SIGNUM: A Graph Algorithm for Terminology Extraction

Axel-Cyrille Ngonga Ngomo

University of Leipzig, Johannisgasse 26, Leipzig D-04103, Germany
ngonga@informatik.uni-leipzig.de
http://bis.uni-leipzig.de/AxelNgonga

Abstract. Terminology extraction is an essential step in several fields of natural language processing such as dictionary and ontology extraction. In this paper, we present a novel graph-based approach to terminology extraction. We use SIGNUM, a general purpose graph-based algorithm for binary clustering on directed weighted graphs generated using a metric for multi-word extraction. Our approach is totally knowledge-free and can thus be used on corpora written in any language. Furthermore it is unsupervised, making it suitable for use by non-experts. Our approach is evaluated on the TREC-9 corpus for filtering against the MESH and the UMLS vocabularies.

1 Introduction

Terminology extraction is an essential step in many fields of natural language processing, especially when processing domain-specific corpora. Current algorithms for terminology extraction are most commonly knowledge-driven, using differential analysis and statistical measures for the extraction of domain specific termini. These methods work well, when a large, well-balance reference corpus for the language to process exists. Yet such datasets exist only for a few of the more than 6,000 languages currently in use on the planet. The need is thus for knowledge-free approaches to terminology extraction. In this work, we propose the use of a graph-based clustering algorithm on graphs generated using techniques for the extraction of multi-word units (MWUs). After presenting work related to MWU extraction, we present the metric for MWU extraction used: SRE. This metric is used to generate a directed graph on which SIGNUM is utilized. We present the results achieved using several graph configurations and sizes and show that SIGNUM improves terminology extraction. In order to evaluate our approach, we used the Medical Subject Headings (MESH), with which the TREC-9 collection was tagged, and the Unified Medical Language System (UMLS) vocabularies as gold standards. Last, we discuss some further possible applications of SIGNUM and the results generated using it.

A. Gelbukh (Ed.): CICLing 2008, LNCS 4919, pp. 85–95, 2008.

2 Related Work

Depending on the amount of background knowledge needed, two categories of approaches for MWU extraction can be distinguished: knowledge-driven and knowledge-free approaches.

Knowledge-driven approaches fall into two main categories: *syntactic* and *hybrid approaches*. Syntactic approaches use linguistic patterns to extract MWU. LEXTER [2] uses an extensive list of predefined syntactic patterns to segment sentences in their components and identify potential nominal phrases and collocations. Furthermore a list of nouns that use certain prepositions as complements is used to filter the initial segmentation results. By applying a learning approach, LEXTER is then able to improve its results. A similar but semi-automatic strategy is implemented in Termight [4], which uses a combination syntactic patterns and frequency analysis for MWU extraction, the most frequent syntactic patterns being seen as more relevant. Purely syntactic approaches are always language-specific due to the patterns they necessitate. In order to improve their flexibility *hybrid approaches* were introduced. They combine syntax and statistics for MWU extraction either by first applying a numerical preprocessing to detect potential candidates for MWU and pruning the resulting list using linguistic patterns (see e.g., XTRACT [19]) or by processing the input in the reserve order, first using syntactic patterns such as NOUN NOUN and ADJ NOUN and subsequently filtering the results using numerical models (see e.g., [11]). Still they have the restrictions of syntactic approaches as they are language-specific as well.

Most *knowledge-free approaches* use probabilistic metrics (e.g., the pure occurrence frequency [9], the Dice formula [6], Pointwise Mutual Information [3], the Symmetric Conditional Probability [8]) to compute the significance of collocations. Schone [17] proposed an approach based on Latent Semantic Analysis to compute the semantic similarity of terms. The extracted similarity values are used to improve the score function during the MWU extraction. This technique shows some improvement, yet is computationally very expensive. Another approach proposed later by Dias [5] yields comparable improvement and is computationally cheaper. Dias uses pattern distributions over positional word n-grams to detect MWUs. He defines a new metric called Mutual Expectation (ME). ME models the non-substitutability and non-modifiability of domain-specific MWU. SRE is similar to ME, yet yields a further component, which takes the distribution of MWU over documents into consideration, modeling their specificity.

