Clustering Documents with Active Learning using Wikipedia

Anna Huang David Milne Eibe Frank Ian H. Witten

Department of Computer Science, University of Waikato
Private Bag 3105, Hamilton, New Zealand
{lh92, dnk2, eibe, ihw}@cs.waikato.ac.nz

Abstract

Wikipedia has been applied as a background knowledge base to various text mining problems, but very few attempts have been made to utilize it for document clustering. In this paper we propose to exploit the semantic knowledge in Wikipedia for clustering, enabling the automatic grouping of documents with similar themes. Although clustering is intrinsically unsupervised, recent research has shown that incorporating supervision improves clustering performance, even when limited supervision is provided. The approach presented in this paper applies supervision using active learning. We first utilize Wikipedia to create a concept-based representation of a text document, with each concept associated to a Wikipedia article. We then exploit the semantic relatedness between Wikipedia concepts to find pair-wise instance-level constraints for supervised clustering, guiding clustering towards the direction indicated by the constraints. We test our approach on three standard text document datasets. Empirical results show that our basic document representation strategy yields comparable performance to previous attempts; and adding constraints improves clustering performance further by up to 20%.

1. Introduction

Text document clustering automatically groups documents with similar themes together while keeping documents with different topics separate. Conventionally, a document is represented using the *bag of words* (*BOW*) document model, consisting of terms that appear in the document and their associated weights. By "terms" we mean words or phrases, but in most cases they are single words. In this model, similarity between documents is usually measured by co-occurrence statistics. Hence the clustering algorithm can only relate documents that use identical terminology, while semantic relations like acronyms, synonyms, hypernyms, spelling variations and related terms are all ignored. Furthermore, the BOW model assumes that terms appear independently and word order is immaterial, which usually conflicts reality.

We therefore propose to represent documents by concepts, so that semantic relations can be captured and utilized. We use *Wikipedia* to identify the concepts appearing within a document. Wikipedia surpasses other structural knowledge bases in its coverage of concepts, rich link information and up-to-date content. We first use Wikipedia to create a semantic representation of documents by mapping phrases to their corresponding Wikipedia articles. Secondly, Wikipedia is also used to facilitate active learning during the the clustering process, by measuring the semantic relatedness between concepts.

The paper is organized as follows. In the next section we describe how to use Wikipedia to create a concept-based document representation and compute semantic relatedness between concepts. Next in Section 3 we propose our active learning algorithm that finds pair-wise constraints. Section 4 briefly reviews the underlying clustering algorithm used in our experiments. Experimental results are discussed in Sections 5 and 6. Related work is reviewed in Section 7 and Section 8 concludes the paper and discusses future work.

2. Extracting Concepts with Wikipedia

When using Wikipedia for text mining, it is common to map document terms to concepts in Wikipedia [6, 17]. Different approaches have been proposed to accomplish this. Gabrilovich and Markovitch [6] map a document to a weighted list of relevant Wikipedia articles, by comparing the textual overlap between each document and article. Banerjee et al. [1] treat the entire document as a query to Wikipedia and associate the document with the top articles in the returned result list. Wang et al. [17] create the mapping by searching for the titles of Wikipedia articles within fixed-length sub-sequences of a document. This last method is efficient but rather brittle, because matches must be exact.

We investigate an alternative method for mapping a document to Wikipedia concepts, by leveraging an informative

and compact vocabulary—the collection of anchor texts in Wikipedia. Each link in Wikipedia is associated with an *anchor text*, which can be regarded as a descriptor of its target article. Anchor texts have great semantic value: they provide alternative names, morphological variations and related phrases for the target articles. Anchors also encode polysemy, because the same anchor may link to different articles depending on the context in which it is found.

Our approach works in three steps: identifying candidate phrases in the given document, mapping them to Wikipedia articles, and selecting the most salient concepts. The outcome is a set of concepts representing the topics mentioned in the input document and each concept is associated with its number of occurrences within the document.

Candidate Identification. Given a plain text document as input, we first break it into sentences. N-grams (up to 10 words) are then gathered and matched (with case-folding) to anchor texts in Wikipedia. Not all matches are considered, because even stop-words such as "and" have articles to describe them. We use Mihalcea and Csomai's [10] keyphraseness feature to discard such unhelpful terms. For each candidate phrase, we calculate its probability of being a concept as $\frac{f_a(p)}{f_a(p)+f_t(p)}$, where p is a candidate phrase, $f_a(p)$ is the number of Wikipedia articles in which it is used as an anchor, and $f_t(p)$ is the number of articles in which it appears in any form. Phrases with low probabilities are discarded. The same feature is used to resolve overlaps. For example, the term "South Africa" matches to three anchors: "South", "Africa", and "South Africa". In such cases only the concept or non-overlapping group of concepts with the highest average keyphraseness is preserved, and "South Africa" is retained in this case.

