Knowledge-Based Systems 74 (2015) 1–13

Contents lists available at ScienceDirect

Knowledge-Based Systems

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

Adaptive Concept Resolution for document representation and its applications in text mining

Lidong Bing [a], Shan Jiang [b], Wai Lam [a], Yan Zhang [c,*], Shoaib Jameel [a]

[a] Key Laboratory of High Confidence Software Technologies, Ministry of Education (CUHK Sub-Lab), Department of Systems Engineering and Engineering Management, The Chinese University of Hong Kong, Hong Kong
[b] Department of Computer Science, University of Illinois at Urbana-Champaign, Urbana, United States
[c] Department of Machine Intelligence, Peking University, China

ARTICLE INFO

Article history:
Received 28 February 2014
Received in revised form 21 July 2014
Accepted 6 October 2014
Available online 1 November 2014

Keywords:
Adaptive Concept Resolution
Ontology
WordNet
Wikipedia
WordNet-Plus

ABSTRACT

It is well-known that synonymous and polysemous terms often bring in some noise when we calculate the similarity between documents. Existing ontology-based document representation methods are static so that the selected semantic concepts for representing a document have a fixed resolution. Therefore, they are not adaptable to the characteristics of document collection and the text mining problem in hand. We propose an Adaptive Concept Resolution (ACR) model to overcome this problem. ACR can learn a concept border from an ontology taking into the consideration of the characteristics of the particular document collection. Then, this border provides a tailor-made semantic concept representation for a document coming from the same domain. Another advantage of ACR is that it is applicable in both classification task where the groups are given in the training document set and clustering task where no group information is available. The experimental results show that ACR outperforms an existing static method in almost all cases. We also present a method to integrate Wikipedia entities into an expert-edited ontology, namely WordNet, to generate an enhanced ontology named WordNet-Plus, and its performance is also examined under the ACR model. Due to the high coverage, WordNet-Plus can outperform WordNet on data sets having more fresh documents in classification.

1. Introduction

Traditionally, the representation of text documents is usually based on the Bag of Words (BOW) approach, which represents the documents with features as weighted occurrence frequencies of individual words. This technique has several drawbacks. First, it breaks a phrase, say " air conditioner", into independent features. Second, it maps synonymous words into different features. Third, it merges a polysemous word's different meanings into a single feature. These drawbacks make the document similarity unable to be computed by BOW accurately. The methods that overcome these drawbacks can be categorized into two classes, namely, linear projection models (including LSA [7], PLSA [17], LDA [4], OPCA [32]), and S2Net [45], and ontology-based methods [19,35]. In this paper, we focus on the latter methodology.

Some expert-edited ontologies include WordNet [29], Cyc [27], Mesh [50], etc. Previous empirical results have shown some improvement in some applications utilizing ontologies [1,5,12,13,19,25,26,35,38,40,50,52]. Recently, the online collaborative encyclopedia Wikipedia[1] provides us another resource to assist the text mining tasks, and its potential has been shown in classification [14,47,48], clustering [20,21,30,36], semantic relatedness computing [44], and taxonomy induction [2,9,34,33]. However, the existing works have an obvious shortcoming: the strategies they adopted are static. For example, one strategy is to use each synset in the WordNet as one dimension in the representation vector of the documents. Therefore, the resolutions for representing the documents belonging to different collections are the same. Suppose we have two document collections, the first one has coarse granularity categories, such as sports and military, while the second one has finer granularity categories, such as football and basketball. In the first collection, football players and basketball players should be regarded as related, while in the second they should be unrelated. So an adaptive strategy is very likely able to outperform the static

* Corresponding author. Tel.: +86 1062755592.
E-mail addresses: ldbing@se.cuhk.edu.hk (L. Bing), sjiang18@illinois.edu (S. Jiang), wlam@se.cuhk.edu.hk (W. Lam), zhy@cis.pku.edu.cn (Y. Zhang), msjameel@se.cuhk.edu.hk (S. Jameel).

[1] http://en.wikipedia.org.

http://dx.doi.org/10.1016/j.knosys.2014.10.003
0950-7051/

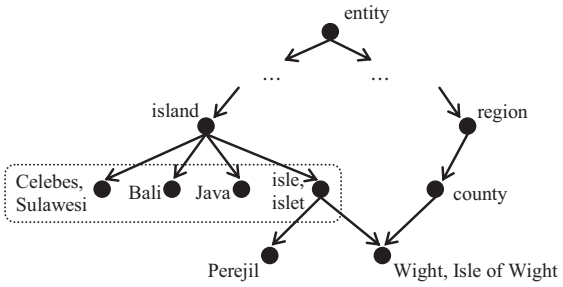

Fig. 1. A fragment of WordNet structure. Each node is a concept, whose synset contains the terms attached to the node.

one. Furthermore, in the existing works only one ontology is employed, either expert-edited one or online collaborative one. Hence they suffer from the former's limited coverage or the latter's noisy information.

In this paper, the proposed Adaptive Concept Resolution (ACR) model can learn a concept border from an ontology taking into the consideration of the characteristics of the particular document collection. Then, this border can provide a tailor-made semantic concept representation for a document coming from the same domain. The structure of an ontology is a hierarchical directed acyclic graph[2] (refer to the example in Fig. 1), and the border is a cross section in the graph. All the concepts located below the border will be merged into one of the concepts on the border. We design a gain value to measure whether a concept is a good candidate for the border. The gain value is calculated based on the characteristics of the given document collection. As a result, our model can generate different tailor-made borders for different collections adaptively. Another advantage of ACR is that it is applicable in both classification task where the groups are given in the training document set and clustering task where no group information is available. To do so, we only need to change the granularity, that is either cluster or individual document, for calculating the gain value. Therefore, ACR can be applied to both classification and clustering. The experimental results show that our model can outperform an existing static method in almost all cases.

Currently, there are more than 4 million English articles (i.e., entities) in Wikipedia, which makes it an extremely valuable linguistic repository. Wikipedia's ability of covering new terms is much better than expert-edited ontologies. Take the term "Bing" as an example, it may refer to a Web search engine from Microsoft, or a soft drink from UK, or others. But this term is not covered by WordNet. However, the abundant information is also a double-edged sword. Because Wikipedia is collaboratively edited by large number of users with different backgrounds and editing capabilities, it involves large amount of noise and its structure is very complicated. To leverage the advantages and eliminate the limitations, we propose a method to merge Wikipedia entities into the structure of an expert-edited ontology, i.e. WordNet, and construct an enriched ontology, called WordNet-Plus. Consider a Wikipedia entity, with the category information of the entity as clues. We can get a set of WordNet concepts which are the potential higher-level semantic meanings of the entity. Then, the similarity between the entity and each candidate concept is calculated to find the most suitable higher-level semantic meaning for the entity. Finally, we attach this Wikipedia entity under the found WordNet concept. Thus, WordNet-Plus keeps WordNet's good structure, meanwhile it encapsulates large amount of information from Wikipedia. Therefore, WordNet-Plus inherits the advantages of both WordNet and Wikipedia. In our experiment, 611,161 Wikipedia entities are integrated into WordNet. For example, a small island "Bacan" in Indonesia is successfully attached under the WordNet concept "island", the search engine "Bing" is attached under "search engine" and "website".

For comparing the performance of WordNet-Plus with the expert-edited ontology, both of them are applied to our proposed ACR model to generate two different representations for the same document. These two representations are applied to two different text mining tasks, namely, classification, and clustering. The results show that the performance of WordNet-Plus in text mining is competitive under ACR model compared with WordNet. In the Web page classification task, WordNet-Plus can outperform WordNet significantly because of its high coverage on new terms. In the clustering experiment, WordNet-Plus performs as good as WordNet on three data sets.

The presented work in this paper substantially extends our previous short paper [3] in several aspects. First, we elaborate the technique details of the proposed ACR model, which cannot be fully given in the short paper [3]. Second, we present a method to integrate Wikipedia entities into an expert-edited ontology, namely WordNet, to generate an enhanced ontology named WordNet-Plus. Third, the performance of WordNet-Plus is investigated under the ACR model. Due to the high coverage, WordNet-Plus can outperform WordNet on data sets having more fresh documents in classification. Fourth, extensive case studies of WordNet-Plus are given to demonstrate the rationality of WordNet-Plus construction. Quantitative evaluation is also conducted to further examine the quality of WordNet-Plus.

In the remainder of this paper, we first review the literature in Section 2. After the preliminary of ontology and the overview of ACR model are introduced in Section 3, two main components of ACR, namely, concept border generation and concept-based document representation, are presented in Sections 4 and 5 respectively. The technique details and the time complexity of ACR are presented in Section 6. The construction of WordNet-Plus is discussed in Section 7. Then, the experiment design and results are given in Sections 8 and 9. Finally, we conclude the paper.

2. Related work

As an important expert-edited ontology, WordNet has been used to improve the performance of clustering and classification. Hotho et al. [18,19] show that incorporating the synset and the hypernym as background knowledge into the document representation can improve the clustering results. Jing et al. [24] construct a term similarity matrix using WordNet to improve text clustering. However, their approach only uses synonyms and hyponyms, and fails to handle polysemy, and breaks the multi-word concepts into a group of single words. In Recupero's work [35], two strategies, namely, WordNet lexical categories (WLC) technique and WordNet ontology (WO) technique, are used to create a new vector space with low dimensionality for the documents. The authors in [39] successfully integrate the WordNet resource for document classification. They show improved classification results with respect to the Rocchio and Widrow-Hoff algorithms. A significant difference between our ACR model and the methods mentioned above is that we adopt a learning process to determine the dimensions in the new representation for the documents, which gives our method more adaptability in dealing with different document collections.

