A proposal for transformation of topic-maps into similarities of topics

ABSTRACT

Newer information filtering and retrieval models like the Fuzzy Set Model or the Topic-based Vector Space Model consider term dependencies by means of numerical similarities between two terms. This leads to the question from what and how these numerical values can be deduced? This paper proposes an algorithm for the transformation of topic-maps into numerical similarities of paired topics. Further the relation of this work towards the above named information filtering and retrieval models is discussed.

Categories and Subject Descriptors

H [**3**]: 3—*Information Search and Retrieval*

Keywords

similarities, topic-maps, information filtering, information retrieval

1. MOTIVATION

The development of digital media and especially of the Internet makes publishing of documents easier than ever before. The number of documents available in business and research is growing very fast. But the fact that more information is on-hand for choosing from several options does not necessarily lead to better decisions. Restricted capacity of human's information processing forces reduction of the amount of information presented to the human. Today one of the greatest challenges in many areas of business, science and administration is the efficient filtering and retrieval of information. While information filtering (IF) selects documents from a dynamic stream of documents using some kind of (static) profile, information retrieval (IR) selects relevant documents from a static set of documents based on a specified (ad hoc) query. [3]

A lot of automated IF or IR systems have been sketched or implemented so far. But the problem of IF and IR is still not finally solved. One reason for this is the fact that machines are not able to 'understand' human language. Therefore heuristics have to be used to deliver a more or less good solution for the IF or IR problem. Those heuristics depend on an abstract and formal model of natural languages. There are two scientific communities working on natural language models. On the one hand there is the computer linguistics community which works on a formal representation for (usually one certain) natural language including syntax, semantics and pragmatics. Those scientists are working on parsers (e.g. like the parser presented in [19] and [9] for the German language) or on dialogue systems (e.g. like *Verbmobil* [18]), which are able to understand the human language in a limited manner. While these approaches are very promising the two main problems of those approaches are, that they still have a too low coverage regarding the processable words and sentence constructions and that they need too much computing power to be efficently used for huge amounts of documents, which are typical for IF and IR problems. The second scientific communitiy has a different approach. They use more efficiently computable representations of natural language documents which have been explicitly developed for the solution of IF and IR problems. These approaches consider only a small number of features of natural languages. The classical approaches in this area are for example the *Standard Boolean Model* [1], the *Vector Space Model* [15, 16] and the *Binary Independence Retrieval* [14]. Those approaches have in common, that they represent natural language documents as a list or set of terms (or words), which are assumed to be pairwise independent. This independency assumption makes the computation very efficient, but it does not reflect the real situation of natural languages. Therefore the quality of IF and IR results of systems using those language models is affected.

For the above named reason a bunch of new models have been presented during the last decades. Those models are for example the *Generalized Vector Space Model* [20], the *Latent Semantic Index* [4] and the *Language Model* [13, 17, 10, 6, 5, 21] just to name a few. Those models have in common, that the terms are not necessarily pairwise independent. The degree of interdependency beween two terms is usually measured by some kind of statistical co-occurrence of term pairs within a set of documents. Because the statistical measurement of term dependency is not unproblematic [8] other models have been presented, which do not rely on statistical methods. Such models are for example the *Fuzzy Set Model* [12] and the *Topic-based Vector Space Model* [2], the latter one can be seen as a generalization of the Gener-

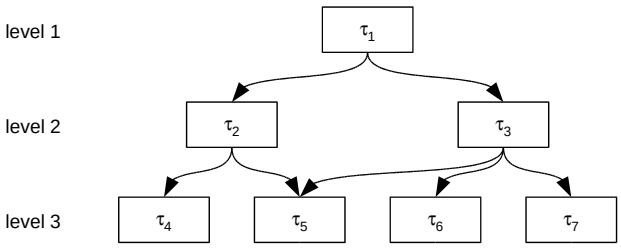


Figure 1: Sample topic-map.

alized Vector Space Model which again a generalization of the Vector Space Model. Those models have in common, that they represent the term dependencies *not* by explicit relationship arcs between term nodes in a graph similar to topic-maps, word nets or graphical ontologies. Rather they represent the term dependencies for performance reasons as a numerical value, which is assigned to a combination of two terms. This leads to the question from what and how those numerical values can be derived? One possibility is to write those values simplay down, but this leads to the problem that a manual and consistent setting of numerical values for a larger amount of terms is very difficult to handle for human beings. One solution for this problem is to use graphical representations which are easier to handle for humans. Later this representations can be transformed into numerical values, which are assigned to pairwise combinations of terms. In this paper an algorithm for this kind of transformation is proposed.

