Evaluating term concept association measures for short text expansion: two case studies of classification and clustering

Alessandro Marco Boutari, Claudio Carpineto, and Raffaele Nicolussi

Fondazione Ugo Bordoni, Rome, Italy
{aboutari, carpinet, rnicolussi}@fub.it

Abstract. The proliferation of Web applications based on short texts represents both an opportunity and a challenge to text mining algorithms, because of sparse representations and lack of shared context. To address this problem, we investigate a term expansion approach based on analyzing the relationships between the term concepts present in the concept lattice associated with the document corpus. We define five term concept association measures: proximity, concept similarity, connection strength, damping-weighted proximity, proximity&strength. By means of two case studies, we evaluate the effectiveness of these measures for expansion-enhanced K-NN classification and K-Means clustering of short texts. The results suggest that the five measures are highly competitive, with the best measure showing a clear improvement over the corresponding unenhanced K-NN and K-Means algorithms, as well as over two alternative term expansion enhancements (i.e., based on Wordnet and on pseudo-relevance feedback).

1 Introduction

The increasingly important role played by short texts in the modern means of Web communication and publishing, such as Twitter messages, blogs, news feeds, and customer reviews, opens new application avenues for text mining techniques but it also raises new scientific challenges. Although text classification and clustering are well established techniques (e.g., [18], [13]), they are not successful in dealing with short and sparse data, because standard text similarity measures require substantial word co-occurrence or shared context.

There are two main approaches to address the problems raised by short texts. Either we try to define new semantic similarity functions by means of external knowledge sources, without changing the underlying document representation (e.g., [15], [2], [16]), or we expand the given texts prior to using the traditional syntactic document similarity functions (e.g., [11], [10], [1]). Our work belongs to the latter research line.

We investigate a method for text expansion that exploits the features of the concept lattice built from the document-term matrix. We model the similarity between two terms as function of the relationships between the corresponding

term concepts in the the concept lattice. In particular, we define five term concept association measures: proximity, concept similarity, connection strength, strength-weighted proximity, proximity&strength. These measures take advantage of both the local statistical co-occurrence of terms and the global structural relationships between overlapping documents, as encoded in the concept lattice. They have an intuitive meaning and are mostly easily computable. We show that the full set of term-term similarities can be generated efficiently from the concept lattice by exploring the nearest concepts of single term concepts.

We study the use of concept lattice-based term expansion with the five association functions to enhance classification and clustering of short texts. We present two experimental studies, using two classical algorithms, namely K-NN and K-Means, on two collections of short texts, namely the Reuters-21578 news data set and the ODP-239 data set (extracted from the ODP Web directory). We evaluate the effectiveness of the unenhanced algorithms and of the same algorithms enhanced with the five variants of concept lattice-based term expansion. We also include, as a reference of comparison, two additional enhanced versions of the basic algorithms using two existing expansion methods based, respectively, on WordNet and pseudo-relevance feedback. We show that classification and clustering with concept lattice-based expansion may be much more accurate than competing methods across a range of evaluation measures, especially using some term concept association functions.

The remainder of the paper has the following organization. We first provide some introductory remarks on the use of term-term associations for semantic document similarity. Then we describe the five term concept association measures and present an efficient algorithm for finding all pairwise term similarities. In the next section we describe our implementation of the expansion methods based on WordNet and pseudo-relevance feedback. The following two sections are dedicated to the experiments performed with the K-NN classifiers and K-Means clusterers on the Reuters and ODP data sets, respectively. We end the paper by discussing related work and we finally provide our conclusions and some directions for future work.

2 Document expansion for semantic similarity

In text mining, a set of documents is usually represented as an n-by-m document-term matrix $\mathcal{D}$, where n is the number of documents and m is the number of unique terms in the document collection. Each row of the matrix $\mathcal{D}$ corresponds to a document d_i and the value of element $\mathcal{D}(i, j)$ denotes the importance (weight) of term j in document i (e.g., computed by the term frequency - inverse document frequency scheme).