3 Smoothed Relative Expectation

To compute n-gram scores, we used the Smoothed Relative Expectation (SRE) [15] given by

$$SRE(w) = p(w)\frac{e^{-\frac{(d(w)-\mu)^2}{2\sigma^2}}}{\sqrt{2\pi\sigma^2}}\frac{nf(w)}{\sum_{i=1}^{n} f(c_1...c_i * c_{i+2}...c_n)}, \quad (1)$$

where

- $w = c_1...c_n$,
- $*$ is the wildcard symbol,
- $d(w)$ returns the number of documents in which w occurs,
- μ and σ^2 are the mean and the variance of the occurrence of an n-gram in a document respectively,
- $p(w)$ is the probability of occurrence of w in the whole corpus,
- $f(w)$ is the frequency of occurrence of w in the whole corpus and
- $c_1...c_i * c_{i+2}...c_n$ are all patterns such that $ham(w, c_1...c_i * c_{i+2}...c_n) = 1$.

SRE computes the expectation of a given word combination relatively to other word combinations at a Hamming distance [10] of 1 and combines it with their distribution over the documents in the corpus. It can be used to compute n-grams of all lengths. The SRE metric was compared to five other state-of-the-art metrics (DICE = dice coefficient, FR = frequency, ME = Mutual Expectation, PMI = Pointwise Mutual Information, SCP = Symmetric Conditional Probability) on the extraction of bi-grams out of the TREC-9 corpus for filtering, which consists of abstracts of publications from the medical domain. The gold standard was the MESH vocabulary. Table 1 and Figure 1 give the precision achieved when ordering bi-grams according to their score and considering the best scoring bi-grams. SRE clearly outperforms all other metrics. A t-test with a confidence level of 99% reveals that the precision achieved by SRE is significantly better than that of all other metrics.

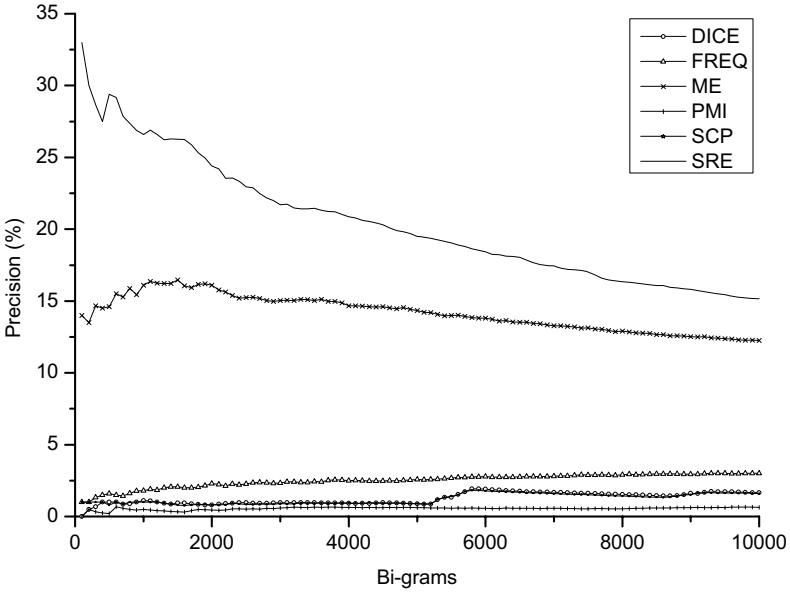


Fig. 1. Precision of six metrics

Table 1. Precision of six metrics for bi-gram extraction

Bi-grams	FR	DICE	SCP	PMI	ME	SRE
500	1.60	1.00	0.80	0.20	14.60	**29.40**
1000	1.80	1.10	1.00	0.50	16.10	**26.60**
1500	2.07	0.93	0.80	0.33	16.40	**26.26**
2000	2.30	0.80	0.80	0.45	16.10	**24.40**
2500	2.28	0.96	0.84	0.52	15.24	**22.96**
3000	2.33	0.97	0.91	0.65	15.03	**21.70**
3500	2.42	0.97	0.91	0.65	15.02	**21.45**
4000	2.50	0.93	0.95	0.63	14.68	**20.88**
4500	2.46	0.96	0.91	0.62	14.60	**20.31**
5000	2.56	0.88	0.90	0.62	14.34	**19.50**