The next section presents the algorithm and the consistency criterias and features of the resulting similarity matrix of the algorithm. Then some practical issues regarding the gaining of the input data (topic-map) and the usage of the output data (similarity matrix) for IF and IR are discussed. Finally a short summary is given.

2. MAIN PART

This section presents the algorithm for the calculation of similarities between two entities (topics) of an acyclic topic-map. To rise comprehensibility the explanations are accompained by some (for didactic reasons artificial) examples. We start our explanations with some definitions followed by the algorithm and some consistency criterias, which are met by the resulting similarity matrix. Finally we will finish with some features of the algorithm.

2.1 Definitions

As a starting point we will define $\Theta = \{\tau_1, \tau_2, ..., \tau_{\#\Theta}\}$ as the set of all possible topics in the topic-map. In case of our sample topic-map (figure 1) the set has the following elements: $\Theta = \{\tau_1, \tau_2, ..., \tau_7\}$. Further the relation structure between all topics is represented by the super-topic-relation $S(\tau_i) \subseteq (\Theta \backslash \tau_i)$. This relation is defined for all topics $\tau_i \in \Theta$. The super-topic-relation defines for every topic τ_i the direct superordinated topics to which τ_i belongs. For the sample topic-map for example the following values are defined among others: $S(\tau_1) = \{\}$, $S(\tau_4) = \{\tau_2\}$, $S(\tau_5) = \{\tau_2, \tau_3\}$, etc.

Using the super-topic-relation $S(\tau_i)$ the transitive p-level super-topic-relation $S^p(\tau_i)$ can be derived. This relation represents the superordinated topics to a topic τ_i, which are positioned exactly p levels above the level of the topic τ_i. For the further description of the algorithm the transitive and not restricted super-topic-relation $S^*(\tau_i)$ is needed. This relation can be derived using $S^p(\tau_i)$ as follows:

$$S^p(\tau_i) = S(\tau_i) \qquad \text{for } p = 1$$

$$S^p(\tau_i) = \bigcup_{\tau_k \in S^{p-1}(\tau_i)} S(\tau_k) \qquad \text{for } p > 1$$

$$S^*(\tau_i) = S^1(\tau_i) \cup S^2(\tau_i) \cup S^3(\tau_i) \cup ...$$

In case of our sample topic-map (figure 1) we have the following results (among others) for the $S^*()$ relation:

$$S^*(\tau_1) = \{\}$$
$$S^*(\tau_4) = \{\tau_1, \tau_2\}$$
$$S^*(\tau_5) = \{\tau_1, \tau_2, \tau_3\}$$

Beneath the set of all topics Θ we need the set of all topic-leaves Θ_B. Topic-leaves are topics, which have no sub-topics. This means that no topic is existing, which has an element from Θ_B as its super-topic. Formally:

$$\Theta_B = \{\tau_i \in \Theta : \nexists \tau_k \in \Theta \text{ with } \tau_i \in S(\tau_k)\}$$

In our sample the set of topic-leaves contains the following entities: $\Theta_B = \{\tau_4, \tau_5, \tau_6, \tau_7\}$. Beneath the set of topic-leaves we can define the set of topic-nodes Θ_K as the set of all topics, which are not topic-leaves. Formally:

$$\Theta_K = \complement\Theta_B = \Theta \setminus \Theta_B$$

2.2 Algorithm

For each topic $\tau_i \in \Theta$ a topic vector $\vec{\tau_i} = (\tau_{i,1}, \tau_{i,2}, ..., \tau_{i,\#\Theta}) \in \mathbb{R}^{\#\Theta}$ is assigned. The values of the vector entries depend on the position of the topics within the ontology. For topic-leaves the following values are assigned to the topic-vector:

$$\forall \tau_i \in \Theta_B : \vec{\tau_i} = |(\tau^*_{i,1}, \tau^*_{i,2}, ..., \tau^*_{i,\#\Theta})|$$

with:

$$\tau^*_{i,d} = \begin{cases} 1 & \text{if } \tau_d \in S^*(\tau_i) \ \vee \ i = d \\ 0 & \text{for all other cases} \end{cases}$$