Classification and clustering algorithms typically rely on a similarity function between pairs of documents. The most common approach is to apply a simple (syntactic) similarity measure to the document term vectors (called *bag of words* representations); e.g., using the inner product:

$$Sim_{DOC}(d_1, d_2) = \langle d_1, d_2 \rangle = d_1 \; d_2^T \tag{1}$$

For short or very short texts, rather than using Equation 1 with the original sparse input terms, it may be more convenient to consider a linear mapping of the document vectors $\phi(d) = d\,\mathcal{S}$. The matrix $\mathcal{S}$ typically encodes pairwise term similarities, thus implying that the mapping $\phi(d) = d\,\mathcal{S}$ allows us to represent each document not only by its original terms but also by the terms that are related to each of them. In this case, the similarity function between documents becomes:

$$Sim_{DOC}(d_1, d_2) = d_1 \; \mathcal{S} \; \mathcal{S}^T \; d_2^T \tag{2}$$

By varying the matrix $\mathcal{S}$ one can obtain different transformations of the document feature space. Term-term associations can be computed using various methods discussed in [5], including linguistic analysis, semantic term relationships, and statistical term co-occurrence. In the next section we describe an approach based on analyzing the relationships between formal term concepts.

3 Attribute concept association measures

Let $\mathcal{C}(G, M, I; \leq)$ be the *concept lattice* of the context (G, M, I).[1] A particular type of concepts relevant to us are *attribute concepts*. The attribute concept of an attribute $m \in M$ is the concept (m', m''), where m' is the attribute extent $\{g \in G \,|\, gIm\}$. The attribute concept of m is thus the smallest concept with m in its intent.

The order relation $\leq$ induces the notion of nearest neighborhood. Let (X_1, Y_1) and (X_2, Y_2) be two concepts in $\mathcal{C}(G, M, I; \leq)$. We say that (X_1, Y_1) is a *nearest neighbor* of (X_2, Y_2) if and only if $(X_1, Y_1) \leq (X_2, Y_2)$ or $(X_2, Y_2) \leq (X_1, Y_1)$, and there does not exist $(X_3, Y_3) \in \mathcal{C}(G, M, I; \leq)$ such that $(X_1, Y_1) \leq (X_3, Y_3) \leq (X_2, Y_2)$ or $(X_2, Y_2) \leq (X_3, Y_3) \leq (X_1, Y_1)$.

We now define five association measures between a pair of attribute concepts. The measures take into account the topological structure of the lattice, the concept descriptions, or both. We assume that the association between two attribute concepts is stronger when they are more similar, when they are closer, and when they are connected with more similar concepts.

Concept similarity. This is a very straightforward measure because it is based only on the description of attribute concepts, regardless of how they are connected. The similarity ($CSim$) between two attribute concepts (m_1', m_1''), (m_2', m_2''), is the average of the similarities of their extents and intents, as measured by the Dice coefficient:

[1] We assume that the reader is familiar with the basic notions and terminology of formal concept analysis.

$$CSim = \frac{1}{2}\left(\frac{2|m'_1 \cap m'_2|}{|m'_1| + |m'_2|} + \frac{2|m''_1 \cap m''_2|}{|m''_1| + |m''_2|}\right) = \frac{|m'_1 \cap m'_2|}{|m'_1| + |m'_2|} + \frac{|m''_1 \cap m''_2|}{|m''_1| + |m''_2|} \quad (3)$$

We consider both the extents and intents, although these features are not independent, to better account for the relative sizes of the set of objects and attributes, and we used the Dice coefficient (rather than e.g., the Jaccard coefficient) because it works well even with a small number of shared elements (as a portion of all non-zero elements). The $CSim$ value is not equal to zero if and only if the two attributes co-occur in at least one object, while the contribution of the intent similarity is greater than zero if and only if one attribute is perfectly associated with the other (i.e., when their mutual information is maximum).