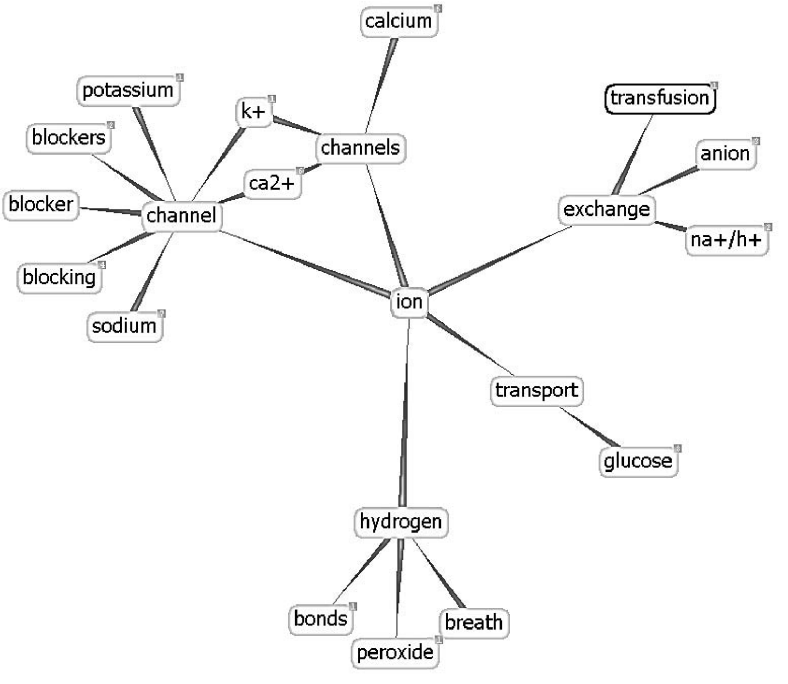


Fig. 2. Bi-gram graph for "ion"

Given any score function *score*() for word sequences, the results achieved by MWU extraction technique can be represented as weighted graphs $G = (V, E, \omega)$, with

- V being the vocabulary of the language,
- $E = \{(u, v) : score(uv) > 0\}$ and
- $\omega(uv) = \varphi(score(uv))$.

Figure 2 displays an example of such a graph. The score function was SRE, $\varphi = -1/log_{10}$.

4 SIGNUM

Collocation graphs display small-world characteristics [20], thus they have a high clustering coefficient. This property of collocation graphs makes them particularly suitable for graph clustering algorithms. Especially, the small mean path length between nodes allows the use of algorithms using exclusively local information for clustering, since the transfer of local information to all other nodes of the graph occurs considerably faster than in purely random graphs [13]. The main advantage of clustering approaches, which use local information lies at hand: they are computationally cheap and can thus deal with very large graphs, such as those usually generated during NLP. Although graphs extracted out of bi-grams (see Figure 2) are directed and thus not collocation graphs as such, they display similar topological characteristics (clustering coefficient, edge degree, etc.) and can thus be clustered using local information as well.

4.1 Basic Idea

SIGNUM was designed to achieve a binary clustering on weighted directed graphs. The basic idea behind SIGNUM originates from the spreading activation principle, which has been used in several areas such as neural networks and information retrieval [1]: the simultaneous propagation of information across edges. In the case of the basic version of SIGNUM, this information consists of the classification of the predecessors of each node in one of the two classes dubbed + and −. Each propagation step consists of simultaneously assigning the predominant class of its predecessors to each node. The processing of a graph using SIGNUM thus consists of three phases: the *initialization phase*, during which each node is assigned an initial class; the *propagation phase*, during which the classes are propagated along the edges until a termination condition is satisfied, leading to the *termination phase*. The resulting categorization is then given out.

4.2 Formal Specification

Phase I: Initialization. Directed weighted graph are triplets G = (V, E, ω) with $E \subseteq V \times V$ and $\omega : E \to \mathbb{R}$. Let

$$\sigma : V \to \{+, -\} \tag{2}$$

be a function, which assign vertices a positive or negative signum. The goal of the initialization phase is the definition of the initial values of this function (i.e., the definition of the value it initially returns for each and every node in V). Depending on the field in which SIGNUM is used, this definition might differ. In

the special case of terminology extraction, the information available about the edges is more suitable to determine the initial values of σ. Thus, let

$$\sigma_e : E \rightarrow \{+, -\} \tag{3}$$

be a function, which assigns a positive or negative signum to edges. The weight of the edge between two terms allows assumptions concerning the domain-specificity of the terms it connects. Let σ_e be fully known. Furthermore, let

$$\Sigma^+(v) = \{u : uv \in E \wedge \sigma_e(uv) = +\} \tag{4}$$

and

$$\Sigma^-(v) = \{u : uv \in E \wedge \sigma_e(uv) = -\}. \tag{5}$$

The initial values of σ are then be given by:

$$\sigma(v) = \begin{cases} + \text{ if } \sum_{u \in \Sigma^+(v)} \omega(uv) \ > \sum_{v \in \Sigma^+(v)} \omega(uv); \\ - \text{ else.} \end{cases} \tag{6}$$

This initialization prioritizes one class (in this case the $-$ class). In the case of lexicon extraction, this implies that a word is considered as initially not belonging to the lexicon when the evidence for its belonging equals the evidence for the opposite.