Wikipedia is an important online linguistic resource, which has been studied quite a lot for different purposes in recent year, such as clustering [21,20], classification [14,47,48], and semantic relatedness computation [15,51]. In clustering [21,20], the researchers

[2] A hierarchical directed acyclic graph is a directed acyclic graph with the layer information on each node. The head node of an edge must have a higher layer than the tail of the edge.

extract several kinds of information for a document from Wikipedia, such as concept, category and synonym. Then the representation of the document is enriched by combining these information with the original BOW vector. Finally, the new representation is used to do clustering. Wang et al. [47,48] extract the relations between Wikipedia concept, and then build a proximity matrix, called semantic kernel. They also adopt the similar method mentioned above to get an enriched vector for each document, then utilize the semantic kernel to calculate the similarity. Departure from general ontology learning [46], taxonomy induction based on Wikipedia was also investigated in previous works [9,34,33]. WordNet-Plus is different from these works because of the utilization of WordNet structure.

Some researchers have observed the necessity of combining the expert-edited ontology and Wikipedia. Medelyan et al. [28] propose a combination method, exact and ambiguous mapping, to map Cyc terms onto Wikipedia articles. 52,690 Cyc terms find their corresponding Wikipedia articles. The similar work focusing on WordNet and Wikipedia is reported by Ruiz-Casado et al. [37]. Researchers also investigated integrating expert-edited ontology and online encyclopedias in other languages such as Chinese [23]. In WordNet-Plus, we focus on enriching WordNet with Wikipedia article's title which is absent from WordNet. The attaching point is determined by matching the head terms of the article's categories with the synsets in WordNet. Thus, WordNet-Plus is different from WorkiNet [22], which also added new categories into WordNet. YAGO [42,41] mainly focuses on finding the "individuals" and "facts" from Wikipedia, then represents them in description logics. The authors try to extract 14 kinds of relation, such as "locatedIn", "hasWonPrize" and "bornInYear". WordNet is utilized to help them to generate "subClassOf" and "means" relations. In the generation of "subClassOf" relation, YAGO attempts to locate a super category in WordNet for a Wikipedia category name. While in our work, we try to find a category for each Wikipedia article title in WordNet. In technique details, they only consider one category name each time, while in our method we utilize the category set which can provide more information. BabelNet [31] is a multilingual semantic network, in which the concepts and relations are generated from WordNet and Wikipedia. Concepts in BabelNet are represented similarly to WordNet by grouping sets of synonyms in the different languages into multilingual synsets with lexicalizations from WordNet synsets, the corresponding Wikipedia pages and additional translations. The relations between synsets are collected from WordNet and Wikipedia hyperlink. BabelNet has been successfully utilized in different cross-lingual tasks such as plagiarism detection, document retrieval, and text categorization [10,11]. The major insight is that the documents in different languages can be connected via presenting them with the help of multilingual synsets in BabelNet.

3. Ontology preliminary and ACR overview

3.1. Ontology preliminary

Concepts are the basic components of an ontology. Each concept may refer to an abstractive entity or a real entity. In each concept, several components are involved, such as a *synset*, *hyponymy* (is-a) relation with other concepts and a gloss. We give a formal definition of a concept:

Definition 1. Concept: A semantic concept π is a quadruple $\langle id,\ \Omega,\ \sigma,\ \Upsilon\rangle$, where *id* is its ID, Ω is the synset, σ denotes the gloss, and Υ is the set of its hyponym concepts. We refer to the items with $\pi \cdot id,\ \pi \cdot \Omega,\ \pi \cdot \sigma$ and $\pi \cdot \Upsilon$ respectively. Ω's element is denoted as ω, and a set of concepts is denoted as Π. If Υ is empty, π is a leaf concept.

WordNet [29] is a popular ontology and it has been extensively utilized in text mining for more than one decade. In this paper, we employ WordNet2.1 as an instance of the ontology to illustrate the framework. In Fig. 1, a fragment of WordNet is given. Take the concept "island" (*id*: 09316454) as an example. Then Ω is {island}, σ is "a land mass (smaller than a continent) that is surrounded by water", and some of its hyponym concepts are shown in the dashed box. The first term in a synset is also used to refer to the concept.

In WordNet, the concept "00001740" with synonymy set {entity} is the root concept. There are two kinds of "is a" relation, "Java" is a real instance of "island", while "islet" is a semantic instance. Some concepts may have more than one hypernym concepts, as exemplified by the concept "Wight" at the bottom right in Fig. 1. We call this kind of concept an *ambiguous concept*. As a result, the structure of an ontology is a hierarchical directed acyclic graph. The depth $d(\pi)$ for the concept π in the graph is defined as follows:

$$d(\pi) = \begin{cases} 0 & \text{if } \pi = root, \\ \max\limits_{\pi' \in \{\pi' | \pi \in \pi' \cdot \Upsilon\}} d(\pi') + 1 & \text{otherwise}. \end{cases} \quad (1)$$

If a single term is contained by more than one concepts, it is an *ambiguous term* since it refers to multiple semantic meanings. The term "Java" refers to an island in Fig. 1, it can also refer to a kind of coffee or a programming language as shown in Fig. 2.

3.2. Overview of ACR model

Fig. 3 depicts the overview of our Adaptive Concept Resolution (ACR) model. There are two main parts in the framework indicated by the dashed boxes, namely, the learning part on the left hand side and the utilizing part on the right hand side. The learning part aims at generating a concept border in an ontology using a training document collection as guidance. After that, the utilizing part employs this border to construct a concept-based representation for a document in the same domain.

In the learning part, given a document collection and an ontology structure, the algorithm learns which concepts have better information gain and generates the elements for the concept border $\mathcal{B}$. The border is composed of all the concepts (represented by empty circles) located on the dashed line. Each concept in $\mathcal{B}$ encapsulates all its descendants concepts in the ontology. For example, the terms in π_1 and π_2 are added into $\pi \cdot \Omega$. Then we get a derived concept, π^b, of π in the border. Thus, the border is tailor-made for the given document collection. In the border utilizing part, $\mathcal{B}$ is used to represent a document coming from the same domain as the training document collection, and each concept in $\mathcal{B}$ is treated as one dimension in the vector. If t is a term in the document and contained by the synset of π^b, its weight will be accumulated into the dimension of π^b. If t is contained by a concept above the border, it will be omitted. In both parts, the concept extraction and matching component is involved to preprocess the documents. We will present the details of this component as well

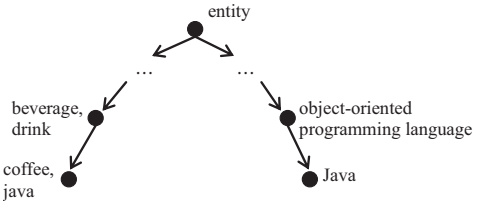

Fig. 2. Ambiguous term "java" in WordNet.

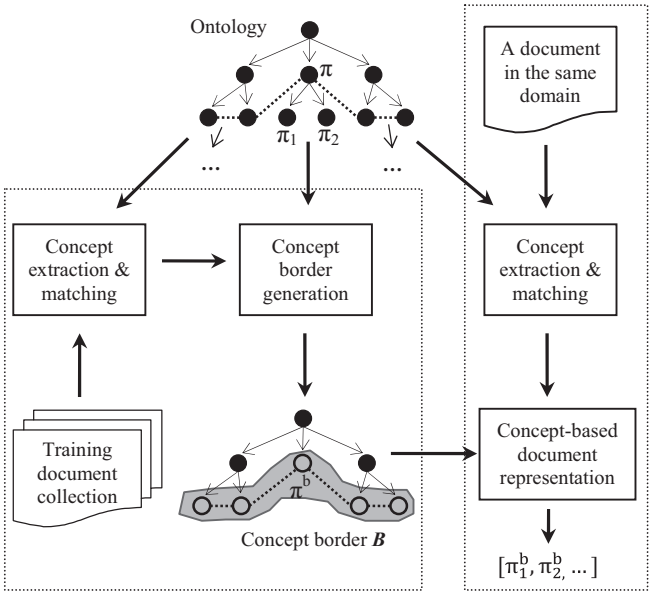

Fig. 3. The overview of our Adaptive Concept Resolution (ACR) model. The learning part is in the left hand side dashed box, and the utilizing part is in the right hand side dashed box.

as the concept-based document representation in Section 5 after the concept border generation is discussed in Section 4.

4. Concept border generation

In this section, we first introduce the definition of generalized entropy for a concept. Then, this entropy is employed to define a *gain* function to guide the concept border generation process.

4.1. Concept entropy

Suppose D is a document set, and it can be partitioned into groups $D = \{u_1, u_2, \ldots\}$. If there exist clusters in the document set, each u_i represents a cluster. Otherwise, each u_i represents a single document.

Traditionally, document frequency (df) is used to indicate whether a term is commonly used in a document set. But df is a simple measure and it ignores the weight of the term in different documents. We propose an entropy based method to measure the popularity of terms in a document set. The entropy of a term set T is calculated as in Eq. (2):

$$entropy(T) = -\sum_{u_i \in D} p(u_i|T) \log p(u_i|T), \tag{2}$$

where $p(u_i|T)$ is the probability of u_i given T and it is calculated as in Eq. (3):

$$p(u_i|T) = \frac{\sum_{t_j \in T} w_{ij}}{\sum_{u_k \in D, t_j \in T} w_{kj}}, \tag{3}$$

where $w_{i,j}$ is the weight of t_j in u_i. If T just contains a single term t, Eq. (2) becomes the term entropy. If t is frequently used in most of u_i's, its entropy tends to be large, i.e., its uncertainty is large. Consequently, t is not a good feature. If t is very frequently used in only one or several u_i's and seldom mentioned in the others, t is a desirable feature.

Each concept's synset contains one or several terms, and these terms have identical or very similar meanings. Thus, we measure the entropy of a concept π, denoted as $entropy(\pi)$, with the entropy of its synset:

$$entropy(\pi) = entropy(\pi \cdot \Omega). \tag{4}$$

If the concept entropy is large, it indicates the fact that the semantic meaning of this concept is frequently mentioned in D.

