonymous versions appear in the jth document, and 0 otherwise. The first term of Eq. (4) computes similarities between candidate phrases, based on their word co-occurrences, while the second term assumes that similarity between two candidate phrases is indirectly related to the number of documents that contain their copies or synonymous versions. A combination of these two terms attempts to capture deeper semantic connections between phrases than each term individually.

Instead of the above procedure, there can be alternative ways for creating these matrices. The users of CEDL can experiment and define different similarity matrices using, for instance, advanced natural language processing and text mining tools, or by taking advantage of expert knowledge and external resources available in their domains. For example, it is possible to compute $\mathbf{S}_{dd}$ by employing linguistically motivated text features instead of simple bag-of-word features (Hatzivassiloglou et al., 2001), or by judging whether two documents are related through citations or hyperlinks (Noel et al., 2003), or by using "related articles" information returned by PubMed (Theodosiou et al., 2008). It is also possible to construct $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$ by considering semantic similarities between the words contained by the documents and phrases, computed based on the distributional representation of words learned by, for example, a deep network (Bengio, Ducharme, Vincent, & Jauvin, 2003; Bengio, Courville, & Vincent, 2013). As will be seen later, we attempt to generalise the design of CEDL so that it can be compatible with any types of input similarity information $\mathbf{S}_{dp}$, $\mathbf{S}_{dd}$ and $\mathbf{S}_{pp}$ suitable to a particular domain or corpus.

Algorithm Description

The working routines of many existing descriptive clustering algorithms, especially those combining a traditional clustering method with post-assignment of cluster labels, e.g., Matsumoto and Hung (2010); Theodosiou et al. (2008), are actually equivalent to first clustering the documents based on $\mathbf{S}_{dd}$, and then choosing as cluster label the phrase that possesses the highest averaged similarity to a cluster of documents based on $\mathbf{S}_{dp}$, where $\mathbf{S}_{dd}$ and $\mathbf{S}_{dp}$ are defined differently according to the algorithm. Such operation takes into account $\mathbf{S}_{dd}$ only in cluster discovery and $\mathbf{S}_{dp}$ only in label induction. It not only ignores the possible impact of $\mathbf{S}_{dp}$ and $\mathbf{S}_{pp}$ in cluster discovery, and the impact of $\mathbf{S}_{dd}$ and $\mathbf{S}_{pp}$ in label induction, but it also completely neglects interactions and latent connections between all three similarity matrices, which could potentially offer significant semantic information gain and thus improve the quality of document clusters and their labels. Considering these issues, we design CEDL so that it is capable of utilising better both the direct connections contained within $\mathbf{S}_{dp}$, $\mathbf{S}_{dd}$ and $\mathbf{S}_{pp}$ and the latent connections implied across all three matrices. In the following subsections, we explain in detail the proposed algorithm in terms of its five main processing components.

Stage 1: Generation of Co-embeddings. Differently from standard clustering tasks, descriptive clustering requires to manipulate two sets of heterogeneous objects according to links between them. Usually, instead of trying to reason from these links directly, it is more convenient to learn a common space to embed both types of objects where the link information can be preserved (Globerson, Chechik, Pereira, & Tishby, 2007; Mu & Goulermas, 2013). This is because such a common space can be viewed as an alternative representation to the similarity-based one that stores the link information. It also enables to treat heterogeneous objects as homogeneous ones, which broadens the range of techniques that can be used to analyse them, and also offers opportunities for more thorough analyses. Specifically, in the first stage of CEDL, a compact, lower dimensional, common description space is sought to embed simultaneously the documents and candidate phrases by preserving the similarities as captured by $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$. In order to effectively blend the link information stored in $\mathbf{S}_{dp}$, $\mathbf{S}_{dd}$ and $\mathbf{S}_{pp}$ so that not only the direct connections contained within them, but the latent connections implied across them are also considered, an enhancing procedure is conducted on these three matrices to output a composite version of $\mathbf{S}_{dp}$, $\mathbf{S}_{dd}$ and $\mathbf{S}_{pp}$ highlighting the latent connections from which embeddings are computed, instead of relying on their original versions.

The composite version of $\mathbf{S}_{dp}$, $\mathbf{S}_{dd}$ is denoted as an $(n+m) \times (n+m)$ similarity matrix $\mathbf{S} = [s_{ij}]$. It treats equally the documents and phrases as homogeneous objects, from which embeddings are computed. A convenient way of formulating this overall similarity matrix is to combine the three partial similarity

matrices $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$ as

$$\mathbf{S}_0 = \begin{bmatrix} w_1\mathbf{S}_{dd} & w_3\mathbf{S}_{dp} \\ w_3\mathbf{S}_{dp}^T & w_2\mathbf{S}_{pp} \end{bmatrix}, \tag{5}$$

where the three parameters $w_1, w_2, w_3 > 0$ are used to adjust the weight of each partial similarity matrix. When elements of the three matrices of $\mathbf{S}_{dp}$, $\mathbf{S}_{dd}$ and $\mathbf{S}_{pp}$ are within a relatively similar range, a convenient setting of $w_1 = w_2 = w_3 = 1$ can be used without enhancing or weakening the contribution of any of the three constituent matrices.

In order to reduce noise and sharpen the potential structure of $\mathbf{S}_0$, we apply a nearest neighbour search to $\mathbf{S}_0$. This procedure removes unimportant information corresponding to distant object pairs in $\mathbf{S}_0$, because the proximity information structure is better controlled by the local geometry of the data patterns. Specifically, we define a binary neighbourhood indication matrix $\mathbf{N}(\mathbf{S}_0, h_1) = [\delta_{ij}]$ by applying an h_1-nearest neighbour search to the similarity matrix $\mathbf{S}_0$. By letting $\boldsymbol{o}_i$ denote the ith from the $(n+m)$ totally available objects, each ijth element of $\mathbf{N}(\mathbf{S}_0, h_1)$ is given by

$$\delta_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{o}_j \in N(\boldsymbol{o}_i, \mathbf{S}_0, h_1) \vee \ \boldsymbol{o}_i \in N(\boldsymbol{o}_j, \mathbf{S}_0, h_1), \\ 0 & \text{otherwise,} \end{cases} \tag{6}$$

where $N(\boldsymbol{o}, \mathbf{S}_0, h_1)$ denotes the set of the h_1-nearest neighbouring objects of $\boldsymbol{o}$ searched using the similarity values in $\mathbf{S}_0$. The disjunction $\vee$ defines nonzero indicators only for instances that are undirected neighbours of each other. Alternatively, the conjunction operation can be used to define nonzero indicators for mutual neighbouring objects. The sharpened proximity matrix is then given as $\mathbf{S}_0 \circ \mathbf{N}(\mathbf{S}_0, h_1)$, where $\circ$ denotes the Hadamard product of matrices.

Subsequently, in order to infer higher order (transitive) relationships that lie latent within the current proximity structure of $\mathbf{S}_0 \circ \mathbf{N}(\mathbf{S}_0, h_1)$, we further construct the following second-order similarity matrix using the cosine function

$$\mathbf{S}_1 = \cos\left(\mathbf{S}_0 \circ \mathbf{N}(\mathbf{S}_0, h_1), \mathbf{S}_0 \circ \mathbf{N}(\mathbf{S}_0, h_1)\right). \tag{7}$$

The rationale behind the latter enhancement, is that it has been demonstrated that second-order similarities can capture latent dataset information better than first-order similarities (Chen, 2002; Cribbin, 2011; Kovács, 2010). In the end, we apply again the nearest neighbour search to further sharpen the second-order proximity structure indicated by $\mathbf{S}_1$, leading to the final composite matrix of

$$\mathbf{S} = \mathbf{S}_1 \circ \mathbf{N}(\mathbf{S}_1, h_2). \tag{8}$$

where $\mathbf{N}(\mathbf{S}_1, h_2)$ is defined for $\mathbf{S}_1$ similarly to Eq. (6) for $\mathbf{S}_0$, using the undirected neighbours search and a different number h_2 of nearest neighbours.

In the final step of this stage, we use the composite $(n+m)\times(n+m)$ similarity matrix $\mathbf{S}$ of Eq. (8) to generate the l-dimensional embeddings. Various techniques can be used to achieve this, for example, multi-dimensional scaling (MDS) (Torgerson, 1952) and Laplacian eigenmaps (Belkin & Niyogi, 2003). Here, we apply the classical MDS framework to perform the task because it involves the least computational operation as compared to others. Let $\mathbf{Z}=[z_{ij}]$ denote the $(n+m)\times l$ embedding matrix for both the n documents and m candidate phrases, $\mathbf{Z}_d$ the $n\times l$ embedding matrix for documents only, and $\mathbf{Z}_p$ the $m\times l$ embedding matrix for candidate phrases only, with the three embedding matrices inter-related as $\mathbf{Z}^T=\left[\mathbf{Z}_d^T,\mathbf{Z}_p^T\right]$. Then, $\mathbf{Z}$ can be computed from $\mathbf{S}$ by minimising the following reconstruction error based on the Frobenius norm

$$\min_{\substack{\mathbf{Z}\in R^{(n+m)\times l}\\ \mathrm{rank}(\mathbf{Z}\mathbf{Z}^T)=l}} \|\mathbf{S}-\mathbf{Z}\mathbf{Z}^T\|_F^2. \tag{9}$$

The optimisation problem in Eq. (9) drives the inner products between the embeddings (rows of $\mathbf{Z}$) closer to the entries of $\mathbf{S}$, so that the sharpened high-order proximity structure represented by $\mathbf{S}$ is preserved in the inner product space of the embeddings.

Based on the Eckart-Young theorem for low-rank matrix approximation (Eckart & Young, 1936), the optimal solution of Eq. (9) can be readily obtained by

$$\mathbf{Z}=\mathbf{P}_l\mathbf{\Sigma}_l^{\frac{1}{2}}, \tag{10}$$

where $\mathbf{\Sigma}_l$ and $\mathbf{P}_l$ are the eigenvalue and eigenvector matrices of $\mathbf{S}$ corresponding to the largest l eigenvalues. The computed embedding matrix preserves not only the original proximity information, but also the higher-order relationships implied by the three input similarity matrices. Additionally, the feature-based representations of $\mathbf{Z}_d$ and $\mathbf{Z}_p$ are more flexible and compact, and more amenable to manipulation than $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$. Since the obtained common embedding space simultaneously encompasses heterogeneous document objects (rows of $\mathbf{Z}_d$) and candidate phrase objects (rows of $\mathbf{Z}_p$), we will refer to these embeddings as the *co-embeddings of documents and phrases*, and the corresponding l-dimensional space as the *co-embedded space*. In order to reduce the information loss to the minimum, we set the co-embedding dimensionality to $l=\mathrm{rank}[\mathbf{S}]$.

Stage 2: Exploration of Cluster Structure. The computed embedding matrix $\mathbf{Z}$ encodes enhanced link information derived from the three similarity matrices of $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$. In later stages, we work directly on these computed embeddings, treating documents and phrases as homogeneous objects. To group the documents, a standard clustering algorithm can naturally be applied with document embeddings $\mathbf{Z}_d$ used as the input features. In the current stage, we only allow single cluster membership so that one document can only belong to one cluster, in order to maintain good cluster separability.

Here, we choose the k-means algorithm because of its simplicity and effectiveness. Other algorithms (R. Xu, 2005) can also be used, e.g., mean shift clustering (Cheng, 1995), fuzzy clustering (Baraldi & Blonda, 1999a, 1999b), Gaussian mixture clustering (Ouyang, Welsh, & Georgopoulos, 2004), or spectral clustering (Luxburg, 2007). In this stage, an $n \times k$ binary cluster membership matrix $\mathbf{Y} = [y_{ij}]$ is finally acquired, where $y_{ij} = 1$ indicates that the ith document belongs to the jth cluster, whereas $y_{ij} = -1$ that it does not. The parameter k denotes the number of clusters and it can be controlled by the user. We assume that each document cluster suggests one potential topic of the studied documents.

Stage 3: Discovery of Topic Phrases. After determining the basic cluster structure in Stage 2, we work on cluster label generation. Instead of choosing directly a single phrase from the whole candidate pool that best describes the cluster topic in one go, it is more reasonable to conduct the search by gradually narrowing down the search range. Thus, the third stage of CEDL seeks a group of phrases from the total set $\{\mathfrak{P}_\mathrm{i}\}_{i=1}^{m}$ so that they are more relevant to the document cluster topics, and possess higher chance to be chosen as the final descriptive labels. For each document cluster, we define this subgroup of phrases as its *topic phrases*. The most intuitive way to determine the topic phrases is perhaps to conduct a clustering procedure on both documents and phrases in the co-embedded space, other than documents only as in the current setup of Stage 2. Then, within each obtained cluster, its member documents constitute the desired document cluster, while its member phrases constitutes the corresponding set of topic phrases. However, this co-clustering procedure creates two problems. First, the incorporation of candidate phrases into the clustering procedure could introduce noise and dependencies to the overall structure recovery of both types of objects, thus affecting the cluster separability within each type. Second, such a procedure provides the same type of membership for both types of objects. For example, it does not allow one set of objects to appear in multiple clusters and the other set in single ones. Considering the fact that one phrase can often be related to multiple topics and the current setup that one document can only belong to one cluster to produce good cluster separability, co-clustering of documents and phrases is not an appropriate choice. Therefore, in order to maintain good quality of document clusters and enable multi-memberships for phrases, we propose to process documents and phrases separately. First, documents are clustered on their own as in Stage 2. Then, a multi-class multi-label classification procedure is employed to ensure multi-membership for phrases, which is trained in the co-embedded space in order to predict whether a candidate phrase belongs to a targeted document cluster.