Sense Disambiguation. As mentioned earlier, anchors may be ambiguous in that they can have multiple target articles. We use machine learning to disambiguate them. The input to the classifier is a set of possible targets for a given anchor text and the set of all unambiguous anchors from the surrounding text, which are used as context. The classifier predicts, for each sense, the probability of it being the intended sense. The one with the highest probability is selected. More details about the algorithm can be found in [12].

Attribute Selection. After the first two steps we have a list of candidate concepts for each document. Despite pruning with the keyphraseness feature this is still a long list, because phrases are matched against a huge vocabulary of anchors. Fortunately, because these terms have been disambiguated to their relevant concepts, Wikipedia's semantics can be used to prune the concepts further to improve both efficiency and accuracy. Here we want to preserve concepts that are better descriptors of the document theme, and discard outliers that are only loosely related to this central thread. To measure the relatedness between concepts we use Milne and Witten's similarity measure [11]. Given two concepts x, y and the sets of hyperlinks X and Y that are made to each of the associated Wikipedia articles, the similarity of x and y is calculated as $SIM(x,y) = 1 - \frac{max(\log|X|,\log|Y|)-\log|X\cap Y|}{\log|N|-min(\log|X|,\log|Y|)}$, where N is the total number of articles in Wikipedia.

We define two concepts x and y to be *neighbors* if the semantic relatedness between them is no less than a prespecified threshold ϵ. We denote the *neighborhood* of a concept c by $N_\epsilon(c)$. The more neighbours a concept has–the larger the size of $N_\epsilon(c)$–the more salient the concept is. This is similar to the density-based clustering algorithm DBSacn [5], where data points are connected into clusters based on the cohesiveness of the neighborhood. We eliminate concepts whose value of $N_\epsilon(c)$ falls below a certain threshold n.

Instead of finding an appropriate threshold n through trial and error, we use an approach that adapts it automatically. If no concept's $N_\epsilon(c)$ contains more than n concepts, we decrement n, until some of the concepts are preserved or n is zero. The latter case indicates that the topics mentioned within the document are diverse; therefore, all the candidate concepts will be selected.

Besides using the concepts alone to represent a document, we also combine them with the words as in [8]. Concepts can be *add*ed into the bag of words, or alternatively can *replace* their mapping terms, resulting in a hybrid set of concepts and terms that failed to associate with a concept. We use ***BOC*** and ***BOW*** to denote the concept-based and word-based document representation respectively. The two hybrid schemes will be denoted ***Combined*** and ***Replaced***.

3. Constraining Clustering using Wikipedia

Recent research found that providing clustering algorithms with a certain amount of supervision significantly improves their accuracy (eg. [16]). Pair-wise instance-level constraint is a type of supervision that has been used widely in different clustering applications [16, 2, 4, 3]. Since labeling is expensive, we propose to actively learn these pair-wise constraints instead of using random selection. We propose to use Wikipedia for identifying these informative document pairs. The selection is based on analyzing the major concept groups–representing the major threads–in the given document collection and finding documents that are more likely to have different/similar themes. Our approach combines instance level constraints in semi-supervised clustering with active learning by selective sampling.

3.1. Active Learning of Constraints

As described in the attribute selection step above, we can cluster the concepts based on the density of their neighbor-

hoods. When given a document cluster, we first select concepts that appear frequently in the cluster, then we cluster those concepts, resulting in a number of concept clusters. These clusters represent different themes mentioned in the collective document group. We rank documents according to their weights for each concept cluster and select query documents from the top of the lists. We describe our active learning algorithm in the following, and denote the input document cluster as C_D.

Clustering Concepts. First, we cluster the m most frequent concepts in C_D according to their semantic relatedness, using the DBScan algorithm [5]. We start by randomly selecting a concept c_i with $N_\epsilon(c_i) \geq n$, and create a concept cluster C_{c_i} to hold c_i. Then c_i's neighborhood is propagated until no more concepts can be added to this cluster. This process repeats until all the concepts have been considered.