π's descendants represent the specialized meanings of π. Considering all these concepts, including π, as a whole, we define the generalized entropy $Gentropy(\pi)$ to measure its popularity in the document collection:

$$Gentropy(\pi) = entropy(\Omega_\pi^g), \tag{5}$$

where $\Omega_\pi^g = \pi \cdot \Omega \cup_{\pi' \in \Pi_\pi^d} \pi' \cdot \Omega$, and $\Pi_\pi^d = \{\pi' | \pi \rightsquigarrow \pi'\}$ is the descendant set of π. Consequently, the generalized entropy indicates the popularity of an entire concept including its descendants. For a leaf concept π^l, we have $entropy(\pi^l) = Gentropy(\pi^l)$.

Suppose we have a universal document set which is extremely large and covers different topics evenly. On this universal document set, the generalized entropy of a general concept should be no less than that of a special concept. It is formally stated as:

Observation 1. If $\pi' \in \pi \cdot \Upsilon$, we have $Gentropy(\pi) \geqslant Gentropy(\pi')$.

The intuitive understanding is as follows: If $|\pi \cdot \Upsilon| = 1$, comparing with $Gentropy(\pi')$, the calculation of $Gentropy(\pi)$ considers the general terms in $\pi \cdot \Omega$, so its value should not be smaller than the former. If $|\pi \cdot \Upsilon| > 1$, $Gentropy(\pi)$ also considers the siblings of π', which makes its value becomes larger.

4.2. Gain-based Border Generation (GBG)

To generate the concept border, the leaf concepts are merged into their hypernyms recursively. In this process, a gain value is defined and employed to measure whether the merging is profitable.

Refer to Fig. 1, let us consider whether we should merge "Perejil" and "Wight" into "isle". Based on Observation 1, $Gentropy(isle)$ is equal or greater than the average of $Gentropy(Perejil)$ and $Gentropy(Wight)$. If we use one feature to represent these three concepts, some noise will be brought in. But at the same time the merging also provides more accurate similarity. For example, the similarity among the documents belonging to the same cluster and talking about "isle", "Perejil" and "Wight" respectively will be increased, which is exactly the desired result. For measuring whether the trade-off is worthwhile, we define the gain function $gain(\pi)$ for a concept π:

$$gain(\pi) = \frac{\frac{1}{|\pi \cdot \Upsilon|} \sum_{\pi' \in \pi \cdot \Upsilon} Gentropy(\pi')}{Gentropy(\pi)}. \tag{6}$$

It can be observed that $0 < gain(\pi) \leqslant 1$. The larger the gain value is, the less the noise is brought in because of merging π' into π. $gain(\pi) = 1$ means that no noise is brought in. So we always prefer the merging with larger $gain(\pi)$. A parameter θ is used as a profitable threshold. If $gain(\pi) \geqslant \theta$, the merging will be performed.

The above discussion follows a bottom-up fashion, which is a generalization process. We can also consider it in a top-down fashion, which is a specialization process. And the merging operation becomes the splitting operation in which $gain(\pi)$ can still be used in the same way. We name these two methods as GBG-g and GBG-s respectively.

GBG-g is summarized in Algorithm 1. In each loop, we attempt to merge the deepest leaves into their hypernyms. First, we get the deepest leaf π^l (line 5), and locate π^l's hypernym π. If π^l is an

ambiguous concept, we select its hypernym which has the largest depth (line 6). If π contains non-leaf and leaf hyponyms at the same time, these hypernyms will not be merged into π, and we set the border flag under π (line 8). Otherwise, if it is profitable to merge π's leaves into it, all leaves' synsets will be added into $\pi \cdot \Omega$ (line 11), then we delete these leaves from G to make π become a leaf (line 12). If the merging is not profitable, we set the border flag under π (line 14). Finally, the border is composed of all leaves with $flg = true$.

Algorithm 1. GBG-g

```
1: input: the HDAG G of an ontology, threshold θ
2: output: concept border B
3: each concept has a flag flg and taking value false initially
4: while G has leaves with flg = false do
5:   get the deepest leaf π^l with flg = false
6:   get π among π^l's hypernyms, which is the deepest
7:   if π has non-leaf hyponym concepts then
8:     set_border_flag(π)
9:   else
10:    if gain(π) ⩾ θ then
11:      set π · Ω ← π · Ω ∪_{π'∈π·Υ} π' · Ω
12:      set π · Υ ← null
13:    else
14:      set_border_flag(π)
15:    end if
16:  end if
17: end while
18: set B = {π | π's flg is true}
1: proc set_border_flag(π)
2: for all π' in π · Υ do
3:   if π' · Υ = null then
4:     if π' is ambiguous then
5:       delete π' from π · Υ
6:       reset the depth of π'
7:     else
8:       set flg ← true for π'
9:     end if
10:  end if
11: end for
```

In the sub-procedure of *set_border_flag*, the unambiguous leaves of π become the members in $\mathcal{B}$ (line 8), while the ambiguous leaves will be removed from $\pi \cdot \Upsilon$ and its depth is reset based on the depth definition (Eq. (1)). Suppose an ambiguous leaf π' has two hypernyms. After π' is removed from $\pi \cdot \Upsilon$, it becomes an unambiguous leaf and can be treated as an ordinary leaf hereafter in the remaining processing of GBG-g. Note that the depth of π' should be reset based on its remaining hypernyms. The larger the depth of a hypernym of π' is, the earlier the hypernym is considered. Thus, we always try to merge π' into its more specialized hypernym.

GBG-s is summarized in Algorithm 2. A recursive splitting operation is performed from the root. If using π to represent all of its descendants is profitable enough, we will encapsulate its descendants into π first (line 3 of *top_down*), and then add π into $\mathcal{B}$ (line 4 of *top_down*). Otherwise each of π's hyponyms will be used as the parameter to invoke the *top_down* procedure (line 8 of *top_down*). Once an ambiguous concept is merged into any one of its hypernyms (direct or undirect), it will be removed from G (line 5 of *top_down*).

Algorithm 2. GBG-s

```
1: input: the HDAG G of an ontology, threshold θ
2: output: concept border B
3: top_down(root)
1: proc top_down(π)
2: if gain(π) ⩾ θ then
3:   set π · Ω ← π · Ω ∪_{π'∈{π'|π⇝π'}} π' · Ω
4:   put π into B
5:   remove all concepts in {π'|π⇝π'} from G
6: else
7:   for all π' in π · Υ do
8:     top_down(π')
9:   end for
10: end if
```

Before π is added into $\mathcal{B}$, all terms contained by $\pi \cdot \Omega$'s descendants are encapsulated into π, see line 11 in Algorithm 1 and line 3 of *top_down* in Algorithm 2. As a result, the descendants' semantic meanings are merged into π. This merging is performed under the guidance of $gain(\pi)$, which guarantees the trade-off is profitable.

5. Concept-based document representation

After the concept border is generated, it can be used to represent a new document as a concept vector, and each concept in $\mathcal{B}$ is one dimension in the vector. In this subsection, the details of the concept extraction and matching component are described first. Then, the method of representing a document in a weighted concept vector is presented.

5.1. Concept extraction

As illustrated in the previous examples, some concepts are represented by phrases, such as "Isle of Wight" and "object-oriented programming language". Instead of performing time consuming noun phase chunking operation, we use the term set contained by the ontology as a dictionary and perform a forward maximum cutting to extract the concepts from the documents. First, we detect sentence boundaries in a document and get a set of sentences denoted as $\{S_1, S_2, \ldots\}$. Let $(\varpi_1^i, \varpi_2^i, \ldots)$ denote a sequence of tokens in the sentence S_i. Then the segment of the first l tokens $(\varpi_1^i, \ldots, \varpi_l^i)$ is treated as a candidate concept and retrieved in the ontology dictionary. If it fails, the token at the end of the segment is omitted to generate a new candidate concept $(\varpi_1^i, \ldots, \varpi_{l-1}^i)$, and then the above retrieval is performed again. When a certain segment $(\varpi_1^i, \ldots, \varpi_k^i)$ is found in the dictionary, it will be treated as a concept in the document. Then we consider the next segment $(\varpi_{k+1}^i, \ldots, \varpi_{k+l}^i)$. If the retrieval of (ϖ_1^i) still fails, we continue to consider $(\varpi_2^i, \ldots, \varpi_{1+l}^i)$. By this procedure, we can get all concepts contained in the document. l is a predefined maximum cutting length. We use "word" refer to a single word, and "term" refer to both a single word and a multi-word phase in this paper.