Proximity. The association between two attribute concepts can be determined using the length of their shortest connecting path (topological distance) in the concept lattice. The closer the two attributes are to each other, the greater their semantic relation due to the properties of near concepts [3]. Thus, by collecting the attribute concepts at increasing distances from a given attribute, we achieve a minimal transitive closure of the initial document-term description. We define the proximity ($Prox$) of two attribute concepts (m'_1, m''_1), (m'_2, m''_2) as an inverse function of the normalized shortest distance (SD) between the two attribute concepts, according to the nearest neighbor relation:

$$Prox \;=\; 1 \;-\; \frac{SD - min(SD)}{max(SD) - min(SD)} \;=\; 1 \;-\; \frac{SD}{max(SD)} \quad (4)$$

because $min(SD) = 0$ (i.e., when the two attributes coincide).

Connection strength. Using the length of shortest paths alone is not enough because some paths are weaker than others. For instance, if a concept happens to cover many objects and a nearest neighbor concept covers few objects, the association between the attributes in the first concept and the attributes in the second concept is weak. This aspect can be taken care of by looking at the connection strength (Str) of two attribute concepts (m'_1, m''_1), (m'_2, m''_2), defined as the average of the concept similarities ($CSim$) of the pairs of consecutive concepts along the shortest connecting path between (m'_1, m''_1), (m'_2, m''_2). When (m'_1, m''_1), (m'_2, m''_2), are nearest neighbor concepts, the connection strength is equal to the concept similarity $CSim$.

Proximity&strength. Proximity and connection strength can be combined in a single measure in various ways. We define the proximity&strength ($Prox\&Str$) of two attribute concepts as a linear combination of the proximity $Prox$ and the connection strength Str:

$$Prox\&Str \;=\; \alpha\, Prox \;+\; (1 - \alpha)\, Str \quad (5)$$

The parameter α allows us to control the relative importance of extents and intents (the default value is 0.5). Note that the *Str* value in Equation 5 is computed after finding the shortest path connecting the two attribute concepts. A tighter combination of *Prox* and *Str*, formulated as a shortest weighted path problem, leads to the following measure.

Damping-weighted proximity. Let $Damp = 1 - Str$ be the *connection damping* between two nearest neighbor concepts. The damping-weighted proximity (*DampW–Prox*) of two attribute concepts is an inverse function of the normalized shortest weighted distance (*SWD*) of the two attribute concepts, according to the nearest neighbor relation weighted with the connection damping:

$$DampW\text{–}Prox \;=\; 1 \;-\; \frac{SWD - min(SWD)}{max(SWD) - min(SWD)} \tag{6}$$

As an illustration, consider the simple context for vertebrate animals shown in Figure 1, first introduced in [7], together with its corresponding concept lattice augmented with the connection strength values between nearest concepts.

		a	b	c	d	e	f	g	h
1	Bat		x			x		x	
2	Eagle		x	x		x			
3	Monkey				x			x	
4	Parrot fish	x		x			x		
5	Penguin			x		x	x		
6	Shark	x					x		
7	Lantern fish	x					x		x

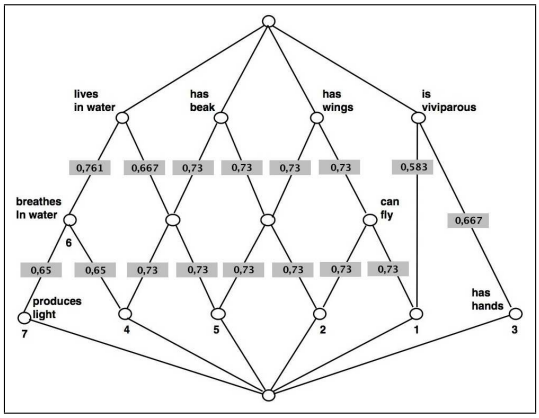

Fig. 1. A context for vertebrate animals (left) and its associated concept lattice (right), with edges labeled by their connection strength. The meaning of the attribute symbols is the following: (a) = breathes in water, (b) = can fly, (c) = has beak, (d) = has hands, (e) = has wings, (f) = lives in water, (g) = is viviparous, (h) = produces light.

