# Semantic-Aware Expert Partitioning

Veselka Boeva[1], Lilyana Boneva[1], and Elena Tsiporkova[2]

[1] Computer Systems and Technologies Department,
Technical University of Sofia, 4000 Plovdiv, Bulgaria
vboeva@tu-plovdiv.bg, lil2@abv.bg

[2] ICT & Software Engineering Group, Sirris, 1030 Brussels, Belgium
elena.tsiporkova@sirris.be

**Abstract.** In this paper, we present a novel semantic-aware clustering approach for partitioning of experts represented by lists of keywords. A common set of all different keywords is initially formed by pooling all the keywords of all the expert profiles. The semantic distance between each pair of keywords is then calculated and the keywords are partitioned by using a clustering algorithm. Each expert is further represented by a vector of membership degrees of the expert to the different clusters of keywords. The Euclidean distance between each pair of vectors is finally calculated and the experts are clustered by applying a suitable partitioning algorithm.

**Keywords:** expert location, expert partitioning, knowledge management.

## 1 Introduction

Expertise retrieval is not something new in the area of information retrieval. Finding the right person in an organization with the appropriate skills and knowledge is often crucial to the success of projects being undertaken [31]. Expert finders are usually integrated into organizational information systems, such as knowledge management systems, recommender systems, and computer supported collaborative work systems, to support collaborations on complex tasks [16]. Initial approaches propose tools that rely on people to self-assess their skills against a predefined set of keywords, and often employ heuristics generated manually based on current working practice [13,36]. Later approaches try to find expertise in specific types of documents, such as e-mails [9,11] or source code [31]. Instead of focusing only on specific document types systems that index and mine published intranet documents as sources of expertise evidence are discussed in [17]. In the recent years, research on identifying experts from online data sources has been gradually gaining interest [4,19,23,37,40,43]. For instance, Tsiporkova and Tourwé propose a prototype of a software tool implementing an entity resolution method for topic-centered expert identification based on bottom-up mining of online sources [40]. The tool extracts information from online sources in order to build a repository of expert profiles to be used for technology scouting purposes.

G. Agre et al. (Eds.): AIMSA 2014, LNAI 8722, pp. 13–24, 2014.

Many scientists who work on the expertise retrieval problem distinguish two information retrieval tasks: expert finding and expert profiling, where expert finding is the task of finding experts given a topic describing the required expertise [10], and expert profiling is the task of returning a list of topics that a person is knowledgeable about [3]. For instance, in [5,10] expertise retrieval is approached as an association finding task between topics and people. In Balog's PhD thesis, the expert finding and profiling tasks are addressed by the application of probabilistic generative models, specifically statistical language models [5].

Document clustering is a widely studied problem with many applications such as document organization, browsing, summarization, classification [1,28]. Clustering analysis is a process that partitions a set of objects into groups, or clusters in such a way that objects from the same cluster are similar and objects from different clusters are dissimilar. A text document can be represented either in the form of binary data, when we use the presence or absence of a word in the document in order to create a binary vector. A more enhanced representation would include refined weighting methods based on the frequencies of the individual words in the document, e.g., TF-IDF weighting [35]. However, the sparse and high dimensional representation of the different documents necessitate the design of text-specific algorithms for document representation and processing. Many techniques have been proposed to optimize document representation for improving the accuracy of matching a document with a query in the information retrieval domain [2,35]. Most of these techniques can also be used to improve document representation for clustering. Moreover, researchers have applied topic models to cluster documents. For example, clustering performance of probabilistic latent semantic analysis (PLSA) and Latent Dirichlet Allocation (LDA) has been investigated in [28]. LDA and PLSA are used to model the corpus and each topic is treated as a cluster. Documents are clustered by examining topic proportion vector.