5.2. Concept matching

Now suppose that the ambiguous term "Java" appears in two documents, "Object-oriented programming language" and "The top 10 populous islands in the world". The above concept extraction method can only tell us that "Java" is a concept in both documents, but it cannot tell us which one is its matching concept in the ontology, a programming language or an island. To match

"Java" to a correct concept, we consider its context in the document, i.e. the surrounding sentences of "Java", denoted as $\mathcal{C}$. Let Π denote the concepts that contain "Java". Then for each π in Π, we calculate the matching score between $\mathcal{C}$ and π with $match(\mathcal{C}, \pi)$:

$$match(\mathcal{C}, \pi) = \alpha sim(\mathcal{C}, \pi) + (1 - \alpha) sim(\mathcal{C}, \Pi^c_\pi), \tag{7}$$

where $\Pi^c_\pi = \{\pi' | \pi \in \pi' \cdot \Upsilon\} \cup \{\pi' | \pi \rightsquigarrow_n \pi'\}$ is the context of π ($\pi \rightsquigarrow_n \pi'$ means π' is a descendant of π in n hops), and α decides the weights of the two parts. Because the available text information in a concept is quite limited, the traditional VSM (Vector Space Model) similarity is not suitable, a variant of Dice's coefficient [8] is used to calculate the similarity between two text fragments str_1 and str_2:

$$dice(str_1, str_2) = \frac{2|N_1 \cap N_2|}{|N_1| + |N_2|}, \tag{8}$$

where N_i is the multiset (or bag) [43] of the nouns in str_i. For example, $dice(\{a,b\}, \{a,a\})$ is $(2 * 1)/4 = 0.5$, and $dice(\{a,a\}, \{a,a\})$ is $(2 * 2)/4 = 1$. The similarity between a text fragment str and π is defined as:

$$sim(str, \pi) = dice(str, Str(\pi)), \tag{9}$$

where $Str(\pi)$ concatenates $\pi \cdot \sigma$ with each term in $\pi \cdot \Omega$ to get a new string. We use Eq. (10) to compute $sim(str, \Pi)$:

$$sim(str, \Pi) = \frac{\sum_{\pi \in \Pi} sim(str, \pi)}{|\Pi|}. \tag{10}$$

Finally, the concept π^m with the maximum matching score with $\mathcal{C}$ is selected as the correct matching concept:

$$\pi^m = \arg\max_{\pi \in \Pi} match(\mathcal{C}, \pi). \tag{11}$$

5.3. Weighted concept-based document representation

The border $\mathcal{B}$ generated by the GBG algorithms is used to represent a document. In the vector space model, each concept π_i in $\mathcal{B}$ is one dimension. Similar to TF-IDF, we introduce CF-IDF to indicate the importance of π_i in a certain document d_j, calculated as follows:

$$cf_{i,j} = \frac{f_{i,j}}{\sum_{\pi_k \in d_j} f_{k,j}}, \tag{12}$$

$$idf_i = \log \frac{|D|}{1 + |\{d | \pi_i \in d\}|}, \tag{13}$$

$$cfidf_{i,j} = cf_{i,j} \times idf_i, \tag{14}$$

where $f_{i,j} = \sum_{t_l \in \pi_i \cdot \Omega} n_{l,j}$ is the frequency of π_i in d_j ($n_{l,j}$ is the frequency of the term t_l in d_j), $\pi_i \in d$ means that at least one term in $\pi_i \cdot \Omega$ is contained by the document d.

6. Technique details and time complexity

6.1. Virtual concept

We find that ontology's structure is quite unbalanced. In the graph on left hand side of Fig. 4, although c, d and e have a common hypernym r, c has more descendants than d and e. Reviewing the structure shown in Fig. 1, the semantic meanings between "Bali" and "Java" are closer than that between "Bali" and "isle" since both "Bali" and "Java" are leaf concepts while "isle" is a non-leaf concept. If the degree of the unbalance becomes severe, as shown in Fig. 4, this kind of difference will increase dramatically. Hence, combining c, d and e together will surely cause a lot of noise. However, d and e are very likely to have similar meanings because both of them are leaves. We introduce a *virtual concept* v to make it possible that d and e are combined and c is excluded, as shown in the graph on the right hand side of Fig. 4. More formally, if the concept π contains more than one leaf concepts and at least one non-leaf concept, we introduce a virtual concept v for π with the following operations:

1. $v \cdot \Upsilon \leftarrow \{\pi' | \pi' \in \pi \cdot \Upsilon \text{ and } \pi' \cdot \Upsilon = null\}$,
2. $\pi \cdot \Upsilon \leftarrow \pi \cdot \Upsilon - v \cdot \Upsilon$,
3. $\pi \cdot \Upsilon \leftarrow \pi \cdot \Upsilon \cup \{v\}$.

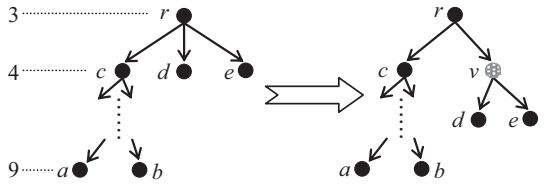

Fig. 4. Unbalanced structure and virtual concept. The depth of the concept is given at the beginning dashed line.

We perform the procedure of adding virtual concepts before the algorithms GBG-g and GBG-s.

6.2. Gain function calculation technique

We first analyze the intrinsic structure of $gain(\pi)$, then propose an efficient algorithm to calculate the gain value for each π in an ontology. Suppose *idx* is an inverted index of the ontology concepts on a document set $D = \{u_1, u_2, \ldots\}$. The index record for π is $\pi \rightarrow \{\langle w^\pi_1, w^\pi_2, \ldots \rangle, w^\pi\}$, where $w^\pi_i = \sum_{t_j \in \pi \cdot \Omega} w_{i,j}$ indicates the weight of π in u_i, and $w^\pi = \sum_{u_k \in D} w^\pi_k$ is the accumulated weight of π in D. Now suppose π has only two leaf hyponyms, π_1 and π_2. The weight of Ω^g_π in u_i, denoted as $w_i^{\Omega^g_\pi}$, is additively separable, $w_i^{\Omega^g_\pi} = \sum_{t_j \in \Omega^g_\pi} w_{i,j} = w_i^{\pi_1} + w_i^{\pi_2} + w_i^\pi$. This property can be generalized to the concepts with any depth, and the recursive calculation formula is:

$$w_i^{\Omega^g_\pi} = \begin{cases} w_i^\pi & if\ \pi \cdot \Upsilon = null, \\ w_i^\pi + \sum_{\pi' \in \pi \cdot \Upsilon} (w_i^{\Omega^g_{\pi'}}) & otherwise. \end{cases} \tag{15}$$

Then the probability $p(u_i | \Omega^g_\pi)$ can be calculated as:

$$p(u_i | \Omega^g_\pi) = \frac{w_i^{\Omega^g_\pi}}{\sum_{u_k \in D} w_k^{\Omega^g_\pi}}. \tag{16}$$

Based on Eqs. (2) and (5), $Gentropy(\pi)$ can be calculated. The calculation of *gain* for each π is summarized in Algorithm 3. In a bottom-up fashion (line 3), each $w_i^{\Omega^g_\pi}$ for π is calculated recursively and saved for reuse in the future (line 6). Then $Gentropy(\pi)$ is calculated and saved in lines 7 and 8. Finally $gain(\pi)$ is calculated in line 10.

Algorithm 3. Gain calculation

```
1: input: An ontology graph G, idx on a document set D
2: output: gain value for each π
3: for depth from maxDepth to 0 do
4:   for all π with d(π) = depth do
5:     retrieve π → {⟨w^π_1, w^π_2, ...⟩, w^π} from idx
6:     calculate and save w_i^{Ω^g_π} for each u_i (Eq. (15))
7:     calculate p(u_i|Ω^g_π) for each u_i (Eq. (16))
8:     calculate and save Gentropy(π)
9:     if π · Υ ≠ null then
10:      calculate gain(π) according to Eq. (6)
11:    end if
13:  end for
14: end for
```

6.3. Time complexity

The overall time complexity of ACR model should include the time consuming of text preprocessing, concept extraction and matching, inverted indexing and the GBG algorithms. The first three parts are well investigated in text mining and information retrieval. So we only analyze the complexity of the GBG algorithms. The most time consuming operation in the GBG algorithms is the gain value calculation for each concept. Now suppose each document in D is treated as an individual u_i, the time complexity of calculating $w_i^{\Omega_\pi^g}$ for all u_i is $O(|D| * |\pi \cdot \Upsilon|)$. Then the calculation of $p(u_i|\Omega_\pi^g)$ for all u_i takes $O(|D|)$. $Gentropy(\pi)$ calculation also takes $O(|D|)$. $gain(\pi)$ calculation takes $O(|\pi \cdot \Upsilon|)$. Thus, the time complexity for calculating one concept's gain value is $O(|D| * |\pi \cdot \Upsilon|)$. Let Π denote the concept set of an ontology, and $avg(|\Upsilon|)$ denote the average size of Υ in Π, the overall time complexity of the GBG algorithm is $O(|D| * avg(|\Upsilon|) * |\Pi|)$.

7. Enhanced ontology

As we know, the expert-edited ontology is rather static. Its coverage and ability of containing new terms are poor compared with Wikipedia. We propose a method to integrate Wikipedia entities into an existing ontology to construct an enriched ontology. We apply this method on WordNet and generate an enhanced ontology called WordNet-Plus. First, for a Wikipedia entity, a set of candidate concepts are selected from the ontology. Then, we calculate the similarity between each candidate concept and the Wikipedia entity. Finally, those similar concepts are determined as the hypernyms of the entity in the ontology.

7.1. Wikipedia overview

Generally speaking, Wikipedia also employs a hierarchical structure to organize its categories and entities. However, because of the collaborative editing, the massive number of editors and the huge quantity of information, Wikipedia's structure is much more complicated than any expert-edited ontology. A small fragment of Wikipedia structure is shown in Fig. 5. It is about a Wikipedia article "Bacan"[3] and part of its related categories. We can see that the flexibility is quite large when the editors assign a category for an article or a sub-category. "Bacan" is an island of Indonesia as well as a part of Maluku province. While "Maluku" also refers to the largest island among Maluku Islands, this leads a path from "Islands of Indonesia" to "Maluku". Consequently, a new 3-hop path exists from "Islands of Indonesia" to "Bacan". From "Islands of Asia" to "Islands of Indonesia", there are also two pathes with hop 1 and 2, respectively. We can see that, compared with the finely defined traditional ontology (refer to Fig. 1), Wikipedia involves much more abundant semantic information.

From the above example, we observe that it is difficult to accurately match a Wikipedia category with an ontology concept. For example, although the entity "Bacan" has the same semantic attribute as the ontology concepts "Bali" and "Java", we cannot attach "Bacan" under the concept "island" directly by matching the category name against the concept "island" since no category of "Bacan" is named "island". To tackle this problem, we propose an entity attaching method to incorporate Wikipedia entities into an ordinary ontology.

