Information Processing and Management 47 (2011) 647–666

Contents lists available at ScienceDirect

Information Processing and Management

journal homepage: www.elsevier.com/locate/infoproman

Multilingual document mining and navigation using self-organizing maps

Hsin-Chang Yang [a,*], Han-Wei Hsiao [a], Chung-Hong Lee [b]

[a] Department of Information Management, National University of Kaohsiung, Kaohsiung 811, Taiwan
[b] Department of Electrical Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung, Taiwan

ARTICLE INFO

Article history:
Received 30 April 2009
Received in revised form 16 November 2009
Accepted 1 December 2009
Available online 8 January 2010

Keywords:
Multilingual Web page navigation
Multilingual text mining
Self-organizing map
Hierarchy alignment

ABSTRACT

One major approach for information finding in the WWW is to navigate through some Web directories and browse them until the goal pages were found. However, such directories are generally constructed manually and may have disadvantages of narrow coverage and inconsistency. Besides, most of existing directories provide only monolingual hierarchies that organized Web pages in terms that a user may not be familiar with. In this work, we will propose an approach that could automatically arrange multilingual Web pages into a multilingual Web directory to break the language barriers in Web navigation. In this approach, a self-organizing map is constructed to train each set of monolingual Web pages and obtain two feature maps, which reveal the relationships among Web pages and thematic keywords, respectively, for such language. We then apply a hierarchy generation process on these maps to obtain the monolingual hierarchy for these Web pages. A hierarchy alignment method is then applied on these monolingual hierarchies to discover the associations between nodes in different hierarchies. Finally, a multilingual Web directory is constructed according to such associations. We applied the proposed approach on a set of Web pages and obtained interesting result that demonstrates the feasibility of our method in multilingual Web navigation.

1. Introduction

Nowadays the users of the World Wide Web (or for simplicity, the Web) try to access the huge amount of documents on the Web (or the Web pages) by searching with a search engine or browsing through hyperlinks existed within Web pages. For users who have no specific goal, browsing Web pages is often the preferred choice. However, many users have difficulty of getting start from a page which will eventually lead to their goals. Hence many portal sites emerge to provide such starting points. These sites often provide, in company with search facility, some sorts of navigating structure which organize Web pages into hierarchies, which are called Web directories henceforth. Users can then achieve a thematic navigation through such hierarchies. However, these hierarchies were generally constructed by human experts manually and were often lack of coverage, redundant, probably inconsistent, and hard to maintain. Besides, different sites often adopt different topic selection and categorization schemes which make them incapable of information exchange.

Another problem of existing Web directories comes from the monolingual nature in the construction process. Most of Web directories categorized only Web pages written in a specific language, such as English. Different Web directories have to be constructed for different native Web pages. Such monolingual interface may limit the spread of users who are unfamiliar with the used language. For example, a native Chinese may not intend to use a Web directory which provides only

[*] Corresponding author.
E-mail addresses: yanghc@nuk.edu.tw (H.-C. Yang), hanwei@nuk.edu.tw (H.-W. Hsiao), leechung@mail.ee.kuas.edu.tw (C.-H. Lee).
URL: http://www.im.nuk.edu.tw/yanghc (H.-C. Yang).

0306-4573/$ - see front matter © 2009 Elsevier Ltd. All rights reserved.
doi:10.1016/j.ipm.2009.12.003

English categorization labels and Web pages. Thus, it will be convenient for users to have a Web directory providing multilingual category labels and categorizing multilingual Web pages.

There are two necessary steps in constructing a multilingual Web directory. The first step is to organize Web pages into hierarchies for easy browsing. Although other structures are possible for Web page navigation, hierarchies were most adopted since they have intrinsic categorization structures that higher-level categories represent superset of lower-level ones. Most users found this convenient since they could achieve their goals by exploiting the structures in a coarse-to-fine manner from the most general theme that meets their goals. Most popular portal sites constructed such hierarchies by human experts. Although these hierarchies have the advantages of precise and consistent, manual construction approach suffers from the enormous amount of time and labor to initiate and maintain those hierarchies and prevents it being applied on large datasets. Thus automatic approach should be more feasible for large datasets such as the Web.

The second step in constructing multilingual Web directories is to obtain the associations between different languages. One popular approach is to apply some machine translation schemes to translate terms in one language to another. Unfortunately, there is still no well recognized scheme to provide precise translation between two languages. A different approach is to match terms in different languages directly without a priori translation. This approach is also difficult since we need some kind of measurements to measure the semantic relatedness between them. Such semantic measurements are generally not able to be explicitly defined, even with human intervention. Thus we need a kind of automated process to discover the relationships between different languages. Such process is often called multilingual text mining (MLTM).

To construct Web directories, human intervention is unavoidable in present time. We need human effort in tasks such as selecting topics and revealing their relationships. Such need is acceptable only when the volume of Web pages is considerably small. However, the volume of Web pages under consideration is generally large enough to prevent manual construction. To expand the applicability of the directories, some kind of automatic process should involve during the construction of the directories. The degree of automation in such construction process may differ for different constructors with different needs. One may only need a friendly interface to automate the authoring process, and another one may try to automatically identify every component from the ground up. We recognize the importance of a Web directory not only as a navigation tool but also a desirable scheme for knowledge acquisition and representation. According to such recognition, we try to develop a scheme based on a proposed text mining approach to automatically construct Web directories. Our approach is opposite to the navigation task performed by an existing Web directory to obtain goal pages. We extract knowledge from a corpus of Web pages to construct a Web directory.

In this work, we will develop an automatic scheme to arrange multilingual Web pages into Web directories based on a multilingual text mining approach. We will first apply a machine learning algorithm (Yang & Lee, 2004) based on self-organizing maps (Kohonen, 1997) on a corpus of monolingual Web pages to identify their topics, discover the relations among them, and construct a Web directory. The construction process consists of two major tasks. The first is topic detection which identifies the major themes existed in a set of close related Web pages. This set of Web pages is called a category hereinafter. The second is the construction of a Web directory for these monolingual Web pages. A second text mining process is then applied on two monolingual directories to discover the associations between categories in different directories. The proposed method will provide users novel multilingual Web directories for easy browsing of Web pages in different languages.

Fig. 1 depicts an overview of architecture of the proposed method. When a set of monolingual Web pages is input, we will first cluster them using self-organizing map algorithm. Two feature maps are obtained to reveal the relationships among

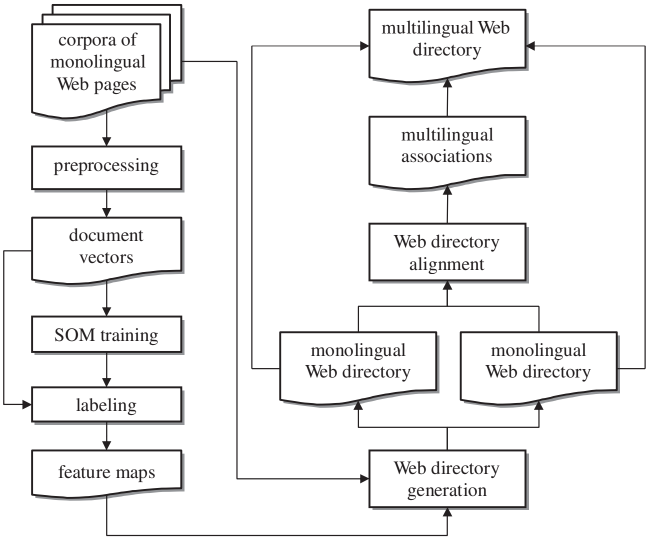

corpora of monolingual Web pages
preprocessing
document vectors
SOM training
labeling
feature maps
multilingual Web directory
multilingual associations
Web directory alignment
monolingual Web directory
monolingual Web directory
Web directory generation

Fig. 1. The processing steps of the proposed method.

terms and Web pages, respectively. A hierarchy generation process is then applied on these maps to construct a Web directory for these monolingual Web pages. After constructing two monolingual directories, a hierarchy alignment process is applied on these directories to discover the relationships between categories in different directories. Finally, a multilingual Web directory is constructed according to such relationships. Our method provides a uniform framework that not only identifies important subjects from an entity of Web pages and specifies the Web pages that are semantically related to a subject, but also explores the relations among these subjects. Besides, incorporating multilingual terms and pages into single directory should provide us a much comprehensive way to navigate and organize Web pages in different languages.

The following text is organized as follows. In Section 2 we will mention some works related to this work. Section 3 describes the Web directory generation process based on self-organizing maps. After generating Web directories, we then show how to find the associations between two monolingual Web directories and obtain a multilingual Web directory in Section 4.1. The experimental result will be described in Section 5. Finally, we give some conclusions and discussions in the last section.

2. Related work

In this section, we will review works in areas related to this work, which are Web directory generation, hierarchy alignment, and multilingual information retrieval/text mining.

2.1. Literature on Web directory generation

Web directory generation belongs to the general task of hierarchy generation. Hierarchical structure is popular when we intend to organize a set of data for the ease of comprehension and access. Most of the related works on hierarchies focus on using them for accessing data. An example is the Cat-a-Cone system developed by Hearst and Karadi (1997). This usage replies on some existing hierarchies which are often constructed manually by domain experts. There is increasing need in constructing hierarchies automatically recently due to the increasing volume of data that prohibits manual handling. Methods for automatically constructing various types of hierarchies were proposed, such as bounding volume hierarchies in computer graphics (Ng & Trifonov, 2003), memory hierarchies in computer system design (Benini, Macii, Macii, & Poncino, 2000), and concept hierarchies in knowledge discovery researches. We will discuss some works regarding to concept hierarchy generation here. In knowledge discovery research, we often like to organize data, especially textual data, into hierarchies since they are well perceived by humans. In one of the early work by Han and Fu (1994), they developed a technique for refining existing concept hierarchies and generating new concept hierarchies. However, their method can only applied to numerical attributes. McCallum and Nigam (1999) used a bootstrapping process to generate new terms from a set of human-provided keywords. This approach was then applied in portal site construction (McCallum, Nigam, Rennie, & Seymore, 2000). Human intervention is still required in their work. Probabilistic methods were widely used in exploiting hierarchy. Weigend, Wiener, and Pedersen (1999) proposed a two-level architecture for text categorization. The first level of the architecture predicts the probabilities of the meta-topic groups, which are groups of topics. This allows the individual models for each topic on the second level to focus on finer discrimination within the group. They used a supervised neural network to learn the hierarchy where topic classes were provided and already assigned. A different probabilistic approach by Hofmann (1999) used an unsupervised learning architecture called Cluster-Abstraction Model to organize groups of documents in a hierarchy. Liu and Yang (2007) used the link information within a Web page to build a topic hierarchy of a Website. Their method relies on the precise analysis and specification of link features which are implausible for normal texts. They improves their work (Yang & Liu, 2009) by modeling a Web site's link structure as a weighted directed structure and estimate the weights of edges by various methods. They then adapted the graph search algorithms to generate the Web directories. A close approach to our work was proposed by Chuang and Chien (2005), which uses the search results of search engines for feature extraction and applies a modified hierarchical agglomerative clustering algorithm to organize short text segments into shallow hierarchies. Their work resembles ours in two aspects: first, hierarchical clustering approach is used. Second, it also creates shallow, multi-way hierarchies instead of binary ones. However, our method does not rely on the support of search engines for feature extraction. Besides, our approach can simultaneously identify themes and organize documents merely using the contents of the textual documents.

Another approach for Web directory generation resembles the generation of topic maps since they both need to identify topics and discover relationships among topics. Rath (1999) discussed a framework for automatic generation of topic maps according to a so-called 'topic map template' and a set of generation rules. The structural information of topics is maintained in the template. They used a generator to interpret the generation rules and extract necessary information that fulfills the template to create the topic map. However, both the rules and the template are to be constructed explicitly and probably manually. Moore (2000) discussed topic map authoring and how software may support it. He argued that the automatic generation of topic maps is a useful first step in the construction of a production topic map. However, the real value of a topic map comes through the involvement of people in the process. This argument is true if the knowledge that contained in the topic maps can only be obtained by human efforts. Fully automatic generation process is possible only when such knowledge may be discovered from the underlying set of information resources through an automated process, which is generally known as knowledge discovery from texts or text mining. Böhm, Heyer, Quasthoff, and Wolff (2002) proposed a text mining infrastructure called Concept Composer using linguistic and statistical analysis techniques based on word collocations, and applied to the generation of topic maps. However, their method is not able to generate the relationships among topics and need human intervention.