Finding Candidate Documents. For each concept cluster, we retrieve a small number of documents and rank them according to their weight for the concept cluster. We compute the weight of a document d for a concept cluster C_c as $w(d, C_c) = \sum_{c_i \in C_c} w(d, c_i)$, where $w(d, c)$ is the weight of concept c in document d (eg. c's TFIDF weight). The weight $w(d, C_c)$ denotes the document's collective representativeness for the theme as represented by the cohesive concepts in C_c. If two documents are highly representative for two different topic groups, it is more likely that they have different themes and belong to different clusters in the first place.

Obtaining Pair-wise Constrains. Therefore, we select two top-ranked documents, each from a different list, as the next query to the noiseless *oracle* that determines which type of relation the given query pair exhibits. According to the oracle's response, a *must-link* constraint will be formed if the oracle determines that the two documents belong to the same category; otherwise, a *cannot-link* constraint will be constructed if the two documents come from different categories. The oracle can also return "unknown", in which case the answer is simply discarded and the pair will not be proposed again. It is possible that the same document appears within the top document list for different concept clusters and is selected to form the next query; in this case we skip to the next candidate document in the list. Moreover, documents that have been labeled before will not be used as a query again. In our experiments, we simulate the oracle by revealing the known class labels for the two documents concerned, as in [4]. The oracle can only be consulted for a limited number of times.

The active learning approach is applicable to any document representation scheme where concept-level semantic relatedness is available. This includes *BOC* and the two hybrid models, but not the *BOW* model.

4. Constrained K-MEANS Clustering

Two clustering algorithms are used in our experiments: K-MEANS and COP-KMEANS [16] for clustering without and with constraints respectively. COP-KMEANS is very similar to K-MEANS, except that when predicting the cluster assignment for an instance, it will check that no existing constraints are violated. When an instance cannot be assigned to the nearest cluster because of violating existing constraints, the next nearest cluster will be checked, until the instance is legally assigned to a cluster without violating any constraints; otherwise the instance will remain as an outlier. However, in order to compare our empirical results with previous ones, we assign the outliers as well, to the cluster that causes the smallest number of constraints to be violated, and the nearest cluster if more than one such cluster exist.

The active learning step and the COP-KMEANS step are performed repeatedly. In each iteration, if new constraints have been found after active learning, COP-KMEANS clustering starts again with the updated set of constraints. This process terminates when either of the following two criteria is satisfied: there is no change in the COP-KMEANS clustering process; or the maximum number of queries have been posed to the oracle. It is worth noting that if no new cannot-link constraints are found in an active learning iteration, the algorithm terminates, because the search for constraints is restricted to be within a cluster.

5. Experiments

We tested the proposed methods with six test sets created from three standard data sets, and a Wikipedia snapshot taken on November 20, 2007. We first collected all anchor texts in the snapshot. After case-folding, just under five million distinct anchor terms remained, linking to almost all of the two million articles in the snapshot.

5.1. Datasets

The following test collections were used. We randomly selected 100 documents from each class for all test sets, except for *Classic3*.

20Newsgroups (20NG) contains messages from 20 different newsgroups, with 1000 messages each. Three test sets were created from this data: *20NG_Diff3* with three substantially different classes, *20NG_Sim3* with three significantly similar classes, and the combined 10 major classes *20NG_Multi10*.

Reuters-21578 consists of short news articles dating back to 1987. We created the *R_Min15Max100* set following [8] and the *R_Top10* consisting of the largest 10 classes.

Classic3 is the least challenging test set. The documents consist of titles and abstracts from academic papers from three different subjects. All document are retained, resulting in about 4000 documents.

5.2. Methodology

We created the four representations: *BOW*, *BOC*, *Combined* and *Replaced*, and also a simple *bi-gram* model. The *BOW* and *bi-gram* models were compared to as baselines. Preprocessing involved selecting only alphabetical sequences and numbers, lowercasing them, and removing stop words and infrequent words/concepts that appeared just once across the data set.

Each document was represented by a vector $\overrightarrow{t_d}$, with attributes being either words or concepts and the attribute value being its *TFIDF* weight. *TFIDF* is defined as $tfidf(d,t) = tf(d,t) \times log(\frac{|D|}{df(t)})$, where $tf(d,t)$ is the number of occurrences of attribute t in document d, $df(t)$ is the number of documents in which t appears, and $|D|$ denotes the total number of documents in the data set. *Cosine similarity* was used as the similarity metric for clustering documents. Since the relatedness measure between two concepts varies from 0 to 1 and is in accordance with human evaluation [11], we set the relatedness threshold ϵ to be 0.6.