Phase II: Propagation. Each node is assigned the class of the majority of its predecessors. The class $-$ is assigned in case of a tie. Formally,

$$\sigma(v) = \begin{cases} + \text{ if } \sum_{\sigma(u)=+} \omega(uv) > \sum_{\sigma(u)=-} \omega(uv) \\ - \text{ else.} \end{cases} \tag{7}$$

Obviously, each edge is used exactly once during a propagation phase, making each step of SIGNUM linear in the number of edges. Furthermore, the reassignment of the classes to the node occurs simultaneously, making SIGNUM easy to implement in a parallel architecture.

Phase III: Termination. The algorithm terminates when the function σ remains constant. Obviously, several graph configurations exist, in which this propagation approach does not terminate. Fig. 3 displays an example of such a configuration. Every edge has a weight of 1. The nodes without relief are assigned to $+$, else to $-$. Yet such examples appear rarely in real life data, due to the fact that collocation graphs extracted from real world data are usually large and scale-free. For other categories of graphs, the simplest ways to ensure that the algorithm terminates is to set a threshold either for the number of iteration or for the number of changes.

4.3 Extensions

The SIGNUM concept can be extended is several ways, of which two are of particular interest for NLP. First, SIGNUM can be extended to be used on all other graph topologies (i.e., undirected and unweighted graphs): Undirected graphs can

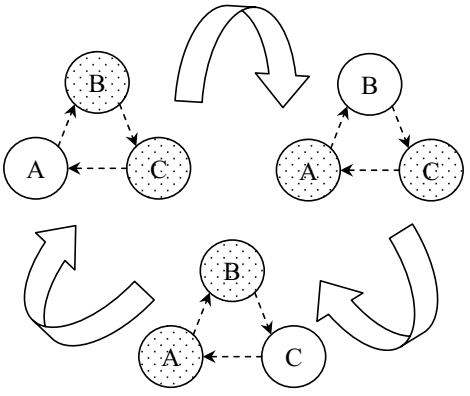


Fig. 3. Example of non termination of SIGNUM

be considered as directed graphs, with the particularity that each edge (u, v) has an equivalent edge (v, u) yielding the same weight. Furthermore unweighted graphs can be modeled as graphs with a constant edge weight function 1.

SIGNUM can also be used to cluster graphs with an unknown number of classes, for example to detect semantic classes. However, the initialization needs to be slightly modified, by assigning the same unique class label to each clique or almost-clique of the graph . An algorithm implementing such a clustering was presented in [14].

5 Using SIGNUM for Lexicon Extraction

For the practical application of lexicon extraction, we use the fact that termini from the same domain tend to appear in the same paradigmatic context, i.e., to collocate [12]. Thus, the predecessors and successors of domain-specific words can be seen as potentially belonging to the same lexicon. The initialization of the graph can thus be based on the information at hand, i.e., the degree to which words collocate. Assuming that we have an ordered list of ordered word pairs extracted from a domain specific corpus, a natural initialization of the graph would consist of selecting the upper half of the list as initially belonging to the same class (i.e., +) and the rest as belonging to the other one (i.e., −). The subsequent use of SIGNUM over the resulting graph to have the predecessors of the node would then confirm the hypothetic classes by their own classification.

5.1 Data Set

The underlying data set for the results presented below was extracted from the TREC-9 corpus [16]. This corpus is a test collection composed of abstracts of publications from the medical domain. The entries in the available test corpus included the abstract text of medical publications (marked in each entry with

Table 2. Topology of n-gram graphs

N-grams	Nodes	Edges	Components	Avg. N/C	Avg. E/C	Max N/C	Max E/C
10,000	11,106	9,969	2,606	4.26	3.83	4,854	6,282
20,000	23,733	19,939	7,415	3.20	2.69	7,136	10,688
50,000	47,905	49,685	14,579	3.29	3.41	13,895	30,204
100,000	79,658	98,893	21,454	3.71	4.61	25,315	65,811

a ".W") and further metadata such as the subject, type of publication, etc. The data extraction process consisted exclusively of the retrieval of all the text entries (i.e. those marked with ".W" in the TREC-9 corpus) and the deletion of punctuation. 233,445 abstracts (244 MB) were retrieved and utilized for the evaluation presented in this section. 355,616 word forms were extracted from the corpus with a mean frequency of 109.08. 6,096,183 different bi-grams were found, their mean frequency being 6.36. The mean occurrence of bi-grams in documents was 5.67 with a standard deviation of 137.27. Figure 2 displays an excerpt of the graph extracted from the data set. The length of the edges is inversely proportional to their weight.