2.2. Literature on hierarchy alignment

Alignment of concept hierarchies resembles the task of ontology mapping or alignment since a concept hierarchy is similar to an ontology in both structure and function. Many works on ontology alignment or mapping have been proposed (Choi, Song, & Han, 2006; Kalfoglou & Schorlemmer, 2005; Noy & Stuckenschmidt, 2005). According to Noy and Stuckenschmidt's classification, the ontology mapping discovery schemes can be based on common reference ontology, lexical similarity, structural similarity, user input, external resources such as Wordnet's annotation, or prior matching. Our method, however, cannot be easily classified in such classification. We may consider our approach a combined scheme of using lexical and structural similarities. Sornlertlamvanich, Kruengkrai, Tongchim, Srichaivattana, and Isahara (2005) proposed an approach to align concepts between two hierarchical ontologies based on term distributions. A close work of our method is the HICAL system (Ichise, Takeda, & Honiden, 2001). The HICAL system generated alignment rules for concepts using the statistic method in a top-down manner. They first examined the similarity of the root concepts of distinct hierarchies and generated a rule if these concepts are similar. This process is recursively applied to lower-level concepts to obtain more specific rules. Their method is similar to ours in respect of the use of data instances within concepts in the hierarchies. However, the alignment algorithm is significantly different.

Research on automatic hierarchy alignment on multilingual domain is still in its infancy. One of the approach is to develop translated version of some existing monolingual hierarchy such as WordNet. For example, one early work is the MultiWordNet project (Pianta, Bentivogli, & Girardi, 2002). They used an automatic procedure to select the most likely synsets in Princeton WordNet. These synsets were then provided to the lexicographers to actually build the Italian synsets. Another early work using translation-based approach was proposed by Daudé, Padró, and Rigau (1999). They used relaxation labeling method to select the corresponding English synset for a Spanish one. Levow, Dorr, and Lin (2000) also adopted translation-based approach in creating multilingual hierarchy based on existing monolingual ontologies, namely the Chinese HowNet and English Levin-based English verb classification. Another multilingual hierarchy alignment approach is based on classification of the concepts. Adar, Skinner, and Weld (2009) used a self-supervised classification process to classify and align Wikipedia infoboxes in four different languages. Total 26 features were extracted and fed in the classifier. Their approach needs no dictionaries to align multilingual infoboxes. However, extensive feature selection and extraction should be performed.

2.3. Literature on multilingual information retrieval/text mining

The aim of multilingual information retrieval (MLIR) is to provide users a way to search documents written in a different language from the query. Query translation thus plays a central role in MLIR research. Three different strategies for query translation were used (Oard & Dorr, 1996), namely dictionary-based, thesaurus-based, and corpus-based methods. We will briefly describe the corpus-based methods which resemble to ours. Corpus-based approach uses knowledge acquisition techniques to discover cross-language relationships and applies them to MLIR. We may divide this approach into three categories, namely word alignment, sentence alignment, and document alignment, based on the granularity of alignment. Word alignment can generate the finest bilingual corpora, which the relationships between words in different languages have been clearly defined automatically or manually. Brown (1996) used this kind of technique to construct a translation table for query translation. Chen and Chen (1994) stated that the precision of alignment will affect the quality of query translation. Good alignment methods should be developed to ensure the quality of corpus-based MLIR. For sentence alignment, an example is the work of Davis and Dunning (1996). Both word alignment and sentence alignment suffer from the fact that such alignments are not easy to achieve. On the other hand, document alignment is much easier. There are two types of corpora that could be used in document alignment, namely parallel corpora and comparable corpora. The former contains documents in multiple translations. The latter contains categories of multi-language documents that have the same topics. Comparable corpora are common since the multi-language copies of a topic are easier to obtain. An example is the work by Sheridan and Ballerini (1996). They aligned German and Italian news articles and constructed a translation dictionary. This dictionary is then used to generate target query. Talvensaari, Juhola, Laurikkala, and Järvelin (2007) proposed corpus-based translation method that might be used to update the references and gain some comparative insight.

Many multilingual text mining techniques, especially for machine learning approaches, are based on comparable or parallel corpora. Chau and Yeh (2004) generated fuzzy membership scores between Chinese and English terms and clustered them to perform MLIR. Lee and Yang (2003) also used self-organizing map (SOM) (Kohonen, 2001) to train a set of Chinese-English parallel corpora and generate two feature maps. The relationships between bilingual documents and terms are then discovered. Rauber, Merkl, and Dittenbach (2002) applied growing hierarchical self-organizing map (GHSOM) to cluster multilingual corpora which contain Russian, English, German, and French documents. However, they translated all documents into one language first before training. Thus they actually performed monolingual text mining. Since our work adopts GHSOM for benchmarking, we will briefly review the model. GHSOM is a neural network model modified from basic SOM. The major advantage of GHSOM is its hierarchical structure of expandable maps. A map could expand its size during training to achieve better result. Any neuron in the map could even expand to a lower level map when necessary. The expansion could proceed to lower layers.

We briefly summarize the GHSOM training algorithm in the following.

Step 1 (Initialization step) layer 0 contains a single neuron. The synaptic weight of this neuron, $\mathbf{w}_0$, is initialized to the average value of the input vectors:

$$\mathbf{w}_0 = \frac{1}{N}\sum_{1\leqslant i\leqslant N}\mathbf{x}_i, \tag{1}$$

where $\mathbf{x}_i$ is the ith training vector and N is the number of training vector. The mean quantization error mqe_0 is calculated as follow:

$$mqe_0 = \frac{1}{N}\sum_{1\leqslant i\leqslant N}\|\mathbf{x}_i - \mathbf{w}_0\|. \tag{2}$$

Step 2 (Map growing step) construct a small SOM $\mathcal{M}$, e.g. containing 2 × 3 neurons, below layer 0. This is layer 1. Train this layer by SOM algorithm in λ steps. Find the neuron which each training vector is labeled to. A training vector is labeled to a neuron if the synaptic weight of this neuron is closest to this vector. Calculate the mean quantization error of neuron n as follow:

$$mqe_n = \frac{1}{|X_n|}\sum_{i\in X_n}\|\mathbf{x}_i - \mathbf{w}_n\|, \tag{3}$$

where X_n is the set of training vectors that label to neuron n and $\mathbf{w}_n$ is the synaptic weight vector of neuron n. The mean quantization error of this map, denoted by $MQE_{\mathcal{M}}$, is the average of the mean quantization error of every neuron in this map. If $MQE_{\mathcal{M}}$ exceeds a fixed percentage of mqe_0, i.e. $MQE_{\mathcal{M}} \geqslant \tau_{\mathcal{M}} \times mqe_0$, a new row or a new column of neurons will be inserted to this SOM. This new row or column is added in the neighbor of the error neuron e with the highest mqe_e. Whether a row or a column is added is guided by the location of the most dissimilar neurons to e. The insertion process proceeds until $MQE_{\mathcal{M}} < \tau_{\mathcal{M}} \times mqe_0$.

Step 3 (Hierarchy expansion step) all neurons in the maps of a layer are examined to determine if they need further expansion to new maps in next layer. A neuron with large mean quantization error, i.e. an error greater than a percentage of mqe_0, will expand to a new SOM in next layer. The percentage is denoted by τ_u. When neuron i is selected to be expanded, the expanded SOM is trained using those vectors labeled to i. Each new SOM could grow in the way described in Step 2.

Step 4 Repeat Step 3 until no neuron in all maps of a layer needs expansion.

3. Web directory generation

In this section we will first describe the method to automatically generate monolingual Web directories. To obtain the Web directories, we first perform a clustering process on a corpus of Web pages. We then apply an automatic generation process to the clustering result and obtain the Web directories. In this section, we will start with the preprocessing steps, follow by the clustering process by SOM learning algorithm. Two labeling processes are then applied to the trained result to obtain two feature maps which characterize the relationships among Web pages and keywords, respectively. The Web directory generation process is then applied to these two maps to develop the Web directory.

3.1. Web page preprocessing and encoding

A document should be converted to proper form before SOM training. Here the proper form is a vector that catches essential (semantic) meaning of the document. In this work we adopt bilingual parallel corpora that contain Chinese and English documents. The encoding of English documents into vectors has been well addressed (Salton, 1989). Processing steps such as word segmentation, stopword elimination, stemming, and keyword selection are often used in extracting representative keywords from a document. In this work, we first used a common segmentation program to segment possible keywords. A part-of-speech tagger is also applied to these keywords so that only nouns are selected. These selected keywords may contain stopwords that have trivial meanings. These stopwords will be removed to reduce the number of keywords. Stemming process will be also applied to convert each keyword to its stem (root). This will further reduce the number of keywords. After these processing steps, we will obtain a set of keywords that should be representative to this document. All keywords of all documents are collected to build a vocabulary for English keywords. This vocabulary is denoted as V_E. A document is encoded into a binary vector according to those keywords that occurred in it. When a keyword occurs in this document, the corresponding element of the vector will have value 1; otherwise, the element will have value 0. With this scheme, a document E_j will be encoded into a binary vector $\mathbf{E}_j$. The size (number of elements) of $\mathbf{E}_j$ is just the size of the vocabulary V_E, i.e. $|V_E|$. We use binary vector scheme to encode the documents and ignore any kind of term weighting schemes. We decide to use the binary vector scheme due to the following reasons. First, we intend to cluster documents according to the co-occurrence of the words, which is irrelevant to the weights of the individual words. Second, our experiments showed no advantage in the clustering result by using term weighting schemes (classical *tf* and $tf \cdot idf$ schemes were used). As a result, we believe the binary scheme is adequate for our need.

The processing of Chinese documents differs significantly from their English counterparts in several aspects. First, a Chinese sentence contains a list of consecutive Chinese letters. A Chinese letter, however, carries little meaning individually without referring to its context. Several letters are often combined to give specific meaning and are basic constituents of a sentence in Chinese. Here we will mention such combined letters as a word for consistency with English words. The segmentation of Chinese words is more difficult than English words because we have to separate these consecutive letters into a set of words. There are several segmentation schemes for Chinese words. We adopt the segmentation program developed by the CKIP team of Academia Sinica to segment words (Chen & Bai, 1998). Another difference between Chinese and English text processing is the need for stemming. Chinese words require no stemming in general. Stopword elimination could be applied to Chinese words as in English. However, unlike English stopwords, there is no standard stopword list available. In this work, we omit this process since we select only nouns as keywords. The reason for selecting nouns as keywords is that they will be used as category themes later in Web directory generation process. We believe that only nouns can provide meaningful guidance for users in navigating the directory. A problem of omitting the stopword elimination is that some nouns such as 'homepage' will occur in lots of documents. This will not affect the clustering process since (1) the distance calculation during clustering is contributed by many keywords that will not be affected much by several common keywords and (2) the distance between documents will not be affected suppose they all contains these common keywords. As in English case, the selected keywords are collected to build Chinese vocabulary V_C. Each Chinese document C_j is encoded into a vector $\mathbf{C}_j$ in the same manner as English. The size of $\mathbf{C}_j$ is just the size of the vocabulary V_C, i.e. $|V_C|$.

3.2. Feature map generation

In this sub-section we will briefly review a method to organize documents into clusters by their co-occurrence similarities using SOM (Lee & Yang, 1999). The documents in the corpus are first encoded into a set of vectors as described in Section 3.1. We intend to organize these documents into a set of clusters such that similar documents will fall into the same cluster or nearby clusters. The unsupervised learning algorithm of SOM networks (Kohonen, 2001) meets our needs. The SOM algorithm organizes a set of high-dimensional vectors into a two-dimensional map of neurons according to the similarities among the vectors. Similar vectors, i.e. vectors with small distances, will map to the same or nearby neurons after the training (or learning) process. That is, the similarity between vectors in the original space is preserved in the mapped space. Applying the SOM algorithm to the document vectors, we actually perform a clustering process on the documents. Here a neuron in the map can be treated as a cluster. Similar documents will fall into the same or neighboring neurons (clusters). Besides, the similarity of two clusters can be measured by the geometrical distance between their corresponding neurons. To decide the cluster to which a document or a word belongs, we apply two labeling processes to the documents and the words, respectively. After the document labeling process, each document will associate with a neuron in the map. We record such associations and obtain the document cluster map (DCM). In the same manner, we label each word to some neuron in the map and obtain the keyword cluster map (KCM). We then use these two maps to generate the hierarchies.

We define some denotations and describe the training process here. Let $\mathbf{x}_i = \{x_{i_n} | 1 \leqslant n \leqslant N\}, 1 \leqslant i \leqslant M$, be the encoded vector of the *i*th document in the corpus, where N is the number of indexed terms and M is the number of the documents. We use these vectors as the training inputs to the SOM network. The network consists of a regular grid of neurons which each has N synapses. Let $\mathbf{w}_j = \{w_{j_n} | 1 \leqslant n \leqslant N\}, 1 \leqslant j \leqslant J$, be the synaptic weight vector of the *j*th neuron in the network, where J is the number of neurons in the network. We train the network by the SOM algorithm:

Step 1 Randomly select a training vector $\mathbf{x}_i$ from the corpus.

Step 2 Find the neuron j with synaptic weight vector $\mathbf{w}_j$ which is the closest to $\mathbf{x}_i$, i.e.

$$\|\mathbf{x}_i - \mathbf{w}_j\| = \min_{1 \leqslant k \leqslant J} \|\mathbf{x}_k - \mathbf{w}_j\|. \tag{4}$$

Step 3 For each neuron l in the neighborhood of neuron j, update its synaptic weights by

$$\mathbf{w}_l^{\text{new}} = \mathbf{w}_l^{\text{old}} + \alpha(t)(\mathbf{x}_i - \mathbf{w}_l^{\text{old}}), \tag{5}$$

where $\alpha(t)$ is the training gain at time stamp t.