In this work, we are concerned with the problem of how to cluster experts into groups according to the degree of their expertise similarity. The cluster hypothesis for document retrieval states that similar documents tend to be relevant to the same request [21]. In the context of expertise retrieval this can be re-stated that similar people tend to be experts on the same topics. Traditional clustering approaches assume that data objects to be clustered are independent and of identical class, and are often modelled by a fixed-length vector of feature/attribute values. The similarities among objects are assessed based on the attribute values of involved objects. However, the calculation of expertise similarity is a complicated task, since the expert expertise profiles usually consist of domain-specific keywords that describe their area of competence without any information for the best correspondence between the different keywords of two compared profiles. Therefore Boeva *et al.* propose to measure the similarity between two expertise profiles as the strength of the relations between the semantic concepts associated with the keywords of the two compared profiles [7]. In addition, they introduce the concept of expert's expertise sphere and show how the subject in question can be compared with the expertise profile of an individual expert and her/his sphere of expertise. In this paper, the problem is approached in a different way. Namely, it proposes a semantic-aware clustering approach for partitioning of a group of experts represented by lists of keywords. Initially, a common set of all different keywords is formed by pooling the keywords of all the expert profiles. Then the semantic distance between each pair of keywords is calculated and the keywords are partitioned by applying a selected clustering algorithm. Further, each expert is represented by a vector of membership degrees of the expert to the different clusters of keywords. Finally, the Euclidean distance between each pair of vectors is calculated and the experts are clustered by using some partitioning algorithm.

The rest of the paper is organized as follows. Section 2 briefly discusses the partitioning algorithms and describes the proposed semantic-aware clustering approach for partitioning of experts. Section 3 presents the initial evaluation of the proposed approach, which is applied to perform partitioning of researchers taking part in a scientific conference, and interprets the obtained clustering results. Section 4 is devoted to conclusions and future work.

## 2 Methods

In this section, we present our clustering method by first reviewing the characteristics of three widely-used groups of partitioning algorithms and then by describing how experts represented by lists of keywords can be clustered.

### 2.1 Partitioning Algorithms

Three partitioning algorithms are commonly used for the purpose of dividing data objects into $k$ disjoint clusters [29]: k-means clustering, k-medians clustering and k-medoids clustering. All three methods start by initializing a set of $k$ cluster centers, where $k$ is preliminarily determined. Then, each object of the dataset is assigned to the cluster whose center is the nearest, and the cluster centers are recomputed. This process is repeated until the objects inside every cluster become as close to the center as possible and no further object item reassignments take place. The expectation-maximization (EM) algorithm [12] is commonly used for that purpose, *i.e.* to find the optimal partitioning into $k$ groups. The three partitioning methods in question differ in how the cluster center is defined. In *k-means* clustering, the cluster center is defined as the mean data vector averaged over all objects in the cluster. For *k-medians* clustering the median is calculated for each dimension in the data vector instead. Finally, in *k-medoids* clustering [24], which is a robust version of the k-means, the cluster center is defined as the object which has the smallest sum of distances to the other objects in the cluster, *i.e.* this is the most centrally located point in a given cluster.

## 2.2 Semantic-Aware Expert Partitioning Approach

We propose herein a semantic-aware clustering approach that is used to partition experts into groups according to degree of their expertise similarity. It consists of three distinctive steps: 1) Construction of expert profiles via the extraction and association with each expert of a set of relevant keywords representing his/hers topics of interest; 2) Topic clustering based on pairwise semantic distance between the different keyword; 3) Clustering of experts based on their degree of association with the different topic clusters.

**Construction of Expert Profiles.** An expert profile may be quite complex and can, for example, be associated with information that includes: e-mail address, affiliation, a list of publications, co-authors etc. In view of this, an expert profile can be defined as a list of keywords, extracted from the available information about the expert in question, describing her/his area of expertise. The data needed for constructing the expert profiles could be extracted from various Web sources, *e.g.*, LinkedIn, the DBLP library, Microsoft Academic Search, Google Scholar Citation etc.

There exist several open tools for extracting data from public online sources. For instance, Python LinkedIn is a tool which can be used in order to execute the data extraction from LinkedIn. This is a package which provides a pure Python interface for the LinkedIn Connection, Profile, Search, Status, Messaging and Invitation APIs [32]. The DBLP database offers an easy access to the researchers' expertise since it describes each publication entry in an XML format and thus allowing easy parsing and information gathering for constructing the expert profiles [8].