In the sample case (figure 1) we get the following values for the topic-vectors of the leaves:

$$\tau_4 = |(1,1,0,1,0,0,0)| = (\frac{1}{\sqrt{3}}, \frac{1}{\sqrt{3}}, 0, \frac{1}{\sqrt{3}}, 0, 0, 0)$$
$$\tau_5 = |(1,1,1,0,1,0,0)| = (\frac{1}{2}, \frac{1}{2}, \frac{1}{2}, 0, \frac{1}{2}, 0, 0)$$
$$\tau_6 = |(1,0,1,0,0,1,0)| = (\frac{1}{\sqrt{3}}, 0, \frac{1}{\sqrt{3}}, 0, 0, \frac{1}{\sqrt{3}}, 0)$$
$$\tau_7 = |(1,0,1,0,0,0,1)| = (\frac{1}{\sqrt{3}}, 0, \frac{1}{\sqrt{3}}, 0, 0, 0, \frac{1}{\sqrt{3}})$$

It is worth mentioning that the arrangement of the topics is not lost in the representation of topics by the topic-vector. The structural relations of topics to their super topics at the different levels is mirrored by the inner composition of the topic-vector. It is possible to visualize this by choosing

	1	2	4	5	3	6	7
1	1.000	0.933	0.734	0.924	0.933	0.741	0.741
2	0.933	1.000	0.888	0.888	0.742	0.513	0.513
4	0.734	0.888	1.000	0.577	0.483	0.333	0.333
5	0.924	0.888	0.577	1.000	0.836	0.577	0.577
3	0.933	0.742	0.483	0.836	1.000	0.871	0.871
6	0.741	0.513	0.333	0.577	0.871	1.000	0.667
7	0.741	0.513	0.333	0.577	0.871	0.667	1.000

Figure 2: Similarity matrix of the sample map.

a proper order of dimensions in the representation of topic-vectors like in the following example:

$$\tau_6 = |(\overbrace{1}^{\text{level 1}}, \overbrace{0,1}^{\text{level 2}}, \overbrace{0,0,1,0}^{\text{level 3}})|$$

For topic-nodes the topic-vector is defined as the normed sum over all direct sub-topics-vectors of the node:

$$\forall \tau_i \in \Theta_K : \vec{\tau_i} = | \sum_{\tau_s \in \Theta:\ \tau_i \in S(\tau_s)} \vec{\tau_s}|$$

This results in the following values for the node topic-vectors of the sample topic-map in figure 1:

$$\tau_1 \approx (0.669, 0.429, 0.495, 0.174, 0.255, 0.120, 0.120)$$
$$\tau_2 \approx (0.607, 0.607, 0.282, 0.325, 0.282, 0, 0)$$
$$\tau_3 \approx (0.642, 0.194, 0.642, 0, 0.194, 0.224, 0.224)$$

Finally we define the similarity $\text{sim}(\tau_a, \tau_b)$ between two topics τ_a and τ_b as the scalar-product between the topic-vectors of both topics. Because the topic-vectors are normed, the scalar-product is equivalent to the cosine of the angle $\omega_{a,b}$ between the topic vectors:

$$\begin{aligned} \text{sim}(\tau_a, \tau_b) &= \vec{\tau_a}\vec{\tau_b} \\ &= \sum_{i=1}^{\#\Theta} \tau_{a,i}\tau_{b,i} \\ &= \cos \omega_{a,b} \end{aligned} \quad (1)$$

The similarity matrix in figure 2 shows the resulting similarities for the topics from the topic-map in figure 1.

2.3 Consistency Criterias

in the categorization of similarity measures consistency criterias play an important role. It can be easily shown, that the following consistency criterias are hold by the similarity measure gained by the above presented algorithm:

1. Norming: $\text{sim}(\tau_a, \tau_b) \in [0...1]$
   Because all entries of the topic-vectors are positive, only angles between 0 and 90 degrees are possible. This leads to similarity values not less than zero.