In Figure 2 we report the shortest distance between any pair of attributes (left), derived from the concept lattice in Figure 1 after the removal of its top and bottom element, and their degree of association, computed from the shortest paths and from the connection strength values using Equation 5 ($\alpha = 0.5$). For instance, for the attribute pair 'has wings' (e), 'is viviparous' (g), the shortest path is {(1 2 5), (has wings)} $->$ {(1 2), (has wings, can fly)} $->$ {(1), (has wings, can fly, is viviparous)} $->$ {(1 3), (is viviparous)}, $Prox = 1 - 3/10 = 0.7$, $Str = (0.73 + 0.73 + 0.583)/3 = 0.681$, $Prox\&Str = (0.7 + 0.681)/2 = 0.69$.

	a	b	c	d	e	f	g	h
a	0	6	3	9	5	1	8	1
b	6	0	3	3	1	5	2	7
c	3	3	0	6	2	2	5	4
d	9	3	6	0	4	8	1	10
e	5	1	2	4	0	4	3	6
f	1	5	2	8	4	0	7	2
g	8	2	5	1	3	2	0	9
h	1	7	4	10	6	7	9	0

	a	b	c	d	e	f	g	h
a	1	0.55	0.71	0.39	0.61	0.83	0.45	0.77
b	0.55	1	0.71	0.68	0.81	0.60	0.72	0.50
c	0.71	0.71	1	0.54	0.76	0.74	0.60	0.65
d	0.39	0.68	0.54	1	0.63	0.44	0.78	0.34
e	0.61	0.81	0.76	0.63	1	0.65	0.69	0.55
f	0.83	0.60	0.75	0.44	0.65	1	0.50	0.75
g	0.45	0.72	0.60	0.78	0.69	0.50	1	0.35
h	0.77	0.50	0.65	0.34	0.55	0.75	0.35	1

Fig. 2. Pairwise shortest distances (left) and Prox&Str association values (right) of the attributes in the context in Figure 1, derived from the corresponding concept lattice.

The associations shown in Figure 2 are meaningful. Each attribute is more strongly associated with the attributes which co-occur with it, but it is also transitively related to the other attributes in the data set. Consider for example 'breathes in water' (a). The degree of association between 'breathes in water' and the other attributes is shown in the first row of the left matrix. The three most associated attributes are 'lives in water' (f), 'produces light' (h), and 'has beak' (c), all of which co-occur with (a). The association ranking of the other (non co-occurring) attributes is the following: 'has wings' (e), 'can fly' (b), 'is viviparous' (g), 'has hands' (d).

4 Practical construction of the term-term association matrix

For text mining applications, objects are documents and attributes are terms. There are three main computational steps involved in the construction of the term similarity matrix $\mathcal{S}$: text pre-processing, construction of the concept lattice from the document-term matrix built in the earlier step, and determination of pairwise term similarities using the term concept association measures defined above.

Text pre-processing consists of text segmentation, punctuation removal, conversion of upper to lower case, and stop-wording. We also remove all the words that appear only in one document because they have a negligible effect on retrieval performance. We do not perform any stemming and we use strict single-word indexing. To build the document lattice, we use the NextNeighbors algorithm, described in [7] on page 35. The only difference is that each edge is labeled using Equation 3 when it is added to the structure. Its computational time complexity is $O(|\mathcal{C}||G||M|^2)$ or $O(|\mathcal{C}||G|^2|M|)$, whichever is smaller, and the number of concepts $|\mathcal{C}|$ is usually linear in the number of objects for sparse contexts. The most critical operation is the subsequent determination of pairwise similarities. Unless the concept lattice is of very limited size, this step cannot be carried out by invoking a shortest path finding algorithm for every pair of term concepts (e.g., Dijkstra's algorithm), because in this case the involved time

complexity would be $O(|\mathcal{C}|^2(|E| + |\mathcal{C}|log|\mathcal{C}|) = O(|\mathcal{C}|^2|E| + |\mathcal{C}|^3log|\mathcal{C}|)$, where $|E|$ is the number of edges in the concept lattice.