The Stanford part-of-speech tagger [39] can be used to annotate the different words in the text collected for each expert with their specific part of speech. Next to the part of speech recognition, the tagger also defines whether a noun is plural, whether a verb is conjugated, etc. Further the annotated text can be reduced to a set of keywords (tags) by removing all the words tagged as articles, prepositions, verbs, and adverbs. Practically, only the nouns and the adjectives are retained and the final keyword set can be formed according to the following simple chunking algorithm:

– *adjective-noun(s) keywords:* a sequence of an adjective followed by a noun is considered as one compound keyword *e.g.* "supervised learning";
– *multiple nouns keywords:* a sequence of adjacent nouns is considered as one compound keyword *e.g.* "mixture model";
– *single noun keywords:* each of the remaining nouns forms a keyword on its own.

**Clustering of Topics (Keywords).** Assume that $n$ different expert profiles are created in total and each expert profile $i$ $(i = 1, 2, \ldots, n)$ is represented by a list of $p_i$ keywords. Further suppose that a set of $m$ $(m << \sum_{i=1}^{n} p_i)$ different keywords is formed by gathering all the keywords of all $n$ expert profiles.

Then we can calculate the semantic distance between each pair of keywords by using, *e.g.*, the WordNet [14,30]. WordNet is a large lexical database of English. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

Initially, the WordNet networks for the four different parts of speech were not linked to one another and the noun network was the first to be richly developed. This imposes some constrains on the use of WordNet ontology. Namely, most of the researchers who use it limit themselves to the noun network. However, not all keywords representing the expert profiles are nouns. In addition, the algorithms that can measure similarity between adjectives do not yield results for nouns hence the need for combined measure.

Let $m_i$ be an arbitrary similarity measure and $v$ be an arbitrary keyword. Then $m_i(v, v)$ gives the maximum possible score of $m_i$. We define $MN_i$ as a normalized measure of $m_i$. Initially, for any two keywords $v$ and $w$ and for each used similarity measure $m_i$ $(i = 1, 2, \ldots, r)$ we compute its normalized measure $MN_i(v, w) = m_i(v, w)/m_i(v, v)$. One can easily see that if $m_i$ takes non-negative values, then $MN_i$ takes values in [0, 1]. In order to compute our own normalized measure $MN$ combined from $r$ different similarity measures $m_1, m_2, \ldots, m_r$, we first normalize independently each $m_i$ using the above method and then define: $\alpha_1, \alpha_2, \ldots, \alpha_r$, such that $\alpha_i$ denotes the weight of $i$-th measure. In addition, $\alpha_1 + \alpha_2 + \ldots + \alpha_r = 1$. Further the normalized measure $MN$ for any two keywords $v$ and $w$ is calculated as follows:

$$MN(v, w) = \alpha_1 MN_1(v, w) + \alpha_2 MN_2(v, w) + \ldots + \alpha_r MN_r(v, w).$$

It is clear that $MN$ takes values in [0, 1].

Once we have the distances (or similarity scores) calculated, the keywords can be clustered by applying the k-means (or other partitioning) algorithm which is explained in Section 2.1. Initially, the number of cluster centers is identified. As discussed in [15,38], this can be performed by running the selected clustering algorithm on the dataset in question for a range of different numbers of clusters. Subsequently, the quality of the obtained clustering solutions needs to be assessed in some way in order to identify the clustering scheme which best fits the considered datasets. For example, the internal validation measure that is presented in Section 3.2 or different one can be used as validity index to identify the best clustering scheme. Suppose that $k$ cluster centers are determined for the set of keywords.