5.2 Initialization

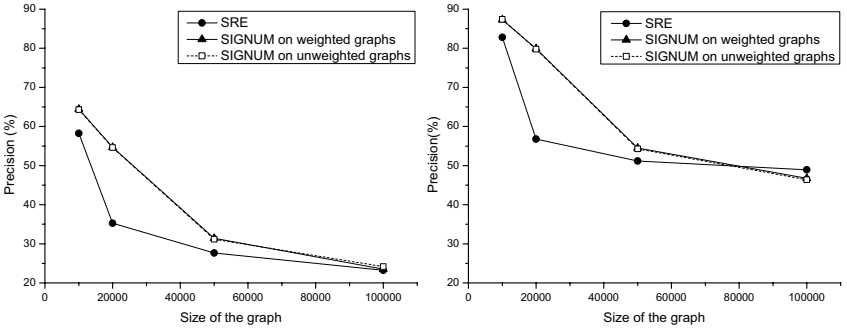
The scores computed using SRE bear small values for large corpora, ranging between 0 and 10^{-5} in the special case at hand. The weight $\omega(w_1 w_2)$ of the edge between two words w_1 and w_2 was thus set to

$$\omega(w_1 w_2) = \frac{-1}{log_{10}(SRE(w_1 w_2))}. \tag{8}$$

In order to compute σ_e, the sum of all scores was computed and halved. Subsequently, the bi-gram list was processed sequentially. Bi-gram were assigned positive signum values until the sum of their scores reached half of the total sum of scores. The residual edges were assigned negative signum values.

5.3 Results

SIGNUM was tested using the n best scoring bi-grams, with n taking values between 10,000 and 100,000 (see Table 2; N/C = nodes/component, E/C = edges/component). The resulting graphs presented a similar topology: they consisted of a large main component and a large number of small components. This topology is similar to that reported by other groups (see e.g., [7]). SIGNUM was tested on two graph configurations against the results of SRE: in the first configuration, the weights were not considered during the propagation phase (i.e., they were all set to 1). In the second configuration, the weights were considered. Two golden standards were used to measure the precision of the results achieved. The MESH vocabulary was selected because it was used to tag the TREC-9 data set. Due to the restrictiveness of the MESH vocabulary, the more complete UMLS



(a) Precision using MESH (b) Precision using UMLS

Fig. 4. Precision measured using the MESH and UMLS vocabularies

Table 3. Comparison of the precision of SRE and SIGNUM. The left column of each block displays the results of the precision using MESH, while the right one displays the same metric when using UMLS.

N-grams	SRE		Unweighted		Weighted	
10,000	58.26	82.81	64.29	**87.37**	**64.44**	87.31
20,000	35.27	56.79	54.63	79.77	**54.71**	**79.89**
50,000	27.67	51.18	31.18	54.29	**31.34**	**54.50**
100,000	23.23	**48,91**	**24.17**	46.40	23.52	48.91

vocabulary was also utilized for measuring the precision achieved by SRE and SIGNUM. The precisions achieved is displayed in Table 3. The left sub-column of each of the method columns displays the precision achieved using MESH as reference vocabulary. The left one display the same metric on the UMLS vocabulary. Both results are displayed graphically in figure4.

As shown by figure4(a) and 4(b), the weighting of the graphs does not significantly alter the performance of SIGNUM on the graph at hand. This hints toward the fact that the topology of the graph is the key influence for the performance of SIGNUM and not the weight distribution over the graph. A significantly high difference between the results of SRE and SIGNUM is observed when the graph is generated out of 20,000 bi-grams. However, the gain in precision then decreases with the size of the graph. This can be explained by the fact that larger graph include more functions words, which tend to collocate with terms from both classes and thus augment the total weight of the intra-cluster edges, leading to more errors as the class labels are transferred over the edges. This is especially clear, when the results achieved on the 100,000 bi-gram graphs are considered.