7.2. Entity attaching for WordNet-Plus generation

Formally, a *Wikipedia Entity (WE)* ϕ is a triple $\langle \tau, \sigma, \Psi \rangle$, where τ is the title of ϕ, σ denotes the text description of ϕ, and Ψ is the set of categories that ϕ belongs to. We refer to the items with $\phi \cdot \tau$, $\phi \cdot \sigma$ and $\phi \cdot \Psi$ respectively. Each element of Ψ is denoted as ψ, and the entire Wikipedia entity set is denoted as Φ. By considering the head terms of categories, we can increase the chance of successfully matching between the category name and ontology concept. For example, "Islands of Indonesia" is a noun phrase, and its head term is "island". Therefore, we also formally introduce *Category Head (CH)* as the head term of a category, denoted as η. Taking the Wikipedia article Bacan as an example, the items of the corresponding entity are: $\phi \cdot \tau$ is "Bacan", $\phi \cdot \sigma$ is "Bacan refers to a group of islands in the Maluku Islands of Indonesia and to that group's largest island ..." and $\phi \cdot \Psi$ is {"Islands of Indonesia", "Maluku", "Landforms of Indonesia", "Islands by country", "Islands of Asia", "Islands of Southeast Asia", "Provinces of Indonesia", "Maluku Islands"}. We employ the first sentence of the article as the description, and the closest two levels of categories (refer to Fig. 5) of the article to compose Ψ.

For each ϕ, we construct a forward index: $\phi \rightarrow \{\eta_{i_1} : f_{i_2}, \eta_{i_2} : f_{i_2}, \ldots\}$, where f_{i_n} means the frequency of η_{i_n} appearing as a CH in all elements of $\phi \cdot \Psi$. The head terms are sorted in the descending order according to their frequencies. This index is named *Head Index (HI)* of Wikipedia entity. We interpret how to construct this index with the above running example. First, the head term of each category is extracted. Then, the duplicated heads are merged, meanwhile the frequency is also aggregated. After sorting the heads based on their frequencies, we get such an index record: $Bacan \rightarrow \{island : 5, landform : 1, maluku : 1, province : 1\}$. Therefore, each head term describes a higher level semantic meaning.

Algorithm 4. Entity attaching for WordNet-Plus generation

input: Wikipedia entity set Φ,
  existing ontology (e.g. WordNet) O
output: Generated WordNet-Plus O^p
initial O^p with O
for all $\phi \in \Phi$ **do**
  concept sets $\Pi_1^{can} \leftarrow \{\}, \Pi_2^{can} \leftarrow \{\}, \Pi^b \leftarrow \{\}$
  for all $\psi \in \phi \cdot \Psi$ **do**
    $\Pi_1^{can} \leftarrow \Pi_1^{can} \bigcup \{\pi | \psi \in \pi \cdot \Omega, \pi \ in \ O\}$
  end for
  if $\Pi_1^{can} = \{\}$ **then**
    construct *IECH idx* for ϕ
    for all $\eta_{i_n} \in idx$ **do**
      $\Pi_2^{can} \leftarrow \Pi_2^{can} \bigcup \{\pi | \eta_{i_n} \in \pi \cdot \Omega, \pi \ in \ O\}$
    end for
  end if
  if $\exists \pi_k \in \Pi_1^{can} \bigcup \Pi_2^{can} s.t. fit\_n(\phi, \pi_k) > 0.5$ **then**
    $\Pi^b \leftarrow \{\pi_k\}$
  else
    $\Pi^b \leftarrow \{\pi_k - \pi_k \in \Pi_1^{can} \bigcup \Pi_2^{can}, \pi_k$ has top-2 *fit_n*$\}$
  end if
  create π^ϕ
  $\pi^\phi \cdot \Omega \leftarrow \{\phi \cdot \tau\}$
  $\pi^\phi \cdot \sigma \leftarrow \phi \cdot \sigma$
  for all $\pi \in \Pi^b$ **do**
    retrieve π's identical concept π^g from O^p
    $\pi^g \cdot \Upsilon \leftarrow \pi^g \cdot \Upsilon \bigcup \{\pi^\phi\}$
  end for
end for

Given a Wikipedia entity ϕ, we first construct a candidate set Π^{can} which contains the potential attaching concepts of ϕ. Then the similarity between each candidate and ϕ is calculated, the most

[3] http://en.wikipedia.org/wiki/Bacan.

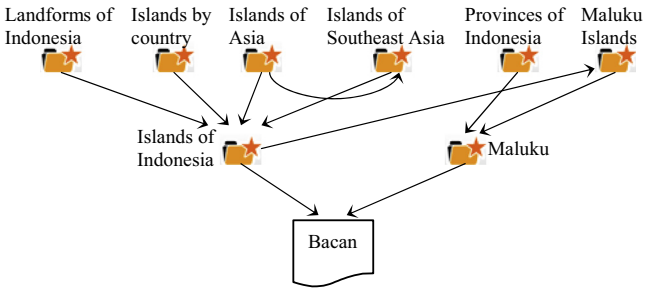

Fig. 5. A fragment of Wikipedia structure.

similar one (or two) is chosen as the semantic hypernym of ϕ in the expert-edited ontology. In the generation of Π^{can}, we employ a two-phase strategy. In the first phase, we retrieve the elements of $\phi \cdot \Psi$ in the target ontology, say WordNet. If a certain concept π satisfies $\exists \psi \in \Psi \rightarrow \psi \in \pi \cdot \Omega$, this concept will be put into Π^{can}. After the first phase, if Π^{can} is not empty, a fitness value between ϕ and each $\pi_k \in \Pi^{can}$ is calculated as $fit'(\phi, \pi_k) = match(\phi \cdot \sigma, \pi_k)$. If Π^{can} is empty, we move to the second phase. We first construct the HI of ϕ. Then, the head terms in HI are retrieved in the ontology to compose the set Π^{can}. In this phase, the fitness value is calculated as $fit''(\phi, \pi_k) = \log(f_{i_n} + 1) * (match(\phi \cdot \sigma, \pi_k) + 1)$, where f_{i_n} is the frequency of η_{i_n} ($\eta_{i_n} \in \pi_k \cdot \Omega$). In this two-phase way, the original category names of ϕ are of the first priority. If the category names do not exist in the ontology, we turn to the head terms of the categories for help. We use $fit(\phi, \pi_k)$ to uniformly denote the fitness value since all π_k's in Π^{can} come from the same phase, either the first phase or the second phase. The normalized fitness value $fit\_n(\phi, \pi_k)$ is calculated as:

$$fit\_n(\phi, \pi_k) = \frac{fit(\phi, \pi_k)}{\sum_{\pi_j \in \Pi^{can}} fit(\phi, \pi_j)}. \tag{17}$$

If any π_k's fit_n value dominates others significantly, say greater than 0.5, it will be selected as the attaching concept for ϕ in the ontology. Otherwise, the top-two fitting concepts will be selected. We create a new concept π^{ϕ} with $\pi^{\phi} \cdot \Omega = \{\phi \cdot \tau\}$ and $\pi^{\phi} \cdot \sigma = \phi \cdot \sigma$. Finally, π^{ϕ} is attached under the best fitting concepts. We apply this method on WordNet and generate an enriched ontology, named WordNet-Plus. The entire method is summarized in Algorithm 4. From the generating process, it can been seen that the original structure of WordNet is preserved in WordNet-Plus, and new terms from Wikipedia are introduced into WordNet-Plus.

8. Experiments of ACR model

8.1. Experimental setting

We evaluate the perform of the ACR model on two text mining tasks, namely, document clustering and document classification. Three previous ontology-based methods are also employed to conduct comparison evaluation. The first ontology-based method is known as the "only" strategy in [19]. In this strategy, each concept's synset is used as one dimension in the document representation vector, and its weight is decided by the terms in the synset. This baseline is referred to as SS. The second ontology-based method is the WordNet lexical categories (WLC) technique proposed in [35]. In WLC, 41 lexical categories for nouns and verbs are used to construct the feature vector, as a result the vector has 41 dimensions. The third ontology-based method is the WordNet ontology (WO) technique proposed in [35]. WO employs the output of WLC as the initial input and relies on the structure of WordNet to group words according to the concepts they are related to. One linear projection model, namely LDA [4], is employed to generate the presentation of document in the semantic space with 50 hidden topics. And then, such presentation is utilized as the input feature vector in the tasks of document clustering and document classification.

Four data sets are used in the experiments: 20 Newsgroups (NG20), TREC data extracted from the document collection Disc 5, ODP page set and OHSUMED (MED). 20 Newsgroups and TREC data are two ordinary document sets; OHSUMED [16] is a professional medical science data; ODP page set contains the Web pages crawled from five ODP categories, including Arts, Business, Computers, Health, and Sports. In OHSUMED, there are 106 queries which have manual labeled results. We use each query as a predefined cluster including the documents which are labeled as "definitely relevant". The documents definitely relevant to more than one queries are eliminated. Finally, we get 101 clusters with 1870 documents. The details of the data sets are given in Table 1.

8.2. Document clustering

In document clustering experiments, we employ K-Means algorithm to perform clustering, and it is executed three times for each data set to get an average result. The entire document collection is used as the input information of the *gain* value calculation in GBG algorithms. No cluster information is used in the calculation, hence, each u_i mentioned in Section 4 refers to an individual document.