**Clustering of Experts.** As discussed above, the $m$ keywords are grouped by the selected clustering algorithm into $k$ clusters, *i.e.* a set of clusters $C_1, C_2, \ldots, C_k$ is produced. Let us denote by $b_{ij}$ the number of keywords from the expert profile of expert $i$ that belong to cluster $C_j$. Now each expert $i$ $(i = 1, 2, \ldots, n)$ can be represented by a vector $e_i = (e_{i1}, e_{i2}, \ldots, e_{ik})$, where $e_{ij} = b_{ij}/p_i$ $(j = 1, 2, \ldots, k)$ and $p_i$ is the total number of keywords in the expert profile representation. In this way, each expert $i$ is represented by a $k$-length vector of membership degrees of the expert to $k$ different clusters of keywords. Then we can calculate, *e.g.*, the Euclidean distance between each pair of vectors and group the experts by applying the k-means or other clustering algorithm.

## 3 Initial Evaluation

### 3.1 Test Data

We need test data that is tied to our specific task, namely the expert clustering. For this task, we use the test collection from a scientific conference [20] devoted to information technology in bio- and medical informatics. For each topic, participants (53 in total) of the corresponding conference session are regarded as experts on that topic. This is an easy way of obtaining topics and relevance judgements. A total of 5 topics (sessions) are created by the conference science committee. A list of researchers for these topics are also supplied, *i.e.*, names that are listed in the conference program on the sessions (topics) information. These researchers are considered as relevant experts, thus, used as the ground truth to benchmark the results of the proposed clustering method.

The data needed for constructing the researcher expertise profiles are extracted from Microsoft Academic Search, *i.e.*, a researcher profile is defined by a list of keywords used in the profile page of the author in question to describe her/his scientific area. Note that some of the used keywords are multiple-word terms, e.g. "Molecular Biology", "Data Mining", "Software Engineering", "Information Retrieval" etc. However, not all the multiple-word terms are present in WordNet ontology. Therefore, these keywords have been divided into their constituting words. The latter does not have effect on the quality of final expert clustering, because even the constituting words have been allocated in different clusters of keywords they are both included into the corresponding expert profiles and further are taken into account by the experts' membership degrees to those clusters.

### 3.2 Cluster Validation Measures

One of the most important issues in cluster analysis is the validation of clustering results. Essentially, the cluster validation techniques are designed to find the partitioning that best fits the underlying data, and should therefore be regarded as a key tool in the interpretation of clustering results. Since none of the clustering algorithms performs uniformly best under all scenarios, it is not reliable to use a single cluster validation measure, but instead to use at least two that reflect different aspects of a partitioning. In this sense, we have used two different validation measures. We apply the *Silhouette Index* (SI) for assessing compactness and separation properties of the obtained clustering solutions [34]. SI is also used as a validity index to identify the clustering scheme which best fits the test data described in the foregoing section. Furthermore, we use the *F-measure* for evaluating the accuracy of the generated clustering solutions [25].

**Silhouette Index.** The *Silhouette index* reflects the compactness and separation of clusters [34]. Suppose $C = \{C_1, C_2, \ldots, C_k\}$ is a clustering solution of the considered data set, which contains the attribute vectors of $m$ objects. Then the SI is defined as

$$s(C) = \frac{1}{m} \sum_{i=1}^{m} (b_i - a_i) / \max\{a_i, b_i\},$$

where $a_i$ represents the average distance of object $i$ to the other objects of the cluster to which the object is assigned, and $b_i$ represents the minimum of the average distances of object $i$ to object of the other clusters. The values of Silhouette Index vary from -1 to 1.

**F-measure.** The *F-measure* is the harmonic mean of the precision and recall values for each cluster. Let us consider two clustering solutions $C = \{C_1, C_2, \ldots, C_k\}$ and $C' = \{C'_1, C'_2, \ldots, C'_l\}$ of the same data set. The first solution $C$ is a known partition of the considered data set while the second one $C'$ is a partition generated by the applied clustering algorithm. The F-measure for a cluster $C'_j$ is then given as

$$F(C'_j) = \frac{2 \left| C_i \bigcap C'_j \right|}{|C_i| + \left| C'_j \right|},$$

where $C_i$ is the cluster that contains the maximum number of objects from $C'_j$. The overall F-measure for clustering solution $C'$ is defined as the mean of cluster-wise F-measure values, *i.e.* $F(C') = \frac{1}{l} \sum_{j=1}^{l} F_j$. For a perfect clustering, when $l = k$, the maximum value of the F-measure is 1.