2. Symmetry: $\text{sim}(\tau_a, \tau_b) = \text{sim}(\tau_b, \tau_a)$
   Results from the symmetry of angles.

3. Maximality: $1 = \text{sim}(\tau_a, \tau_a) \geq \text{sim}(\tau_a, \tau_b)$
   Results from the fact, that a vector has always the angle of value zero towards itself.

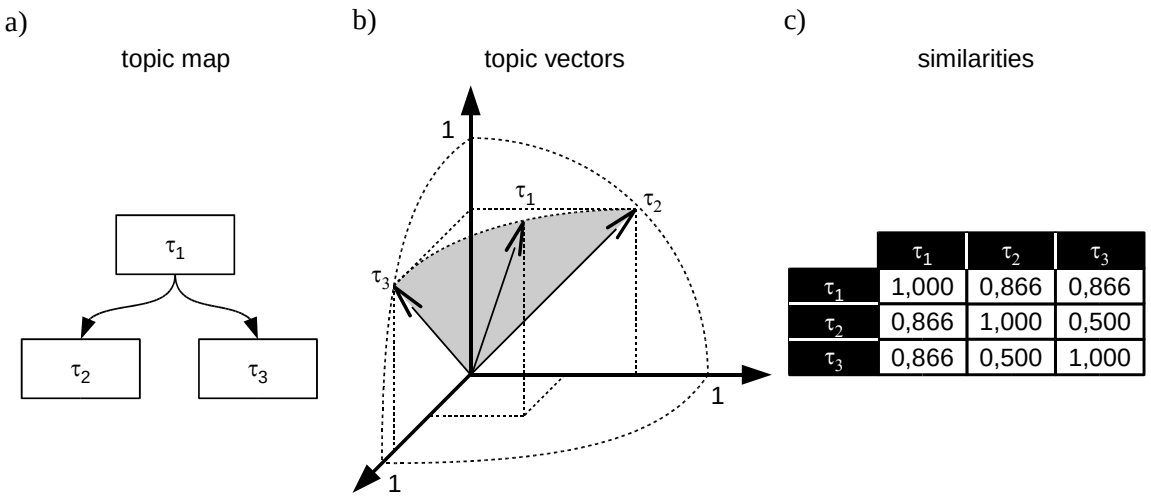
4. Weak Transitivity:
   $$|\omega_{a,b} - \omega_{a,c}| \leq \omega_{b,c} \leq \min(\omega_{a,b} + \omega_{a,c}, 90)$$
   Due to the geometrical representation of the topics in a vector space and the fact that the similarites bases on the cosinus function (e.g.: $\omega_{a,b} = \cos^{-1}(\text{sim}(\tau_a, \tau_b))$) weak transitivity can be observed.

2.4 Features

Figure 3 shows the representation of a very simple topic-map in the vector space and the resulting similarity matrix. It is apparent that the vectors of the topic-leaves are building a sub-space in the vector space. The vectors of topic-nodes are embedded in this sub-space. Hence the representation of the topic-map is reduced in fact to a $\#\Theta_B$-dimensional vector space altough for technical reasons a $\#\Theta$-dimensional vector space is used to represent the spanning vectors. Further it is notable that a topic-map can consist of several not connected submaps. In this case the topic vectors of the submap will be orthogonal to the topic vectors of the other submap. As a result the topic similarities between two topics of two different submaps have always the value null.

An other interesting feature of the presented similarity measure is, that the addition of all vectors of those topics which are a direct subtopic to a particular topic adds up to a vector, which points into the same direction as the vector of the particular topic. This feature is useful in combination with vector-based IF and IR models like for example the already mentioned Topic-based Vector Space Model [2]. Such models usually represent documents by a vector which is the result of the summation of all word vectors of the documents. To combine this similarity measure with such a model, words should be assigned to topics and consequently the document vector should be derived from the normed summation of the topic vectors. Having this scenario a document vector containing words which are assigned to the topics 'Linux', 'Windows' and 'Mac OS X' would have a similarity of value one in relation to the topic 'Operating System', if 'Linux', 'Windows' and 'Mac OS X' are modelled as direct subtopics of the topic 'Operating System', which is quite natural for this context.