8.2.1. Evaluation criteria

The purity measure is employed to evaluate the clustering performance. Let $G = \{g_1, g_2, \ldots\}$ denote the cluster set generated by K-Means, and $C = \{c_1, c_2, \ldots\}$ denote the predefined clusters. To calculate the purity, each g_k is assigned to the predefined cluster c_i which is the most frequent in g_k, and then the purity is measured by counting the number of correctly assigned documents and divided by $|D|$. Formally, $Pur(C, G)$ is calculated as:

$$Pur(C, G) = \frac{1}{|D|} \sum_k \max_i |g_k \cap c_i|. \tag{18}$$

8.2.2. Results

The clustering result is given in Table 2. Both of our methods can outperform the comparison methods. Considering NG20 and TREC data sets, our methods outperform the ontology-based methods with lager margins, about 0.05 (7%) to 0.09 (20%). Compared with the linear projection model LDA, our methods are also able to achieve better results on all data sets. The results demonstrate that our adaptive manner of generating document representations captures the characteristics of the corpus more precisely. Of the first three data sets, the performances of GBG-g and GBG-s are similar. For the fourth data MED, GBG-g outperforms GBG-s by more than 0.06 (8%). In Table 2, the bold results of our methods indicate that the performance of our method on the corresponding data sets is significantly better than that of all the comparison methods under paired t-test with $P < 0.05$.

Table 1
Details of the data sets.

	# Doc.	# Cate.	Categories
NG20	19,997	20	ALL
TREC	12,637	20	354, 362, 365, 376, 393, 394, 397, 398, 401, 417, 422, 423, 432, 433, 434, 442, 446, 617, 625, 627
ODP	5000	5	Arts, Business, Computers, Health, Sports
MED	1870	101	ALL

Table 2
Clustering performance comparison.

	NG20	TREC	ODP	MED
GBG-g	**.808**	**.516**	**.825**	**.795**
GBG-s	**.807**	**.536**	**.830**	**.731**
SS	.752	.449	.783	.711
WLC	.701	.425	.716	.608
WO	.724	.438	.730	.659
LDA	.768	.497	.787	.707

8.2.3. Parameter sensitivity analysis

The effect of θ in the clustering is shown in Table 3. The performance of GBG-g algorithm is not sensitive to θ, and it can outperform the existing method SS under any θ value. It is because GBG-g algorithm merges the concepts in a bottom-up fashion, and each merging is performed among the concepts with high semantic relation to each other. Therefore, the consistency of the semantic meaning of a derived new concept can be guaranteed, even when the θ value is small. When the θ value is too large, say 1.0, the constraint becomes too strict and the related concepts cannot be merged sufficiently. Consequently, the result is not as good as the result under a smaller θ, say 0.7.

GBG-s is relatively more sensitive to θ than GBG-g. When θ is small, the top-down splitting will stop early at some general concepts and these concepts are added into $\mathcal{B}$. As a result, the general meaning of the concepts in $\mathcal{B}$ brings in more noise to the similarity calculation. Therefore, a larger θ value can generally achieve better results than a smaller value in GBG-s. Generally speaking, GBG-g is better and more stable than GBG-s. It is because GBG-g considers the specialized concepts first in generating the concept border, which are more important than the general ones from the semantic point of view. Furthermore, GBG-g can deal with the unbalanced structure more effectively, because it does not merge the leaf concepts with the non-leaf concepts.

Based on the above discussions, the value of θ used in GBG-g is 0.7, and in GBG-s is 0.9 in the comparative result depicted in Table 2.

8.3. Document classification

In the classification experiments, LibSVM [6] with linear kernel is employed to conduct the classification and 5-fold cross-validation is adopted. Note that the *gain* calculation only needs the training set. Because there exist many small classes, with less than 5 documents, in MED, we do not use this data set in the classification experiment.

8.3.1. Evaluation criteria

The overall *F*-measure score of the classification result can be computed in two manners, namely, micro-average and macro-average [49]. In the macro-average manner, the precision and recall for each category c_i are calculated first, denoted as P_i and R_i. Then *F*-measure for each category c_i is calculated as:

$$F_i = \frac{2P_iR_i}{P_i + R_i}, \tag{19}$$

The macro-averaged *F*-measure is the average of *F*-measure for each category:

$$F^{ma} = \frac{\sum_i F_i}{|C|}, \tag{20}$$

where C denotes the predefined classes. In the micro-averaging manner, *F*-measure is computed globally over all category decisions. The global precision and recall are calculated as:

$$P = \frac{\sum_i TP_i}{\sum_i (TP_i + FP_i)}, \tag{21}$$

and

$$R = \frac{\sum_i TP_i}{\sum_i (TP_i + FN_i)}, \tag{22}$$

where TP_i, FP_i and FN_i are true positive, false positive and false negative numbers for category c_i. Micro-averaged *F*-measure is defined as:

$$F^{mi} = \frac{2PR}{P + R}. \tag{23}$$

8.3.2. Results

The classification result is given in Table 4. Except for F^{ma} of GBG-s on NG20, our methods dominate all other cases. On the TREC data, both of our methods can outperform the comparison ontology-based methods with margins from 0.055 (9%) to 0.113 (18%). LDA method is very competitive and achieves better results than the ontology-based comparisons on two data sets, namely, TREC, and ODP. GBG-g performs better than GBG-s on NG20 and TREC, while GBG-s achieves better results on the ODP data. The bold results of our methods indicate that the performance of our method on the corresponding data sets is significantly better than that of all the comparison methods under paired *t*-test with $P < 0.05$.

8.3.3. Parameter sensitivity analysis

The effect of θ in the classification is shown in Table 5. Again we find that GBG-s is more sensitive to θ than GBG-g because of the same reasons discussed above. Without exception, the best results for both GBG-g and GBG-s are achieved when θ is 1. Under the cluster granularity of calculating the gain value, a larger θ can prevent the concepts that can bring in much noise to be added into $\mathcal{B}$. At the same time, because the *gain* value is calculated considering the cluster information, the related semantic meaning in the same cluster can still be merged. Thus, the value of θ used in both GBG-g and GBG-s is 1 in the classification experiment.

Table 3
Parameter θ's effect in ACR model for the clustering.

θ	GBG-g				GBG-s			
	NG20	TREC	ODP	MED	NG20	TREC	ODP	MED
0.1	.795	.494	.807	.766	.710	.447	.729	.599
0.2	.795	.503	.804	.740	.726	.450	.732	.754
0.3	.765	.502	.824	.753	.779	.503	.718	.744
0.4	.792	.499	.815	.772	.781	.507	.751	.756
0.5	.800	.501	.815	.780	.755	.515	.781	.734
0.6	.799	.500	.819	.773	.795	.497	.785	.761
0.7	.808	.516	.825	.795	.792	.500	.780	.762
0.8	.803	.531	.828	.770	.789	.521	.766	.739
0.9	.806	.502	.837	.762	.807	.536	.830	.731
1.0	.802	.497	.809	.785	.805	.535	.826	.741

Table 4
Classification performance comparison.

	NG20		TREC		ODP	
	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}
GBG-g	**.934**	**.818**	**.696**	**.643**	**.809**	.677
GBG-s	.907	.780	**.692**	**.635**	**.825**	**.689**
SS	.904	.793	.631	.580	.783	.653
WLC	.859	.737	.583	.542	.723	.604
WO	.876	.764	.602	.567	.754	.636
LDA	.891	.782	.643	.608	.785	.661

Table 5
Parameter θ's effect in ACR model for the classification.

θ	GBG-g						GBG-s					
	NG20		TREC		ODP		NG20		TREC		ODP	
	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}
0.1	.930	.814	.659	.607	.796	.666	.824	.708	.537	.493	.688	.575
0.2	.932	.818	.655	.604	.802	.671	.820	.699	.530	.486	.692	.579
0.3	.930	.816	.657	.608	.799	.669	.830	.715	.544	.499	.657	.549
0.4	.921	.808	.665	.613	.805	.674	.875	.749	.524	.479	.666	.555
0.5	.917	.802	.655	.604	.791	.661	.887	.764	.565	.515	.764	.639
0.6	.919	.804	.669	.617	.796	.666	.883	.757	.595	.538	.802	.670
0.7	.917	.803	.664	.613	.792	.662	.902	.772	.642	.587	.806	.673
0.8	.930	.816	.680	.626	.807	.675	.898	.772	.666	.615	.799	.667
0.9	.930	.812	.686	.633	.807	.675	.896	.768	.685	.631	.800	.667
1.0	.934	.818	.696	.643	.809	.677	.907	.780	.692	.635	.825	.689

Interestingly, we find that on NG20, GBG-g can perform slightly better with both small and large θ values than with the medium values. It is because after the concepts are significantly merged under a small θ, the benefit obtained for calculating the similarity within a cluster overwhelms the noise brought in at the same time. While a larger θ achieves a better result by suppressing the amount of noise. For other data sets, this exceptional situation does not happen. Therefore, we adopt a larger θ value for all data sets.

9. Experiments of enhanced ontology

In this section, we first give some general information of WordNet-Plus construction. Then we show some case studies of WordNet-Plus in Section 9.2. After that, the quality evaluation of WordNet-Plus is conducted in Section 9.3. The performance of WordNet-Plus under the ACR model is investigated in Section 9.4.

9.1. Information of WordNet-Plus construction

After eliminating the articles without category information or article content, 3,012,229 articles are collected from a Wikipedia dump. For each article (i.e. entity), the first paragraph is used as its discerption, and only its direct categories are considered in the construction of *HI*. If we consider two or more levels, some very general categories will be included, such as sports and arts, which will make it difficult to locate an accurate concept for the entity in WordNet. Finally, WordNet-Plus incorporates 1,060,126 new leaves coming from 611,161 distinct Wikipedia entities. In WordNet 2.1 we used, there are 89,646 concepts. Thus, on average each concept has 6.8 new hyponyms coming from Wikipedia.

9.2. Case study in WordNet-Plus

Table 6 presents some examples about under which concept one Wikipedia entity is attached. The entities are given in the second column, and their attaching WordNet concepts are shown in the third column. Note that the synset is used to represent a concept.