We used stratified 10-fold cross-validation and report results as the average of 5 runs. In each fold, the clustering model is built on 90% of the entire data and then tested on the remaining 10% data. Because we are interested in relative performance only, we set the desired number of clusters in both KMEANS and COP-KMEANS (k) equal to the number of classes in the data.

We used *Purity*–the degree to which a cluster contains members from a single class–to evaluate clustering performance. Given a particular cluster C_i of size n_i, the purity of C_i is defined as $P(C_i) = \frac{1}{n_i} \max_h(n_i^h)$, where n_i^h denotes the number of members in C_i belonging to the hth class. The overall purity is the weighted average over all clusters: $\sum_{i=1}^{k} \frac{n_i}{n} P(C_i)$.

6. Results and Discussion

In this section we report and discuss the results obtained. We investigate how the different representations affect the dimensionality of the feature space, and the performance of the clustering algorithm. We also separately evaluate the effectiveness of our active learning strategy.

6.1. Document Representations and Dimensionality

High dimensionality is a substantial problem for tasks involving text documents. Considering that the number of distinct concepts appearing in a document is usually much lower than the number of words, we expect a significant reduction in dimensionality by using the *BOC* model and a positive correlation between the reduction and performance lift.

Dataset	Words	Bi-grams	Concepts
20NG_Diff3	4,487	9,148	2,409
20NG_Sim3	3,819	6,805	1,808
20NG_Multi10	9,872	22,762	6,281
R_Min15Max100	7,615	35,075	6,085
R_Top10	5,029	16,397	3,972
Classic3	11,570	56,633	8,435

Table 1. Comparison of Dimensionality

As shown in Table 1, the *BOC* model is significantly more compact than the word-based ones, as expected. However, it is worth noting that although *R_Min15Max100* has more categories and significantly more documents than *20NG_Multi10*, it has a much smaller vocabulary, in both the *BOW* and *BOC* models. This also holds for the *R_Top10* set. This indicates that the vocabularies used in different categories in the two Reuters data sets overlap significantly, which makes these two sets more challenging than the others.

6.2. Document Representations and Clustering Performance

Table 2 lists the performance for different representations, without employing any constraints during clustering. The results are disappointing at first sight. The *BOW* model often outperforms the *BOC* model, except for one test set (the *20NG_Sim3*). However, combining the two models improves clustering, with a maximum increase of 14.8% in our experiments.

It is interesting that the largest improvement was achieved in the most difficult test set (the *20NG_Sim3*), and it is the only case where using the concepts alone produces substantially better clustering performance. Taking a closer look at the *BOC* and *BOW* representations of the documents in *20NG_Sim3*, we found that using concepts to retain the semantics between words makes the three very similar categories more distinguishable; such semantics was discarded by the *BOW* model which assumes that all terms are independent of each other. For example, "screen" is a common term in both the *comp.graphics* and *comp.os.ms-windows.misc* categories. In the *comp.graphics* class it often appears as a single term, referring to the screen types, whereas in *comp.os.ms-windows.misc* it appears more frequently as part of the phrase "screen resolution". However, the two obviously different semantics are treated the same in the *BOW* representation, whereas with the *BOC* representation they are differentiated.

On *20NG_Multi10* and *Classic3*, even the combined

Dataset	Baseline BOW	Baseline bi-grams	BOC	Combined	Replaced	Impr.
	avg ± std	avg ± std	avg ± std	avg ± std	avg ± std	
20NG_Diff3	0.757 ± 0.18	0.420 ± 0.07	0.711 ± 0.128	**0.793 ± 0.131**	0.767 ± 0.086	4.76%
20NG_Sim3	0.443 ± 0.13	0.370 ± 0.06	0.479 ± 0.074	**0.497 ± 0.086**	0.453 ± 0.106	**14.8%**
20NG_Multi10	**0.467 ± 0.05**	0.179 ± 0.03	0.427 ± 0.060	0.464 ± 0.063	0.410 ± 0.058	-0.64%
R_Min15Max100	0.560 ± 0.02	0.454 ± 0.02	0.553 ± 0.028	**0.576 ± 0.041**	0.532 ± 0.019	2.86%
R_Top10	0.538 ± 0.03	0.522 ± 0.05	0.539 ± 0.048	**0.564 ± 0.044**	0.539 ± 0.044	4.83%
Classic3	**0.965 ± 0.08**	0.904 ± 0.07	0.964 ± 0.077	0.940 ± 0.103	0.964 ± 0.078	-0.10%

Table 2. Results of the effect of different document representations for clustering in terms of *Purity*

model still loses to *BOW*, but with a trivial decrease when compared to the gains on other data sets. A possible cause for the performance reduction is the curse of dimensionality: these two data sets have the most dimensions in their combined document model, more than 16 and 20 thousand dimensions respectively.