6 Conclusion and Outlook

We presented SIGNUM, a novel graph-based approach for the extraction of domain-specific terminology and showed that it improves the results achieved

using state-of-the-art techniques in the task of extracting one-word dictionaries from word collocation graphs. As the technique for MWU extraction and SIGNUM are independent, our approach can be used for the improvement of any of the metrics for MWU extraction presented above. The results achieved using SIGNUM can be used to filter MWU results and improve the quality of automatically generated multi-word dictionaries. SIGNUM furthermore bears the advantage of using solely local information available to each node, making it computationally cheap. Therefore, it is able to handle very large graphs. Due to the simultaneous reclassification of nodes, SIGNUM can be easily implemented in a parallel architecture.

A category of graphs which was not considered in this work presented are link graphs, which can be used for disambiguation (see e.g., [18,7]) and thus for further improvement of the MWU extraction. This work is currently being undertaken.

References

1. Baeza-Yates, R.A., Ribeiro-Neto, B.A.: Modern Information Retrieval. ACM Press, Addison-Wesley (1999)
2. Bourigault, D.: Lexter: A terminology extraction software for knowledge acquisition from texts. In: 9th Knowledge Acquisition for Knowledge-Based Systems Workshop, Banff, Canada (1995)
3. Church, K.W., Hanks, P.: Word association norms, mutual information, and lexicography. In: Proceedings of the 27th. Annual Meeting of the Association for Computational Linguistics, Vancouver, B.C, pp. 76–83. Association for Computational Linguistics (1989)
4. Dagan, I., Church, K.: Termight: identifying and translating technical terminology. In: Proceedings of the fourth conference on Applied natural language processing, pp. 34–40. Morgan Kaufmann, San Francisco (1994)
5. Dias, G.: Extraction Automatique dAssociations Lexicales partir de Corpora. PhD thesis, New University of Lisbon (Portugal) and LIFO University of Orléans (France), Lisbon, Portugal (2002)
6. Dice, L.R.: Measures of the amount of ecological association between species. Ecology 26, 297–302 (1945)
7. Dorow, B.: A Graph Model for Words and their Meanings. PhD thesis, University of Stuttgart, Stuttgart, Germany (2006)
8. da Silva, J.F., Lopes, G.P.: A local maxima method and a fair dispersion normalization for extracting multi-words units from corpora. In: Sixth Meeting on Mathematics of Language, Orlando, USA, pp. 369–381 (1999)
9. Giuliano, V.E.: The interpretation of word associations. In: Stevens, M.E., et al. (eds.) Proceedings of the Symposiums on Statistical Association Methods for Mechanical Documentation, Washington D.C., number 269, NBS (1964)
10. Hamming, R.: Error-detecting and error-correcting codes. Bell System Technical Journal 29(2), 147–160 (1950)
11. Justeson, J., Katz, S.: Co-occurrences of antonymous adjectives and their contexts. Computational Linguistics 17(1), 1–20 (1991)
12. Manning, C., Schütze, H.: Foundations of Statistical Natural Language Processing, 1st edn. MIT Press, Cambridge (1999)

13. Milgram, S.: The small-world problem. Psychology Today 2, 60–67 (1967)
14. Ngonga Ngomo, A.-C.: CLIque-based clustering. In: Proceedings of Knowledge Sharing and Collaborative Engineering Conference, St. Thomas, VI, USA (November 2006)
15. Ngonga Ngomo, A.-C.: Knowledge-free discovery of domain-specific multi-word units. In: Proceedings of the 2008 ACM symposium on Applied computing, ACM, New York (to appear, 2008)
16. Robertson, S.E., Hull, D.: The TREC 2001 filtering track report. In: Text REtrieval Conference (2001)
17. Schone, P.: Toward Knowledge-Free Induction of Machine-Readable Dictionaries. PhD thesis, University of Colorado at Boulder, Boulder, USA (2001)
18. Schütze, H.: Automatic word sense discrimination. Computational Linguistics 24(1), 97–123 (1998)
19. Smadja, F.A.: Retrieving collocations from text: Xtract. Computational Linguistics 19(1), 143–177 (1993)
20. Steyvers, M., Tenenbaum, J.: The large-scale structure of semantic networks: Statistical analyses and a model of semantic growth. Cognitive Science: A Multidisciplinary Journal 29(1), 41–78 (2005)