Modelling of topic-maps is not always a trivial task. There are some cases existing where the 'right' structure of the topic-maps may depend on the point of view of the modeller. This difficulty and its possible consequences on the similarity matrix are presented by figure 4. This figure examplifies the problem on three topics: 'water', 'ice' and 'snow'. One possibility to arrange those topics is an alignment regarding their *is-a* relationship (classification). The result of this approach is presented in figure 4a. Ice and snow are in this case subtopics of water. Therefore the similarity between ice and snow (0.500) is lower then the similarity between ice and water or snow and water (0.866). Furthermore the normed vector sum of ice and snow adds up to the vector of water. This results in the similarity value one between water and the normed vector sum of ice and snow. In cases where this is not desired, a dummy topic can be introduced like in fig-



	τ_1	τ_2	τ_3
τ_1	1,000	0,866	0,866
τ_2	0,866	1,000	0,500
τ_3	0,866	0,500	1,000

Figure 3: Visualisation of topic representation.

ure 4b. This may be useful if it is known that there are more subtopics existing to a topic, but if for e.g. for complexity reasons the topic-map shall not be fully modelled.

Figures 4c to 4e take a different point of view on the things: Here the structure of water, ice and snow is modelled by the use of the *consist-of* relationship. This leads naturally to a different structure of the topic map. As shown in figure 4c water is modelled as a subtopic to ice and snow. This results in the fact that all three topics are assigned to the same topic vector. So the pairwise similarity of all three topics has the value one, which is in fact not very useful. A reason for this is that the modelling of the relationship between water, ice and snow in this scenario is wrong: In reality ice and snow naturally consist primary of water, but there are some more things needed for ice and snow which makes the difference like for e.g. low temperature. Further there is a difference between ice and snow, for example regarding their crystalline structure which is also not modelled in figure 4c. Figure 4d shows the same model with two dummies representing the special but not named ingredients which are needed to create ice respectively snow. In this case we get a useable similarity matrix. One feature of this matrix is the fact, that the similarity of the added vectors of ice and snow to the water vector is lesser than the value one. Further similarities between ice respectivly snow and water has been lowered a little in comparision to figure 4a, while the similarity between ice and snow themself has been highered a bit. But on the whole the proportions in the similarity matrix of figure 4d has not change in comparision to figure 4a.

The comparison between figure 4d and figure 4e shows that the similarities of two super-topics is directly depending from the relative amount of common subtopics in ratio to the total number of subtopics of the super-topic. In figure 4e this ratio is lower for the ice and snow topics than in figure 4d. For this reason the similarities (except the similarity between one and the same topic) are generally lower.

At a closer look we see that neither the similarity matrix in figure 4a nor in figure 4d is satisfying. It is intuitively not comprehensible why the similarity between ice and snow is lower than e.g. the similarity between ice and water. Figure 4f shows an topic-map representing the relationships between the three topics in a better way. Ice is defined as a subtopic to water because it is a special physical condition of water, which includes that it is a specialization. Because water may have several more physical states (e.g. like steam) dummy1 has been introduced to represent them. Snow is modelled as a subtopic to ice, because snow is ice with a special crystalline structure. This is in fact another specialization. Likewise water snow may also have further specializations e.g. like artificial snow which are represented here by dummy2. As shown in the similarity matrix of figure 4f ice and snow have the highest similarity value (0.913) followed by ice and water (0.851) and snow and water (0.777) which is fitting well to the intuitive feeling most people have regarding those similarities.

3. PRACTICAL USAGE AND POTENTIAL

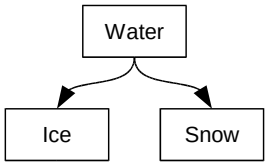
The resulting similarity matrix of the topic-map transformation algorithm presented here can be used in IF and IR algorithms, which are able to handle similarities between paired terms, words or topics. For instance the *Fuzzy Set Model* [12] and the *Topic-based Vector Space Model* [2] can be considered. Like already mentioned in the above section one useful approach is to map all terms or words of the documents to topics. For example the terms 'Microsoft Windows', 'MS-Windows' and 'Windows' could be assigned to the topic 'Microsoft Windows Operating System'. For the further processing by the IF or IR algorithms only the topics and their similarities should be used instead of the terms or words as usual. One sample implementation using this approach can be found in [7].