Step 4 Increase time stamp t. If t reaches the preset maximum training time T, halt the training process; otherwise decrease $\alpha(t)$ and the neighborhood size, goto Step 1.

The training process stops after time T which is sufficiently large so that every vector may be selected as training input for certain times. The training gain and neighborhood size both decrease when t increases.

When the document clustering process is accomplished, we then perform a labeling process on the trained network to establish the association between each document and one of the neurons. The labeling process is described as follows. Each document's feature vector $\mathbf{x}_i, 1 \leqslant i \leqslant M$ is compared to the synaptic weight vectors of every neuron in the map. We then label the ith document to the jth neuron if they satisfy Eq. (4). After the labeling process, each document is labeled to some neuron or, from a different point of view, each neuron is labeled by a set of documents. We record the labeling result and obtain the DCM. In the DCM, each neuron is labeled by a list of documents which are considered similar and are in the same cluster.

We construct the KCM by labeling each neuron in the trained network with certain keywords. Such labeling is achieved by examining the neurons' synaptic weight vectors, and is based on the following observation. Since we adopt binary representation for the document feature vectors, ideally the trained map should consist of synaptic weight vectors with component values near either 0 or 1. Since a value of 1 in a document vector indicates the presence of a corresponding word in that document, a component with value near 1 in a synaptic weight vector also shows that such neuron has recognized the importance of the word and tries to 'learn' the word. According to such interpretation, we design the following word labeling process. For the weight vector of the jth neuron $\mathbf{w}_j$, if its nth component exceeds a predetermined threshold, the corresponding word of that component, i.e. the nth word in the vocabulary, is labeled to this neuron. By virtue of the SOM algorithm, a neuron may be labeled by several keywords which often co-occur in a set of documents, making a neuron a keyword cluster.

The KCM autonomously clusters keywords according to their similarity of co-occurrence. Keywords that tend to occur simultaneously in the same document will be mapped to neighboring neurons in the map. For example, the translated Chinese words for 'neural' and 'network' often occur simultaneously in a document. They will map to the same neuron, or neighboring neurons, in the map because their corresponding components in the encoded document vector are both set to 1. Thus a neuron will try to learn these two keywords simultaneously. On the contrary, words that do not co-occur in the same document may map to much distant neurons in the map. Thus we can reveal the relationship between two keywords according to their corresponding neurons in the KCM.

3.3. Web directory generation

To generate a hierarchy, we should first cluster documents by the SOM using the method described in Section 3.2 to obtain the DCM and the KCM. As we mentioned before, a neuron in the DCM represents a cluster of documents so we will use the terms 'cluster' and 'neuron' interchangeably in the following text. Documents which are labeled to the same neuron, or neighboring neurons, usually contain keywords that often co-occur in these documents. By virtue of the SOM algorithm, the synaptic weight vectors of neighboring neurons have the least difference comparing to those of distant neurons. That is, similar document clusters will correspond to neighboring neurons in the DCM. Thus we may generate a cluster of similar clusters, or a super cluster, by congregating neighboring neurons. This should essentially create a two-level hierarchy such that the parent node is the constructed super cluster and the child nodes are the clusters that compose the super cluster. The hierarchy generation process can be further applied to every super cluster to establish the next higher level of this hierarchy. The overall hierarchy can then be established iteratively until a stop criterion is satisfied.

3.3.1. Super cluster construction

Here we use a top-down approach to generate hierarchies based on the construction of super clusters. To form a super cluster, we first define the distance between two clusters:

$$D(i,j) = \|\mathbf{G}_i - \mathbf{G}_j\|, \tag{6}$$

where i and j are the neuron indices of the two clusters and $\mathbf{G}_i$ is the two-dimensional grid coordinates of neuron i in the map. For a square formation of neurons, $\mathbf{G}_i = (i \bmod \sqrt{J}, i \operatorname{div} \sqrt{J})$. $\|\mathbf{G}_i - \mathbf{G}_j\|$ measures the Euclidean distance between the two coordinates $\mathbf{G}_i$ and $\mathbf{G}_j$. We also define the dissimilarity between two clusters:

$$\mathcal{D}(i,j) = \|\mathbf{w}_i - \mathbf{w}_j\|, \tag{7}$$

where $\mathbf{w}_i$ is the synaptic weight vector of neuron i. Now we try to find some significant clusters among all clusters as super clusters. Since neighboring clusters in the map represent similar clusters, we can determine the significance of a cluster by examining the overall similarity that is contributed by its neighboring clusters. Such similarity, namely the *aggregated cluster similarity*, is defined by:

$$S_i = \frac{1}{|B_i|} \sum_{j \in B_i} \frac{doc(i)doc(j)}{F(D(i,j), \mathcal{D}(i,j))}, \tag{8}$$

where $doc(i)$ is the number of documents associated with neuron i, i.e. cluster i, in the DCM and B_i is the set of neuron indices in the neighborhood of neuron i. The function $F : R^+ \to R^+$ is a monotonically increasing function which takes $D(i,j)$ and $\mathcal{D}(i,j)$ as arguments.

3.3.2. Determining dominating clusters

The super clusters are developed from a set of dominated clusters in the map. A dominated cluster is a cluster which has locally maximal aggregated cluster similarity. We may select all dominated clusters in the map by the following algorithm:

Step 1 Find the cluster which has the largest aggregated cluster similarity among all clusters under consideration. This cluster is a dominated cluster.

Step 2 Eliminate its neighboring clusters so that they will not be considered as dominated clusters in succeeded steps.

Step 3 If

1. there is no cluster left, or
2. the number of dominated clusters N_D exceeds a predetermined value, or
3. the level of generated hierarchy so far exceeds a predetermined depth σ,

stops the process. Otherwise, decrease the neighborhood size in Step 2 and goto Step 1.

The algorithm finds dominated clusters from all clusters under consideration and creates a level of the hierarchy. The overall process is depicted in Fig. 2. As shown in the figure, we may develop a two-level hierarchy by using a super cluster (or dominated cluster) as the parent node and the clusters which are similar to the super cluster as the child nodes. In Fig. 2, super cluster k corresponds to the dominating cluster k. The neighboring clusters of cluster k are used as the child nodes in the hierarchy.

A dominated cluster is the centroid of a super cluster, which contains several child clusters. We will use the neuron index of a neuron as the index of the cluster associated with it. For consistency, the index of a dominated cluster is used as the index of its corresponding super cluster. The child clusters of a super cluster can be found by the following rule: cluster i belongs to super cluster k if for all l super clusters: $\mathcal{D}(i,k) < \mathcal{D}(i,l)$.

3.3.3. Constructing hierarchy

The above process creates a two-level hierarchy. In the following we will show how to obtain the overall hierarchy. In the first application of the super cluster generation process (denoted by STAGE-1), we obtain a set of super clusters. We aggregate these super clusters under one pseudo root node and form the first and second levels of the hierarchy. To find the children of the super clusters obtained in STAGE-1, we may apply the super cluster generation process on each super cluster (STAGE-2). Notice that in STAGE-2 we only consider clusters which belong to the same super cluster. A set of child nodes will be obtained and be used as the third level of the hierarchy. The overall hierarchy can then be revealed by recursively applying the same process to each new-found super cluster (STAGE-n). We decrease the size of neighborhood in selecting dominated clusters when the super cluster generation process proceeds. This will produce a reasonable number of levels for the hierarchies, as we will discuss later.

3.3.4. Parameter setting and discussions

The neighborhood B_i in calculating supporting cluster similarity of a cluster i may be arbitrarily selected. Two common selections are circular neighborhood and square neighborhood. In our experiments, the shapes of the neighborhood are not crucial. It is the size of the neighborhood, denoted by N_{c1}, which matters. Different sizes of neighborhood may result in different selections of dominated clusters. Small neighborhood may not capture the necessary support from similar clusters. On the other hand, without proper weighting, a large N_{c1} will incorporate the support from distant clusters which may not be similar to the cluster under consideration. Besides, large neighborhoods have the disadvantage of costing much computation time.

Another usage of neighborhood is to eliminate similar clusters in the super cluster generation process. In each stage of the process, this neighborhood size, denoted by N_{c2}, has a direct influence to the number of dominated clusters. Large neighborhoods will eliminate many clusters and result in less dominated clusters. Oppositely, a small neighborhood produces a large number of dominated clusters. We must decrease the neighborhood size when the process proceeds because the number of neurons under consideration is also decreased.

In Step 3 of the super cluster generation process algorithm we set three stop criteria. The first criterion stops finding super clusters if there is no neuron left for selection. This is a basic criterion but we need the second criterion, which limits the

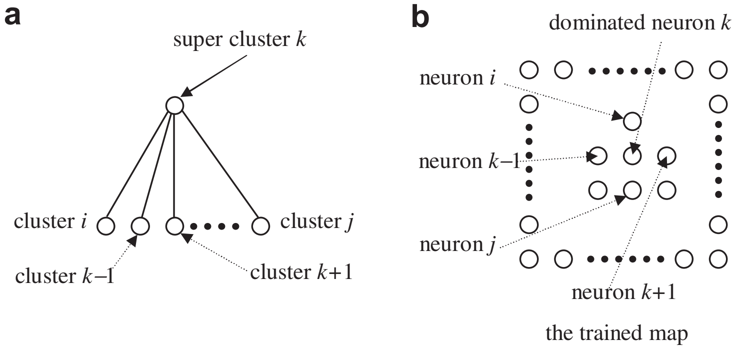


Fig. 2. (a) A two-level hierarchy comprises a super cluster as root node and several clusters as child nodes. (b) The dominated cluster k is selected and used as a super cluster. Its neighboring clusters compose the super cluster. We only show a possible construction of the hierarchy here.

number of dominated clusters, to constrain the breadth of hierarchies. The lack of the second criterion may result in shallow hierarchies with too many nodes in each level if the neighborhood size is considerably small. An extreme case happens when the neighborhood size is 0. In such case Step 2 of the algorithm will not eliminate any cluster. As a result, every cluster will be selected as dominated clusters and we will obtain a two-level hierarchy with J level-2 nodes. Determining an adequate neighborhood size as well as a proper number of dominated clusters is crucial for obtaining an acceptable result. The third criterion constrains the depth of a hierarchy. If we allow a hierarchy having large depth then we will obtain a 'slim' hierarchy. Notice that setting large depth may cause no effect because the neighborhood size and the number of dominated clusters may already satisfy the stop criterion. An ad-hoc heuristic rule used in our experiments is to determine the maximum depth d if it satisfies the following rule:

$$\text{Find the smallest natural number } d \text{ such that} \frac{J}{K^{2d}} < 1, \tag{9}$$

where K is the dimension of the neighborhood and is defined as a ratio to the map's dimension. For example, if the map contains an 8×8 grid of neurons, $K = 4$ means that the dimension of the neighborhood is fourth of the map's dimension, which is two in this case. The depth d which satisfies Eq. (9) is then 2.

Notice that there may exist some 'spare' clusters which are not used after the hierarchy generation process. These clusters are not significant enough to be a dominated cluster of a super cluster in any stage of the process. Although we can extend the depths in the hierarchy generation process to enclose all clusters into the hierarchy, sometimes we may decide not to do so because we want a higher document-cluster ratio. That is, a cluster should contain a significant amount of documents. For example, if all clusters contain very few documents, it is not wise to use all clusters in the hierarchy because we may have a hierarchy which contains many nodes without too much information. To avoid producing such over-sized hierarchy, we may adopt a different approach. When the hierarchy has been created and there still exists some spare clusters, we simply assign each spare cluster to its nearest neighbors. This in effect merges the document clusters associated with these spare clusters into the hierarchy. The merging process is necessary to achieve a reasonable document-cluster ratio.

In the generated hierarchy, a node represents an individual neuron which is associated with a cluster of documents. We also assign labels to each node to exhibit the themes of the clusters. The labels of a node are those keywords associated with the same neuron in the KCM. In this regard, a node in the hierarchy is labeled by a document cluster as well as a keyword cluster. Nodes in higher levels should have more general coverage than those in lower levels. We will use these labels to align hierarchies, which will be discussed in the next section.