Specifically, the classification procedure begins with a set of training instances, which in our case are the n documents. Their $n \times k$ cluster membership indication matrix $\mathbf{Y}$, obtained in Stage 2, is used for the training as the known class labels, while the $n \times l$ document embeddings $\mathbf{Z}_d$, computed in Stage 1, are used as the input features. The entire procedure derives a prediction model that estimates a new $m \times k$ phrase

cluster membership matrix $\mathbf{Y}_p$ from the candidate phrases represented by the $m \times l$ phrase embedding matrix $\mathbf{Z}_p$.

If we define our prediction model to correspond to some mapping $F : \mathbb{R}^l \to \{-1, +1\}^k \in \mathfrak{F}$, defined from the feature space of phrase co-embeddings to the binary indication space of document clusters, then the training procedure of the classifier can be given by the optimisation

$$F^* = \arg\max_{F \in \mathfrak{F}} L(F, \mathbf{Z}_d, \mathbf{Y}), \tag{11}$$

where $L : \mathfrak{F} \times \mathbb{R}^{n \times l} \times \{-1, +1\}^{n \times k} \to \mathbb{R}$ is the objective function underlying the classification, designed to compare possible prediction models. Then, the final output in matrix form is given by

$$\mathbf{Y}_p = F^*(\mathbf{Z}_p). \tag{12}$$

To enable multi-class multi-label classification, one binary classifier is trained for each document cluster. We employ a set of k linear discriminant functions $\left\{ f_i(\boldsymbol{z}) = \text{sgn}\left(\boldsymbol{w}_i^T \boldsymbol{z} + b_i\right) \right\}_{i=1}^{k}$, one for each of the k document clusters, to despatch an one-against-all discrimination of the phrase embedding input. The function sgn$(\cdot)$ is the sign function, $\{\boldsymbol{w}_i\}_{i=1}^{k}$ are the l-dimensional weight vectors of the k linear decision boundaries, $\{b_i\}_{i=1}^{k}$ their biases (or threshold weights), while $\boldsymbol{z}$ denotes a phrase instance in the co-embedded space. To implement the objective function L of Eq. (11), Fisher criterion (Duda, Hart, & Stork, 2001) is adapted to the current setup and derive the following formulations (Mu, Jiang, Wang, & Goulermas, 2012) in order to compute the optimal weight vectors $\boldsymbol{w}_i$

$$\boldsymbol{w}_i = \left(\mathbf{Z}_d^T \left(\text{diag}\left(\mathbf{G} \times \mathbf{1}_{n \times 1} \right) - \mathbf{G} \right) \mathbf{Z}_d \right)^{-1} \mathbf{Z}_d^T \left(\text{diag}(\mathbf{G}) \circ \boldsymbol{y}_i \right),$$

and biases b_i

$$b_i = -\frac{1}{2} \boldsymbol{w}_i^T \mathbf{Z}_d^T \text{diag}(\mathbf{G}).$$

The class-based similarity matrix used above is defined as

$$\mathbf{G} = \mathbf{Y}_i \left(\mathbf{Y}_i^T \mathbf{Y}_i \right)^{-1} \mathbf{Y}_i^T, \tag{13}$$

and is based on the $n \times 2$ binary matrix $\mathbf{Y}_i = \frac{1}{2}[1 + \boldsymbol{y}_i, 1 - \boldsymbol{y}_i]$ with $\boldsymbol{y}_i$ denoting the ith column of the main document-cluster membership matrix $\mathbf{Y}$. The notation $\mathbf{1}_{m \times n}$ denotes the $m \times n$ matrix with all elements equal to 1. The individually computed weight vectors $\boldsymbol{w}_i$ can then compose an $l \times k$ weight matrix $\mathbf{W}$, with each $\boldsymbol{w}_i$ being its ith column, while the biases can compose a k-dimensional column vector $\boldsymbol{b} = [b_1, b_2 \ldots, b_k]^T$. In this way, the finally estimated $m \times k$ candidate phrase membership matrix $\mathbf{Y}_p$ is calculated as

$$\mathbf{Y}_p = \text{sgn}\left(\mathbf{Z}_p \mathbf{W} + \mathbf{1}_{m \times 1} \boldsymbol{b}^T \right), \tag{14}$$

where the sign function is applied element-wise. The interpretation of $\mathbf{Y}_p$, is that its ijth element indicates whether the ith candidate phrase is the topic phrase of the jth document cluster. This bears the significant advantage that the contents of different document clusters can be compactly summarised with their corresponding topic phrases, while the same candidate phrases can describe multiple topics.

The classification procedure employed here acts as a filtering step that takes advantage of the co-embedding information of documents and phrases to smoothen out the approximate document cluster information and form more robust relationships between documents and topic phrases. A final remark is that apart from the simple linear discriminant functions f_i optimised via the Fisher criterion, other classification algorithms could be used; examples include k-nearest neighbours (Cover & Hart, 1967), Bayesian classification (Domingos & pazzani, 1997), support vector machines (Cristianini & Shawe-Taylor, 2002) and neural networks (Haykin, c 1999). However, we recommend here the use of a simple and efficient classifier that does not require extra effort for determining complex model parameters. Using more powerful classification boundaries, in addition to being redundant for this task, could overfit the topic phrase recovery and produce false or extraneous cluster memberships.

Stage 4: Determination of Cluster Descriptions. The stages so far accomplish the summarisation of each document cluster with a set of topic phrases that provide a compact description of the contents of each cluster. In this stage, we attempt to extract the final cluster labels from these topic phrases. To assure the quality of the chosen cluster labels, we keep following the strategy of gradually narrowing down the search range of phrases that can be potentially utilised. Also, for further quality assurance, we evaluate the candidate phrases based on multiple measures assessing different properties.

Before further explanation, we first extend the definition of some previous quantities to cover cluster membership and make the exposition clearer. Specifically, we use $\left\{\mathfrak{P}_{\mathrm{j}}^{(\mathrm{i})}\right\}_{j=1}^{m_i}$ and $\left\{\mathfrak{D}_{\mathrm{j}}^{(\mathrm{i})}\right\}_{j=1}^{n_i}$ to denote the set of the m_i topic phrases and the set of the n_i documents, respectively, that belong to the ith cluster. We also define two row submatrices of $\mathbf{Z}^T = \left[\mathbf{Z}_d^T, \mathbf{Z}_p^T\right]$ to correspond to these two sets; namely, the $n_i \times l$ embedding matrix $\mathbf{Z}_d^{(i)}$ for the n_i documents, and the $m_i \times l$ embedding matrix $\mathbf{Z}_p^{(i)}$ for the m_i topic phrases. We similarly introduce for the m_i topic phrases of the ith cluster, the $m_i \times k$ per-cluster unsigned phrase membership matrix $\widehat{\mathbf{Y}}_p^{(i)}$. This is a row submatrix for the overall $m \times k$ matrix $\widehat{\mathbf{Y}}_p = \mathbf{Z}_p\mathbf{W} + \mathbf{1}_{m\times 1}\boldsymbol{b}^T$, which is defined as $\mathbf{Y}_p$ in Eq. (14) but without the discretisation imposed by the $\mathrm{sgn}(\cdot)$ function.

The first step of this stage is to compute the following m_i-length score vector for the m_i topic phrases of each ith cluster

$$\mathrm{score}_1^{(i)} = \cos\left(\mathbf{Z}_p^{(i)}, \mathbf{Z}_d^{(i)}\right)\mathbf{1}_{n_i\times 1}. \tag{15}$$

Each jth element of the above vector evaluates the overall closeness between the jth topic phrase (jth row

of $\mathbf{Z}_p^{(i)}$) and all of the n_i member documents (rows of $\mathbf{Z}_d^{(i)}$) of the cluster the phrase appears in. This closeness is simply estimated as the sum of the elements of the jth row of the $m_i \times n_i$ similarity matrix $\cos(\mathbf{Z}_p^{(i)}, \mathbf{Z}_d^{(i)})$. The crucial advantage of this score is that, because it is computed through the second-order neighbour-based similarities in the co-embedded space (as described in Stage 1), it takes into account the combined semantic and syntactic connections between documents and phrases.

Although the above score can be used directly for our final purpose, in order to increase its robustness we combine it with a secondary score vector, defined for each ith document cluster as $\text{score}_2^{(i)}$. This score is taken to be the ith column of $\widehat{\mathbf{Y}}_p^{(i)}$ that contains the continuous linear prediction values from the classification model introduced in Stage 3. Each jth element of $\text{score}_2^{(i)}$ gives the signed degree of confidence that the jth topic phrase of the ith cluster is assigned to that cluster. The benefit of this score is that it reveals the latent connections between candidate phrases and document clusters, as it uses the core information extrapolated in the discriminant learning stage to perform the phrase to cluster allocation. This information is effectively used here again to amplify or attenuate the values of $\text{score}_1^{(i)}$.

The two scores can then be scaled within the range $[0, 1]$ and combined to a single vector defined for each ith cluster as

$$\text{score}^{(i)} = \text{score}_1^{(i)} \circ \left(\text{score}_2^{(i)}\right)^a. \tag{16}$$

The user-defined parameter $a \geq 0$ modifies the effect of the weight $\text{score}_2^{(i)}$. This score has the potential to include phrases that are semantically connected to the topic in a latent way, but may also introduce more noisy phrases compared to $\text{score}_1^{(i)}$. Therefore, we use the parameter a to control the degree that $\text{score}_2^{(i)}$ influences the phrase selection procedure. When $a = 0$, $\text{score}_2^{(i)}$ is not considered and only the primary score $\text{score}_1^{(i)}$ is in effect. In the present work, both scores are considered equally important by setting $a = 1$.

The ranking procedure based on $\text{score}^{(i)}$, selects the $m_i^* \leq m_i$ top phrases possessing the highest scores from the total of m_i topic phrases in each ith cluster. We extract the cluster labels only from these top m_i^* phrases. A typical pattern in human languages is that phrases that are used to describe the same topic share certain common words. Therefore, we extract a set of common words from the top m_i^* phrases, which are defined as uni-gram words that appear in at least two of those m_i^* phrases. Only topic phrases that contain more than two such common words are allowed to be used as cluster labels; these are referred to as the *final candidates*. Finally, we rank these final candidates based on their corresponding $\text{score}_1^{(i)}$ values (as this score is the primary one), and choose the one with the highest value as the *final cluster label*. In this procedure, we recommend setting $m_i^* = \min(10, m_i)$, as we found it experimentally sufficient to summarise the topic of a document cluster with approximately ten phrases.

The proposed label induction strategy gradually reduces the content of each document cluster in terms of the sets of phrases potentially utilised. For example, the total of m phrases was reduced, for the

ith cluster, to $m_i < m$ topic phrases, which in turn led to $m_i^* \leq m_i$ top phrases with the aid of ranking. Then, an even smaller set of final candidates was obtained using common words, and finally, a single cluster label was selected (see the example in Figure 2). This gradual multi-stage process of narrowing down the residual phrase search range contributes to the stability of the algorithm and the reduction of the negative effects from noisy candidates, and hence, enables us to obtain more meaningful and more accurate cluster descriptions.

Stage 5: Verification of Document Clusters. From Stage 2, we obtain an initial cluster structure of the studied documents where only single memberships are allowed in order to maintain good cluster separability. However, many text mining and information retrieval services demand overlapping clusters of documents, because of the existence of documents relevant to multiple topics. Thus, the main function of this final stage is to trim the k document clusters obtained in Stage 2 to allow overlappings between clusters.

First, we combine document clusters that possess identical, similar or synonymous cluster labels. Subsequently, we modify the content of each document cluster by adding relevant documents from other clusters. Using our co-embedding formulation, this is straightforward to achieve as follows. Letting $\boldsymbol{z}_p^{(i)}$ denote the l-dimensional row co-embedding vector of the cluster label of the ith cluster, we compute the following relevance threshold

$$c^{(i)} = \max\left(\cos\left(\mathbf{Z}_d^{(i)}, \boldsymbol{z}_p^{(i)}\right)\right). \tag{17}$$

This is the maximum value of the similarity between the cluster label and all the n_i documents in that cluster and reflects the lowest bound for inter-cluster document membership. If $\boldsymbol{z}_d^{(j)}$ denotes the co-embedding of some document from any other cluster of index $j \neq i$, we consider that document relevant to the ith cluster if and only if $\cos(\boldsymbol{z}_d^{(j)}, \boldsymbol{z}_p^{(i)}) \geq c^{(i)}$. In such case, we assign it to also belong to the ith cluster (in addition to its original jth cluster allocation).

Another step in this stage is to remove clusters that do not contain enough documents (e.g., less than three) as those can be considered trivial. The final polishing step here needed to improve the internal consistency of each document cluster, is to reallocate those documents that are within the bottom 10% of the least relevant documents to their corresponding cluster label. As before, this is done using the relevance values evaluated with the cosine similarities between documents and cluster label co-embeddings, so that those dubious documents are re-allocated to other more relevant clusters with higher similarity.

In summary, this stage includes the three operations of firstly, merging highly related clusters, secondly, allowing multi-membership for highly related inter-cluster documents, and finally, reallocating the least related documents from each cluster. Differently from DKM that completely abandons the original cluster structure obtained by k-means, the proposed trimming procedure relies on the cluster structure

obtained in Stage 2, while allowing modifications to outlier and ambiguous documents. This is because such structure is derived from a carefully designed composite similarity matrix, which stores more advanced link information than those frequency-based links as used by DKM. The composite similarities are also used to drive the trimming procedure by computing the cosine score in the co-embedded space.