Our algorithm for finding the pairwise similarities efficiently is the following. We map each term onto the concept lattice to find the corresponding term concept. This operation takes constant time using an appropriate data structure; e.g., a trie or a hash table. Then, for each term concept, an exhaustive breadth-first search through the lattice is performed, without generating the concepts that have already been encountered. Term concepts are collected along the way as soon as they are encountered. This requires at most one pass over the concept lattice, and thus the computational time complexity of finding all pairwise term similarities reduces to $O(|M||\mathcal{C}|)$. In practice, it is not necessary to explore the whole lattice. We halt the search at a fixed depth value, because this is much more efficient and it may also improve performance due to noise reduction. From preliminary tests on the Reuters data set, we found that the performance reaches a peak for a small depth value (i.e., usually 2 or 3), after which it declines. Using this cut off value, only a small fraction (less than 5%) of all theoretically possible pairs obtained a nonzero value in the term-term similarity matrix. In the experiments reported in this paper, we set the cut off value to 2.

5 Two other standard approaches to document expansion

We have implemented, as a reference of comparison, two other text expansion methods relying on standard term-term association techniques. One is based on WordNet synsets. To look up synonyms defined by WordNet, we used a package provided by Lucene[2], and then we selected only those synsets terms related to more than one original term for improving disambiguation. The resulting binary expanded representation was used to compute pairwise document similarities with Equation 1. The second expansion method, based on pseudo relevance feedback, consists of selecting the terms which mostly contribute to the Kullback-Leibler divergence (KLD) between the top ranked documents retrieved in response to the original text from a corpus and the corpus itself. To find KLD-weighted expansion terms, we used the query expansion facility offered by the Terrier platform [17].[3] The KLD-weighted expansion terms were generated by Terrier while scoring the text to be expanded against the TREC WT10g collection, that was previously indexed using Terrier itself.

6 Experiments with expansion-enhanced K-NN

Text categorization is one of the most successful data mining technique. Among many existing classifiers, the K-NN algorithm usually delivers top performance, unless fairly little training data is available [13]. The (nonlinear) K-NN classifier determines the category of an unknown document based on the categories of the

[2] http://lucene.apache.org/
[3] http://terrier.org/

K training documents that are nearest to it in the document space (usually by means of a simple majority vote). For our purposes, the most important thing is that K-NN explicitly computes pair-wise document similarities as a central step of its algorithm. We used *cosine similarity* of the binary document vectors as a similarity measure. In expansion-enhanced K-NN, the similarity between documents was computed by Equation 2 rather than by Equation 1.

We now describe the experimental setting. We used the ten most numerous classes of the well known Reuters-21578 news dataset, totaling about 8,000 documents. As we were interested in short or very short texts, we considered only the news headlines (i.e., the title field of each news item) as input documents and applied the pre-processing steps listed in Section 4. We randomly split the data set into two subsets, for training and test, then we built the concept lattice associated with the training subset, which contained 14,760 concepts and was rather wide and flat.

We ran seven versions of the K-NN classifier, one unenhanced (denoted as KNN), four enhanced with concept lattice-based expansion (one for each term concept association measure except for DampW-Prox due to computational reasons), and two enhanced respectively with WordNet (KNN-WN) and pseudo-relevance feedback (KNN-KLD). To evaluate the performance, we used the recall (R), precision (P), and F-measure, i.e. $F = 2PR/(P + R)$. In Table 1 we report the results obtained by each method for recall, precision, and F-measure, averaged over the set of classes (the best values are in bold). NA stands for not available, due to computational issues.