3.4. Evaluation of the quality of generated hierarchies

We introduce a measure to evaluate the quality of the constructed bilingual hierarchies. Let L_1 and L_2 be the corpora of two different languages. Let H_1 and H_2 be the hierarchies constructed using L_1 and L_2, respectively. Let C_i be a document in L_1 and E_i be its counterpart in L_2. We also denote $\mathcal{C}_k$ and $\mathcal{E}_k$ as the document cluster associated with node k in H_1 and H_2, respectively. We then calculate the mean inter-document path length between each pair of documents in $\mathcal{C}_k$ or $\mathcal{E}_k$ as follows:

$$P_k = \frac{\sum_{1 \leqslant i \leqslant j \leqslant |\mathcal{C}_k|} dist(C_i, C_j)}{\binom{|\mathcal{C}_k|}{2}}, \tag{10}$$

where function $dist(C_i, C_j)$ returns the shortest path length between E_i and E_j, which are the counterparts of C_i and C_j, in H_2 and $|\mathcal{C}_k|$ denotes the number of documents associated with node k. An example is depicted in Fig. 3. In the figure, $\mathcal{C}_k$ contains

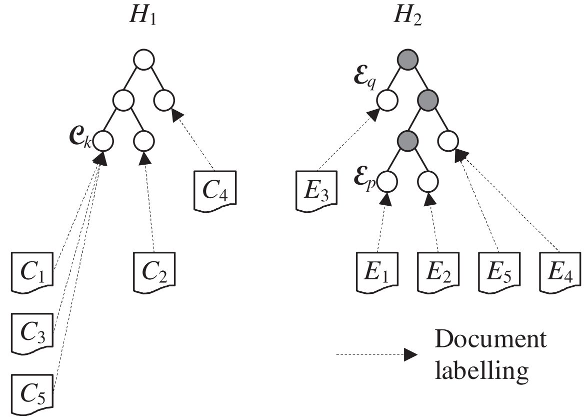


Fig. 3. An example of the bilingual hierarchies.

documents C_1, C_3, and C_5. Therefore, P_k is the average of $dist(C_1, C_3), dist(C_1, C_5)$, and $dist(C_3, C_5)$, which are the path lengths between E_1 and E_3, E_1 and E_5, and E_3 and E_5, respectively. In this example, $P_k = (4+3+3)/3 = 3.33$. The quality of the bilingual hierarchies can then be measured by the average of all P_k, denoted by $\overline{P_k}$, over entire H_1. A small $\overline{P_k}$ means better quality since it shows that the counterparts of related documents are also related.

$\overline{P_k}$ could also be used to justify the quality of the generated hierarchies. Since SOM uses random initial weights, the result could be varied for different trainings. To ensure that the two hierarchies have comparable result, we may generate a hierarchy H_1 using L_1 first and repeatedly generate hierarchies using L_2. The $\overline{P_k}$ between H_1 and the newly-created hierarchy H_2 using L_2 is then measured. We stop SOM training for L_2 as soon as $\overline{P_k}$ is lower than some threshold. When there is no such H_2 could yield $\overline{P_k}$ that is lower than this threshold, the one with the lowest $\overline{P_k}$ should be selected. Although this scheme will spend additional time in generating acceptable hierarchies, we should apply it to ensure that both hierarchies are comparable and the discovered associations are accurate.

4. Multilingual Web directory generation

Monolingual Web directories such as those generated by the method in Section 3.3 should provide users a convenient way to find the Web pages they like. However, most users should rely on some translation engines to obtain Web pages in the same topic written in non-native languages of them. It will be convenient for users to have a multilingual Web directory that contains labels and Web pages written in different languages. For example, when a user tried to find a manual for a foreign-made digital camera, he may want to collect all possible Web pages written in various languages besides his native language. If only monolingual directories were available, he is forced to navigate through a foreign-language directory to find the goal pages even he is unfamiliar with the language. This often causes lots of time waste or, in worse, failure to reach his goal. On the other hand, when a multilingual Web directory was available, users can navigate through the directory and reach his goal pages, no matter written in what languages, using his familiar language. Such navigation process may also broaden his search, especially for uncommon language users. For native English users, multilingual Web directories may not be so attractive since most of the Web pages were written in English. However, non-English users will find multilingual Web directories convenient. Users who intensively access multilingual information sources such as reporters and technicians may also find multilingual directories useful. Construction of multilingual Web directories is not an easy task. In this section we will demonstrate a method to align two monolingual hierarchies generated using the method described in Section 3.3. Let $H_1 = \{V_1, E_1\}$ and $H_2 = \{V_2, E_2\}$ be the hierarchies generated using Web pages of two different languages, where V_1 and E_1 denote the set of vertices (nodes) and edges, respectively, of H_1. Likewise, V_2 and E_2 denote the set of vertices and edges, respectively, of H_2. We will first introduce an alignment method to map nodes between H_1 and H_2. That is, a mapping $\mathcal{M} : V_1 \to V_2$ will be derived. According to such mapping, we can construct a multilingual Web directory to access Web pages of both languages. We will discuss the alignment and Web directory construction processes in the following sub-sections.

4.1. Alignment of monolingual Web directories

To map a category $\mathcal{C}_k \in V_1$ to some category $\mathcal{E}_l \in V_2$, a simple solution is to translate keywords associated with $\mathcal{C}_k$ to the language used to construct H_2 and obtain their associations. This is implausible here since we could not use any dictionary or thesaurus in the whole process. What we should rely on are the trained hierarchies. We consider $\mathcal{C}_k$ and $\mathcal{E}_l$ to be related if they have similar themes. Meanwhile, the theme of a category could be determined by the documents labeled to it. Thus we could associate two categories according to their corresponding document clusters. To define such associations, we use parallel corpora to train the SOM in this work. The advantage of using parallel corpora is that the correspondence between a document and its counterpart in another language is known a priori. We should then use such correspondences to associate categories of different languages.

To define the association between two categories in different directories, two types of similarity measurements were defined here, namely semantic similarity and structural similarity. The first measurement measures the semantic similarity between two categories. The other measures the structural similarity between categories in different hierarchies. First we should define the semantic similarity between two categories in different hierarchies. Here we define two categories being semantically similar if they have similar themes, which are discovered by method described in Section 3.2. The problem is that these themes are discovered from monolingual documents and the relationships between those themes in different hierarchies cannot be found directly. A typical solution is to translate themes in one language to another. In this work, we would not adopt such approach and would develop a more 'self-discovery' approach. We argue that two categories have similar themes if they contain similar documents. Since we use parallel corpora to develop the hierarchies, the associations between documents in different languages are intrinsically defined. We should use such associations to define the similarity between categories in different hierarchies.

4.1.1. Calculating semantic similarity

To obtain the semantic similarity between $\mathcal{C}_k$ in H_1 with $\mathcal{E}_l$ in H_2, we should refer to the DCMs that reveal the semantic similarity between documents in each language. For each document C_i in $\mathcal{C}_k$, we first locate the neuron to which its English counterpart E_i labeled. The similarity between $\mathcal{C}_k$ and $\mathcal{E}_l$ can then be obtained by calculating the average distance between each $E_j \in \mathcal{E}_l$ and each E_i, where $C_i \in \mathcal{C}_k$, in the DCM for English documents. That is, we should measure the semantic similarity between $\mathcal{C}_k$ and $\mathcal{E}_l$, denoted by $S_s(\mathcal{C}_k, \mathcal{E}_l)$, as follow:

$$S_s(\mathcal{C}_k, \mathcal{E}_l) = \frac{1}{|\mathcal{C}_k||\mathcal{E}_l|} \tag{11}$$

In Eq. (11), E_i is the counterpart document of C_i and $G(E_i)$ is the two-dimensional coordinates of the neuron to which E_i is labeled in the DCM for English documents. $\|\cdot\|$ is the norm of a vector. We add 1 to the nominator to avoid the case of dividing by zero. The equation is in inverse form so that a large value of S_s means that $\mathcal{C}_k$ and $\mathcal{E}_l$ are similar since the locations of neurons to which every $E_j \in \mathcal{E}_l$ labeled are all close to E_i, which is the counterpart of $C_i \in \mathcal{C}_k$. In the extreme case, when all documents in $\mathcal{C}_k$ have their counterparts all in $\mathcal{E}_l, S_s(\mathcal{C}_k, \mathcal{E}_l)$ will be 1 which means that $\mathcal{C}_k$ and $\mathcal{E}_l$ are semantically identical. Fig. 4 depicts the calculation of S_s.

4.1.2. Incorporating structural similarity

Two related clusters may have insignificant semantic similarity because related documents in these clusters are labeled to neighboring neurons instead of the same neuron. Besides, there may exist some spurious documents in these related

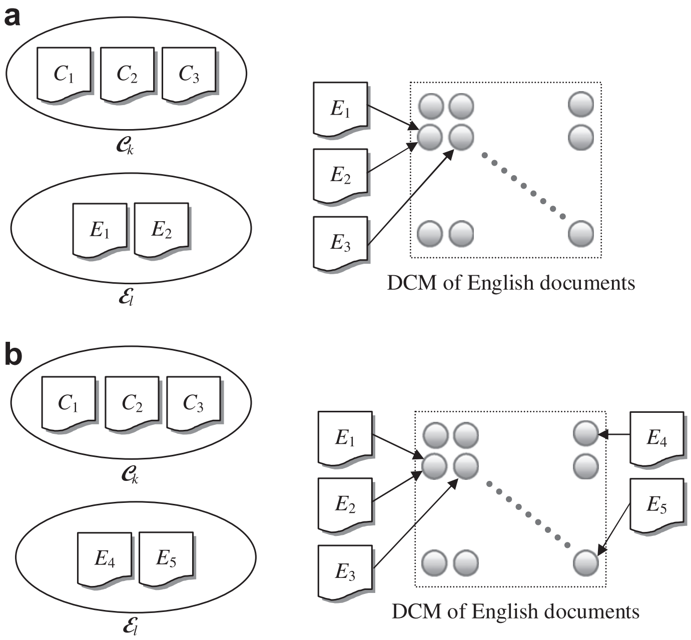


Fig. 4. The calculation of semantic similarity between (a) semantic related clusters and (b) semantic unrelated clusters. The circles in the DCM depict the neurons and the arrows depict the document labeling.

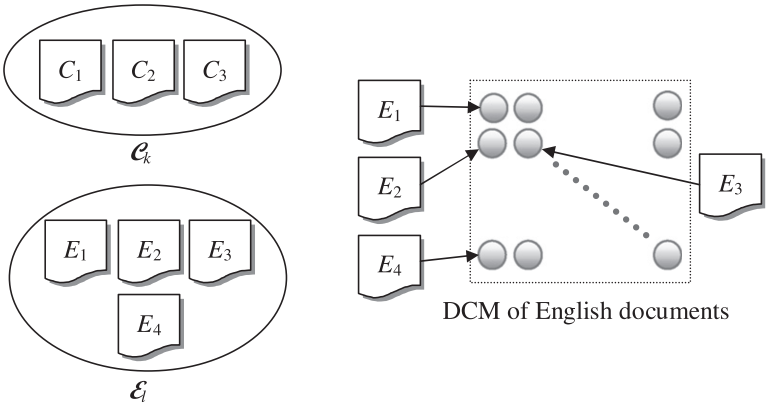


Fig. 5. Two related clusters may have insignificant semantic similarity due to spurious labeling.

clusters. An example is shown in Fig. 5. To compensate the effect of such misplacement, we should allow topological relatedness being incorporated into similarity calculation. The second similarity measures the topological relatedness, or structural similarity, between $\mathcal{C}_k$ and $\mathcal{E}_l$. Here we define the topological relatedness as the amount of related documents in neighboring clusters on the hierarchies. For a cluster $\mathcal{C}_k$ in H_1, we will calculate the structural similarity of it from all its neighboring clusters. Let $\mathcal{N}_{\mathcal{C}_k}$ be the set of neighboring clusters of $\mathcal{C}_k$. The structural similarity between $\mathcal{C}_k$ and a cluster $\mathcal{E}_l$ in H_2, denoted by $S_t(\mathcal{C}_k, \mathcal{E}_l)$, is defined as follow:

$$S_t(\mathcal{C}_k, \mathcal{E}_l) = \frac{1}{|C_k| \sum_{\mathcal{C}_p \in \mathcal{N}_{\mathcal{C}_k}} |\mathcal{C}_p|} \sum_{\mathcal{C}_p \in \mathcal{N}_{\mathcal{C}_k}} \sum_{C_i \in \mathcal{C}_k, C_j \in \mathcal{C}_p} Q(\mathcal{C}_k, \mathcal{C}_p) D(E_i, E_j), \tag{12}$$

where

$$Q(\mathcal{C}_k, \mathcal{C}_p) = \begin{cases} q_1 & \text{if } \mathcal{C}_p \text{ is the parent of } \mathcal{C}_k \\ q_2 & \text{if } \mathcal{C}_p \text{ is a child of } \mathcal{C}_k \end{cases}, \tag{13}$$

and

$$D(E_i, E_j) = \begin{cases} 2 & \text{if } E_i \in \mathcal{E}_l, E_j \in \mathcal{E}_m \in \mathcal{N}_{\mathcal{E}_l} \text{ and } L(\mathcal{E}_m) - L(\mathcal{E}_l) = L(\mathcal{C}_p) - L(\mathcal{C}_k) \\ 1 & \text{if } E_i \in \mathcal{E}_l, E_j \in \mathcal{E}_m \in \mathcal{N}_{\mathcal{E}_l} \text{ and } L(\mathcal{E}_m) - L(\mathcal{E}_l) \neq L(\mathcal{C}_p) - L(\mathcal{C}_k) \\ 0 & \text{otherwise} \end{cases}. \tag{14}$$

The function L returns the level of a cluster in the hierarchy. We use the function $Q(\mathcal{C}_k, \mathcal{C}_p)$ to differentiate the contributions from parent and child clusters. When we prefer the case that the two clusters in comparison have their parents also related to the case that their children are related, we should set $q_1 > q_2$ to allow parent clusters to contribute more in calculating the structural similarity. Typically we will set q_1 and q_2 to both positive values and let $q_1 + q_2 = 1$. $D(E_i, E_j)$ has nonzero values when E_i, which is the counterpart of C_i, and E_j are in neighboring clusters. When these clusters have the same parent/child relationship as $\mathcal{C}_k$ and $\mathcal{C}_p$, $D(E_i, E_j)$ will have larger value of 2. On the other hand, when they are neighbors but their parent/child relationship is different from that of $\mathcal{C}_k$ and $\mathcal{C}_p$, the value of $D(E_i, E_j)$ will be smaller, i.e. 1.