Little research has been done on using features generated using Wikipedia for clustering. The only directly comparable related result is on the *R_Min15Max100* data set, where Hotho et al. [8] achieved 0.618 purity after utilizing the concept hierarchy in WordNet and adding five hypernyms to *BOW*. However, they cluster the data set into 60 clusters instead of the actual number of classes in the data. After setting the number of clusters to 60 in our experiments, we achieved a purity score of 0.623 on the *R_Min15Max100* set, which is comparable to Hotho's.

6.3. Clustering with Constraints

The active learning algorithm discussed in Section 3 is applicable to all models except for *BOW* and *bi-gram*. We experimented with each of them. Performance is compared with the situation when no constraints are used, i.e. results in Table 2.

As Table 3 demonstrates, employing constraints improves clustering performance, to different extents for different data sets. Again, the largest improvement comes from the *20NG_Sim3* test set. We can also see that constrained clustering achieves little improvement on the Reuters data set, which is not unexpected because this data is known to be hard to categorize [13]. Because of the strong resemblance in the vocabulary used for different categories, the categories are less distinguishable. In contrast, on the *20NG_Multi10* data set, which is equivalent in scale to the *R_Top10* test set, an average 5.6% increase in prediction accuracy is achieved by using constraints.

7. Related Work

In contrast to extensive research on using Wikipedia for text categorization [1, 6, 17], little work can be found on exploiting it for clustering. The most closely related work to our clustering approach includes Hu et al. [9], Hotho et al. [8] and Recupero [14]. Hu et al. also utilized Wikipedia for creating a semantically enriched document representation and they developed a semantic document similarity function. In contrast, WordNet is used as the knowledge base instead of Wikipedia by Hotho et al. [8] and Recupero [14]. However, their approaches do not explicitly consider the relatedness between concepts when creating document representations and during subsequent document clustering/categorization.

Using Wikipedia to predict semantic relatedness between concepts has recently attracted a significant amount of interest. Alternatives to the measure from Milne and Witten [11] used in our experiments include WikiRelate! [15], which utilizes the Wikipedia category structure to compute similarity between articles; *explicit semantic analysis* from Gabrilovich and Markovitch [7], where sophisticated content analysis is used; and Wang et al. [17]'s work, which models relatedness between two concepts as a linear combination of the similarity between multiple aspects.

Pair-wise instance level constraints have been reported as effective supervision that improves clustering performance in many different applications [2, 16, 4, 3]. There has been less work on active learning for clustering. Most active learning algorithms are for supervised learning, where certain objective functions can be formulated based on the existing category structure. Few active learning algorithms have been proposed for unsupervised learning where class labels are not as readily available. Basu et al. [2] proposed an active learning algorithm that searches for the two instances that are farthest from each other and poses them to the oracle as the query. Our approach has a similar motivation—to find the documents that are most likely to be different by analyzing the concepts they contain.

8. Conclusions and Future Work

In this paper we utilized Wikipedia and the semantic information therein for text document clustering: to create more informative document representations and to facilitate active learning of pair-wise relations between documents by explicitly analyzing the topic distributions within document groups. Empirical results on three standard document data sets show that the effectiveness of our approach is comparable to previous work.