	Unexp	CL (Prox)	CL (CSim)	CL (Str)	CL (DampW-Prox)	CL (Prox&Str)	WN	KLD
Recall	0.77	0.80	0.79	0.81	NA	**0,83**	0.76	0.79
Precision	**0.92**	0.90	0.89	0.90	NA	0.90	0.88	0.91
F-measure	0.84	0.84	0.84	0.84	NA	**0.86**	0.80	0.83

Table 1. Classification performance of K-NN classifiers on the Reuters-21578 collection, (macro-) averaged over the ten most numerous classes.

In general, all five concept lattice-based versions did very well. The version with Prox&Str achieved the overall best results. Compared to the baseline, it was worse on precision, markedly superior on recall, and better on the combined F-measure. It was better than the other lattice-based association measures for all data points. and much better than KNN-WN and KNN-KLD (except for precision, where its result is nearly equal to that of KLD). On the other hand, KNN-WN and KNN-KLD did not compare favorably to the baseline: KNN-KLD improved over the baseline only in one case, while KNN-WN achieved the worst performance for all evaluation measures. The large increase in recall due to the use of Prox&Str is especially noteworthy because in many domains (e.g., legal, medical, patent decisions) an incorrect assignment could be easily discarded by an expert, whereas missing a relevant category could have serious consequences.

7 Experiments with expansion-enhanced K-Means

Clustering is another well known and long standing data mining technique. It is being increasingly applied to various types of short texts present on the Internet, including web pages, news, blogs, news feed, and Twitter data (see [4] for a recent survey focused on clustering search results). The K-Means algorithm is probably the most famous clustering algorithm and is frequently used for clustering text data. It uses an iterative algorithm that minimizes the sum of distances from each object to its cluster centroid, over all clusters. This algorithm moves objects between clusters until the sum cannot be decreased further. As distance we used the Euclidean distance between the vector representations of documents and centroids. In expansion-enhanced K-Means, the documents were preliminarily expanded, i.e., we computed a new document-term matrix $\mathcal{D}_{exp} = \mathcal{D}\mathcal{S}$ for each term concept association measure, where $\mathcal{D}$ is the original document-term matrix, and then applied K-Means to $\mathcal{D}_{exp}$.

We used the ODP-239 test collection,[4] first introduced in [6], which has 239 topics, each with about 10 subtopics and 100 documents. ODP-239 thus consists of many small collections, each with a comparatively large set of classes, as opposed to having one large collection of documents with a small number of classes. Each document is represented by a title and a short snippet. The topics, subtopics, and their associated documents were selected from the top levels of the Open Directory Project (http://www.dmoz.org), in such a way that the distribution of documents across subtopics reflects the relative importance of subtopics. As the data sets on which clustering was to be performed were very small (about 100 documents each), for the experiments we used the topic snippets rather than the titles, and considered only the ten topics with the largest number of unique terms after pre-processing.

We ran eight versions of the K-Means algorithm on the test collection (rather than seven). Given the very limited size of each data set and concept lattice, we were able to compute the DampW-Prox term-term association matrix using the Dijkstra algorithm for finding the shortest weighted distance for any pair of terms. To evaluate performance, we assessed how successful the K-Means clusterers were at recovering the known subtopics (classes) of each ODP-239 topic. We used the *purity* and *normalized mutual information* (NMI) measures. To compute purity, each document is assigned to the class which is most frequent in the cluster, and then the number of correctly assigned documents is counted and divided by the number of documents. By contrast, NMI is an information theoretic measure that allows us to trade off the quality of the clustering against the number of clusters (for its precise definition see e.g., [13]). The results are shown in Table 2.