4.1.3. Overall similarity

The overall similarity between $\mathcal{C}_k$ and $\mathcal{E}_l$ is a weighted sum of their semantic similarity and structural similarity:

$$S(\mathcal{C}_k, \mathcal{E}_l) = S_s(\mathcal{C}_k, \mathcal{E}_l) + \beta S_t(\mathcal{C}_k, \mathcal{E}_l), \tag{15}$$

where β is the weighting coefficient to scale the contributions from the two similarities. Note that $S_s(\mathcal{C}_k, \mathcal{E}_l)$ has been normalized to the interval (0,1]. However, $S_t(\mathcal{C}_k, \mathcal{E}_l)$ has no fixed upper bound although we have tried to normalize it. Different values of q_1 and q_2 as well as the number of cluster pairs with/without the same parent/child relationships will all affect the upper bound of $S_t(\mathcal{C}_k, \mathcal{E}_l)$. This insufficiency should not deteriorate the overall result since the upper bound will not far from 1 in average after normalization.

Actually, we may measure the structural similarity from the DCM directly using the locations of neurons. However, this will only consider horizontal relationships without vertical ones. Hierarchical relationships provide not only proximity relatedness between two categories, but also subset relatedness. Therefore, we decide to derive the structural similarity from the developed hierarchies rather than DCMs directly.

Cluster $\mathcal{C}_k$ is associated with cluster $\mathcal{E}_l$ if the following condition holds:

$$S(\mathcal{C}_k, \mathcal{E}_l) = \arg\max_m S(\mathcal{C}_k, \mathcal{E}_m). \tag{16}$$

In this section we demonstrate method to find the associated English cluster of a Chinese cluster. We could also reverse the roles of them and obtain the associated Chinese cluster of an English cluster. Note that the resulting associations may be different but still comply with the requirement of being semantically related. By reversing the roles, we can actually obtain two sets of associations for each training. However, we believe choosing one of them will be enough since the experiments exhibited similar result, as shown in Section 5.

4.2. Multilingual Web directory generation

A multilingual Web directory can be easily constructed after the association between each Chinese cluster and some English cluster is found. The monolingual Web directories can be merged into a multilingual one using the discovered associations. A simple merging scheme is to merge each $\mathcal{C}_k$ with its associated $\mathcal{E}_l$ which satisfies Eq. (16) and form a new cluster $\mathcal{C}'_k$. All documents labeled to either $\mathcal{C}_k$ or $\mathcal{E}_l$ are all labeled to the new cluster $\mathcal{C}'_k$. It is clear that the new hierarchy has the same structure as H_1. The labels on $\mathcal{C}'_k$ are the union of the two keyword clusters labeled to neuron k and neuron l in the KCMs of Chinese and English documents, respectively. Fig. 6 demonstrates the merging process of a Chinese cluster and its associated English cluster. We can also construct the Web directory based on the English hierarchy and obtain a multilingual one with structure the same as the English hierarchy.

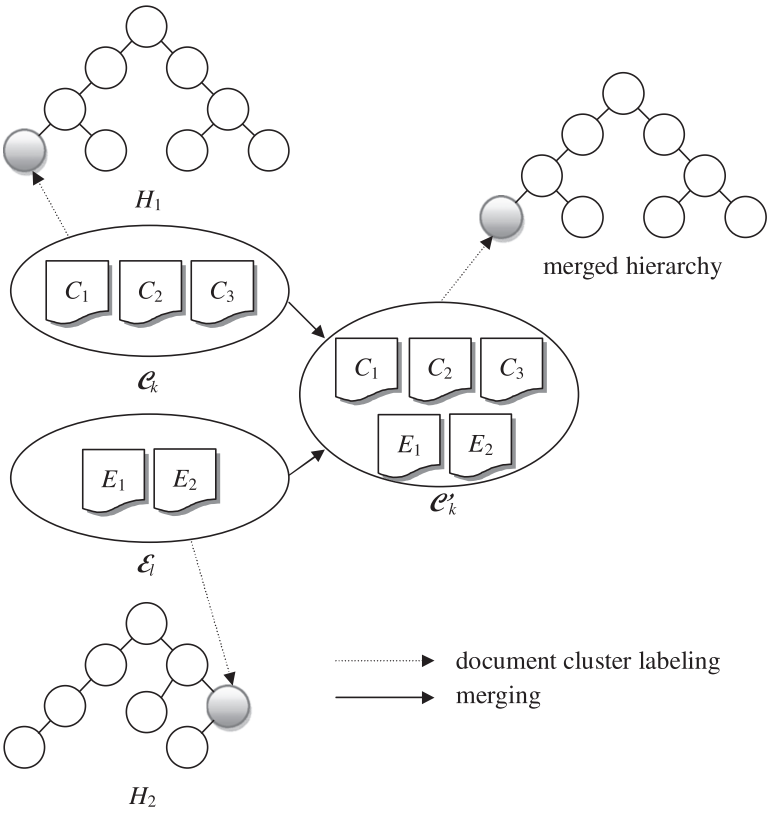


Fig. 6. Two related clusters are merged into single cluster in multilingual hierarchy.

5. Experimental result

We constructed the bilingual parallel corpora by collecting parallel documents from Sinorama corpus.[1] The corpus contains segments of bilingual articles of Sinorama magazine.[2] An example article is shown in Fig. 7. An advantage of this corpus is that each Chinese article was faithfully translated into English in a sentence-by-sentence manner. We collected two sets of parallel documents in our experiments to demonstrate the scalability of our method. Each document is a segment of an article in the original corpus. The first set, denoted as Corpus-1, contains 976 parallel documents. This is a relatively small corpus that we majorly used for explanation purpose. The other set, denoted as Corpus-2, contains 10672 parallel documents. This corpus resembles to a medium-size corpus that may fit the scale of a real application. Each Chinese document had been segmented into a set of keywords though the segmentation program developed by the CKIP team of Academia Sinica.[3] The program is also a part-of-speech tagger. We selected only nouns and discarded stopwords. As a result, we have vocabularies of size 3436 and 12941 for Corpus-1 and Corpus-2, respectively. For English documents, common segmentation program and part-of-speech tagger are used to convert them into keywords. Stopwords were also removed. Furthermore, Porter's stemming algorithm (Porter, 1980) was used to obtain stems of English keywords. Finally, we obtained two English vocabularies of size 3711 and 13723 for Corpus-1 and Corpus-2, respectively. These vocabularies were then used to convert each document into a vector, as described in Section 3.1.

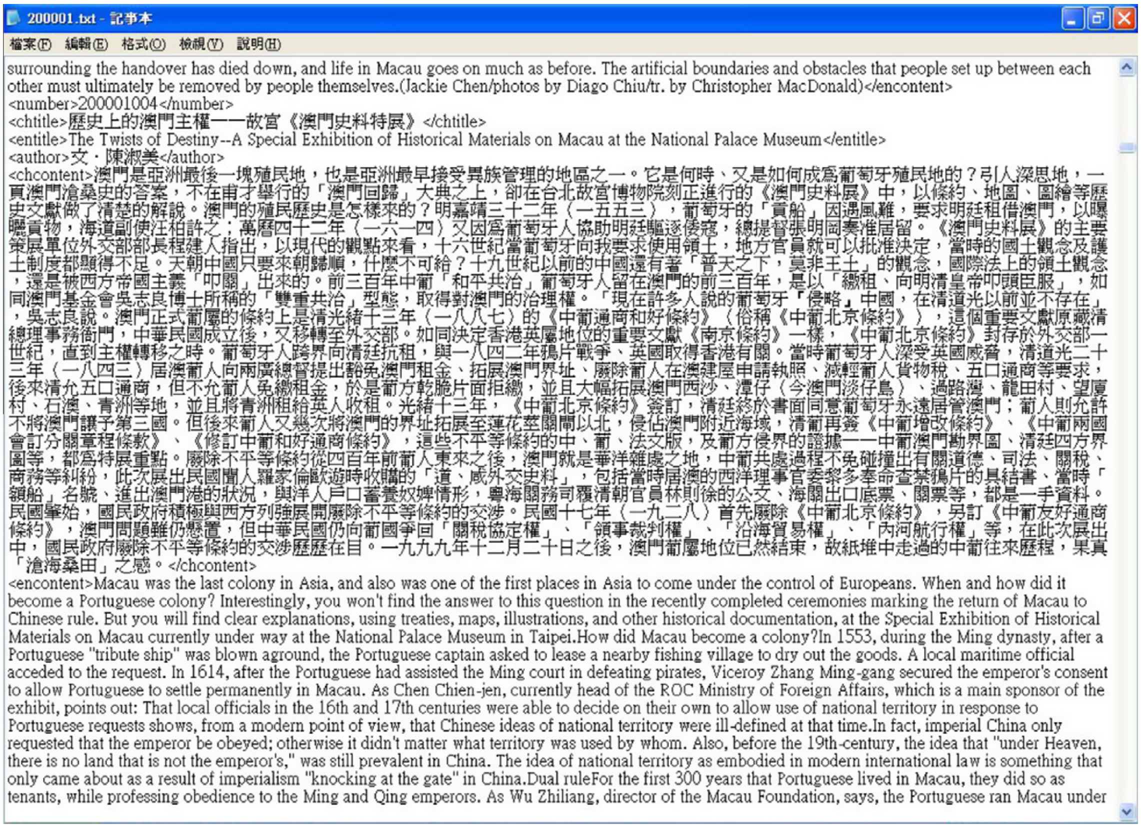
5.1. SOM training

To train Corpus-1, we constructed a self-organizing map which contains 100 neurons in 10 × 10 grid format. The number of neurons was determined experimentally such that a better clustering can be achieved. Each neuron in the map contains 3436 and 3711 synapses for training Chinese documents and English documents, respectively. The initial training gain was set to 0.4 and the maximal training time was set to 200 for both trainings. These settings were also determined experimentally. We tried different gain values ranged from 0.1 to 1.0 and various training time setting ranged from 50 to 500. We simply adopted the setting which achieved the most satisfying result. After training we labeled the map by documents

[1] Available from The Association for Computational Linguistics and Chinese Language Processing: http://www.aclclp.org.tw/index.php.

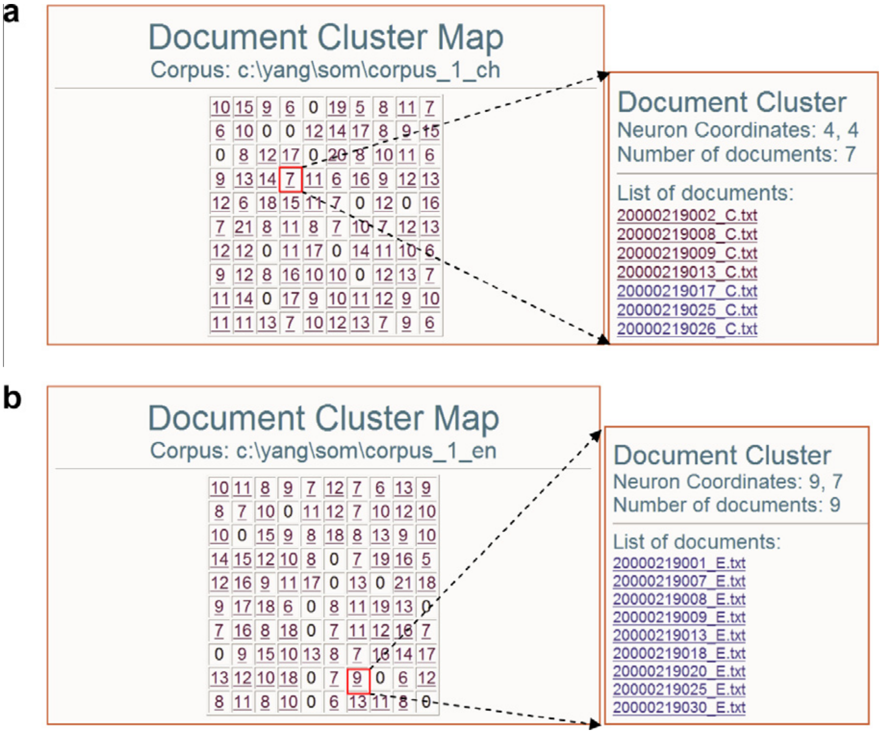
[2] http://www.taiwan-panorama.com/en/index.php.