We can see that reasonable higher-level concepts for the entities are found. In example 7, we attach a Japanese fruit "dekopon" under the concept "citrous fruit", which is a rare term and it cannot even be retrieved in the dictionary. Example 8 shows that "Friends" is correctly recognized as a series, and attached under the concept ⟨06621447, {serial, series}, "a serialized set of programs, 'a comedy series' ", {"soap opera", ...}⟩. "J2EE application" is recognized as a software platform, and attached under ⟨03962685, {platform}, "the combination of a particular computer and a particular operating system", {}⟩. The famous "Marshall Plan" is attached under ⟨00250259, {development}, "act of improving by expanding or enlarging or refining, 'he congratulated them on their development of a plan to meet the emergency' ", {...}⟩. It is very interesting that "Water 1st International" (a non-profit organization) is treated as an instance of "water system", although not correct, it still gives us some positive information.

Table 6
Case study 1.

#	Wikipedia entity	WordNet concept
1	Albertville micropolitan area	Alabama
2	Book of cool	Entertainment
3	Bug bash	Testing
4	Centaurus	Constellation
5	Chinese calendar	Culture
6	Conditional entropy	Information, entropy, selective information
7	Dekopon	Citrus, citrus fruit, citrous fruit
8	Friends	Serial, series
9	IPTV	Television, television system
10	J2EE application	Platform
11	LaTeX	Software, software program, computer software, software system, software package, package
12	Marshall Plan	Development
13	Standard molar entropy	Information, selective information, entropy
14	Standard template library	Library, program library, subroutine library
15	Sydney film festival	Festival
16	Thenar eminence	Hand
17	Twitter	Network, web
18	Visual Basic.NET	BASIC
19	Water 1st International	Water system, water supply, water
20	Yahoo! Meme	Network, web

Table 7 gives some examples from another perspective to show the meaning consistency of the entities attached under the same concept. In example 1, different BASIC IDEs are found and attached under the concept "BASIC". In examples 2 and 3, chemistry-related phrases are finely separated into "electrochemistry" and "organic chemistry". In example 4, "Windows Journal" (an application for PC) and "TabletKiosk" (a manufacturer of PC) are mis-attached under the concept "PC". This is because they are members of the category "Tablet PC", obviously they are wrong category allocations. In example 5, we find more than 20 kinds of tea, even some are quite unacquainted to Chinese.

9.3. Quality evaluation of WordNet-Plus

9.3.1. Evaluation setup

For evaluation purpose, we define five grades to score the correctness of the semantic relation between one Wikipedia entity

Table 7
Case study 2.

#	WordNet concept	Wikipedia entities
1	BASIC	Microsoft BASIC; Visual Basic.NET; Dartmouth BASIC; Galaksija BASIC; Visual Basic; Microsoft Small Basic
2	Electrochemistry	Electrochemical window; Mercury beating heart; Faraday-efficiency effect; Electrochemical reaction mechanism; Proton coupled electron transfer
3	Organic chemistry	Intramolecular reaction; On water reaction; Free radical reaction; Alpha and beta carbon; Trapp mixture; Combustion analysis; Diradical; Schlenk equilibrium
4	Microcomputer PC, personal computer	Tablet PC; Vulcan FlipStart; EO Personal Communicator; Sony U-series; VoodooPC; Ultra-Mobile PC; Sony Vaio UX Micro PC; Pepper Pad; HP Compaq TC1100; HP Compaq TC4200; HP Compaq TC4400; NanoBook; TabletKiosk; Gateway C-series; Comparison of Windows Journal; Tablet PCs; HP TouchSmart; JooJoo; HP Pavilion TX1000 series Tablet PC
5	Tea	Chinese tea; Junshan Yinzhen tea; Roasted barley tea; Enviga; Cannabis tea; Rhododendron groenlandicum; Qi Lan tea; Jin Suo Chi tea; Fo Shou tea; Huang Guanyin tea; Bu Zhi Chun tea; Jiaogulan; Coca tea; Rhododendron tomentosum; Ilex guayusa; Sungnyung; Chamei; Hibiscus tea; Lei cha; Xia Sang Ju; Matcha; Konacha tea

and its hypernym (the corresponding WordNet concept) in WordNet-Plus. The details and examples are given in Table 8. In grades Excellent and Good, the matching quality is very high, and these cases can come to an agreement with human being's general knowledge. In Fair grade, the relation between the entity and the concept is not that strong but still reasonable. If the case is difficult to judge, we put it into Neutral grade. Finally, the wrong cases are put into Bad. We have 5 volunteers to do the evaluation. For a certain case, each volunteer gives a score, then we average all scores and round the average to the nearest grade.

We randomly sample 100 cases from each of the following domains: Business, Science, and Sports. Take Sports domain as an example, we first get the sub-graph with root concept "Sports" from the HDAG of WordNet-Plus. Then, a sample space containing all the inserted entities in the sub-graph is constructed. Finally, we perform sampling without replacement to generate the sample set.

9.3.2. Evaluation result

The evaluation result is given in Table 9. The overall average score is 4.35. Science domain achieves the best result, 93% of the cases are Excellent or Good. Business domain is not as good as the other two. It is because the meanings of the entities in Science and Sports domains are less ambiguous, and the Wikipedia contributors can come to an agreement on the category allocation. The information in Business domain is much more complicated, the contributors may give a batch of category names to an entity, which will mislead our method.

We examine the Bad and Neutral cases, and find that the poor category allocation in Wikipedia is the main incentive. Take "La pelota vasca ⇒ politics, political science" as an example, "La pelota vasca" is a politics film, but is wrongly categorized into "Basque politics" category. In the entity attaching algorithm, both "film" and "politics" concepts are put into Π^b. As a result, "La pelota vasca" is attached under "politics". Another incentive is that some extracted head terms may represent much more general meanings than the original categories. For example, "Logic simulation" is a technique of "design automation". Because the concept "design automation" does not exist in WordNet, we use the head term "automation", which is too general.

Table 8
Evaluation grades.

Grade (Score)	Examples (Wikipedia entity ⇒ WordNet concept)
Excellent (5)	Square foot gardening ⇒ gardening, horticulture Technical diving ⇒ dive, diving
Good (4)	Plastics engineering ⇒ industry, manufacture Earnings call ⇒ finance
Fair (3)	Code project open license ⇒ law, practice of law Decimal dozen ⇒ storage
Neutral (2)	Logic simulation ⇒ automation, mechanization Medical resident work hours ⇒ education
Bad (1)	La pelota vasca ⇒ politics, political science National library week ⇒ science, scientific discipline

Table 9
Quality evaluation result of WordNet-Plus.

	Excellent	Good	Fair	Neutral	Bad
Business	45	22	15	14	4
Science	86	7	4	0	3
Sports	70	13	5	12	0

9.4. WordNet-Plus performance in text mining

In this subsection, we investigate whether WordNet-Plus can work as good as WordNet or even better by applying them in the ACR model. We perform experiments on two text mining tasks, namely, document clustering and document classification. The experimental settings are the same as given in Section 8.

9.4.1. Document classification

The classification result is given in Table 10. On ODP, the performance of WordNet-Plus can dominate that of WordNet. Because the Web data contains more new terms, such as computer terminology and sports stars, the high coverage of WordNet-Plus can improve the result, while WordNet suffers from the limitation of its low coverage. NG20 was constructed nearly 20 years ago and some terminologies in it have been included by WordNet. Therefore, the improvement of WordNet-Plus for NG20 is not as much as for ODP. After checking the documents in our TREC data, we find that their contents are about very common things, and the amount of new terms is less. Thus, on TREC WordNet, outperforms WordNet-Plus because of the benefit from its pure content and not suffering from its low coverage. In summary, WordNet-Plus is good at dealing with newly-minted data, which is coincident with our expectation. The bold results indicate that the performance of WordNet-Plus is significantly better than that of WordNet under the corresponding algorithm with paired t-test ($P < 0.05$).

For WordNet, the value of the parameter θ is 1 for both GBG-g and GBG-s. While for WordNet-Plus, θ takes 0.9 and 1 for GBG-g and GBG-s respectively. Notice that the value of θ is 1 for WordNet

Table 10
Classification performance comparison between WordNet and WordNet-Plus.

		NG20		TREC		ODP	
		F^{mi}	F^{ma}	F^{mi}	F^{ma}	F^{mi}	F^{ma}
WordNet-Plus	GBG-g	**.941**	**.823**	.696	.636	**.850**	**.710**
	GBG-s	**.919**	.783	.685	.624	**.830**	**.695**
WordNet	GBG-g	.934	.818	.696	.643	.809	.677
	GBG-s	.907	.780	.692	.635	.825	.689

in GBG-g, larger than that for WordNet-Plus. After new entities are encapsulated into WordNet-Plus, loosing the parameter slightly can get more benefit from merging the leaf nodes into their hypernyms, meanwhile the negative effect caused by the included noise is not that much.

10. Conclusions

In this paper, we propose an Adaptive Concept Resolution model to adaptively learn a concept border from an ontology taking into the consideration of the characteristics of the particular document collection. Then this border can provide a tailor-made semantic concept representation for a document coming from the same domain. Another advantage of ACR is that it is applicable in both classification task where the groups are given in the training document set, and clustering task where no group information is available. Two algorithms are proposed, namely, GBG-g and GBG-s, to generate the concept border. In the experiments, GBG-g performs better and is more stable than GBG-s. We also construct an enhanced ontology, WordNet-Plus, by integrating the information of Wikipedia into an expert-edited ontology WordNet. Generally, the performance of WordNet-Plus in text mining is competitive, and can outperform WordNet in the Web page classification task because of its high coverage on new terms.

Acknowledgements

The work described in this paper is supported by grants from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: CUHK413510) and the Direct Grant of the Faculty of Engineering, CUHK (Project Code: 4055034). This work is also supported by NSFC with Grant No. 61370054 and 973 Program with Grant No. 2014CB340405.