### 3.3 Implementation and Availability

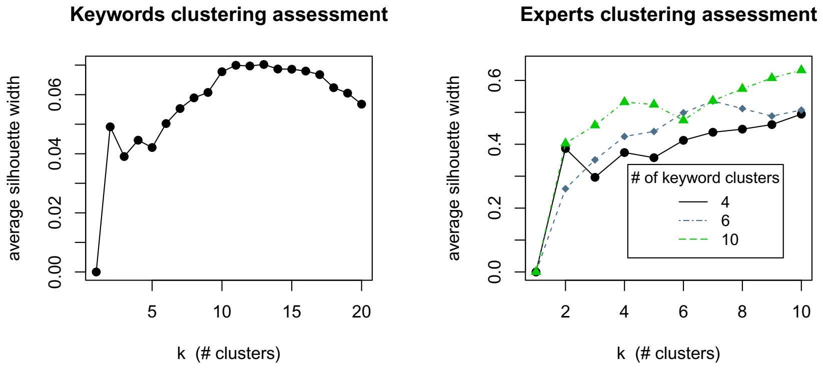
A free distributed Java library has been used to measure the word similarity - WordNet Similarity for Java (WS4J) [41]. A Java program using WS4J API has been applied to calculate a word similarity matrix for the keywords describing the expert profiles. The semantic relatedness algorithms implemented by the library have been used in our experiments [6,18,22,26,27,33,42]. As the score ranges of the algorithms vary in different intervals we have performed a normalization on all scores in order to obtain a final score in one and the same range - [0,1] (see Section 2.2). The weights are evenly distributed among the algorithms that produce a score for a given word pair. Some algorithms work for noun pair and other can be used on other parts of the speech. When an algorithm is not applicable an error score of -1 is returned and the corresponding algorithm is excluded from the calculation of the normalized measure. The algorithms that produce scores for a given word pair are used for calculating its normalized score as a mean of the produced scores. We do not give preference to any algorithm, because of the automation and the lack of any preliminary knowledge about the words being compared.

R scripts have been used to implement all the other experiments and to generate the result plots.

### 3.4 Experimental Results

Initially, a set of 44 different keywords is formed by gathering all the keywords of all 53 expert profiles. Then the semantic distance between each pair of keywords is calculated by using WordNet.

Once we have the normalized similarity scores calculated using the method presented in Section 2.2, the keywords are partitioned by applying k-means clustering algorithm. The partitioning algorithms as k-means contain the number of clusters ($k$) as a parameter and their major drawback is the lack of prior knowledge for that number to construct. Therefore, we have run k-means clustering algorithm for all values of $k$ between 2 and 20 and plot the values of the selected index (Silhouette Index) obtained by each $k$ as the function of $k$ (see Figure 1(a)). We search for the values of $k$ at which a significant local change in value of the index occurs [15]. These values are 4, 6 and 10. Thus, we apply the k-means on the set of keywords for three different values of $k$ ($k = 4, 6, 10$). In this way, three different clustering solutions for the set of keywords are produced. The partition generated for $k = 10$ can be seen in Table 1. In fact, $k = 10$ is more proper number of clusters for the set of keywords than $k = 4$ and $k = 6$. This is supported by the higher SI scores produced on the clustering solutions of the set of experts when the keywords are partitioned in 10 clusters (see Figure 1(b)).



(a) Clustering of keywords (b) Clustering of experts

**Fig. 1.** SI values generated by k-means clustering method on the set of keywords (a) and on the set of experts (b) for different number of clusters

Further, each expert is represented by a vector of membership degrees of the expert to the different clusters of keywords. Finally, the Euclidean distance between each pair of vectors is calculated and the experts are clustered by using the selected clustering algorithm. Thus k-means clustering algorithm has been run on the set of experts for all values of $k$ between 2 and 10 for each of the three clustering solutions of the keywords. For each generated clustering solution a Silhouette Index score is calculated and plotted in Figure 1(b). As can be noticed, the optimal number of clusters for the set of experts are 4 and 7.