Experimental Results and Analysis

Databases

In order to assess the performance of the proposed descriptive clustering algorithm CEDL, we use documents from three different databases. These include a large clinical trial collection provided by the UK's National Centre for Text Mining (NaCTeM) (Korkontzelos, Mu, & Ananiadou, 2012), the "Reuters-21578 Text Categorisation Test Collection" containing articles taken from the Reuters newswire (Lewis, 1997), and a collection of systematic reviews of research evidence conducted by the Evidence for Policy and Practice Information and Co-ordinating Centre (EPPI-Centre) (Ananiadou, Okazaki, Procter, Rea, & Thomas, 2009; Thomas, McNaught, & Ananiadou, 2011). We refer to these three databases as CT, Reuters and EEP, respectively.

These databases include documents of varying length collected from different fields. For instance, the average length of a clinical trial document is around a few hundreds of words, while the Reuters articles vary from a few hundred to several thousands of words. The review articles from the EEP database are quite lengthy with approximate sizes of 250KB plain text document files. To process these documents, text in their title and body was part-of-speech tagged and lemmatised using the GeniaTagger (Tsuruoka et al., 2005) and finally lower-cased. Lemmas that occur in a stop-list for general English were removed.

The documents in the Reuters collection are originally annotated with zero or more from a set of 135 topic tags, amongst which the ten largest topics are "earn", "acq", "crude", "trade", "money-fx", "interest", "ship", "sugar", "money-supply" and "coffee". The documents in the EEP collection were manually assigned zero or more topic tags from a topic hierarchy created by the EPPI-Centre, where the most popular topic is "management, governance and finance". From the CT collection, we retrieved nine sets of documents based on nine different queries, including "asthma", "breast cancer", "lung cancer", "prostate cancer", "cardiovascular", "HIV", "leukemia", "depression" and "schizophrenia". These queries are used as the topic tags for the retrieved documents. Finally, all the documents contained two types of information, that is input text and topic tags.

Experimental Setup

One reliable and convenient way for assessing the cluster quality is to compare the clustering results against a set of predefined classes that are also known as the ground truth partitioning of the documents. In our case, the original topic tags of the documents can be used as the reference set of this ground truth. To allow for an informative and quantitative analysis, specific numerical measures can be used. Here, we use f-measure, cluster purity, entropy and cluster contamination to evaluate the difference between the obtained clusters and the ground truth topics (Weiss, 2006). Letting $\{U_i\}_{i=1}^{g}$ denote g sets of documents each corresponding to a document cluster obtained by the algorithm, while $\{A_i\}_{i=1}^{h}$ denote h sets of documents each corresponding to a ground truth topic, the f-measure is computed by

$$F_\beta = \sum_{i=1}^{h} \frac{|A_i|}{n} \max_{j=1}^{g} \frac{(1+\beta^2)\text{Precision}_{ij}\text{Recall}_{ij}}{\beta^2\text{Precision}_{ij} + \text{Recall}_{ij}}, \tag{18}$$

where $|\cdot|$ denote the number of documents in the input set, and the precision and recall are defined as

$$\text{Precision}_{ij} = \frac{|A_i \cap U_j|}{|U_j|}, \ \text{Recall}_{ij} = \frac{|A_i \cap U_j|}{|A_j|}. \tag{19}$$

The parameter β is used to control the weights of precision and recall, set according to the need of the given application, e.g., an $F_{0.05}$ score gives more emphasis on precision over recall, and an F_1 score puts equal weight on them leading to the traditional f-score. The purity of the jth cluster is defined by

$$P_j = \max_{i=1}^{h} \frac{|A_i \cap U_j|}{|U_j|}. \tag{20}$$

The overall purity is computed as the average of P_j, given as $P = \frac{1}{g}\sum_{j=1}^{g} P_j$. The entropy is defined as

$$E = -\sum_{i=1}^{h} \frac{|A_i|}{n \log g} \sum_{j=1}^{g} \frac{|A_i \cap U_j|}{|U_j|} \log\left(\frac{|A_i \cap U_j|}{|U_j|}\right). \tag{21}$$

It can be seen from Eqs. (18), (20) and (21) that these three measures are linked to each other in that they all rely on precisions between the computed clusters and their dominating ground truth topics, which encourages documents from the same ground truth topic to be grouped into the same cluster. Additionally, the f-measure also relies on the quantity of recall, which prevents a complete ground truth topic from being split to different clusters. The contamination measure is defined in a slightly different way, which computes a degree of closeness from the computed clusters to the worst grouping situation, given for the jth cluster as follows

$$C_j = \frac{\sum_{i_1=2}^{h}\sum_{i_2=1}^{i_1-1} |A_{i_1} \cap U_j||A_{i_2} \cap U_j|}{\sum_{i_1=2}^{h}\sum_{i_2=1}^{i_1-1} h_{i_1 j} h_{i_2 j}}, \tag{22}$$

where the quantity h_{ij} is defined as

$$h_{ij} = \begin{cases} \left\lfloor \frac{\sum_{t=1}^{h} |A_t \cap U_j|}{g} \right\rfloor + 1, & \text{if } i < \sum_{t=1}^{h} |A_t \cap U_j| \mod g, \\ \left\lfloor \frac{\sum_{t=1}^{h} |A_t \cap U_j|}{g} \right\rfloor, & \text{otherwise,} \end{cases} \tag{23}$$

where $\lfloor \cdot \rfloor$ denotes the largest integer less than or equal to the input number. According to their definitions, all these four measures range from 0 to 1. Higher values of f-measure and cluster purity, and lower values of entropy and cluster contamination indicate better cluster quality.

For evaluation purpose we created six document subsets from the three databases, which are referred as CT1, CT2, Reuters1, Reuters2, Reuters3 and EEP1, respectively. The ground-truth topics contained within each document subset and the number of documents included for each topic are listed in Table 2. We include topics containing almost equal numbers of documents (e.g., CT1 and EEP), as well as topics containing quite different numbers of documents (e.g., CT2). To analyse and verify the class discriminability between the ground truth topics, we formulated a separate classification-based evaluation for each document subset by using its topic tags as the document categories. The tf-idf values of the uni-gram words were used as the input features. Two classifiers were used to observe the classification performance; the linear support vector machine (LSVM) (Cristianini & Shawe-Taylor, 2002) and the linear Fisher's discriminant analysis (LFDA) combined with the cosine-based relation features (Mu, Goulermas, Tsujii, & Ananiadou, 2012; Mu, Jiang, et al., 2012). Ten permutations of 5-fold cross validation (CV) were executed to determine the ground truth quality for each document set. The averaged CV classification accuracies of LFDA and LSVM are recorded in the last row of Table 2. Ideally, document sets with higher classification accuracies contain more distinguishable document topics, e.g., CT1 has more distinct topics compared to EEP1.

We compare the proposed CEDL with two other state-of-the-art descriptive clustering methods of DKM and Lingo. To achieve an objective comparison under the same environment, we let the three methods extract cluster labels from the same pool of candidate phrases. In order to keep balanced sizes between the three similarity matrices $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$ and avoid unexpected bias, we maintain equal document and candidate phrase sizes by including only the top $m = n$ terms extracted by Termine as the candidates. To set the two parameters h_1 and h_2 of CEDL for nearest neighbour search, we test neighbour numbers ranging from 5% to 90% of the total object number $n + m$ with a 10% step size. Experiments showed that as the neighbour number increases, the clustering performance first gets better, and then stabilises without obvious performance change after 15%. To maintain a good sparsity of the composite matrix, the 15% level was used by CEDL so that $h_1 = h_2 = \lfloor 15\% \times (n + m) \rfloor$, where $\lfloor \cdot \rfloor$ denotes the floor function. To set the snippet assignment threshold parameter for Lingo, we test the recommended range of $[0.15, 0.3]$ obtained through empirical verifications by Osiński and Weiss (2005), and observe that the clustering performance does not vary obviously within this range, and around the middle point of the range 0.2 was used as the threshold. According to the design, the number of document clusters is directly controlled by the integer number k of k-means clustering for CEDL and DKM, and via the number of

remaining singular vectors k for Lingo. We set k as the same value for the three methods so that they can produce similar numbers of document clusters; however, due to the cluster post-processing all methods perform the final number of clusters may be less than k. All experiments reported here were conducted using MATLAB R2011a, running on a 3.4 GHz Intel i7 CPU with 8 GB memory machine, under Mac OS X 10.7.4.

Results and Analysis

Experiment 1. We first examine whether and how well the three competing methods CEDL, DKM and Lingo can discover the ground truth topics as listed in Table 2. To achieve this, we initially set the cluster parameter k of the three methods equal to the total number of topics contained in each document subset, which is 5 for CT1, 8 for CT2, 3 for Reuters1, 6 for Reuters2, 10 for Reuters3 and 2 for EEP1. Table 3 compares the clustering performance with the aforementioned four quality measures of f-measure (F_1 score), cluster purity, entropy and cluster contamination.

From Table 3, it can be seen that CEDL provides the best performance for all the document sets yielding the highest cluster purity and f-measure scores, as well as the lowest cluster contamination and entropy values. For many datasets, CEDL performs significantly better than DKM. DKM performs comparable to CEDL for only the CT1 dataset, which is easier to be clustered than the other sets, as it contains more distinguishable document topics indicated by its higher LFDA/LSVM classification accuracies as shown in Table 2. As for Lingo, it exhibits worse clustering performance than both CEDL and DKM for all the document sets in terms of the four employed measures. Considering the fact that Lingo is designed for clustering short documents, the lengths of the CT and Reuters documents may exceed its processing capability. We did not apply Lingo to EEP1 (see N/A in Table 3), because the EEP1 documents are much longer than the CT and Reuters ones and lead to worse performance and very long processing times.

We can investigate the better clustering performance of CEDL over DKM and Lingo, as follows. As explained in previous sections, DKM discovers an initial structure of document clusters using k-means clustering, but only utilises this discovered structure for label generation and re-assigns new member documents for each cluster based solely on the occurrences of the generated label (including its distorted or synonymous versions) in the documents. As for Lingo, it first generates cluster labels according to a set of assumed similarities between candidate phrases and latent topics computed based on matrix factorisation. Then, it discovers member documents for each cluster label using a similar procedure to the one of DKM by examining common words shared between the label phrases and documents. The motivation behind

such determination procedure of the cluster member documents used by DKM and Lingo is the expectation for the user to be able to tell which elements of this description label can be found in the corresponding cluster's documents (Weiss, 2006). Although this creates a clear and evident relationship between a cluster label and its member documents which is beneficial, in reality the connections between documents and cluster topics are not necessarily based only on the occurrences of a certain phrase or the number of common words shared between a phrase and a document. Such operation has high chances of ignoring latent semantic connections between documents and topics, and can thus produce inaccurate document clusters. Apart from this, the validity of the similarity assumption used by Lingo may not be reliable for long documents. Consequently, its generated cluster labels may represent an inappropriate cluster structure from the beginning, which thus may lead to even worse clustering performance. This is partly evidenced by its poorer performance than DKM in Table 3. Differently, CEDL does not follow such label-driven determination procedure of cluster member documents. It first constructs an initial cluster structure and then refines such structure other than abandoning it. To obtain good initial cluster structure, CEDL carefully designs a composite similarity matrix based on second-order local neighbourhood links so that it not only captures the latent semantic information of the document contents better, but also provides sharper proximity structure with reduced noise. This offers the k-means clustering algorithm higher quality of input information. Because of these reasons, the document clusters generated by CEDL are more accurate than the ones obtained by DKM and Lingo, as presented in Table 3.

As mentioned above, an important component in the design of CEDL is the composite similarity matrix $\mathbf{S}$ used by CEDL for co-embedding generation. This directly affects the quality of the generated document clusters and their descriptive labels. Here, we demonstrate its effectiveness by visualising this proposed matrix $\mathbf{S}$ based on second-order neighbour-based similarities in Figures 3(a), 3(b) and 3(c) using the CT1, Reuters2 and EEP1 document sets, respectively. We compare it directly with the first-order matrix $\mathbf{S}_0$ in Eq. (5), which is a simple combination of the three input similarity matrices of $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$ without any processing, in Figures 3(d), 3(e) and 3(f) for the corresponding document sets. It is apparent that the proposed similarity matrix can reveal much more distinct block patterns than $\mathbf{S}_0$, indicating clearer and more informative topic structures. Thus, the embeddings generated from $\mathbf{S}$ are more promising and effective and, as our experimentations confirmed, can subsequently lead to much higher document cluster quality.

Experiment 2. In this experiment, we increase the value of k to observe how well the three methods can discover more and finer topics from the input documents. First, we initially set the cluster parameter k of the three methods as 18 for CT1, 15 for Reuters2, and 12 for EEP1, which are much higher than the numbers of ground truth topics of these three document sets that are 5, 6 and 2, respectively. The

obtained document clusters and their corresponding descriptive labels are displayed in Figures 4(a), 4(b) and 4(c) for CEDL, DKM and Lingo, respectively, with the CT1 document set, Figures 5(a), 5(b) and 5(c) for CEDL, DKM and Lingo, respectively, with the R2 document set, and Figure 6 for CEDL and DKM with the EEP1 document set. In these figures, each bar corresponds to a single document cluster. The descriptive label for summarising the content of each cluster is displayed on the horizontal axis under its corresponding bar, along with its cluster purity value. Each colour encodes the original topic tag of the documents. The vertical axis measures the number of documents in the clusters. Ideally, a coherent cluster should include only documents with the same topic tag, that is, correspond to a bar of one solid colour with its cluster purity reaching the upper bound of 100%. Second, in addition to the comparison with fixed values of k, we also compare variations of the clustering performance of CEDL, DKM and Lingo with increasing values of k in Figures 4(d) and 5(d), with the CT1 and Reuters2 document sets, respectively, in terms of the f-measure and cluster contamination. The experiment examines how well a ground truth topic can be split into small clusters, and the more and finer topics it finds, the lower the recall becomes. So, in order to dilute the effect of recall so that it does not interrupt the observation, a small value of $\beta = 0.05$ is used to compute the f-measure.