References

[1] K. Amailef, J. Lu, Ontology-supported case-based reasoning approach for intelligent m-government emergency response services, Decis. Support Syst. 55 (1) (2013) 79–97.
[2] L. Bing, W. Lam, T.-L. Wong, Wikipedia entity expansion and attribute extraction from the web using semi-supervised learning, in: Proceedings of the Sixth ACM International Conference on Web Search and Data Mining, WSDM '13, 2013, pp. 567–576.
[3] L. Bing, B. Sun, S. Jiang, Y. Zhang, W. Lam, Learning ontology resolution for document representation and its applications in text mining, in: Proceedings of CIKM, 2010, pp. 1713–1716.
[4] D.M. Blei, A.Y. Ng, M.I. Jordan, Latent dirichlet allocation, J. Mach. Learn. Res. 3 (2003) 993–1022.
[5] C. Bouras, V. Tsogkas, A clustering technique for news articles using WordNet, Knowl.-Based Syst. 36 (0) (2012) 115–128.
[6] C.C. Chang, C.J. Lin, LIBSVM: a library for support vector machines, 2001 <http://www.csie.ntu.edu.tw/∼cjlin/libsvm>.
[7] S. Deerwester, S. Dumais, G. Furnas, T. Landauer, R. Harshman, Indexing by latent semantic analysis, J. Am. Soc. Inf. Sci. 41 (6) (1990) 391–407.
[8] L.R. Dice, Measures of the amount of ecologic association between species, Ecology 26 (3) (1945) 297–302.
[9] T. Flati, D. Vannella, T. Pasini, R. Navigli, Two is bigger (and better) than one: the Wikipedia bitaxonomy project, in: Proceedings of ACL, 2014, pp. 945–955.
[10] M. Franco-Salvador, P. Gupta, P. Rosso, Cross-language plagiarism detection using a multilingual semantic network, in: Proceedings of ECIR, 2013, pp. 710–713.
[11] M. Franco-Salvador, P. Rosso, R. Navigli, A knowledge-based representation for cross-language document retrieval and categorization, in: Proceedings of EACL, 2014, pp. 414–423.
[12] H. Fujita, J. Hakura, M. Kurematsu, Virtual doctor system (vds): framework on reasoning issues, in: Proceedings of SOMET, 2010, pp. 481–489.
[13] H. Fujita, I.J. Rudas, Mental ontology model for medical diagnosis based on some intuitionistic fuzzy functions, in: Proceedings of the 10th IEEE Jubilee International Symposium on Intelligent Systems and Informatics, 2012, pp. 55–62.
[14] E. Gabrilovich, S. Markovitch, Overcoming the brittleness bottleneck using Wikipedia: enhancing text categorization with encyclopedic knowledge, in: Proceedings of AAAI, 2006, pp. 1301–1306.
[15] E. Gabrilovich, S. Markovitch, Computing semantic relatedness using Wikipedia-based explicit semantic analysis, in: Proceedings of IJCAI, 2007, pp. 1606–1611.
[16] W. Hersh, C. Buckley, T.J. Leone, D. Hickam, Ohsumed: an interactive retrieval evaluation and new large test collection for research, in: Proceedings of SIGIR, 1994, pp. 192–201.
[17] T. Hofmann, Probabilistic latent semantic indexing, in: Proceedings of SIGIR, 1999, pp. 50–57.
[18] A. Hotho, A. Maedche, S. Staab, Ontology-based text document clustering, Data Knowl. Eng. 16 (4) (2002) 48–54.
[19] A. Hotho, S. Staab, G. Stumme, WordNet improves text document clustering, in: Proceeding of SIGIR 2003 Semantic Web Workshop, 2003, pp. 541–544.
[20] J. Hu, L. Fang, Y. Cao, H.-J. Zeng, H. Li, Q. Yang, Z. Chen, Enhancing text clustering by leveraging wikipedia semantics, in: Proceedings of SIGIR, 2008, pp. 179–186.
[21] X. Hu, X. Zhang, C. Lu, E.K. Park, X. Zhou, Exploiting Wikipedia as external knowledge for document clustering, in: Proceedings of KDD, 2009, pp. 389–396.
[22] S. Jiang, L. Bing, B. Sun, Y. Zhang, W. Lam, Ontology enhancement and concept granularity learning: keeping yourself current and adaptive, in: Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD '11, 2011, pp. 1244–1252.
[23] S. Jiang, L. Bing, Y. Zhang, Towards an enhanced and adaptable ontology by distilling and assembling online encyclopedias, in: Proceedings of the 22nd ACM International Conference on Information and Knowledge Management, CIKM '13, 2013, pp. 1703–1708.
[24] L. Jing, L. Zhou, M.K. Ng, J.Z. Huang, Ontology-based distance measure for text clustering, in: Proceedings of SDM Text Mining Workshop, 2006.
[25] J. Kohler, S. Philippi, M. Specht, A. Ruegg, Ontology based text indexing and querying for the semantic web, Knowl.-Based Syst. 19 (8) (2006) 744–754.
[26] J. Lu, C. Wang, G. Zhang, J. Ma, Collaborative management of web ontology data with flexible access control, Expert Syst. Appl. 37 (5) (2010) 3737–3746.
[27] C. Matuszek, J. Cabral, M. Witbrock, J. Deoliveira, An introduction to the syntax and content of cyc, in: Proceedings of the 2006 AAAI Spring Symposium on Formalizing and Compiling Background Knowledge and Its Applications to Knowledge Representation and Question Answering, 2006, pp. 44–49.
[28] O. Medelyan, C. Legg, Integrating Cyc and Wikipedia: folksonomy meets rigorously defined common-sense, in: Wikipedia and Artificial Intelligence: An Evolving Synergy, Papers from the 2008 AAAI Workshop, 2008.
[29] G.A. Miller, WordNet: a lexical database for english, Commun. ACM 38 (1995) 39–41.
[30] J.A. Nasir, I. Varlamis, A. Karim, G. Tsatsaronis, Semantic smoothing for text clustering, Knowl.-Based Syst. 54 (0) (2013) 216–229.
[31] R. Navigli, S.P. Ponzetto, Babelnet: the automatic construction, evaluation and application of a wide-coverage multilingual semantic network, Artif. Intell. 193 (2012) 217–250.
[32] J.C. Platt, K. Toutanova, S.W. tau Yih, Translingual document representations from discriminative projections, in: Processing of EMNLP, 2010, pp. 251–261.
[33] S.P. Ponzetto, R. Navigli, Large-scale taxonomy mapping for restructuring and integrating Wikipedia, in: Proceedings of IJCAI, 2009, pp. 2083–2088.
[34] S.P. Ponzetto, M. Strube, Deriving a large scale taxonomy from Wikipedia, in: Proceedings of AAAI, vol. 2, 2007, pp. 1440–1445.
[35] D. Reforgiato Recupero, A new unsupervised method for document clustering by using WordNet lexical and conceptual relations, Inform. Retrieval (2007) 563–579.
[36] F. Role, M. Nadif, Beyond cluster labeling: semantic interpretation of clusters contents using a graph representation, Knowl.-Based Syst. 56 (0) (2014) 141–155.
[37] M. Ruiz-casado, E. Alfonseca, P. Castells, Automatic assignment of Wikipedia encyclopedic entries to WordNet synsets, in: Proceedings of AWIC, 2005, pp. 380–386.
[38] D. Sanchez, M. Batet, D. Isern, Ontology-based information content computation, Knowl.-Based Syst. 24 (2) (2011) 297–303.
[39] S. Scott, S. Matwin, Text classification using WordNet hypernyms, in: Workshop on usage of WordNet in NLP Systems (COLING-ACL '98), 1998, pp. 45–51.
[40] A. Sole-Ribalta, D. Sanchez, M. Batet, F. Serratosa, Towards the estimation of feature-based semantic similarity using multiple ontologies, Knowl.-Based Syst. 55 (0) (2014) 101–113.
[41] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: a core of semantic knowledge, in: Proceedings of WWW, 2007, pp. 697–706.
[42] F.M. Suchanek, G. Kasneci, G. Weikum, Yago: a large ontology from Wikipedia and WordNet, Web Semantic 6 (3) (2008) 203–217.
[43] A. Syropoulos, Mathematics of multisets, in: Proceedings of the Workshop on Multiset Processing, 2001, pp. 347–358.
[44] M.A.H. Taieb, M.B. Aouicha, A.B. Hamadou, Computing semantic relatedness using Wikipedia features, Knowl.-Based Syst. 50 (0) (2013) 260–278.
[45] W. tau Yih, K. Toutanova, J. Platt, C. Meek, Learning discriminative projections for text similarity measures, in: Proceedings of CoNLL, 2011, pp. 247–256.
[46] P. Velardi, S. Faralli, R. Navigli, Ontolearn reloaded: a graph-based algorithm for taxonomy induction, Comput. Linguist. 39 (3) (2013) 665–707.
[47] P. Wang, C. Domeniconi, Building semantic kernels for text classification using Wikipedia, in: Proceedings of KDD, 2008, pp. 713–721.
[48] P. Wang, J. Hu, H.-J. Zeng, Z. Chen, Using Wikipedia knowledge to improve text classification, Knowl. Inform. Syst. 19 (3) (2009) 265–281.

[49] Y. Yang, X. Liu, A re-examination of text categorization methods, in: Proceedings of SIGIR, 1999, pp. 42–49.
[50] I. Yoo, X. Hu, I.-Y. Song, Integration of semantic-based bipartite graph representation and mutual refinement strategy for biomedical literature clustering, in: Proceedings of KDD, 2006, pp. 791–796.
[51] T. Zesch, C. Müller, I. Gurevych, Using wiktionary for computing semantic relatedness, in: Proceedings of AAAI, 2008, pp. 861–866.
[52] Y. Zhao, Z. Li, X. Wang, W.A. Halang, Decision support in e-business based on assessing similarities between ontologies, Knowl.-Based Syst. 32 (0) (2012) 47–55.