**Table 1.** Clustering of the set of keywords for $k = 10$

| Clusters | Keywords |
|---|---|
| 1 | Algorithm, Engineering |
| 2 | Artificial Intelligence, Computer Science, Electrical Engineering, Computing |
| 3 | Mathematics, Electronics, Physiology, Neuroscience,<br>Biochemistry, Chemistry, Biology, Molecular Biology |
| 4 | Database, Information, Software, Graphics, Botany |
| 5 | Medicine, Pharmacology, Ophthalmology, Toxicology, Distribute, Pattern |
| 6 | Data Mining, Retrieval, Energy |
| 7 | Learning, Theory, Pattern |
| 8 | World Wide Web, Machine |
| 9 | Security, Recognition, Privacy, Parallel |
| 10 | Zoology |

**Table 2.** F-measure scores generated by k-means clustering method on the set of experts for $k = 4, 7$ for three different partitions of keywords ($k$ = 4, 6, 10)

| keywords clustering<br>experts clustering | $k = 4$ | $k = 6$ | $k = 10$ |
|---|---|---|---|
| $k = 4$ | 0.439 | 0.439 | 0.432 |
| $k = 7$ | 0.373 | 0.421 | 0.428 |

Next the F-measure is used to assess the accuracy of the clustering solutions generated on the set of experts for $k = 4, 7$ for three different partitions of keywords ($k = 4, 6, 10$), see Table 2. Each produced clustering solution is benchmarked to the (known) partition of the researchers explained in Section 3.1. The obtained scores are between 0.373 and 0.439. Notice that higher values are produced by the expert partitions generated for $k = 4$. However, there are no superior results with respect to the different clustering solutions of the keywords.

Finally, let us consider the clustering solution generated on the set of experts for $k = 4$ when the keywords are partitioned in 6 clusters. The experts have been grouped into four main clusters. **Cluster 1** contains 27 researchers most of who have expertise in Bioinformatics & Computational Biology, Artificial Intelligence, Data Mining and Machine Learning. Note that all the scientists with expertise in Bioinformatics & Computational Biology are allocated in this cluster. In addition, a clear sub cluster is formed by four experts all with only competence in Biochemistry. In fact, these are grouped in a separate cluster for

$k = 7$. Cluster 1 is the most heterogeneous cluster. This might be due to the fact that it contains many experts (20 such researchers) who have competence in more than two scientific areas. **Cluster 2** contains 9 experts who have competence in Engineering, Artificial Intelligence and Computer Science. This cluster is divided in two separate clusters for $k = 7$. **Cluster 3** contains 12 experts whose main expertise is in Databases and Software Engineering. This is very homogeneous cluster consisting of experts all having the keyword "Database" in her/his expertise profile. **Cluster 4** contains only 5 experts: three with expertise in Medicine, one in Ophthalmology and one in Toxicology, Pharmacology and Molecular Biology. Evidently, the considered clustering solution is a good partition of the researchers with respect to their scientific expertise.

## 4 Conclusion and Future Work

This paper elaborates on a novel semantic-aware approach for clustering of experts represented by lists of keywords. The proposed approach has initially been evaluated by applying the algorithm to partition of researchers taking part in a scientific conference. The produced clustering solutions have been validated by two different cluster validation measures. The obtained results demonstrate that the proposed approach is a robust clustering technique that is able to produce good quality clustering solutions.

For future work, the aim is to pursue further enhancement and validation of the proposed clustering approach applying alternative partitioning methods e.g. hierarchical clustering on richer expert profiles extracted from online sources e.g. LinkedIn, Google Scholar, the DBLP library, Microsoft Academic Search, etc. In addition, our future intention is to evaluate the scalability of the proposed approach. Presently, the method consists of two different clustering phases, which can be rather computationally expensive when the number of targeted experts grows. Another impact on scalability is also the degree of heterogeneity among the experts in terms of expertise. The higher this degree, the more topic clusters will be generated and therefore the vectors representing the experts will have higher dimension. It can also occur in this situation that many topic clusters are of little relevance to all of the experts. One possible way to tackle this problem is adapt the method to deal with sparse data.