As can be seen from Figures 4(a) to 4(c), 5(a) to 5(c) and 6, CEDL produces purer clusters than DKM and Lingo, while Lingo generates the least pure clusters. The plots in Figures 4(d) and 5(d) show that, no matter how the parameter k for controlling the cluster number changes, CEDL consistently generates document clusters possessing the highest f-measure and least contamination. As discussed in previous sections, the superior clustering performance of CEDL is obtained through its multi-stage design that first constructs an accurate initial cluster structure and then enhances it further.

Judging from the descriptive labels as shown in Figures 4 to 6, CEDL discovers more reasonable subtopics than DKM and Lingo. For instance, for the CT1 document set, CEDL groups those documents related to the "cardiovascular" topic to the three subtopics of "coronary artery disease", "blood pressure measurement" and "health care use", whereas DKM finds only two main subtopics of "cardiovascular disease risk" and "cardiovascular risk marker" that are very similar to each other. As for Lingo, the first five bars in Figure 4(c) all contain quite a lot of documents from the "cardiovascular" topic, but their descriptive labels indicate subtopics spanning other topic tags as well, e.g., "lung cancer patient" and "breast cancer risk". For the Reuters2 document set, examining the descriptive labels generated for documents from the "money-fx" topic, it shows that CEDL discovers three main subtopics of "currency exchange rate", "mln stg late assistance" and "new zealand banking group" from them. DKM discovers the two main subtopics of "common currency exchange rate" and "mln stg bank", and Lingo discovers only one main subtopic of "currency exchange rate". For the EEP1 document set, both CEDL and DKM discover

six subtopics for articles related to “work based learning” and six subtopics for those on “primary education”. However, the two subtopics of “primary school child” and “primary school pupil” found by DKM seem too similar to each other. Finally, we also observe that, although the three methods result in quite different clusters and descriptive labels, there exist commonalities between the subtopics discovered by them. For example, from CT1 documents related to “lung cancer”, CEDL, DKM and Lingo all discover subtopics related to “nonsmall cell lung cancer”. Both CEDL and DKM discover subtopics related to “small cell lung cancer”. One common subtopic related to “currency exchange rate” is found by all three methods from the Reuters2 documents.

We can analyse why CEDL is able to discover more reasonable subtopics than DKM and Lingo, as follows. Both CEDL and DKM share the same strategy of generating topic descriptions based on an initial cluster structure produced by k-means clustering. However, one main difference is the similarity information that drives the k-means algorithm. Specifically, DKM clusters the documents employing simple cosine similarities between documents based on word occurrences, while CEDL works on a more sophisticated composite similarity matrix representing a sharper and noise reduced proximity structure between documents. Thus, CEDL has higher chances of producing a better initial cluster structure than DKM. Another difference is the label generation procedure given the derived initial cluster structure. Specifically, DKM directly finds the best matched label phrase according to one single query score, while CEDL follows a more sophisticated procedure by gradually narrowing down the search range of the best label, e.g., from the whole candidate pool to a small set of topic phrases, then to a smaller set of final candidates, and finally to the single selected label, based on multiple assessment scores characterising different properties of the phrases. Compared to CEDL and DKM, Lingo works upon a very different routine. As explained in previous sections, it generates cluster labels by assuming an SVD based similarity score between phrases and latent topics, which may become unreliable for long documents, thus leading to the least reasonable topic descriptions among the three methods.

In Table 4, we display 15 subtopics discovered by CEDL and DKM, from 600 EEP documents that are related to “management, governance and finance”, and in Tables 5 and 6 we display 6 subtopics discovered by CEDL, DKM and Lingo from the top 200 CT documents retrieved from each of the nine queries. It can be observed from Tables 5 and 6 that the sizes of the clusters generated by CEDL are more uniformly distributed than those generated by DKM and Lingo. For example, CEDL groups the 200 clinical trials, retrieved via the “HIV” query, to six clusters with their sizes varying between 19 to 54. However, for the same set of clinical trials, the largest cluster obtained by DKM includes as many as 144 documents, whereas the smallest cluster merely 6 documents. Lingo creates even more dramatic discrepancies between cluster sizes, e.g., from 3 to 200. Usually, more uniform distribution of cluster sizes

is preferred by the users. It is not very typical for the user to accept very large subtopics including almost all the targeted documents, while at the same time be comfortable with very small subtopics of very few documents. We also asked domain experts from the EPPI-Centre for their degree of satisfaction on the subtopics obtained for the EEP documents, and physicians to examine whether the subtopics obtained for the CT documents are reasonable. Based on their subjective and qualitative feedback, the subtopics discovered by CEDL were deemed more sensible than DKM and Lingo.

Other Tests. CEDL is a sophisticated text analysis framework, including multiple processing stages such as co-embedding, clustering, classification and ranking. In addition to the clustering and ranking operations that constitute the main processing procedures of DKM, CEDL also requires the neighbour search and matrix decomposition operations for computing co-embeddings, and the classification operation for determining topic phrases. Computational cost of these additional operations are dominated by the number of documents. It is worth to mention that, although CEDL requires additional effort on computing co-embeddings, it can effectively save computational cost in the later clustering stage. This is because CEDL improves the quality of its similarity computation between documents, especially when processing documents containing hidden cohort structure that cannot be directly captured by simple similarity measures based on word co-occurrences, e.g., the cosine similarity measure used by DKM. Such an improvement can lead to faster convergence in the clustering stage. As for Lingo, it involves two main operations of matrix decomposition by SVD and ranking, of which the cost is dominated by both document number and the total number of uni-gram words (d) used to characterise the documents; the longer the documents are, the higher the value of d.

To investigate how running time of these three methods changes with increasing input data sizes, e.g., increasing number and length of documents, we conduct two sets of tests using the Reuters document set with medium length documents and the EEP set containing very lengthy documents. In Figure 7(a), the running time variations of the three methods are compared using from 150 to over 6000 Reuters documents, of which the total number of words used to characterise the documents increases from around 4 to 46 thousands. It can be seen from the figure that the computational costs of the three methods are comparable when processing comparatively small number of documents. As the input document number increases, the computational cost of Lingo increases much more rapidly than CEDL and DKM. For CEDL, it takes around seven minutes to process over 6000 documents containing over 46 thousands of words, which is within the scalable range. In Figure 7(b), comparisons are conducted using 331 EEP documents characterised by increasing number of words from 8 to 218 thousands. A similar observation is made that the computational cost of Lingo increases much steeper than CEDL and DKM. Also, CEDL is around nine times faster than DKM. As previously explained, this is because CEDL improves the computation of

between-document similarities that drive the k-means clustering, so that they better highlight the cohort structure hidden in the EEP documents, thus leading to much less iterations for the clustering algorithm to converge, and consequently, less running time than DKM. The advantage of CEDL being less sensitive to document length makes it suitable for analysing lengthy document containing complex information, as opposed to Lingo that is developed for processing short text snippets.

Finally, we provide an example of how CEDL can be extended to create hierarchical cluster structures. Sometimes, in a document browsing service this is more practical than using flat clusters. To achieve hierarchical clustering, the input preparation and co-embedding generation stage of CEDL remain the same, while in the cluster exploration stage, the k-means clustering is simply replaced by a hierarchical clustering algorithm, e.g., hierarchical k-means (Arai & Ridho, 2007) as used in our experiment. The obtained hierarchical cluster structure is then a dendrogram. For each of those node clusters having the root as its parent, the whole candidate phrase pool is used as its candidate phrases, while for the remaining node clusters (excluding the root one) all the topic phrases determined for their parent node are used as the candidate phrases, but excluding the chosen phrase as the descriptive label for the parent node. Then, we follow exactly the same procedures as in Stages 3 to 5 to determine the descriptive label for each node cluster. The extended CEDL is applied to group clinical trial documents retrieved based on the four queries of "asthma", "prostate cancer", "breast cancer" and "cardiovascular". The obtained hierarchical document clusters are illustrated in Figure 8. Four clusters of "mild persistent ashma", "permanent prostate cancer", "breast cancer treatment" and "cardiovascular disease" are discovered, which match the four ground truth topics. Under these clusters, sub-clusters are discovered representing relevant sub-topics, which are shown in different colours of tree structures in Figure 8. We compute the average cluster purity and f-measure for the obtained clusters, which are both over 0.96.

Conclusion

We have proposed a novel descriptive clustering framework CEDL for grouping semantically interrelated documents and summarising the content of the obtained document clusters with compact and comprehensive labels. Different from many existing methods, the proposed CEDL treats descriptive clustering as a heterogeneous data analysis problem, jointly analysing documents and phrases by effectively combining connections within and between the two object sets. Such strategy effectively improves the quality of document clusters and cluster labels simultaneously. Technical contributions of this work are summarised as follows.

First, a unique pre-processing scheme for constructing the three input similarity matrices $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$ has been proposed. Its ability to take into account both semantic and syntactic connections

between and within the targeted documents and candidate phrases is very advantageous to the quality of the resulting document clusters and their labels. The link information contained explicitly and implicitly in these matrices dominates the later clustering and label generation procedures. In addition to constructing them based on VSM and tf-idf weighting ($\mathbf{X}_{dw}$, $\mathbf{X}_{pw}$ and $\mathbf{X}_{dp}$), the framework is made adequately generic to accommodate $\mathbf{S}_{dd}$, $\mathbf{S}_{pp}$ and $\mathbf{S}_{dp}$ constructed through alternative ways and information resources. This offers the users the opportunity to experiment and define their own similarity data structures that could suit better a particular domain or corpus.

Second, a very effective descriptive clustering algorithm has been introduced through CEDL to boost the grouping and summarisation performance. The three similarity matrices are effectively blended so that not only the direct links within, but also the latent interactions across them are taken into account. The algorithm advances by deriving a co-embedded space from the blended similarities stored in matrix $\mathbf{S}$. Specifically, by preserving the composite neighbour-based second-order similarity information in $\mathbf{S}$, this critical component commonly maps both documents and candidate phrases into a co-embedded space (Stage 1). The resulting co-embeddings are used as an alternative representation of the documents and phrases, which serves as a convenient platform for simultaneously manipulating heterogeneous objects, and meanwhile establishing connections between the clustering and label generation procedures to achieve a more reliable output. Then, each document is assigned into a single cluster by conducting simple k-means clustering in the co-embedded space (Stage 2), which smoothens out the approximate document cluster information. Subsequently, a gradual multi-stage process of narrowing down the residual phrase search range of the optimal descriptive labels is performed to assure the quality of descriptive labels. It first generates topic phrases through a multi-class multi-label classification procedure to facilitate the multi-topic nature of phrases, reduce noise interference between object types, and form more robust relationships between documents and topic phrases (Stage 3). It then discovers a small set of top phrases from the selected topic phrases according to a ranking procedure based on multiple assessing measures, and finally selects the descriptive label from the top phrases by combining the assessing measures with examination of the common words (Stage 4). This gradual search process contributes to the robustness and stability of the algorithm and reduces the negative effects from noisy candidates. In the end, a trimming procedure is employed in order to further improve the cluster quality and introduce overlapping between document clusters to accommodate practical demands (Stage 5). This prevents significant reduction of cluster separability without breaking the main cluster structure derived from the previous clustering stage.

The effectiveness, accuracy and robustness of the proposed method was demonstrated via quantitative and qualitative evaluations and comparisons with two state-of-the-art competitors using document databases from different application fields. It was shown to produce not only higher quality

document clusters evaluated by various cluster performance measures, such as f-measure, cluster purity, entropy and contamination, but also more reasonable sub-topics judging from the descriptive labels. Also, compared to the competing methods, the improved performance of CEDL is not traded by sacrificing its computational cost.

However, there are still limitations in the proposed system. Given the rapid growth of electronic literature and media information and the demands of online text mining services for faster analysis of large amounts of text data, future work will focus on several improvements of CEDL. First, the system can be enhanced with more scalable data processing capabilities to accommodate large-scale corpora. Second, document content summarisation can be improved by allowing multi-phrase descriptions and by taking into account distributional semantics between words to better capture the semantic topics of the document sets. Third, dynamic and incremental learning ability could be incorporated to process growing text information more effectively. Also, the system could benefit from active self-improving capabilities through feedback information from the users, e.g., user-specified document clusters and user-preferred descriptive label phrases.

Acknowledgment

This research was conducted within the Digging into Data project “Integrated Social History Environment for Research" (ISHER) funded by JISC/ESRC/AHRC.

References

Ananiadou, S., Okazaki, N., Procter, R., Rea, B., & Thomas, J. (2009). Supporting systematic reviews using text mining. *Social Science Computer Review*, *27*(4), 509-523.

Arai, K., & Ridho, B. A. (2007). Hierarchical k-means: an algorithm for centroids initialization for k-means. *Reports of the Faculty of Science and Engineering*, *36*(1), 25-31.

Baeza-Yates, R. A., & Ribeiro-Neto, B. A. (Eds.). (1999). *Modern information retrieval*. ACM Press, Addison-Wesley.

Baraldi, A., & Blonda, P. (1999a). A survey of fuzzy clustering algorithms for pattern recognition-part i. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *2*(6), 778-785.

Baraldi, A., & Blonda, P. (1999b). A survey of fuzzy clustering algorithms for pattern recognition-part ii. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, *2*(6), 786-801.