These findings confirm, to a larger extent, the main results of the classification experiment, namely the improvement of all the methods enhanced with concept lattice-based expansion over the baseline, as well as their superiority over the other expansion methods. A topic-by-topic analysis showed that they achieved

[4] http://credo.fub.it/odp239

	Unexp	CL (Prox)	CL (CSim)	CL (Str)	CL (DampW-Prox)	CL (Prox&Str)	WN	KLD
Purity	0.49	0.54	**0.55**	0.53	0.51	0,54	0.49	0.51
NMI	0.25	0.34	**0.35**	0.33	0.30	**0.35**	0.26	0.28

Table 2. Clustering performance of K-Means clusterers on the ODP-239 collection, averaged over the ten topics with the highest number of terms.

the best purity and NMI results for all classes, with a uniformly distributed gain across topics. Among the five concept lattice-based versions, one striking result is the good performance of $CSim$, given its simplicity. Note that using the $CSim$ measure, only the pairs of terms that co-occur in at least one document will receive a nonzero similarity value. In this respect, the transitive closure of the initial document description is restricted to few terms and hidden similarities cannot be discovered. On the other hand, as in the ODP-239 the initial texts are considerably longer than the Reuters title and there is a much smaller number of documents, this simple criterion may be more effective than considering explicitly the expansion terms that are implied by transitivity.

8 Related Work

There has been a certain amount of work on using text expansion for improving classification and clustering. The expansion features can be extracted from a knowledge source, or they can be generated by analyzing a corpus. The former techniques include compiling WordNet concepts into the document representation [11], and using related titles of Wikipedia articles [10]. The latter techniques make use, among others, of terms of the language model associated with Web search results [14], or hidden topics extracted from a corpus [1]. Unlike most existing techniques, our term-term similarity functions are able to exploit both the statistical co-occurrence of terms in the same documents and the structural relationships between such documents. In a sense, they perform a (weighted) minimal transitive closure of the initial document descriptions.

Another relevant area is the application of concept lattices to information retrieval and information science, discussed in e.g. [7], [19]. Most related to this paper is [3]. In this earlier work we applied a similar approach to SVM text classification, but this research was limited by the use of a very simple term concept association measure and by the inefficiency of the algorithm for computing the term-term similarity matrix. Furthermore, the evaluation test was performed on a very small data set under specific assumptions. By contrast, in this paper we have shown the potential of this approach for both classification and clustering under more standard and difficult experimental conditions, including a comparative evaluation with existing techniques.

It is also relevant to this paper the work done on concept similarity. In an attempt to go beyond simple edge counting, some recent approaches tried to combine the structural relationships of concepts with their specific descriptions;

e.g., using sibling concepts [8], the least upper bound of two concepts [9], join-irreducible and meet-irreducible elements of the lattice [20], overlapping concept boundaries [12]. However, these approaches may be computationally demanding because they usually require an exploration of neighbor concepts for any pair of term concepts. Furthermore, their effectiveness has not been demonstrated experimentally.

9 Conclusions

In this paper we studied the use of five term concept association measures to drive text expansion prior to performing classification and clustering of short texts. The main results of our case studies are the following.

- Through a range of evaluation measures, we showed that the K-NN classifier and the K-Means clusterer, enhanced with expanded formal term concepts, were, in general, remarkably more effective than both the unenhanced algorithms and the algorithms enhanced with two different expansion techniques.
- The term-term similarity matrix can be computed efficiently from the concept lattice associated with the document corpus.
- Among the various term concept association measures tested in the experiments, the linear combination of proximity and connection strength exhibited the best classification accuracy, with no additional computational costs.

10 Acknowledgments

We would like to thank Giambattista Amati for helping with the Terrier-based experiments and three anonymous reviewers for their comments and suggestions.