A problem which automatically occurs when the above approach is used is the question where the topic-map can be derived from. One possibility is naturally to model the topic-map by hand, which is quite laborious if the documents are not restricted to a controlled vocabulary. A second approach is to use algorithms for ontology learning [11] to derive an

topic-map

similarity matrix

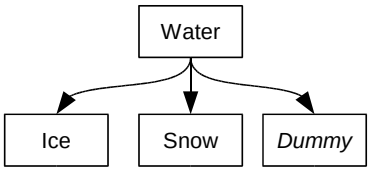
a)



	Water	Ice	Snow
Water	1,000	0,866	0,866
Ice	0,866	1,000	0,500
Snow	0,866	0,500	1,000

sim(Ice+Snow, Water) = 1,000

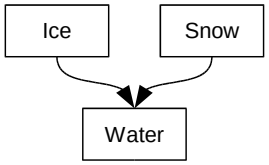
b)



	Water	Ice	Snow
Water	1,000	0,816	0,816
Ice	0,816	1,000	0,500
Snow	0,816	0,500	1,000

sim(Ice+Snow, Water) = 0,943

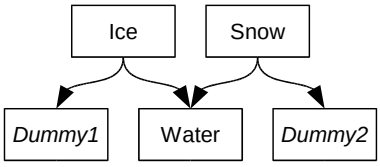
c)



	Water	Ice	Snow
Water	1,000	1,000	1,000
Ice	1,000	1,000	1,000
Snow	1,000	1,000	1,000

sim(Ice+Snow, Water) = 1,000

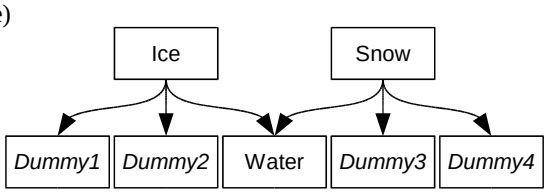
d)



	Water	Ice	Snow
Water	1,000	0,839	0,839
Ice	0,839	1,000	0,644
Snow	0,839	0,644	1,000

sim(Ice+Snow, Water) = 0,925

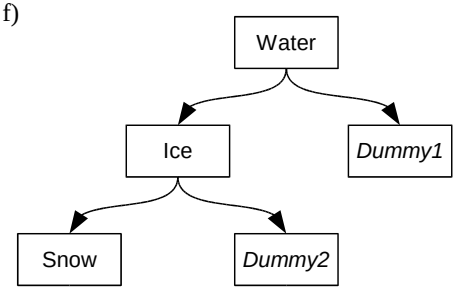
e)



	Water	Ice	Snow
Water	1,000	0,765	0,765
Ice	0,765	1,000	0,467
Snow	0,765	0,467	1,000

sim(Ice+Snow, Water) = 0,894

f)



	Water	Ice	Snow
Water	1,000	0,851	0,777
Ice	0,851	1,000	0,913
Snow	0,777	0,913	1,000

sim(Ice+Snow, Water) = 0,832

Figure 4: Similarities and topic-maps.

topic-map automatically. However those algorithms are still a bleeding edge technology and they may produce suboptimal results. As a third proposition, the topic-map could also be gained by reusing existing ontologies e.g. like *WordNet*[1]. The idea is to take some of the defined relations between the word meanings in the WordNet and to translate them into a topic-map, which uses word meanings of the WordNet as topics. A more detailed sketch for the implementation of this approach is described in [7].

The usage of topic similarities in IF and IR has the potential to improve the quality of filtering and searching of natural language documents, because the relationships between words and topics can now be considered efficently. The algorithm presented in this paper allows to reuse existing topic-maps for the IF and IR task by transforming them into a topic similarity matrix. Thus this makes it easier to implement high quality IF and IR systems.

4. SUMMARY AND FUTURE WORK

This paper proposes an algorithm for tranformation of acyclic topic-maps into similarites of topics. After a formal presentation of the tranformation algorithm, which is illustrated by an example, the consistency criterias and the features of the resulting similarities measures are discussed. Finally the issues of practical usage and the potential of the presented algorithm for IF and IR are identified. Even the application of the presented algorithm looks promising from the theoretical perspective, extensive quantitative evaluations have to be provided in the near future.