[3] http://ckip.iis.sinica.edu.tw/CKIP/engversion/index.htm.



```
surrounding the handover has died down, and life in Macau goes on much as before. The artificial boundaries and obstacles that people set up between each
other must ultimately be removed by people themselves.(Jackie Chen/photos by Diago Chiu/tr. by Christopher MacDonald)</encontent>
<number>200001004</number>
<chtitle>歷史上的澳門主權——故宮《澳門史料特展》</chtitle>
<entitle>The Twists of Destiny--A Special Exhibition of Historical Materials on Macau at the National Palace Museum</entitle>
<author>文・陳淑美</author>
<chcontent>澳門是亞洲最後一塊殖民地，也是亞洲最早接受異族管理的地區之一。它是何時、又是如何成爲葡萄牙殖民地的？引人深思地，一
頁澳門滄桑史的答案，不在甫才舉行的「澳門回歸」大典之上，卻在台北故宮博物院刻正進行的《澳門史料展》中，以條約、地圖、圖繪等歷
史文獻做了清楚的解說。澳門的殖民歷史是怎樣來的？明嘉靖三十二年（一五五三），葡萄牙的「貢船」因遇風難，要求明廷租借澳門，以曝
曬貢物，海道副使汪柏許之；萬曆四十二年（一六一四）又因爲葡萄牙人協助明廷驅逐倭寇，總提督張明岡奏准居留。《澳門史料展》的主要
策展單位外交部部長程建人指出，以現代的觀點來看，十六世紀當葡萄牙向我要求使用領土，地方官員就可以批准決定，當時的國土觀念及護
土制度都顯得不足。天朝中國只要來朝歸順，什麼不可給？十九世紀以前的中國還有著「普天之下，莫非王土」的觀念，國際法上的領土觀念
，還是被西方帝國主義「叩關」出來的。前三百年中葡「和平共治」葡萄牙人留在澳門的前三百年，是以「繳租、向明清皇帝叩頭臣服」，如
同澳門基金會吳志良博士所稱的「雙重共治」型態，取得對澳門的治理權。「現在許多人說的葡萄牙『侵略』中國，在清道光以前並不存在」
，吳志良說。澳門正式葡屬的條約上是清光緒十三年（一八八七）的《中葡通商和好條約》（俗稱《中葡北京條約》），這個重要文獻原藏清
總理事務衙門，中華民國成立後，又移轉至外交部。如同決定香港英屬地位的重要文獻《南京條約》一樣，《中葡北京條約》封存於外交部一
世紀，直到主權轉移之時。葡萄牙人跨界向清廷抗租，與一八四二年鴉片戰爭、英國取得香港有關。當時葡萄牙人深受英國威脅，清道光二十
三年（一八四三）居澳葡人向兩廣總督提出豁免澳門租金、拓展澳門界址、廢除葡人在澳建屋申請執照、減輕葡人貨物稅、五口通商等要求，
後來清允五口通商，但不允葡人免繳租金，於是葡方乾脆片面拒繳，並且大幅拓展澳門西沙、潭仔（今澳門淡仔島）、過路灣、龍田村、望廈
村、石澳、青洲等地，並且將青洲租給英人收租。光緒十三年，《中葡北京條約》簽訂，清廷終於書面同意葡萄牙永遠居管澳門；葡人則允許
不將澳門讓予第三國。但後來葡人又幾次將澳門的界址拓展至蓮花莖關閘以北，侵佔澳門附近海域，清葡再簽《中葡增改條約》、《中葡兩國
會訂分關章程條款》、《修訂中葡和好通商條約》，這些不平等條約的中、葡、法文版，及葡方侵界的證據——中葡澳門勘界圖、清廷四方界
圖等，都爲特展重點。廢除不平等條約從四百年前葡人東來之後，澳門就是華洋雜處之地，中葡共處過程不免碰撞出有關道德、司法、關稅、
商務等糾紛，此次展出民國闇人羅家倫歐遊時收購的「道、咸外交史料」，包括當時居澳的西洋理事官委黎多奉命查禁鴉片的具結書、當時「
額船」名號、進出澳門港的狀況，與洋人戶口蓄養奴婢情形，粵海關務司覆清朝官員林則徐的公文、海關出口底票、關票等，都是一手資料。
民國肇始，國民政府積極與西方列強展開廢除不平等條約的交涉。民國十七年（一九二八）首先廢除《中葡北京條約》，另訂《中葡友好通商
條約》，澳門問題雖仍懸置，但中華民國仍向葡國爭回「關稅協定權」、「領事裁判權」、「沿海貿易權」、「內河航行權」等，在此次展出
中，國民政府廢除不平等條約的交涉歷歷在目。一九九九年十二月二十日之後，澳門葡屬地位已然結束，故紙堆中走過的中葡往來歷程，果真
「滄海桑田」之感。</chcontent>
<encontent>Macau was the last colony in Asia, and also was one of the first places in Asia to come under the control of Europeans. When and how did it
become a Portuguese colony? Interestingly, you won't find the answer to this question in the recently completed ceremonies marking the return of Macau to
Chinese rule. But you will find clear explanations, using treaties, maps, illustrations, and other historical documentation, at the Special Exhibition of Historical
Materials on Macau currently under way at the National Palace Museum in Taipei.How did Macau become a colony?In 1553, during the Ming dynasty, after a
Portuguese "tribute ship" was blown aground, the Portuguese captain asked to lease a nearby fishing village to dry out the goods. A local maritime official
acceded to the request. In 1614, after the Portuguese had assisted the Ming court in defeating pirates, Viceroy Zhang Ming-gang secured the emperor's consent
to allow Portuguese to settle permanently in Macau. As Chen Chien-jen, currently head of the ROC Ministry of Foreign Affairs, which is a main sponsor of the
exhibit, points out: That local officials in the 16th and 17th centuries were able to decide on their own to allow use of national territory in response to
Portuguese requests shows, from a modern point of view, that Chinese ideas of national territory were ill-defined at that time.In fact, imperial China only
requested that the emperor be obeyed; otherwise it didn't matter what territory was used by whom. Also, before the 19th-century, the idea that "under Heaven,
there is no land that is not the emperor's," was still prevalent in China. The idea of national territory as embodied in modern international law is something that
only came about as a result of imperialism "knocking at the gate" in China.Dual ruleFor the first 300 years that Portuguese lived in Macau, they did so as
tenants, while professing obedience to the Ming and Qing emperors. As Wu Zhiliang, director of the Macau Foundation, says, the Portuguese ran Macau under
```

Fig. 7. An example bilingual document in the Sinorama corpus.



Document Cluster Map
Corpus: c:\yang\som\corpus_1_ch

10	15	9	6	0	19	5	8	11	7
6	10	0	0	12	14	17	8	9	15
0	8	12	17	0	20	8	10	11	6
9	13	14	7	11	6	16	9	12	13
12	6	18	15	11	7	0	12	0	16
7	21	8	11	8	7	10	7	12	13
12	12	0	11	17	0	14	11	10	6
9	12	8	16	10	10	0	12	13	7
11	14	0	17	9	10	11	12	9	10
11	11	13	7	10	12	13	7	9	6

Document Cluster
Neuron Coordinates: 4, 4
Number of documents: 7

List of documents:
20000219002_C.txt
20000219008_C.txt
20000219009_C.txt
20000219013_C.txt
20000219017_C.txt
20000219025_C.txt
20000219026_C.txt

Document Cluster Map
Corpus: c:\yang\som\corpus_1_en

10	11	8	9	7	12	7	6	13	9
8	7	10	0	11	12	7	10	12	10
10	0	15	9	8	18	8	13	9	10
14	15	12	10	8	0	7	19	16	5
12	16	9	11	17	0	13	0	21	18
9	17	18	6	0	8	11	19	13	0
7	16	8	18	0	7	11	12	16	7
0	9	15	10	13	8	7	16	14	17
13	12	10	18	0	7	9	0	6	12
8	11	8	10	0	6	13	11	8	0

Document Cluster
Neuron Coordinates: 9, 7
Number of documents: 9

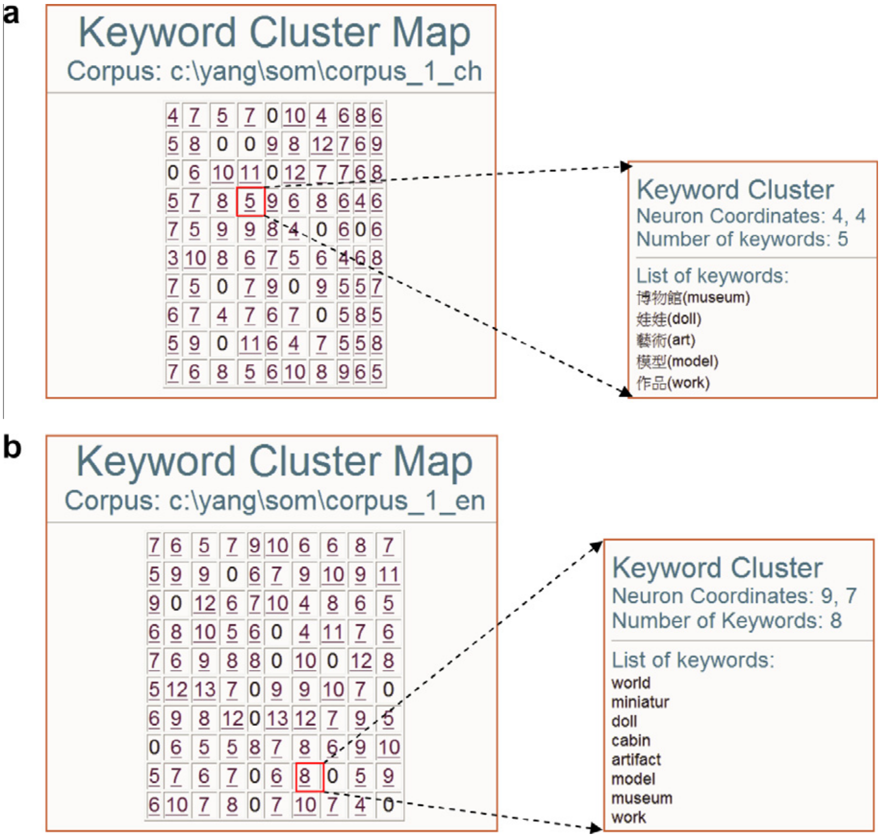
List of documents:
20000219001_E.txt
20000219007_E.txt
20000219008_E.txt
20000219009_E.txt
20000219013_E.txt
20000219018_E.txt
20000219020_E.txt
20000219025_E.txt
20000219030_E.txt

Fig. 8. The document cluster maps of Corpus-1 trained by (a) Chinese documents and (b) English documents. Selected neurons are expanded to show the documents labeled on them. Note that the original maps were formatted in HTML and shown on a Web browser. Clips of the browser are shown here.

and keywords, respectively, by the methods described in Section 3.2 and obtained the DCMs and the KCMs for both languages. The above process was also applied to Corpus-2 and obtained the DCMs and the KCMs for Corpus-2. The resulted DCMs and the KCMs for Corpus-1 are depicted in Figs. 8 and 9, respectively. Due to space limit, we only show selected neurons of the maps. In Fig. 8 the number in each grid in the DCM indicates the number of documents associated with the corresponding neuron. We then show the documents associated with these neurons by listing their filenames. We use a convention of the filenames that should effectively depict the similarity of the documents. For example, a document with filename '20000103024_C.txt' indicates that it is the Chinese version of the 24th segment of the article appeared at the page 3 in the issue published at Jan., 2000. Note that an article is referred by the starting page number in the issue. Therefore, the documents with filenames started with '20000103', e.g. '20000103014_C.txt', should be considered similar to it since they were extracted from the same article. The convention is also true for English documents except that character 'C' is replaced by 'E'. It is clear that similar documents were clustered together in Fig. 8. Similarly, in Fig. 9, we list the keywords associated with each neuron which location is the same as in Fig. 8. It is also clear that the keywords in Fig. 9 reflect the themes of the documents labeled to the corresponding neurons in Fig. 8. Table 1 shows the parameter setting and statistics of the training.

5.2. Hierarchy generation

After the clustering process, we applied the hierarchy generation process to the DCMs to obtain the hierarchies. In our experiments we limited the number of dominated clusters to 10. Both neighborhood sizes N_{c1} and N_{c2} are set to fourth of



Keyword Cluster Map

Corpus: c:\yang\som\corpus_1_ch

4	7	5	7	0	10	4	6	8	6
5	8	0	0	9	8	12	7	6	9
0	6	10	11	0	12	7	7	6	8
5	7	8	5	9	6	8	6	4	6
7	5	9	9	8	4	0	6	0	6
3	10	8	6	7	5	6	4	6	8
7	5	0	7	9	0	9	5	5	7
6	7	4	7	6	7	0	5	8	5
5	9	0	11	6	4	7	5	5	8
7	6	8	5	6	10	8	9	6	5

Keyword Cluster

Neuron Coordinates: 4, 4

Number of keywords: 5

List of keywords:

博物館(museum)
娃娃(doll)
藝術(art)
模型(model)
作品(work)

Keyword Cluster Map

Corpus: c:\yang\som\corpus_1_en

7	6	5	7	9	10	6	6	8	7
5	9	9	0	6	7	9	10	9	11
9	0	12	6	7	10	4	8	6	5
6	8	10	5	6	0	4	11	7	6
7	6	9	8	8	0	10	0	12	8
5	12	13	7	0	9	9	10	7	0
6	9	8	12	0	13	12	7	9	5
0	6	5	5	8	7	8	6	9	10
5	7	6	7	0	6	8	0	5	9
6	10	7	8	0	7	10	7	4	0

Keyword Cluster

Neuron Coordinates: 9, 7

Number of Keywords: 8

List of keywords:

world
miniatur
doll
cabin
artifact
model
museum
work

Fig. 9. The keyword cluster maps of Corpus-1 trained by (a) Chinese documents and (b) English documents.