References

1. Somnath Banerjee. Improving text classification accuracy using topic modeling over an additional corpus. In *SIGIR '08: Proceedings of the 31st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 867–868, New York, NY, USA, 2008. ACM.
2. Danushka Bollegala, Yutaka Matsuo, and Mitsuru Ishizuka. Measuring semantic similarity between words using web search engines. In *WWW '07: Proceedings of the 16th international conference on World Wide Web*, pages 757–766, New York, NY, USA, 2007. ACM.
3. C. Carpineto, C. Michini, and R. Nicolussi. A concept-lattice based kernel for SVM text classification. In *Proceedings of the 7th International Conference on Formal Concept Analysis (ICFCA 2009), Darmstadt, Germany*, pages 237–250. Springer, 2009.
4. C. Carpineto, S. Osiński, G. Romano, and D. Weiss. A survey of Web clustering engines. *ACM Computing Survey*, 41(3), 2009.
5. C. Carpineto and G. Romano. A Survey of Automatic Query Expansion in Information Retrieval. To appear in ACM Computing Surveys.

6. C. Carpineto and G. Romano. Optimal Meta Search Results Clustering. To appear in Proceedings of the 33rd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval, Geneva, Switzerland, 2010.
7. Claudio Carpineto and Giovanni Romano. *Concept Data Analysis — Theory and Applications*. Wiley, 2004.
8. J. Ducrou and P. W. Eklund. SearchSleuth: The Conceptual Neighbourhood of an Web Query. In *Proceedings of the Fifth International Conference on Concept Lattices and Their Applications, CLA 2007, Montpellier, France*. CEUR Workshop Proceedings 331 CEUR-WS.org 2008, 2007.
9. A. Formica. Concept similarity in formal concept analysis: An information content approach. *Knowledge-Based Systems*, 21(1):80–87, 2008.
10. Evgeniy Gabrilovich and Shaul Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of the 20th international joint conference on Artifical intelligence, Hyderabad, India*, pages 1606–1611. Morgan Kaufmann Publishers Inc., 2007.
11. A. Hotho, S. Staab, and G. Stumme. Ontologies Improve Text Document Clustering. In *Proceedings of the Third IEEE International Conference on Data Mining, Melbourne, Florida, USA*, pages 541–544. IEEE Computer Society, 2003.
12. Dandan Li and Dik Lun Lee. A lattice-based semantic location model for indoor navigation. In *MDM '08: Proceedings of the The Ninth International Conference on Mobile Data Management*, pages 17–24, Washington, DC, USA, 2008. IEEE Computer Society.
13. C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.
14. D. Metzler, S. dumais, and C. Meek. Similarity measures for short segments of text. In *Proceedings of the 29th European Conference on IR Research, ECIR 2007, Rome, Italy*, volume 2633 of *Lecture Notes in Computer Science*, pages 16–27. Springer, 2007.
15. Rada Mihalcea and Courtney Corley. Corpus-based and knowledge-based measures of text semantic similarity. In *IProceedings of the 21st national conference on Artificial intelligence - Volume 1, Boston, Massachusetts*, pages 775–780, 2006.
16. David Milne and Ian H. Witten. Learning to link with wikipedia. In *Proceeding of the 17th ACM conference on Information and knowledge management*, pages 509–518. ACM Press, 2008.
17. Iadh Ounis, Gianni Amati, Vassilis Plachouras, Ben He, Craig Macdonald, and Douglas Johnson. Terrier Information Retrieval Platform. In *Proceedings of the 27th European Conference on IR Research, ECIR 2005*, volume 3408 of *Lecture Notes in Computer Science*, pages 517–519. Springer, 2005.
18. M. Steinbach P.-N. Tan and V. Kumar. *Introduction to Data Mining*. Addison Wesley, 2005.
19. U. Priss. Formal Concept Analysis in Information Science. *Annual Review of Information Science and Technology (ARIST)*, 40, 2006.
20. L. Wang and X. Liu. A new model of evaluating concept similarity. *Knowledge-Based Systems*, 21(4):842–846, 2008.