5. REFERENCES

[1] Baeza-Yates, R., and Ribeiro-Neto, B. *Modern Information Retrieval*. Addision Wesley Publishing Company, 1999.

[2] Becker, J., and Kuropka, D. Topic-based vector space model. In *Proceedings of the 6th International Conference on Business Information Systems* (2003), pp. 7–12.

[3] Belkin, N., and Croft, W. Information filtering and information retrieval: two sides of the same coin? *Communications of the ACM 35*, 12 (1992), 29–38.

[4] Furnas, G. W., Deerwester, S., Dumais, S. T., Landauer, T. K., Harshman, R. A., Streeter, L. A., and Lochbaum, K. E. Information retrieval using a singular value decomposition model of latent semantic structure. In *Proceedings of the 11th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval* (1988), pp. 465–480.

[5] Hiemstra, D. Term-specific smoothing for the language modeling approach to information retrieval: The importance of a query term. In *Proceedings of the 25st ACM SIGIR Conference on Research and Development in Information Retrieval* (2002), pp. 35–41.

[6] Jin, R., Hauptmann, A. G., and Zhai, C. X. Title language model for information retrieval. In *Proceedings of the 25st ACM SIGIR Conference on Research and Development in Information Retrieval* (2002), pp. 42–48.

[7] Kuropka, D. *Modelle zur Repräsentation natürlichsprachlicher Dokumente – Information-Filtering und -Retrieval mit relationalen Datenbanken*. Logos Verlag, Berlin, 2004.

[8] Kuropka, D. Uselessness of simple co-occurrence measures for if&ir – a linguistic point of view. In *Proceedings of the 8th International Conference on Business Information Systems* (Poznan, Poland, 2005).

[9] Langer, H. *Parsing-Experimente: Praxisorientierte Untersuchungen zur automatischen Analyse des Deutschen*. Peter Lang, Frankfurt (Main), 2001.

[10] Lavrenko, V., and Croft, W. B. Relevance-based language models. In *Proceedings of the 24st ACM SIGIR Conference on Research and Development in Information Retrieval* (2001), pp. 120–127.

[11] Maedche, A. *Ontology Learning for the Semantic Web*. Kluwer Academic Publishers, 2002.

[12] Ogawa, Y., Morita, T., and Kobayashi, K. A fuzzy document retrieval system using the keyword connection matrix and a lerning method. *Fuzzy Sets and Systems*, 39 (1991), 163–179.

[13] Ponte, J., and Croft, W. A language modeling approach to information retrieval. In *Proceedings of the 21st ACM SIGIR Conference on Research and Development in Information Retrieval* (1998), pp. 275–281.

[14] Robertson, S. E., and Jones, K. S. Relevance weighting of search terms. *Journal of the American Society for Information Sciences 27*, 3 (1976), 129–146.

[15] Salton, G. *Automatic Information Organization and Retrieval*. McGraw-Hill, New York, 1968.

[16] Salton, G. *The SMART Retrieval System — Experiments in Automatic Document Processing*. Prentice Hall Inc., Englewood Cliffs (NJ), 1971.

[17] Song, F., and Croft, W. B. A general language model for information retrieval. In *Proceedings on the 8th International Conference on Information and Knowledge Management (CIKM'99)* (1999), pp. 316–321.

[18] Wahlster, W. Verbmobil: Translation of face-to-face dialogues. In *Proceedings of the 3rd European Conference on Speech Communication and Technology* (Berlin, 1993).

[19] Wauschkuhn, O. Ein werkzeug zur partiellen syntaktischen analyse deutscher textkorpora. In *Natural Language Processing and Speech Technology — Results of the 3rd KONVENS Conference (Bielefeld)* (Berlin, 1996), D. Gibbon, Ed., Mouton de Gryter, pp. 356–368.

[1] http://www.cogsci.princeton.edu/~wn

[20] Wong, S. K. M., Ziarko, W., Raghavan, V. V., and Wong, R. C. N. On modeling of information retrieval concepts in vector spaces. *ACM Transactions on Database Systems 12*, 2 (1987), 299–321.

[21] Zhai, C. X., and Lafferty, J. Two-stage language models for information retrieval. In *Proceedings of the 25st ACM SIGIR Conference on Research and Development in Information Retrieval* (2002), pp. 49–56.