Our method of exploiting semantic information for doc-

Dataset	BOC		Combined		Replaced	
	avg $\pm$ std	impr.	avg $\pm$ std	impr.	avg $\pm$ std	impr.
20NG_Diff3	0.70 ± 0.12	6.59%	0.82 ± 0.16	3.41%	0.79 ± 0.09	2.61%
20NG_Sim3	0.51 ± 0.11	11.5%	0.56 ± 0.06	5.09%	0.51 ± 0.08	12.9%
20NG_Multi10	0.46 ± 0.06	7.03%	0.48 ± 0.07	3.01%	0.44 ± 0.05	6.83%
R_Min15Max100	0.57 ± 0.03	3.32%	0.58 ± 0.04	1.04%	0.55 ± 0.02	2.63%
R_Top10	0.55 ± 0.04	1.48%	0.57 ± 0.03	0.02%	0.54 ± 0.05	0.00%
Classic3	0.96 ± 0.08	0.20%	0.94 ± 0.10	0.00%	0.96 ± 0.08	0.00%

Table 3. Comparison of *Purity* on test data between constrained and unconstrained clustering

ument clustering is only a first step. Devising new document similarity measures based on concept similarities is an interesting and fundamental problem for document clustering. In future work, we will compare our method with Hu et al's [9] and investigate the effect of using different concept-based semantic relatedness measures [15, 7, 17] in clustering documents. Moreover, the supervision employed in our approach is at the instance level. Recent research on transforming instance-level constraints to have global impact is also of interest.

9. Acknowledgments

We thank Olena Medelyan for very helpful discussions and BuildIT New Zealand for funding the trip to the conference.

References

[1] S. Banerjee, K. Ramanathan, and A. Gupta. Clustering short texts using Wikipedia. In *Proceedings of the 30th annual international ACM SIGIR Conference*, pages 787–788, 2007.

[2] S. Basu, A. Banerjee, and R. J. Mooney. Active semi-supervision for pairwise constrained clustering. In *Proceedings of the SIAM International Conference on Data Mining (SDM-2004)*, pages 333–344, 2004.

[3] S. Basu, M. Bilenko, and R. J. Mooney. A probabilistic framework for semi-supervised clustering. In *Proceedings of the 10th International Conference on Knowledge Discovery and Data Mining (KDD-2004)*, pages 59–68, 2004.

[4] D. Cohn, R. Caruana, and A. McCallum. Semi-supervised clustering with user freedback. Technical report, 2000.

[5] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu. A density-based algorithm for discovering clusters in large spatial databases with noise. In *Proceedings of 2nd International Conference on Knowledge Discovery and Data Mining*, pages 226–231, 1996.

[6] E. Gabrilovich and S. Markovitch. Overcoming the brittleness bottleneck using Wikipedia: Enhancing text categorization with encyclopedic knowledge. In *Proceedings of the 21st National Conference on Artificial Intelligence (AAAI)*, pages 1301–1306, 2006.

[7] E. Gabrilovich and S. Markovitch. Computing semantic relatedness using Wikipedia-based explicit semantic analysis. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1606–1611, 2007.

[8] A. Hotho, S. Staab, and G. Stumme. WordNet improves text document clustering. In *Proceedings of the Semantic Web Workshop at the 26th Annual International ACM SIGIR Conference*, pages 541–544, 2003.

[9] J. Hu, L. Fang, Y. Cao, H. J. Zeng, H. Li, Q. Yang, and Z. Chen. Enhancing text clustering by leveraging Wikipedia semantics. In *Proceedings of the 31st International ACM SIGIR Conference*, pages 157–164, 2008.

[10] R. Mihalcea and A. Csomai. Wikify! linking documents to encyclopedic knowledge. In *Proceedings of the 16th ACM Conference on Information and Knowledge Management*, 2007.

[11] D. Milne and I. H. Witten. An effective, low-cost measure of semantic relatedness obtained from Wikipedia links. *In the AAAI 2008 Workshop on Wikipedia and Artificial Intelligence (WIKIAI 2008)*, 2008.

[12] D. Milne and I. H. Witten. Learning to link with Wikipedia. In *To appear in CIKM 2008*, 2008.

[13] Z. Minier, Z. Bodó, and L. Csató. Wikipedia-based kernels for text categorization. In *Proceedings of the 9th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, pages 157–164, 2007.

[14] D. R. Recupero. A new unsupervised method for document clustering by using WordNet lexical and conceptual relations. *Information Retrieval*, (10):563–479, 2007.

[15] M. Strube and S. P. Ponzetto. Wikirelate! computing semantic relatedness using Wikipedia. In *Proceedings of the 21st Conference on Artificial Intelligence*, pages 1419–1424, 2006.

[16] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Proceedings of the 18th International Conference on Machine Learning*, pages 557–584, 2001.

[17] P. Wang, J. Hu, H.-J. Zeng, L. Chen, and Z. Chen. Improving text classification by using encyclopedia knowledge. In *Proceedings of the 7th IEEE International Conference on Data Mining*, pages 332–341, 2007.