## References

1. Aggarwal, C., Zhai, C.: A survey of text clustering algorithms. In: Mining Text Data, pp. 77–128 (2012)
2. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval - the concepts and technology behind search, 2nd edn. Pearson Education Ltd., Harlow (2011)
3. Balog, K., et al.: Broad expertise retrieval in sparse data environments. In: 30th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval. ACM Press, New York (2007)

4. Balog, K., de Rijke, M.: Finding similar experts. In: 30th Annual Int. ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 821–822. ACM Press, New York (2007)
5. Balog, K.: People search in the enterprise. PhD thesis, Amsterdam University (2008)
6. Banerjee, S., Pedersen, T.: An Adapted Lesk Algorithm for Word Sense Disambiguation Using WordNet. In: Gelbukh, A. (ed.) CICLing 2002. LNCS, vol. 2276, pp. 136–145. Springer, Heidelberg (2002)
7. Boeva, V., Krusheva, M., Tsiporkova, E.: Measuring Expertise Similarity in Expert Networks. In: 6th IEEE Int. Conf. on Intelligent Systems, pp. 53–57. IEEE (2012)
8. Buelens, S., Putman, M.: Identifying experts through a framework for knowledge extraction from public online sources. Master thesis, Gent University, Belgium (2011)
9. Campbell, C.S., Maglio, P.P., Cozzi, A., Dom, B.: Expertise identification using Bibliography 189 email communications. In: 12th International Conference on Information and Knowledge Management. ACM Press (2003)
10. Craswell, N., et al.: Overview of the TREC-2005 Enterprise Track. In: 14th Text Retrieval Conference (2006)
11. D'Amore, R.: Expertise community detection. In: 27th Annual Int. ACM SIGIR Conf. on Research and Development in Information Retrieval. ACM Press (2004)
12. Dempster, A.P., Laird, N.M., Rubin, D.B.: Maximum Likelihood from Incomplete Data via the EM Algorithm. J. of the Royal Statistical Society B 39(1), 1–38 (1977)
13. ECSCW99 Workshop. Beyond knowledge management: Managing expertise, `http://www.informatik.uni-bonn.de/~prosec/ECSCW-XMWS/`
14. Fellbaum, C.: WordNet: An Electronic Lexical Database. MIT Press, Cambridge (2001)
15. Halkidi, M., Batistakis, Y., Vazirgiannis, M.: On clustering validation techniques. Journal of Intelligent Information Systems 17(2) (2001)
16. Hattori, F., et al.: Socialware: Multiagent systems for supporting network communities. Communications of the ACM 42(3), 55–61 (1999)
17. Hawking, D.: Challenges in enterprise search. In: 15th Australasian Database Conference. Australian Computer Society, Inc. (2004)
18. Hirst, G., St-Onge, D.: Lexical Chains as Representations of Context for Detection and Correction of Malapropisms. In: WordNet: An Electronic Lexical Database, pp. 305–332. MIT Press (1998)
19. Hristoskova, A., Tsiporkova, E., Tourwé, T., Buelens, S., Putman, M., De Turck, F.: A Graph-based Disambiguation Approach for Construction of an Expert Repository from Public Online Sources. In: 5th IEEE Int. Conf. on Agents and Art. Int. (2013)
20. Böhm, C., Khuri, S., Lhotská, L., Pisanti, N. (eds.): ITBAM 2011. LNCS, vol. 6865. Springer, Heidelberg (2011)
21. Jardine, N., van Rijsbergen, C.J.: The use of hierarchic clustering in information retrieval. Information Storage and Retrieval 7, 217–240 (1971)
22. Jiang, J., Conrath, D.: Semantic Similarity Based on Corpus Statistics and Lexical Taxonomy. In: International Conference Research on Computational Linguistics, pp. 19–33 (1997)
23. Jung, H., Lee, M., Kang, I., Lee, S., Sung, W.: Finding topic-centric identified experts based on full text analysis. In: 2nd International ExpertFinder Workshop at the 6th International Semantic Web Conference, ISWC (2007)
24. Kaufman, L., Rousseeuw, P.J.: Finding Groups in Data: An Introduction to Cluster Analysis. Wiley, New York (1990)