Beil, F., Ester, M., & Xu, X. (2002). Frequent term-based text clustering. In *Proceedings of the eighth acm sigkdd international conference on knowledge discovery and data mining.*

Belkin, M., & Niyogi, P. (2003). Laplacian eigenmaps for dimensionality reduction and data representation. *Neural Computation*, *15*(6), 1373-1396.

Bengio, Y., Courville, A., & Vincent, P. (2013). Representation learning: a review and new perspectives. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *35*(8), 1798-1828.

Bengio, Y., Ducharme, R., Vincent, P., & Jauvin, C. (2003). A neural probabilistic language model. *Journal of Machine Learning Research*, *3*, 1137-1155.

Bharambe, U., & Kale, A. (2011). Landscape of web search results clustering algorithms. *Communications in Computer and Information Science*, *125*(Part 1), 95-107.

Carmel, D., Roitman, H., & Zwerdling, N. (2009). Enhancing cluster labeling using wikipedia. In *Proceedings of the 32nd international acm sigir conference on research and development in information retrieval* (p. 139-146).

Carpineto, C., Osiński, S., Romano, G., & Weiss, D. (2009). A survey of Web clustering engines. *ACM Computing Surveys*, *41*(3), 1-38.

Chang, Y. F., & Chen, C.-M. (2011). Classification and visualization of the social science network by the minimum span clustering method. *Journal of the American Society for Information Science and Technology*, *62*(12), 2404-2413.

Chen, C. H. (2002). Generalized association plots: Information visualization via iteratively generated correlation matrices. *Statistica Sinica*, *12*(1), 7-29.

Cheng, Y. (1995). Mean shift, mode seeking, and clustering. *IEEE Transactions on Pattern Analysis and*

Machine Intelligence, *17*(8), 790-799.

Cover, T., & Hart, P. (1967). Nearest neighbor pattern classification. *IEEE Transactions on Information Theory*, *13*(1), 21-27.

Cribbin, T. (2011). Discovering latent topical structure by second-order similarity analysis. *Journal of the American Society for Information Science and Technology*, *62*(6), 1188-1207.

Cristianini, N., & Shawe-Taylor, J. (2002). *An introduction to support vector machines*. Cambridge, UK: Cambridge University Press.

Cutting, D. R., Karger, D. R., Pedersen, J. O., & Tukey, J. W. (1992). Scatter/gather: A cluster-based approach to browsing large document collections. In *Proceedings of the 15th annual international acm sigir conference on research and development in information retrieval* (p. 318-329).

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*, 391-407.

Domingos, P., & pazzani, M. (1997). On the optimality of the simple bayesian classifier under zero-one loss. *Machine Learning*, *29*, 103-130.

Dongen, S. (2000). *A cluster algorithm for graphs* (Tech. Rep. No. INS-R0010). Amsterdam: National Research Institute for Mathematics and Computer Science in the Netherlands.

Drineas, P., Frieze, A., Kannan, R., Vempala, S., & Vinay, V. (1999). Clustering in large graphs and matrices. In *Proceedings of the tenth annual acm-siam symposium on discrete algorithms.*

Dubin, D. (2004). The most influential paper gerard salton never wrote. *Library Trends*, *52*(4), 748-764.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification* (2nd ed.). New York, NY: John Wiley and Sons.

Eckart, C., & Young, G. (1936). The approximation of one matrix by another of lower rank. *Psychometrika*, *1*(3), 211Ð218.

Frantzi, K., Ananiadou, S., & Mima, H. (2000). Automatic recognition of multi-word terms: the c-value/nc-value method. *International Journal of Digital Libraries*, *3*(2), 117-132.

Geraci, F., Pellegrini, M., Maggini, M., & Sebastiani, F. (2006). Cluster Generation and Cluster Labelling for Web Snippets: A Fast and Accurate Hierarchical Solution. In *String processing and information retrieval* (Vol. 4209, p. 25-36).

Globerson, A., Chechik, G., Pereira, F., & Tishby, N. (2007). Euclidean embedding of co-occurrence data. *Journal of Machine Learning Research*, *8*, 2265-2295.

Hatzivassiloglou, V., Gravano, L., & Maganti, A. (2001). An investigation of linguistic features and clustering algorithms for topical document clustering. In *Proceedings of the 24th annual international acm sigir conference on research and development in information retrieval, sigir* (p. 224-231).

Haykin, S. (c 1999). *Neural networks: a comprehensive foundation.* London: Upper Saddle River, N. J.: Prentice Hall.

Hearst, M. A., & Pedersen, J. O. (1996). Reexamining the cluster hypothesis: scatter/gather on retrieval results. In *Proceedings of the 19th annual international acm sigir conference on research and development in information retrieval* (p. 76-84).

Hussain, S. F., Bisson, G., & Grimal, C. (2010). An improved co-similarity measure for document clustering. In *Proceedings of the 9th int'l conference on machine learning and applications.* Washington, D.C., USA.

Jain, A. K. (2010). Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, *31*, 651-666.

Janruang, J., & Guha, S. (2011). Semantic suffix tree clustering. In *Proceedings of the 1st irast international conference on data engineering and internet technology, deit.*

Janruang, J., & Kreesuradej, W. (2006). A new web search result clustering based on true common phrase label discovery. In *Proceedings of the international conference on computational intelligence for modelling control and automation, and international conference on intelligent agents, web technologies and internet commerce, cimca-iawtic* (p. 242).

Jayabharathy, J., Kanmani, S., & Parveen, A. A. (2011). A survey of document clustering algorithms with topic discovery. *Journal of Computing*, *3*(2), 21-27.

Jiang, P., Zhang, C., Guo, G., Niu, Z., & Gao, D. (2009). A k-means approach based on concept hierarchical tree for search results clustering. In *Proceedings of the 6th international conference on fuzzy systems and knowledge discovery* (p. 380-386).

Jiang, Z., Joshi, A., K., R., & Yi, L. (2000). *Retriever: Improving Web Search Engine Results Using Clustering* (Tech. Rep.). University of Maryland Baltimore County.

Jones, S., & Paynter, G. W. (2002). Automatic extraction of document key phrases for use in digital libraries: evaluation and applications. *Journal of the American Society for Information Science and Technology*, *53*(8), 653-677.

Kohonen, T. (1990). The self-organizing map. *Proceedings of the IEEE*, *78*(9), 1464 - 1480.

Korkontzelos, I., Mu, T., & Ananiadou, S. (2012). ASCOT: a text mining-based web-service for efficient search and assisted creation of clinical trials. *BMC Medical Informatics and Decision Making*, *12*(Suppl 1), S3.

Koshman, S., Spink, A., & Jansen, B. J. (2006). Web searching on the vivisimo search engine. *Journal of the American Society for Information Science and Technology*, *57*(14), 1875-1887.

Kovács, B. (2010). A generalized model of relational similarity. *Social Networks*, *32*(3), 197-211.

Krishnapuram, R., Joshi, A., & Yi, L. (1999). A fuzzy relative of the k-medoids algorithm with application

to web document and snippet clustering. In *Proceedings of the ieee international conference on fuzzy systems, fuzz-ieee* (p. 1281-1286).

Lagus, K., & Kaski, S. (1999). Keyword selection method for characterizing text document maps. In *Proceedings of the 9th int'l conference on artificial neural networks, icann* (p. 371-376).

Lagus, K., Kaski, S., & Kohonen, T. (2004). Mining massive document collections by the WEBSOM method. *Information Sciences: an International Journal - Special issue: Soft computing data mining*, *163*(1-3), 135-156.

Lam, W., Tsang, I., & Wong, T. (2013). Discovering low-rank shared concept space for adapting text mining models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*. (DOI:10.1109/TPAMI.2012.243)

Lan, M., Tan, C.-L., Su, J., & Lu, Y. (2009). Supervised and traditional term weighting methods for automatic text categorization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *31*(4), 721-735.

Lewis, D. D. (1997). Reuters-21578 text categorization test collection. (http://www.daviddlewis.com/resources/testcollections/reuters21578/)

Li, Y., Chung, S. M., & Holt, J. D. (2008). A general framework for dimensionality-reducing data visualization mapping. *Text document clustering based on frequent word meaning sequences*, *64*(1), 381-404.

Li, Z., Peng, H., & Wu, X. (2008). A new descriptive clustering algorithm based on nonnegative matrix factorization. In *Proceedings of the ieee int'l conference on granular computing, grc* (p. 407-412). Hangzhou, China.

Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395-416.

Matsumoto, T., & Hung, E. (2010). Fuzzy clustering and relevance ranking of web search results with differentiating cluster label generation. In *Proceedings of the ieee international conference on fuzzy systems, fuzz-ieee* (p. 1-8).

Mu, T., & Goulermas, J. Y. (2013). Automatic generation of co-embeddings from relational data with adaptive shaping. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *35*(10), 2340-2356.

Mu, T., Goulermas, J. Y., Tsujii, J., & Ananiadou, S. (2012). Proximity-based frameworks for generating embeddings from multi-output data. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, *34*(11), 2216-2232.

Mu, T., Jiang, J., Wang, Y., & Goulermas, J. Y. (2012). Adaptive data embedding framework for multi-class classification. *IEEE Trans. on Neural Networks and Learning Systems*, *23*(8), 1291-1303.

Noel, S., Raghavan, V., & Chu, C.-H. H. (2003). Document clustering, visualization, and retrieval via link

mining. In *Clustering and information retrieval* (Vol. 11, p. 161-194).

Osiński, S. (2006). Improving quality of search results clustering with approximate matrix factorisations. In *Proceedings of the 28th european conference on information retrival research, ecir.* London, UK.

Osiński, S., Stefanowski, J., & Weiss, D. (2004). Lingo: Search results clustering algorithm based on singular value decomposition. In *Advances in soft computing, intelligent information processing and web mining, proceedings of the international iis: Iipwm'04 conference* (p. 359-368).

Osiński, S., & Weiss, D. (2005). A concept-driven algorithm for clustering search results. *IEEE Intelligent Systems*, *20*(3), 48-54.

Ouyang, M., Welsh, W. J., & Georgopoulos, P. (2004). Gaussian mixture clustering and imputation of microarray data. *Bioinformatics*, *20*(6), 917-92.

Rooneya, N., Pattersona, D., Galushkaa, M., & Dobryninb, V. (2006). A scaleable document clustering approach for large document corpora. *Information Processing and Management*, *42*(5), 1163-1175.

Salton, G., & Buckley, C. (1987). *Term weighting approaches in automatic text retrieval* (Research Report). Ithaca, NY, USA: Cornell University.

Stefanowski, J., & Weiss, D. (2007). Extending k-means with the description comes first approach. *Control and Cybernetics*, *36*(4), 1010-1035.

Syed, Z., Finin, T., & Joshi, A. (2008). Wikipedia as an ontology for describing documents. In *Proceedings of the 2nd international conference on weblogs and social media, icwsm.*

Theodosiou, T., Darzentas, N., Angelis, L., & Ouzounis, C. A. (2008). PuReD-MCL: a graph-based PubMed document clustering methodology. *Bioinformatics*, *24*(17), 1935-1941.

Thomas, J., McNaught, J., & Ananiadou, S. (2011). Applications of text mining within systematic reviews. *Research Synthesis Methods*, *2*(1), 1-14.

Torgerson, W. S. (1952). Multidimensional scaling: I. theory and method. *Journal Psychometrika*, *17*(4), 401-419.

Treeratpituk, P., & Callan, J. (2006). Automatically labeling hierarchical clusters. In *Proceedings of the int'l conference on digital government research.*

Tseng, Y. (2010). Generic title labeling for clustered documents. *Expert Systems with Applications*, *37*, 2247-2254.

Tsuruoka, Y., Tateishi, Y., Kim, J., Ohta, T., McNaught, J., Ananiadou, S., & J.Tsujii. (2005). Developing a robust part-of-speech tagger for biomedical text. In *Proceedings of the advances in informatics-10th panhellenic conference on informatics, lncs* (p. 117-132).

Wang, J., Mo, Y., Huang, B., Wen, J., & He, L. (2008). Web search results clustering based on a novel suffix tree structure. *Autonomic and Trusted Computing - Lecture Notes in Computer Science*, *5060*,

540-554.

Weiss, D. (2006). *Descriptive clustering as a method for exploring text collections.* PhD dissertation, Poznan University of Technology Institute of Computing Science, Poland.

Xu, R. (2005). Survey of clustering algorithms. *IEEE Transactions on Neural Networks*, *16*(3), 645-678.

Xu, W., Liu, X., & Gong, Y. (2003). Document clustering based on non-negative matrix factorization. In *Proceedings of the 26th annual international acm sigir conference on research and development in information retrieval* (p. 267-273).

Zamir, O. E. (1999). *Clustering web document: A phrase-based method for grouping search engine results.* Unpublished doctoral dissertation, University of Washington.

Zeng, J., Cheung, W., & Liu, J. (2013). Learning topic models by belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* (DOI:10.1109/TPAMI.2012.185)

Zhang, D., & Dong, Y. (2004). Semantic, hierarchical, online clustering of web search results. *Advanced Web Technologies and Applications, Lecture Notes in Computer Science*, *3007*, 69-78.

Table 1

Summary of the main variables and quantities used in the various stages of CEDL.