Table 1
The parameter and statistics of the training process.

	Corpus-1		Corpus-2	
	Chinese	English	Chinese	English
Map size	10 × 10	10 × 10	20 × 20	20 × 20
Max training epoch	200	200	500	500
Initial training gain	0.4	0.4	0.4	0.4
# Of unlabeled neuron	15	18	44	49

the maps' dimensions for the two test corpora. We limited the depths of hierarchies to 3 and 4 for Corpus-1 and Corpus-2, respectively. In Fig. 10 we show part of the Chinese hierarchy developed from Corpus-1. Each leaf node in a hierarchy represents a cluster in the corresponding DCM. The parent node of some child nodes in level *n* of a hierarchy represents a super cluster found in STAGE-(*n*-1). In this example, the root node is one of the six dominated clusters found in STAGE-1. This node

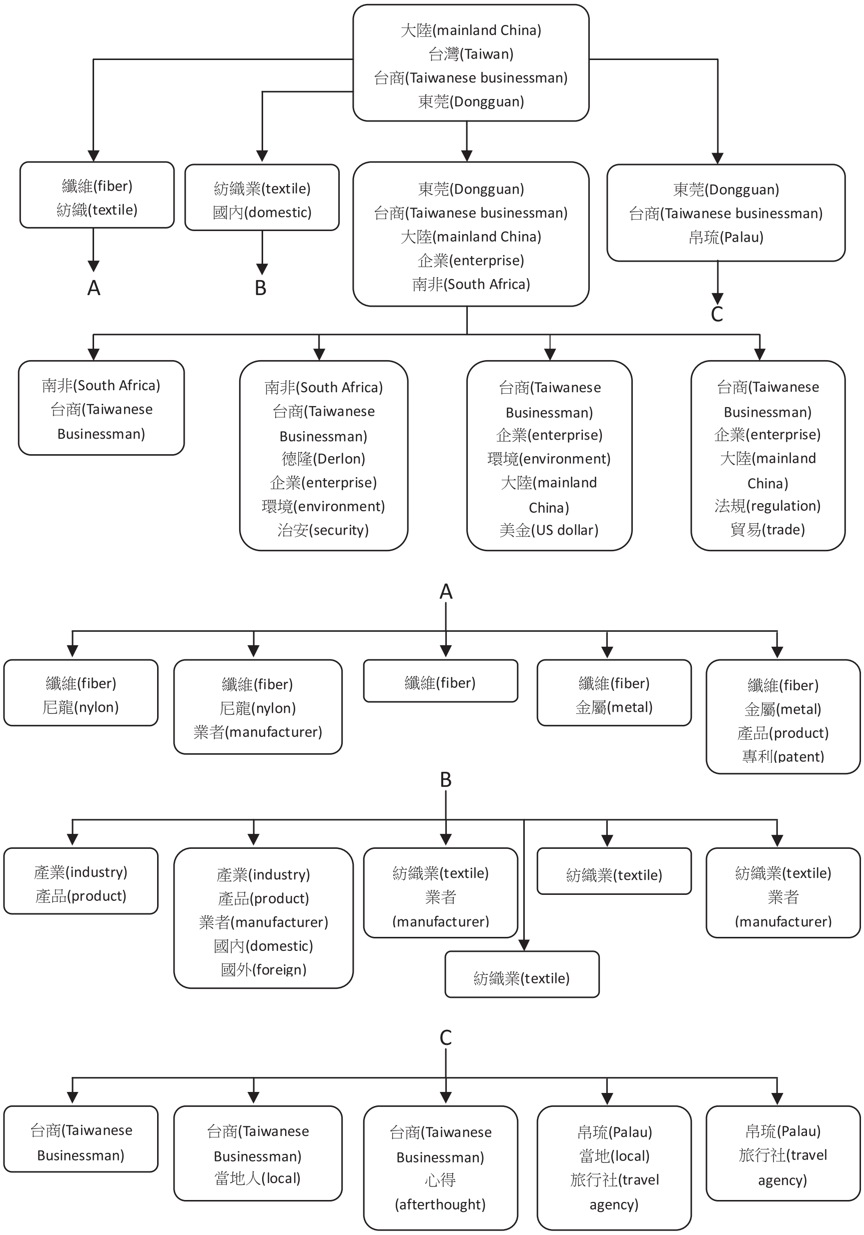


Fig. 10. One of the generated Chinese hierarchies using Corpus-1. English translations are enclosed in the parentheses for reference.

Table 2
The statistics for the generated hierarchies.

	Corpus-1		Corpus-2	
	Chinese	English	Chinese	English
Number of categories	50	47	258	279
Average depths of leaf categories	2.33	2.56	2.43	2.67
Average branching factor	3.26	2.87	3.81	3.21

Table 3
The average $\overline{P_k}$ values for the generated hierarchies.

	Corpus-1		Corpus-2	
	Chinese	English	Chinese	English
Our method	2.31	2.45	2.76	2.96
GHSOM	2.65	2.77	2.97	3.23
Our method on Chinese-to-English translated corpora	2.37	2.53	2.81	3.12
Our method on English-to-Chinese translated corpora	2.39	2.60	2.82	3.17

has four children which are the four dominated clusters obtained in STAGE-2. These child nodes comprise the third level of the hierarchy. Likewise, the child nodes of the third-level nodes are the found dominated clusters after STAGE-3. The keywords in the KCM are used to label every node in the hierarchies. We merge all keywords that belong to those clusters that are not dominated clusters into their nearest dominated clusters, as described in Section 3.3. It is clear in this example that the generated hierarchy comprises similar clusters which have related keywords. We omitted the rest of the hierarchies as well as the English hierarchy and entire hierarchies for Corpus-2 due to space limitation. Table 2 lists some statistics of the generated hierarchies.

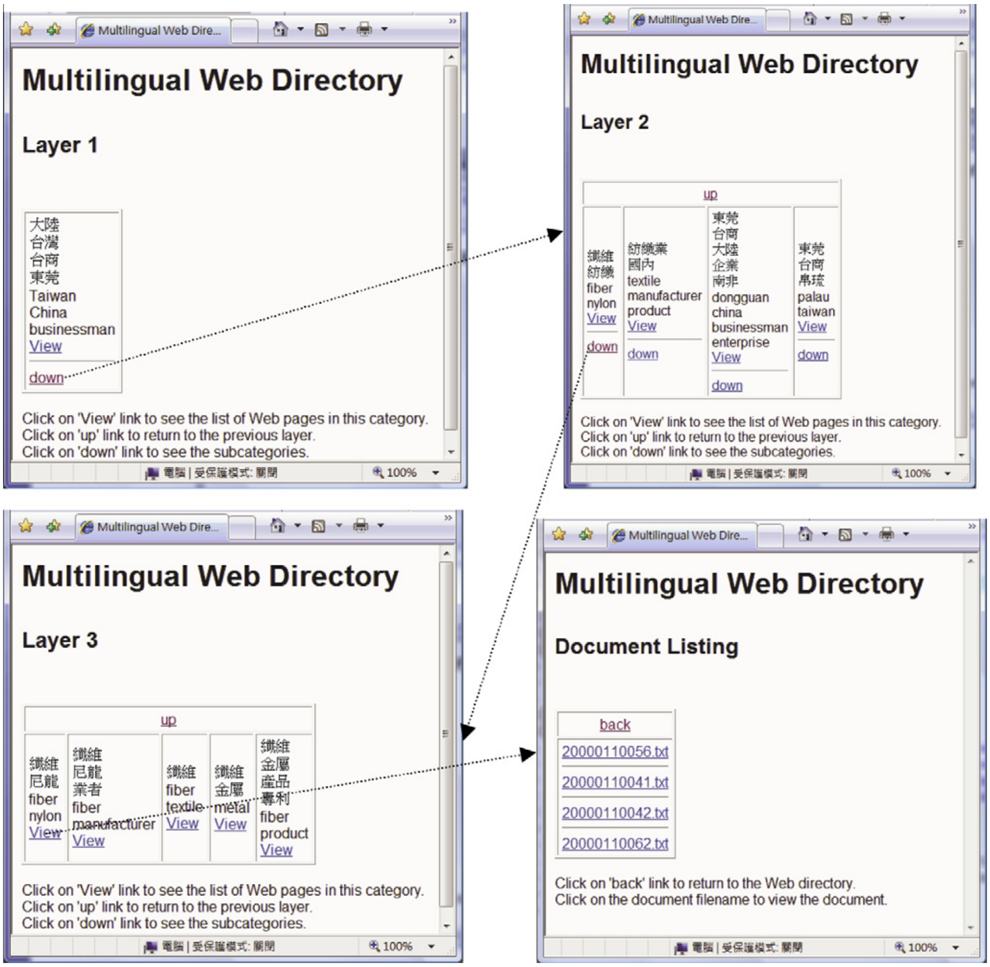
The quality of the generated hierarchies is evaluated using the $\overline{P_k}$ measured defined in Section 3.3 since $\overline{P_k}$ could reflect the likeliness of related documents being clustered together in different hierarchy. That is, when most of the related documents in one language were also related (being clustered together) in another language, the value of $\overline{P_k}$ will tend to be small. We conducted experiments on hierarchy generation process, preceded by SOM training, for 100 times using Corpus-1 and Corpus-2, respectively. The average $\overline{P_k}$ values were then calculated for both corpora. The result is summarized in Table 3. For comparison, we applied the GHSOM model on the same set of corpora. We chose the GHSOM because it is also based on the SOM and could generate hierarchies as our method. The same $\overline{P_k}$ values were also calculated. Another experiment used our method on multilingual corpora which one of the monolingual corpus is direct translation of another using Google translation service.[4] We constructed two corpora which one of them contains English documents translated from Chinese ones and the other contains Chinese documents translated from English ones. We can see that our method improves the $\overline{P_k}$ values for about 10% in average comparing to GHSOM. There are not much difference in $\overline{P_k}$ values for both direct translated corpora. This may due to that we only used nouns for training, which are often translated faithfully by the translation engine. It is also interesting to note that the average $\overline{P_k}$ values are smaller for Chinese corpora in all experiments. This may cause by the difference in the term selection processes between two languages. Also note that in the direct translation scheme we translated the documents before they were segmented. Another possibility is to segment the documents first and translate the resulting terms directly. This approach will produce exact duplicates without considering the barriers between languages, thus was not considered in our experiments.

The parameters in applying our method were set as follows. In the super cluster generation process, we used a square neighborhood in calculating aggregated cluster similarities. The dimension of the neighborhood used in generating Fig. 10 is a half of the dimension of the map. That is, the neighborhood includes at most fourth of the clusters in the map. For simplicity, we let N_{c1} denote the dimension of this neighborhood. Another important parameter is the neighborhood in eliminating neighboring clusters of a dominated cluster. We denote the dimension of this neighborhood by N_{c2} also for simplicity. We let N_{c2} be equal to fourth of the dimension of the map in generating Fig. 10. The third parameter is the maximum number of dominated clusters N_D that we can found in each stage. We allowed tenth of the number of neurons as dominated clusters in our experiments.

5.3. Hierarchy alignment and Web directory generation

After generating hierarchies for two languages, we applied the hierarchy alignment algorithm described in Section 4.1 on these hierarchies. The quality of the alignment is measured by the amount of parallel documents in each pair of associated clusters. For example, let Chinese cluster $\mathcal{C}_k$ be associated with English cluster $\mathcal{E}_l$ in which $\mathcal{C}_k$ contains documents C_1, C_3, C_5, and $\mathcal{E}_l$ contains documents E_1, E_2, E_3, and E_4. Here C_i and E_i are a pair of parallel documents. In this case, the number of parallel documents in this pair of clusters is then 2, namely C_1/E_1 and C_3/E_3. The total number of such parallel document pairs over all associated clusters is summed and divided by the total number of parallel document pairs in the training set. We have ratios 0.72 and 0.66 for Corpus-1 and Corpus-2, respectively, in average after 100 times of alignment of generated hierarchies. This means that about two thirds of parallel document pairs fall in associated clusters. This suggests that about one third of the parallel documents fall in different categories. We also conducted another experiment to see the difference between mapping Chinese clusters to English clusters and mapping English clusters to Chinese clusters. By reversing the role of Chinese clusters and English clusters, the above ratios are 0.74 and 0.69 for Corpus-1 and Corpus-2, respectively. This shows that our method differs not much in the roles during mapping hierarchies.