25. Larsen, B., Aone, C.: Fast and Effective Text Mining Using Linear Time Document Clustering. In: KDD 1999, pp. 16–29 (1999)
26. Leacock, C., Chodorow, M.: Combining Local Context and WordNet Similarity for Word Sense Identification, pp. 265–283. MIT Press, Cambridge (1998)
27. Lin, D.: An Information-Theoretic Definition of Similarity. In: 15th International Conference on Machine Learning, ICML, pp. 296–304 (1998)
28. Lu, Y., Mei, Q., Zhai, C.: Investigating task performance of probabilistic topic models: an empirical study of plsa and lda. Information Retrieval 14(2), 178–203 (2011)
29. MacQueen, J.B.: Some methods for classification and analysis of multivariate observations. In: 5th Berkeley Symp. Math. Stat. Prob., vol. 1, pp. 281–297 (1967)
30. Miller, G.A.: WordNet: A Lexical Database for English. Communications of the ACM 38(11), 39–41 (1995)
31. Mockus, A., Herbsleb, J.D.: Expertise browser: a quantitative approach to identifying expertise. In: 24th Int. Conf. on Software Engineering. ACM Press (2002)
32. Python LinkedIn - a python wrapper around the LinkedIn API, `http://code.google.com/p/python-linkedin/`
33. Resnik, P.: Using Information Content to Evaluate Semantic Similarity in a Taxonomy. In: 14th International Joint Conference on Artificial Intelligence, vol. 1, pp. 448–453 (1995)
34. Rousseeuw, P.: Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. Journal of Computational Applied Mathematics 20, 53–65 (1987)
35. Salton, G., Buckley, C.: Term Weighting Approaches in Automatic Text Retrieval. Information Processing and Management 24(5), 513–523 (1988)
36. Seid, D., Kobsa, A.: Demoir: A hybrid architecture for expertise modeling and recommender systems (2000)
37. Stankovic, M., Jovanovic, J., Laublet, P.: Linked data metrics for flexible expert search on the Open Web. In: Antoniou, G., Grobelnik, M., Simperl, E., Parsia, B., Plexousakis, D., De Leenheer, P., Pan, J. (eds.) ESWC 2011, Part I. LNCS, vol. 6643, pp. 108–123. Springer, Heidelberg (2011)
38. Theodoridis, S., Koutroubas, K.: Pattern recognition. Academic Press (1999)
39. Toutanova, K.: Enriching the knowledge sources used in a maximum entropy partofspeech tagger. In: Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora, EMNLP/VLC-2000 (2000)
40. Tsiporkova, E., Tourwé, T.: Tool support for technology scouting using online sources. In: De Troyer, O., Bauzer Medeiros, C., Billen, R., Hallot, P., Simitsis, A., Van Mingroot, H. (eds.) ER Workshops 2011. LNCS, vol. 6999, pp. 371–376. Springer, Heidelberg (2011)
41. WordNet Similarity for Java (WS4J), `https://code.google.com/p/ws4j/`
42. Wu, Z., Palmer, M.: Verbs semantics and lexical selection. In: 32nd Annual Meeting on Association for Computational Linguistics, pp. 133–138 (1994)
43. Zhang, J., Tang, J., Li, J.: Expert finding in a social network. In: Kotagiri, R., Radha Krishna, P., Mohania, M., Nantajeewarawat, E. (eds.) DASFAA 2007. LNCS, vol. 4443, pp. 1066–1069. Springer, Heidelberg (2007)