Variable name	Range	Description
$\{\mathfrak{D}_i\}_{i=1}^n$	NA	A set of n targeted documents to be clustered.
$\{\mathfrak{P}_i\}_{i=1}^m$	NA	A set of m phrases to be used as candidate labels.
$\mathbf{S}_{dd}$	$\mathbb{R}^{n\times n}$	Similarity matrix between the n targeted documents.
$\mathbf{S}_{pp}$	$\mathbb{R}^{m\times m}$	Similarity matrix between the m candidate phrases.
$\mathbf{S}_{dp}$	$\mathbb{R}^{n\times m}$	Similarity matrix between the n targeted documents and m candidate phrases.
$\mathbf{X}_{dw}$	$\mathbb{R}^{n\times d}$	Word-based feature matrix of the n targeted documents.
$\mathbf{X}_{pw}$	$\mathbb{R}^{m\times d}$	Word-based feature matrix of the m candidate phrases.
$\mathbf{X}_{dp}$	$\mathbb{R}^{n\times m}$	Phrase-based feature matrix of the n targeted documents.
$\mathbf{F}_{pd}$	$\{0,1\}^{m\times n}$	Binary matrix indicating whether a candidate phrase and its distorted or synonymous versions appear in a targeted document.
$\mathbf{S}$	$\mathbb{R}^{(n+m)\times(n+m)}$	A composite similarity matrix between the combined set of n targeted documents and m candidate phrases.
$\mathbf{S}_0$	$\mathbb{R}^{(n+m)\times(n+m)}$	A composite similarity matrix between the combined set of targeted documents and candidate phrases, computed by simple combination in Eq. (5).
w_1, w_2, w_3	$\mathbb{R}^+$	Positive combining parameters for computing $\mathbf{S}_0$ in Eq. (5).
$\mathbf{N}(\mathbf{S}_0, h_1)$	$\{0,1\}^{(n+m)\times(n+m)}$	Binary neighbourhood indication matrix by applying h_1-nearest neighbour search to $\mathbf{S}_0$ in Eq. (5).
$\mathbf{N}(\mathbf{S}, h_2)$	$\{0,1\}^{(n+m)\times(n+m)}$	Binary neighbourhood indication matrix by applying h_2-nearest neighbour search to $\mathbf{S}$ in Eq. (8).
$\mathbf{Z} = [z_{ij}]$	$\mathbb{R}^{(n+m)\times l}$	Co-embedding matrix of both the n targeted documents and m candidate phrases.
$\mathbf{Z}_d$	$\mathbb{R}^{n\times l}$	Co-embedding matrix of the n targeted documents.
$\mathbf{Z}_p$	$\mathbb{R}^{m\times l}$	Co-embedding matrix of the m candidate phrases.
$\mathbf{\Sigma}_l$	$\mathbb{R}^{l\times l}$	Eigenvalue matrix of $\mathbf{S}\circ\mathbf{N}(\mathbf{S}, h_2)$ corresponding to the largest l eigenvalues.
$\mathbf{P}_l$	$\mathbb{R}^{(n+m)\times l}$	Eigenvector matrix of $\mathbf{S}\circ\mathbf{N}(\mathbf{S}, h_2)$ corresponding to the largest l eigenvalues.
k	$\mathbb{Z}^+$	Cluster number used by k-means clustering algorithm.
$\mathbf{Y} = [y_{ij}]$	$\{-1,+1\}^{n\times k}$	Binary cluster membership matrix indicating whether a document belonging to a document cluster.
$\mathbf{Y}_p$	$\{-1,+1\}^{m\times k}$	Binary cluster membership matrix indicating whether a candidate phrase is relevant to the topic of a document cluster.
$\boldsymbol{w}_i$	$\mathbb{R}^l$	The weight vector of the discriminant function for the ith cluster.
$\mathbf{W}$	$\mathbb{R}^{l\times k}$	Weight matrix $\mathbf{W} = [\boldsymbol{w}_1, \boldsymbol{w}_2, \ldots, \boldsymbol{w}_k]$.
b_i	$\mathbb{R}$	The bias of the discriminant function for the ith cluster.
$\boldsymbol{b}$	$\mathbb{R}^k$	The k-dimensional bias vector $\boldsymbol{b} = [b_1, b_2, \ldots, b_k]^T$.
$\boldsymbol{y}_i$	$\{-1,+1\}^n$	The ith column of the cluster membership matrix $\mathbf{Y}$ for documents.
$\mathbf{Y}_i$	$\{0,1\}^{n\times 2}$	Binary matrix computed from $\boldsymbol{y}_i$.
$\left\{\mathfrak{D}_j^{(i)}\right\}_{j=1}^{n_i}$	NA	The set of n_i documents belonging to the ith cluster.
$\left\{\mathfrak{P}_j^{(i)}\right\}_{j=1}^{m_i}$	NA	The set of m_i topic phrases of the ith cluster.
$\mathbf{Z}_d^{(i)}$	$\mathbb{R}^{n_i\times l}$	Co-embedding matrix of the n_i documents of the ith cluster.
$\mathbf{Z}_p^{(i)}$	$\mathbb{R}^{m_i\times l}$	Co-embedding matrix of the m_i topic phrases of the ith cluster.
$\widehat{\mathbf{Y}}_p$	$\mathbb{R}^{m_i\times k}$	Unsigned cluster membership matrix for the m_i topic phrases.
a	$\mathbb{R}^+$	CEDL parameter for modifying the effect of the weight $\text{score}_2^{(i)}$.
m_i^*	$\mathbb{Z}^+$	The number of top phrases selected based on the ranking score in Eq. (16).
$\boldsymbol{z}_p^{(i)}$	$\mathbb{R}^{1\times l}$	Co-embedding vector of the cluster label of the ith cluster.

Table 2

Information of the used document sets, including the number of documents (n), the number of uni-gram words used for characterising the documents (d), and the ground-truth topic information for which the total number of topics (c) is shown in the third row. The number of documents included for each topic is shown in parentheses.

Dataset	CT1	CT2	Reuters1	Reuters2	Reuters3	EEP1
n	1000	580	443	547	1000	331
d	10231	8038	4244	5492	7649	55253
c	5	8	3	6	10	2
Topic tag (ground truth partition)	asthma (200) breast cancer (200) cardiovascular (200) lung cancer (200) prostate cancer(200)	depression (125) leukemia (110) HIV (95) cardiovascular (80) prostate cancer (65) lung cancer (50) breast cancer (35) asthma (20)	money-fx (200) sugar (143) earn (100)	iron-steel (47) earn (100) money-fx (100) sugar (100) ship (100) trade (100)	earn (60) acq (100) crude (50) trade (100) money-fx (80) interest (200) ship (150) sugar (100) money-supply (100) coffee (60)	work-based learning (164) primary education (167)
Classification accuracy	LFDA: 0.9728 LSVM: 0.9896	LFDA: 0.8610 LSVM: 0.9596	LFDA: 0.9816 LSVM: 0.9757	LFDA: 0.9372 LSVM: 0.9467	LFDA: 0.8208 LSVM: 0.8808	LFDA: 0.8157 LSVM: 0.8060

Table 3

Evaluation of the cluster quality for the competing descriptive clustering algorithms, using the six document sets and the four measures. The best performance for each measure and test set is marked in bold.

	f-measure		
	CEDL	DKM	Lingo
CT1	**0.9780**	0.9745	0.5239
CT2	**0.9673**	0.8860	0.5556
Reuters1	**0.9435**	0.8055	0.5185
Reuters2	**0.7856**	0.7094	0.4238
Reuters3	**0.7603**	0.5714	0.3791
EEP1	**0.7282**	0.7141	N/A

	Cluster Purity		
	CEDL	DKM	Lingo
CT1	**0.9786**	0.9749	0.5986
CT2	**0.9614**	0.8973	0.4860
Reuters1	**0.9363**	0.8063	0.6695
Reuters2	**0.7828**	0.6853	0.5603
Reuters3	**0.7803**	0.6436	0.3639
EEP1	**0.7406**	0.7168	N/A

	Cluster Entropy		
	CEDL	DKM	Lingo
CT1	**0.0702**	0.0784	1.8875
CT2	**0.0765**	0.1566	1.6633
Reuters1	**0.2001**	0.3783	0.9312
Reuters2	**0.3454**	0.5095	0.9772
Reuters3	**0.3268**	0.5714	1.7560
EEP1	**0.8324**	0.9438	N/A

	Cluster Contamination		
	CEDL	DKM	Lingo
CT1	**0.0519**	0.0605	0.7732
CT2	**0.0845**	0.1889	0.7519
Reuters1	**0.1762**	0.3687	0.6155
Reuters2	**0.3686**	0.4911	0.6820
Reuters3	**0.3649**	0.5269	0.8384
EEP1	**0.7601**	0.7788	N/A

Table 4

Cluster labels generated for the EEP documents related to the "management, governance and finance" topic, for $k = 15$*. The actual number of documents included in each cluster is shown in parentheses.*

CEDL	DKM
secondary school teacher (123)	high education student (109)
eligible young people (87)	school age child (104)
high education funding (54)	primary school teacher (89)
ental health service (50)	case study school (85)
student total income (47)	family support service (77)
community development learning fund (36)	high education course (55)
local authority interviewee (36)	ema eligible young people (53)
individual learning account (35)	high education qualification (48)
formal childcare provider (29)	secondary school teacher (41)
skill development fund (28)	sure start local programme area (36)
case study partnership (25)	basic skill project (30)
administrative support staf (25)	individual learning account centre (25)
sure start local programme (21)	pupil teacher ratio (20)
minority ethnic pupil (17)	recognition employment sector organisation type (6)

Table 5

Cluster labels generated for different sets of clinical trial documents retrieved using different input queries, for $k = 6$. The number of documents included in each cluster is shown in parentheses.

CEDL	DKM	Lingo
Query: "asthma"		
acute asthma exacerbation (47)	stable asthma patient (65)	smoking=pack year (158)
self management education(40)	moderate persistent allergic asthma (49)	chronic obstructive pulmonary disease (157)
chronic inflammatory airway (37)	moderate acute asthma exacerbation (39)	health care use (151)
symbicort single inhaler therapy (31)	childhood asthma management program (34)	persistent allergic rhinitis (94)
university asthma clinical (27)	house dust mite allergen (13)	ragweed allergy season (56)
house dust mite allergen (18)		unfavorable Th1/Th2 balance (3)
		others (4)
Query: "breast cancer"		
upper normal limit (54)	metastatic breast cancer patient (129)	adjuvant therapy (200)
expanded breast cancer (39)	early stage breast cancer patient (71)	treatment related fatigue (148)
sentinel lymph node (32)		core needle biopsy (70)
behavioral sleep intervention (31)		axillary lymph node(41)
random periareolar fine needle aspiration (23)		EGFR tyrosine kinase (4)
CD44+/CD24 breast (21)		
Query: "lung cancer"		
symptomatic sclerodermarelated interstitial lung disease (48)	small cell lung cancer patient (115)	non small cell (165)
previous lung cancer (47)	stage small cell lung cancer (67)	advanced nonsmall cell (165)
nonsmall cell lung (37)	limited stage small cell lung cancer (20)	extensive stage disease(147)
extensive stage disease small cell (25)		stage IIIB/IV (105)
stage IIIB/IV (23)		dose thoracic CT (92)
minimum dose CT (20)		5th congressional district (2)
		others (8)
Query: "prostate cancer"		
resistant prostate cancer (43)	prostate cancer patient (121)	hormone refractory disease (150)
positive bone scan (42)	androgen independent prostate cancer (31)	serum PSA level (137)
lycopene supplementation lower serum (33)	hormone refractory prostate cancer (28)	androgen deprivation therapy (136)
external beam radiotherapy (32)	high risk prostate cancer (22)	alanine amino transferase (2)
chemotherapy naive hormone (26)		others (14)
androgen deprivation therapy (24)		
Query: "cardiovascular"		
CVD health disparity (41)	cardiovascular disease risk (119)	ischaemic heart disease (199)
progressive aerobic exercise (38)	depressed coronary artery disease patient (39)	cardiovascular disease event (199)
coronary artery disease (35)	cardiovascular disease event (26)	cardiovascular risk factor (190)
control cardiovascular risk (33)	minor cardiovascular risk factor (16)	peripheral biological mechanism (66)
food guide diet (29)		food guide diet (44)
polycystic kidney disease (24)		motor incomplete spinal cord (7)

Table 6

Cluster labels generated for different sets of clinical trial documents retrieved using different input queries, for $k = 6$. The number of documents included in each cluster is shown in parentheses.