[4] http://translate.google.com.



Multilingual Web Directory

Layer 1

大陸
台灣
台商
東莞
Taiwan
China
businessman
View
down

Click on 'View' link to see the list of Web pages in this category.
Click on 'up' link to return to the previous layer.
Click on 'down' link to see the subcategories.

Multilingual Web Directory

Layer 2

up			
纖維 紡織 fiber nylon View down	紡織業 國內 textile manufacturer product View down	東莞 台商 大陸 企業 南非 dongguan china businessman enterprise View down	東莞 台商 帛琉 palau taiwan View down

Click on 'View' link to see the list of Web pages in this category.
Click on 'up' link to return to the previous layer.
Click on 'down' link to see the subcategories.

Multilingual Web Directory

Layer 3

up				
纖維 尼龍 fiber nylon View	纖維 尼龍 業者 fiber manufacturer View	纖維 fiber textile View	纖維 金屬 metal View	纖維 金屬 產品 專利 fiber product View

Click on 'View' link to see the list of Web pages in this category.
Click on 'up' link to return to the previous layer.
Click on 'down' link to see the subcategories.

Multilingual Web Directory

Document Listing

back
20000110056.txt
20000110041.txt
20000110042.txt
20000110062.txt

Click on 'back' link to return to the Web directory.
Click on the document filename to view the document.

Fig. 11. Part of the generated Web directory using Corpus-1. The dotted lines traverse the links during the access of the directory.

The Web directory generation process described in Section 4.2 were then applied on the generated hierarchies. Fig. 11 depicts part of a generated multilingual Web directory which complies with the Chinese hierarchy in Fig. 10. This directory was evaluated by a set of human subjects who participated in an information retrieval course in two semesters. When be asked if they preferred the multilingual directory to monolingual directory such as Yahoo!, 68 out of 90 subjects, i.e. 76%, gave positive answers. Moreover, 72% (65/90) of subjects thought the multilingual directory is feasible in arrangement of category themes and documents. The subjects were asked to give comments freely. Among 28 subjects who gave comments, eight of them stated that some categories do not have coherent Chinese and English themes. Five of them felt the hierarchical relations between categories are not so relevant. These comments suggest the possible weakness of our method and indicate the direction for improving our method.

6. Conclusions

In this work, we proposed an automatic method to generate and align multilingual hierarchies to construct a multilingual Web directory. Our method applies the self-organizing map model to cluster bilingual documents and creates two feature maps for each language. A hierarchy generation process is then applied on these maps to create two monolingual hierarchies. These hierarchies are then aligned according to the relatedness of their nodes. Two kinds of similarity, namely semantic similarity and structural similarity, between nodes were defined in this work. These similarities are determined by the documents associated with the nodes. We conducted experiments on two sets of corpora and obtained promising result.

The major advantages of our method are the development of multilingual hierarchy alignment method. Our approach is fully automated and requires no human intervention. The quality of the generated hierarchies and alignment result are both evaluated and justified. The result of the alignment can be applied to tackle tasks such as multilingual information retrieval. Besides, our method can be applied to any sets of symbols which document associations can be specifically defined.

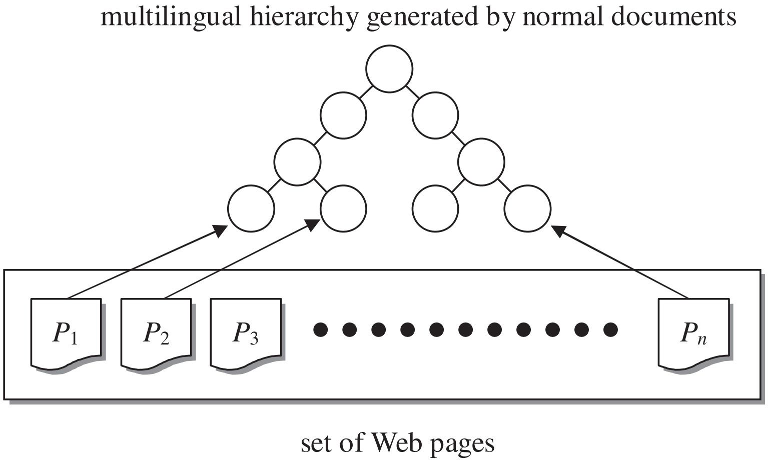


Fig. 12. Bootstrapping of Web directory from hierarchies generated by normal documents. P_i could be a Web page in either language.

A limitation of our method comes from the need of parallel corpora. We need the knowledge of the correspondences between documents of different language to discover the associations. It is uncommon that such parallel corpora containing Web pages are available. Two approaches could be applied to tackle this deficiency. The first approach is to use comparable corpora as long as the correspondence between documents could be defined. When both parallel and comparable corpora are unavailable, we may also apply another correspondence discovery process to find such correspondence prior to our method. The second approach is to use some standard parallel corpora, such as the Sinorama corpora used in this work, to generate the monolingual hierarchies and obtain the associations between these hierarchies. A multilingual hierarchy could also be generated using our method. This hierarchy is then used to 'bootstrap' the generation of the multilingual Web directory. A set of Web pages could then be labeled to these monolingual hierarchies, i.e. labeled to the multilingual hierarchy simultaneously, to obtain the multilingual Web directory. The labeling method is the same as described in Section 3.2. The bootstrapping process is depicted in Fig. 12.

References

Adar, E., Skinner, M., & Weld, D. S. (2009). Information arbitrage across multi-lingual wikipedia. In *Proceedings of the second ACM international conference on Web search and data mining* (pp. 94–103). Barcelona, Spain.

Benini, L., Macii, A., Macii, E., & Poncino, M. (2000). Increasing energy efficiency of embedded systems by application-specific memory hierarchy generation. *IEEE Design and Test of Computers, 17*(2), 74–85.

Böhm, K., Heyer, G., Quasthoff, U., & Wolff, C. (2002). Topic map generation using text mining. *Journal of Universal Computer Science, 8*(6), 623–633.

Brown, R. D. (1996). Example-based machine translation in the pangloss system. In J. Tsujii (Ed.), *Proceedings of the 16th international conference on computational linguistics* (pp. 169–174). San Francisco, CA: Morgan Kaufmann.

Chau, R., & Yeh, C. H. (2004). A multilingual text mining approach to web cross-lingual text retrieval. *Knowledge-Based Systems, 17*(5–6), 219–227.

Chen, K. H. & Chen, H. H. (1994). A part-of-speech-based alignment algorithm. In Y. Wilks (Ed.), *Proceedings of 15th international conference on computational linguistics* (pp. 166–171). Somerset, NJ: Association for Computational Linguistics.

Chen, K. J., & Bai, M. H. (1998). Unknown word detection for Chinese by a corpus-based learning method. *International Journal of Computational linguistics and Chinese Language Processing, 3*(1), 27–44.

Choi, N., Song, I. Y., & Han, H. (2006). A survey on ontology mapping. *SIGMOD Record, 35*(3), 34–41.

Chuang, S. L., & Chien, L. F. (2005). Taxonomy generation for text segments: A practical web-based approach. *ACM Transactions on Information Systems, 23*(4), 363–396.

Daudé, J., Padró, L., & Rigau, G. (1999). Mapping multilingual hierarchies using relaxation labeling. In P. Fung, & J. Zhou (Eds.), *Joint SIGDAT conference on empirical methods in natural language processing and very large corpora* (pp. 12–19). Maryland: College Park.

Davis, M. W., & Dunning, T. (1996). A trec evaluation of query translation methods for multi-lingual text retrieval. In D. K. Harmon (Ed.), *Proceedings of TREC-4* (pp. 483–497). Darby, PA: DIANE Publishing Company.

Han, J., & Fu, Y. (1994). Dynamic generation and refinement of concept hierarchies for knowledge discovery in database. In *Proceedings of AAAI'94 workshop on knowledge discovery in database* (pp. 157–168). AAAI Press.

Hearst, M. A., & Karadi, C. (1997). Cat-a-cone: An interactive interface for specifying searches and viewing retrieval results using a large category hierarchy. In *Proceedings of the 20th international ACM SIGIR conference research and development in information retrieval* (pp. 246–255).

Hofmann, T. (1999). The cluster-abstraction model: Unsupervised learning of topic hierarchies from text data. In *Proceedings of the sixteen international joint conference on artificial intelligence* (pp. 682–687). Morgan Kaufmann.

Ichise, R., Takeda, H., & Honiden, S. (2001). Rule induction for concept hierarchy alignment. In *Proceedings of the workshop on ontology learning at the 17th international joint conference on artificial intelligence (IJCAI)*. CEUR-WS.org.

Kalfoglou, Y., & Schorlemmer, W. M. (2005). Ontology mapping: The state of the art. In Y. Kalfoglou, W. M. Schorlemmer, A. P. Sheth, S. Staab, & M. Uschold (Eds.), *Semantic interoperability and integration*. Germany: IBFI, Schloss Dagstuhl.

Kohonen, T. (1997). *Self-organizing maps*. Berlin: Springer-Verlag.

Kohonen, T. (2001). *Self-organizing maps*. Berlin: Springer-Verlag.

Lee, C. H. & Yang, H. C., (1999). A web text mining approach based on self-organizing map. In *Proceedings of the ACM CIKM'99 2nd workshop on web information and data management* (pp. 59–62). Kansas City, Missouri.

Lee, C. H., & Yang, H. C. (2003). A multilingual text mining approach based on self-organizing maps. *Applied Intelligence, 18*(3), 295–310.

Levow, G. A., Dorr, B. J., & Lin, D. (2000). Construction of Chinese–English semantic hierarchy for information retrieval. In *International conference of chinese language computing*. Chicago, IL.

Liu, N. & Yang, C. C. (2007). A link classification based approach to website topic hierarchy generation. In *Proceedings of WWW 2007* (pp. 1127–1128). Association for Computing Machinery.

McCallum, A. & Nigam, K. (1999). Text classification by bootstrapping with keywords, em and shrinkage. In *Proceedings of ACL'99 workshop for unsupervised learning in natural language processing* (pp. 52–58). Morgan Kaufmann.

McCallum, A., Nigam, K., Rennie, J., & Seymore, K. (2000). Automating the construction of internet portals with machine learning. *Information Retrieval, 3*(2), 127–163.

Moore, G. (2000). Topic map technology – The state of the art. In *XML Europe 2000*. Paris, France.

Ng, K. & Trifonov, B. (2003). Automatic bounding volume hierarchy generation using stochastic search methods. In *Mini-workshop on stochastic search algorithms* (pp. 147–162). Canada: Vancouver.

Noy, N. F., & Stuckenschmidt, H. (2005). Ontology alignment: An annotated bibliography. In Y. Kalfoglou, W. M. Schorlemmer, A. P. Sheth, S. Staab, & M. Uschold (Eds.), *Semantic interoperability and integration*. Germany: IBFI, Schloss Dagstuhl.

Oard, D. W. & Dorr, B. J., (1996). A survey of multilingual text retrieval. Tech. Rep. UMIACS-TR-96-19, Institute for Advanced Computer Studies, University of Maryland, College Park, MD.

Pianta, E., Bentivogli, L., & Girardi, C. (2002). Multiwordnet: Developing an aligned multilingual database. In *Proceedings of the first international conference on global WordNet* (pp. 55–63). CIIL.

Porter, M. F. (1980). An algorithm for suffix stripping. *Program, 14*(3), 130–137.

Rath, H. H. (1999). Technical issues on topic maps. In *Proceedings of metastructures 99 conference*, GCA.

Rauber, A., Merkl, D., & Dittenbach, M. (2002). The growing hierarchical self-organizing map: Exploratory analysis of high-dimensional data. *IEEE Transactions on Neural Networks, 13*(6), 1331–1341.

Salton, G. (1989). *Automatic text processing: The transformation analysis and retrieval of information by computer*. Reading, MA: Addison-Wesley.

Sheridan, P., & Ballerini, J. P. (1996). Experiments in multilingual information retrieval using the spider system. In H. P. Frei (Ed.), *Proceedings of the 19th ACM SIGIR conference on research and development in information retrieval* (pp. 58–65). Seattle, WA: Association for Computing Machinery.

Sornlertlamvanich, V., Kruengkrai, C., Tongchim, S., Srichaivattana, P., & Isahara, H. (2005). Term-based ontology alignment. *Research on Computing Science, 12*, 138–144.

Talvensaari, T., Juhola, M., Laurikkala, J., & Järvelin, K. (2007). Corpus-based cross-language information retrieval in retrieval of highly relevant documents. *Journal of the American Society for Information Science and Technology, 58*(3), 322–334.

Weigend, A. S., Wiener, E. D., & Pedersen, J. O. (1999). Exploiting hierarchy in text categorization. *Information Retrieval, 1*(3), 193–216.

Yang, H. C., & Lee, C. H. (2004). A text mining approach on automatic generation of web directories and hierarchies. *Expert Systems with Applications, 27*(4), 645–663.

Yang, C. C., & Liu, N. (2009). Website topic hierarchy generation based on link structure. *Journal of the American Society for Information Science and Technology, 60*(3), 495–508.