CEDL	DKM	Lingo
Query: "HIV"		
current antiretroviral regimen (54)	HIV infected patient (144)	HIV infected patient (200)
VA Long Beach (42)	HIV infected person(27)	current antiretroviral regimen (156)
sexual health education program (34)	HIV infected TB patient (17)	coronary artery disease (130)
HIV infected inner city (30)	HIV/HBV coinfected individual (7)	CD4 cell count (101)
active pulmonary TB (21)	eban HIV/STD risk reduction intervention (6)	peginterferon alpha 2a (3)
chronic HCV infection (19)		
Query: "leukemia"		
standard/high risk all (45)	chronic lymphocytic leukemia (53)	chronic lymphocytic leukemial (198)
Karnofsky performance status (42)	acute lymphoblastic leukemia (49)	acute myeloid leukemia (198)
bone marrow aspirate (37)	early allogeneic stem cell transplantation (35)	allogeneic hematopoietic cell transplantation (146)
chronic phase chronic myeloid leukemia (32)	lymphoid blast phase chronic myeloid leukemia (35)	myeloid blast phase (128)
allogeneic stem cell transplantation (26)	philadelphia chromosome positive acute lymphoblastic	philadelphia chromosome positive (66)
	leukemia (19)	
week standard/high risk Low (25)	white blood cell transfusion (12)	others (25)
Query: "depression"		
selective serotonin reuptake inhibitor (53)	current depressive disorder (86)	antidepressant drug trial (147)
cognitive behavioral intervention (48)	current major depressive episode (49)	week washout period(144)
poor right hemispheric functioning (34)	recurrent major depressive disorder(44)	cognitive behavioral therapy (134)
current major depressive episode (31)	collaborative depression care management (21)	childhood sexual abuse (101)
recurrent major depressive disorder (29)		fatty acid ethyl (12)
collaborative depression care management (17)		multivariable Cox proportional hazard(5)
		others (6)
Query: "schizophrenia"		
behavioral social skill (60)	schizophrenia/schizoaffective disorder (107)	schizophrenia/schizoaffective disorder (200)
total PANSS score (54)	risperidone oral tablet (70)	substance use disorder (183)
oral atypical antipsychotic (38)		routine clinical care (137)
cognitive adaptation training (28)		social cognitive deficit (100)
smoking cessation treatment session (28)		smooth pursuit eye (4)
electronic schizophrenia treatment adherence (22)		

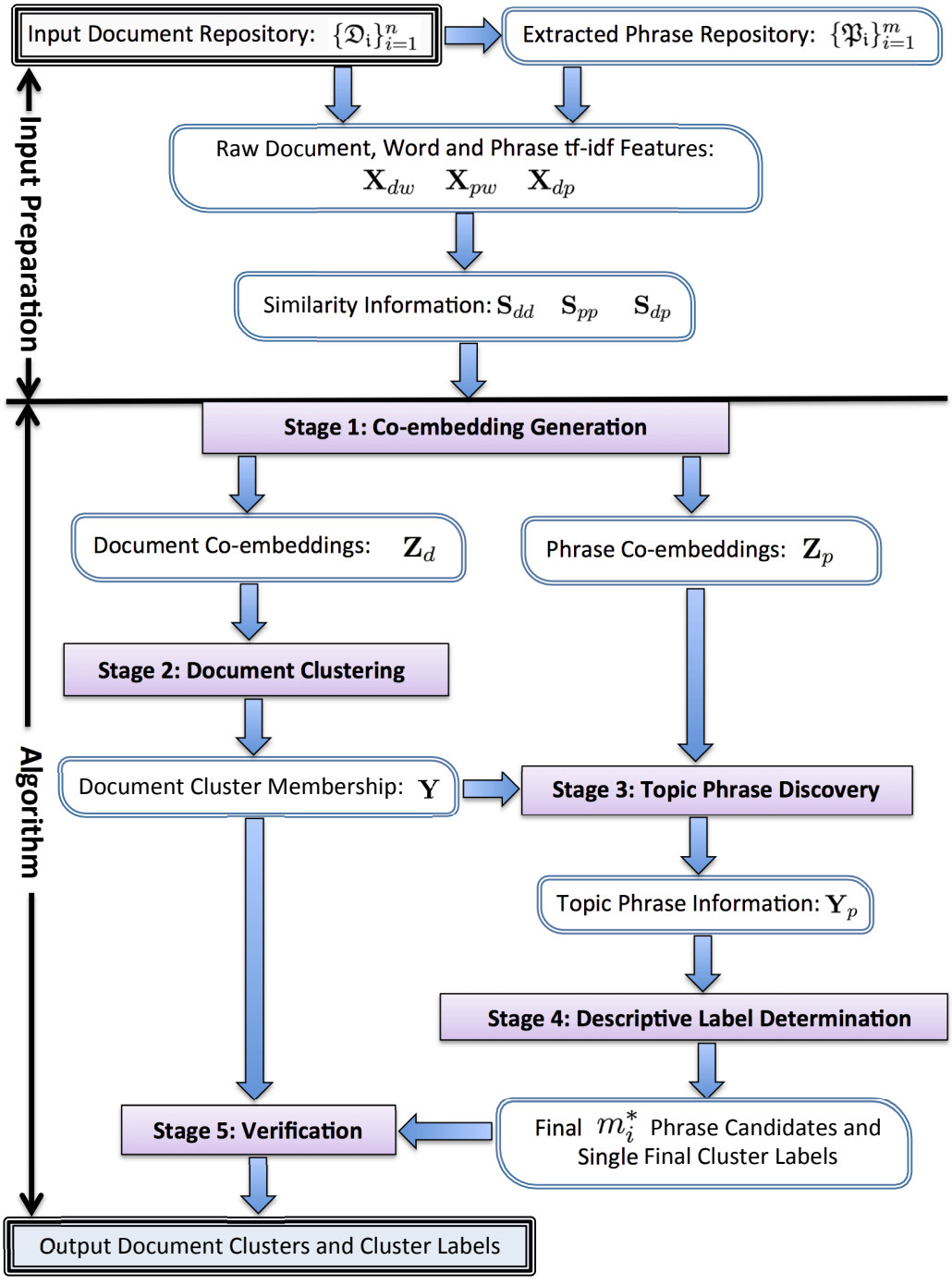


Figure 1. The main processing stages and data structures of CEDL. A detailed summary for the notation of variables and quantities used here can be found in Table 1.

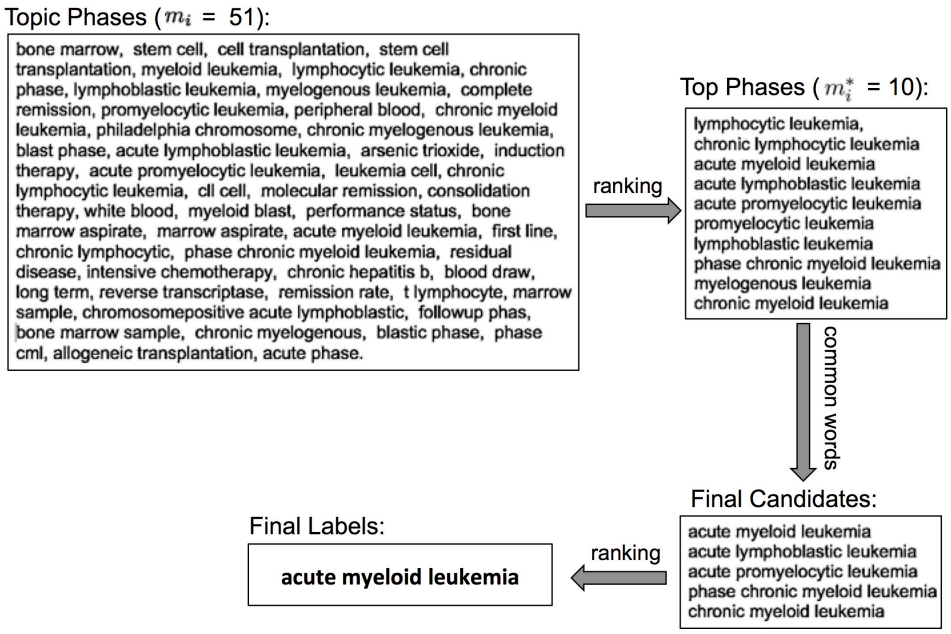
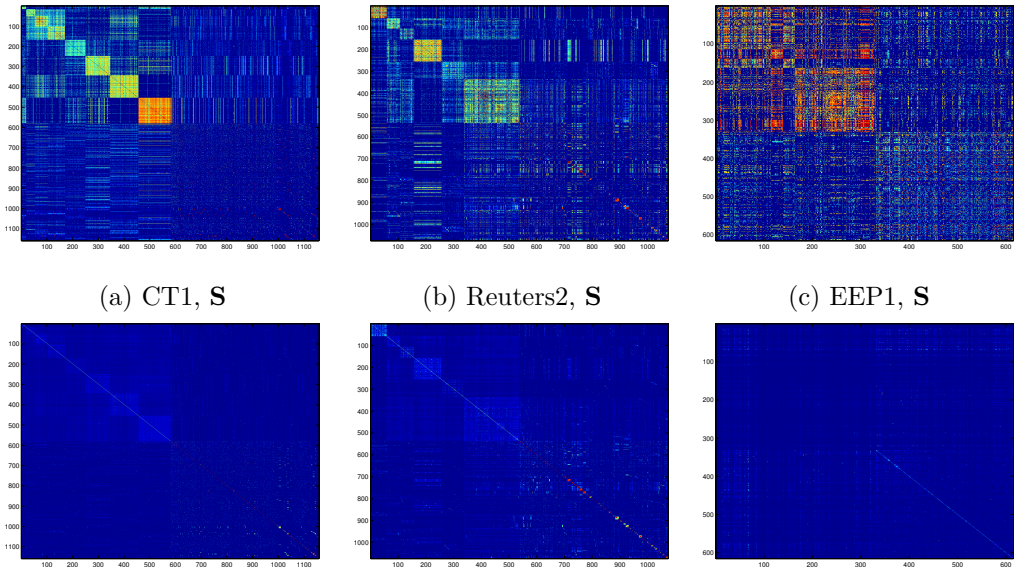


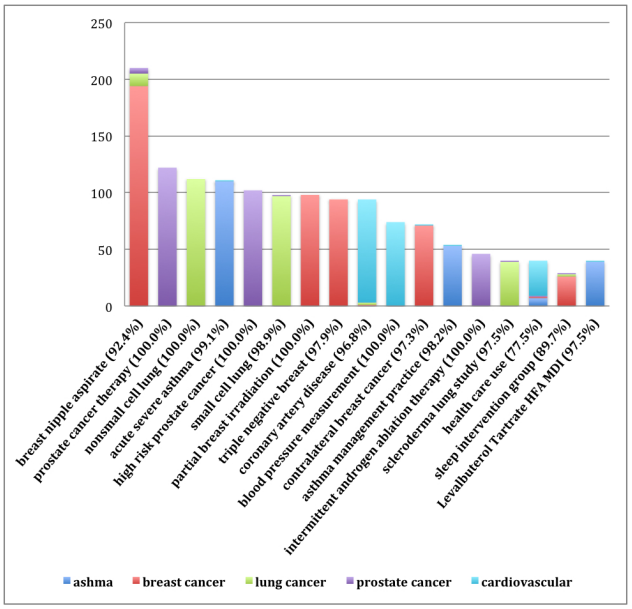
Figure 2. Example of the gradual multi-stage process of determining the descriptive cluster label, generated using the CT2 document set for the leukemia cluster.



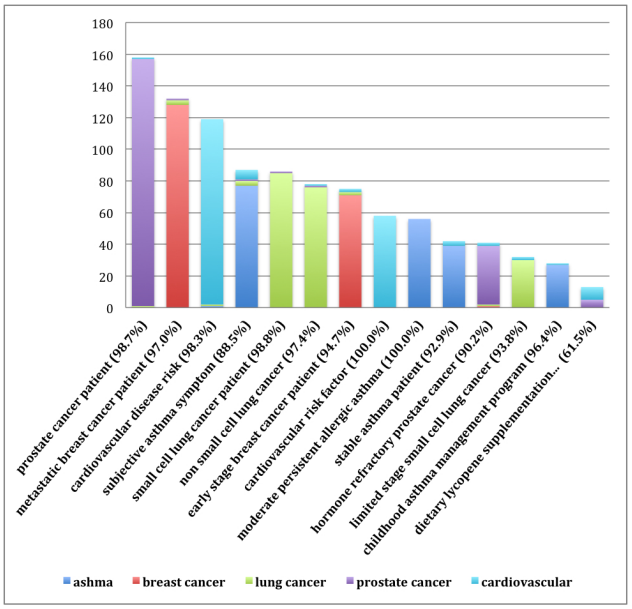
(a) CT1, $\mathbf{S}$ (b) Reuters2, $\mathbf{S}$ (c) EEP1, $\mathbf{S}$

(d) CT1, $\mathbf{S}_0$ (e) Reuters2, $\mathbf{S}_0$ (f) EEP1, $\mathbf{S}_0$

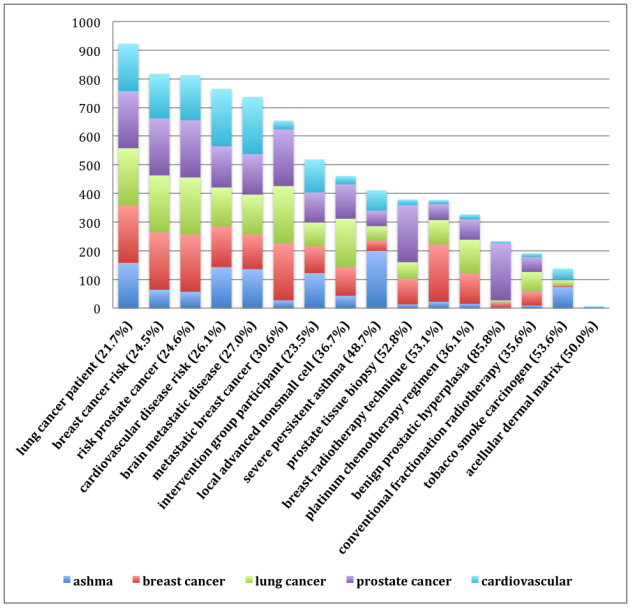
Figure 3. Comparison of the proposed second-order neighbour-based similarity matrix $\mathbf{S}$ used within Eq. (9), with the first-order $\mathbf{S}_0$ of Eq. (5), presented as image matrices in the top and bottom rows, respectively. The matrices are computed with the CT1, Reuters2 and EEP1 document sets.



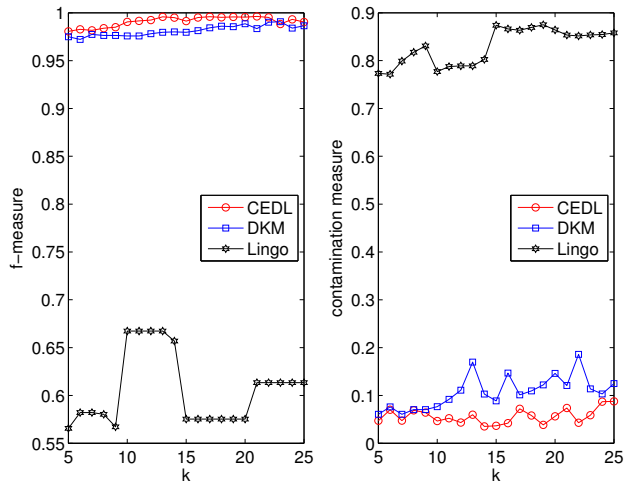
(a) CEDL (ACP=96.6%).



(b) DKM (ACP=93.5%).

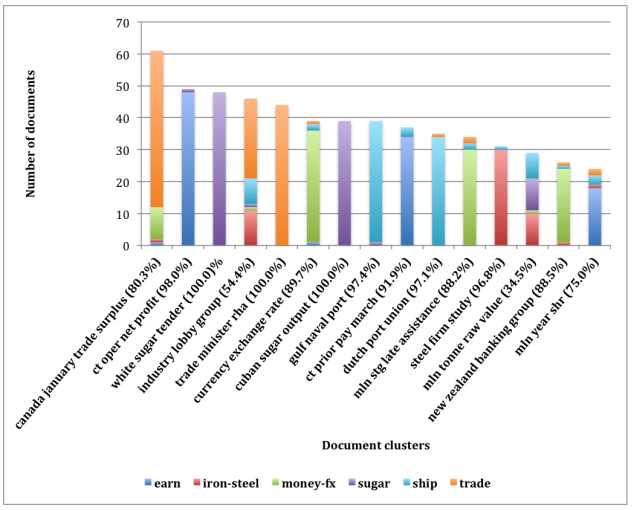


(c) Lingo (ACP= 39.4%).

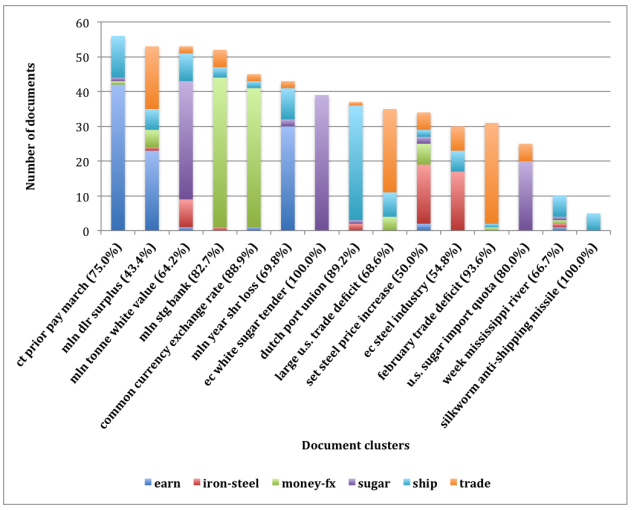


(d) Cluster measures.

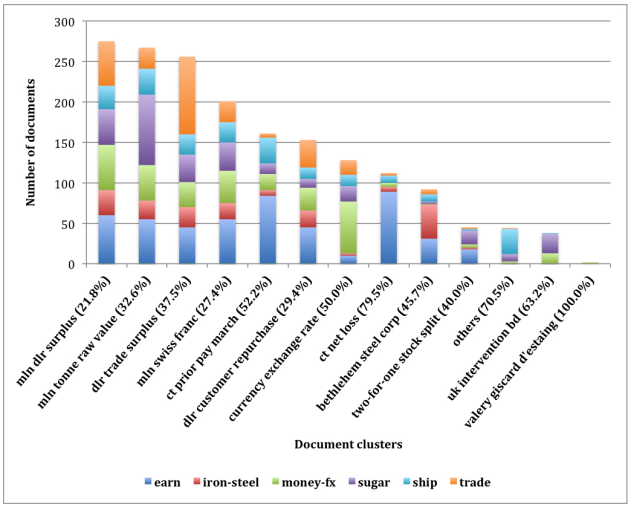
Figure 4. The first three figures display the document clusters and their corresponding labels for the CT1 set, obtained from: (a) CEDL, (b) DKM and (c) Lingo. The parameter k for controlling the cluster number was set as 18. The quality of each obtained cluster is quantified by the cluster purity measures displayed next to each descriptive label in parenthesis. The averaged cluster purity (ACP) measures are also displayed. The last figure (d) plots the values of f-measure and cluster contamination for the three competing methods and increasing values of k that control the number of document clusters.



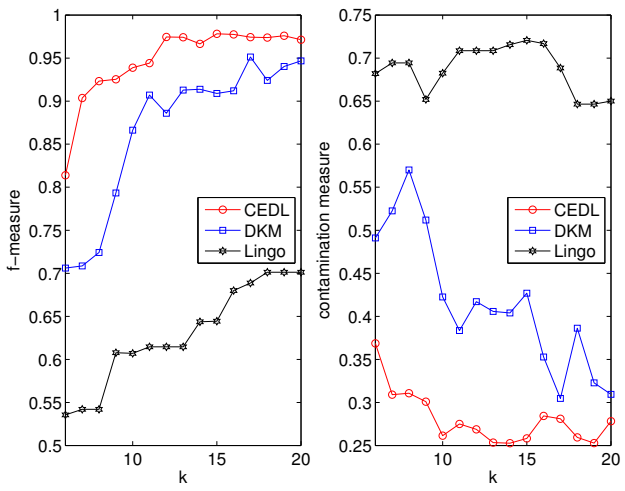
(a) CEDL (ACP= 86.1%).



(b) DKM (ACP= 75.1%).

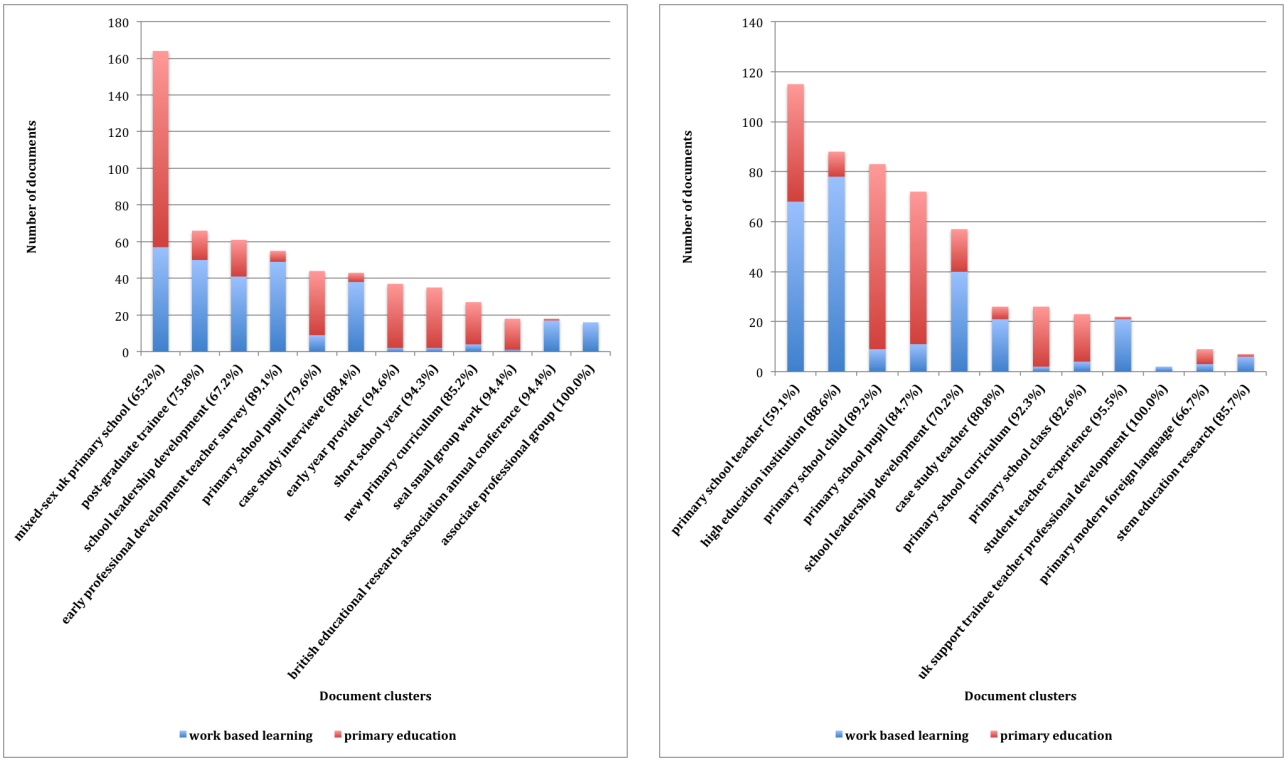


(c) Lingo (ACP= 50.0%).



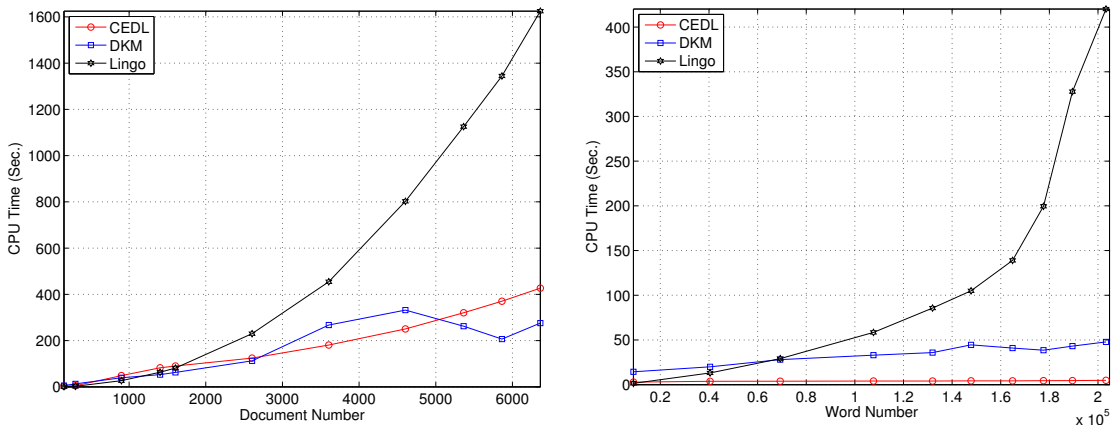
(d) Cluster measures.

Figure 5. Document clusters and their corresponding labels for the Reuters2 set, obtained from: (a) CEDL, (b) DKM and (c) Lingo, for $k = 15$. The individual and averaged cluster purity measures are also displayed. Figure (d) plots the values of f-measure and cluster contamination for all competing methods and increasing values of k that control the number of document clusters.



(a) CEDL (ACP= 85.7%). (b) DKM (ACP= 82.9%).

Figure 6. Document clusters and their corresponding labels for the EEP1 set, obtained from: (a) CEDL and (b) DKM, for $k = 12$. The individual and averaged cluster purity measures are also displayed.



(a) Increasing document number. (b) Increasing word number.

Figure 7. Variations of CPU times in seconds from running the three methods of CEDL, DKM and Lingo with: (a) increasing the number of documents using the Reuters set, and (b) increasing the number of words characterising a set containing a total of 331 EEP documents.

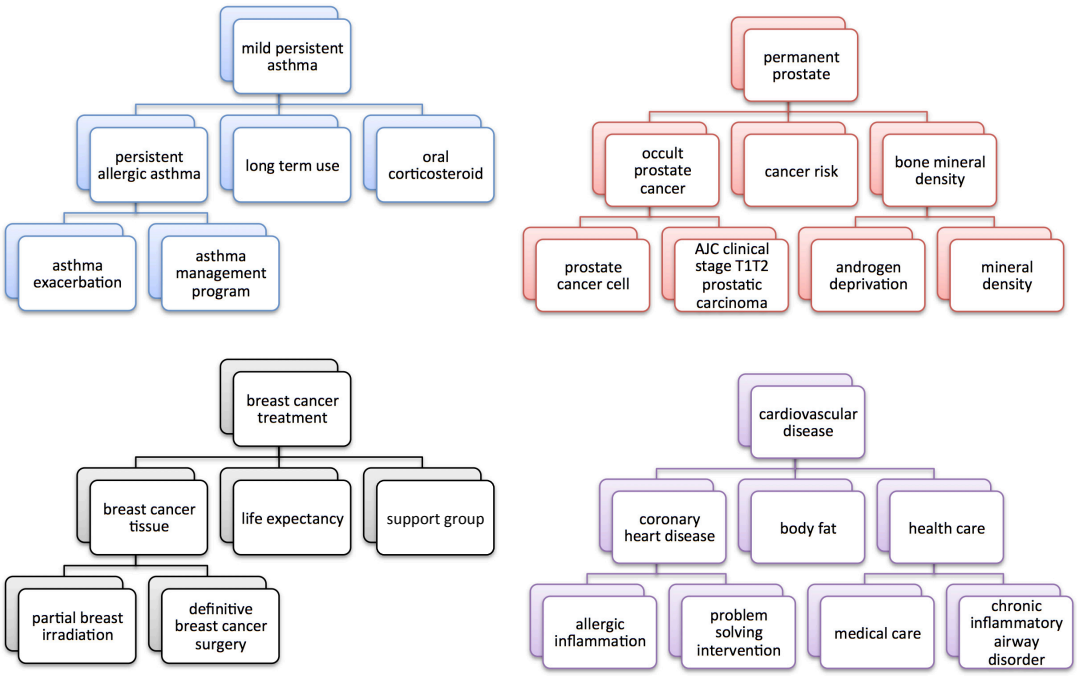


Figure 8. Example of the hierarchical cluster structure along with the descriptive cluster labels generated by CEDL, for clinical trial documents retrieved based on the four queries of “asthma”, “prostate cancer”, “breast cancer”and “cardiovascular”.