Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

A hybrid similarity concept for browsing semi-structured product items

Markus Zanker, Sergiu Gordea, Markus Jessenitschnig
and Michael Schnabl

University Klagenfurt
9020 Klagenfurt, Austria
markus.zanker@uni-klu.ac.at

Abstract. Personalization, information filtering and recommendation are key techniques helping online-customers to orientate themselves in e-commerce environments. Similarity is an important underlying concept for the above techniques. Depending on the representation mechanism of information items different similarity approaches have been established in the fields of information retrieval and case-based reasoning. However, many times product descriptions consist of both, structured attribute value pairs and free-text descriptions. Therefore, we present a hybrid similarity approach from information retrieval and case-based recommendation systems and enrich it with additional knowledge-based concepts like threshold values and explanations. Furthermore, we implemented our hybrid similarity concept in a service component and give evaluation results for the e-tourism domain.

1 Introduction

Browsing assistance, recommendation and personalization are key characteristics of e-commerce sites to reduce the information overload of users. Implementations of such sales support functionalities base fundamentally on the concept of similarity. Instance-based browsing allows users to retrieve similar items or items that are in principle comparable to a reference instance but should be different with respect to some specific features [1–3], i.e. tweaking & critiquing.
Recommendation applications differentiate themselves by the amount and form of knowledge they require about the product domain and the user situation[4–6]. Pure collaborative filtering approaches require no knowledge about the product domain at all. Items are solely described by an unique identifier, while extensive preference information represented by user/item ratings is required. Recommendations for a specific user are computed by determining his neighborhood of other users based on similar ratings and those items are proposed that his nearest neighbors liked. Contrastingly, content-based approaches build on data-intensive product representations, such as full-text documents or Web pages [7]. Characterizing terms from preferred document instances are used to build a user model. The further retrieval is guided by the similarity between the user model and candidate instances. Case-based [8,

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

9] and knowledge-based [10, 11] recommender systems assume structured product representations such as attribute-value pairs. Both approaches support value elicitation for user interaction. In case-based recommendation the product item most similar to the ideal case according to the explicit input of user preferences is presented. Whilst, knowledge-based recommenders possess explicit domain knowledge that maps subjective user requirements onto technical item characteristics[11]. This deep domain knowledge allows to propose product instances due to some form of business rules or constraints.
However, in application domains like e-tourism, consumer electronics or real estates such a dependency between recommendation technique and item representation is too restrictive. Typically items are semi-structured as they are represented by features as well as full-text descriptions, e.g. facilities and descriptions of a hotel or technical features and free text information about accessories of a digital camera model.
Therefore, we propose an hybrid similarity measure that encompasses similarity measures from case-based recommendation and content-based document filtering to support semi-structured item representations. Furthermore, we employ positive and negative preferences for full-text retrieval and enrich the approach with techniques from knowledge-based advisory. We add threshold values on the level of feature-similarity to explicitly model non-similarity and add explanatory facilities.
For practical applicability we developed an editor environment that ensures easy setup and maintainability. We choose the domain of e-tourism for demonstration and conducted several experimental evaluations. The paper is organized as follows: First we discuss different similarity concepts and related work in Section 2. In Section 3 we present our hybrid similarity measure. We continue by sketching our prototype and present results from an experimental evaluation in the domain of e-tourism. Finally we conclude and give an outlook on future work.

2 Related work

Case-based recommendation [8] is a similarity-based retrieval of objects based on user preferences and product descriptions. There exists a large body of research on how to define similarity between entities and how it can be measured. Osborne and Bridge [12, 13] distinguished similarity metrics according to their return type. When computing the similarity between a reference or query case and a set of cases, absolute or nominal similarity measures return a symbol for each case from an alphabet such as boolean values. Relative or ordinal similarity measures produce a lattice that represents a partial ordering of cases. Typically, several atomic similarity measures on different features can be combined by specific composition metrics [13, 14]. However, defining knowledgeable composition operators is an effort-intensive task.
Cardinal similarity measures produce numeric similarity values that at first sight can be combined more easily by standard arithmetic operators like $+$, $-$ or $\times$. But numeric similarity values need to be chosen carefully to ensure a correct interpretation. Nevertheless, case-based recommendation systems [8] rely on weighted sum approaches for cardinal

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

similarity metrics as their setup and definition resembles widely known multi-attribute utility schemes.

Tweaking & critiquing is a browsing assistance mechanism that like case- and knowledge-based recommender systems also builds on structured item representations [2, 15]. Users may formulate a critique concerning some of the feature values of a specific product instance c and the system returns several other product instances that fulfill the critique but are also similar to c. Newer work in this field proposes that returned items should be in addition very dissimilar towards each other [16]. Stahl [17] enhances the utility-oriented matching of a similarity-based recommendation strategy by learning from past cases. The Wasabi Personal Shopper [18] experimented with full-text wine descriptions and extracted explicit features such as *sweetness* or *tastiness*, but similarity itself is still computed on the structured case representation.

Text analysis and automatic feature selection is an important issue in the information retrieval community. There, many types of feature selection algorithms implying different computing complexities were proposed [19], e.g. based on stop words lists, minimum frequency, stemming and/or latent semantic indexing. **Content-based filtering** systems employ these techniques and learn the information seeking behavior of users by building models from preferred documents [7]. They represent documents, i.e. full-text information items, as vectors of stemmed terms. The term frequency - inverse document frequency (TFIDF) algorithm [20] for instance counts the occurrences of a term within a document and divides it by the frequency of the term throughout all indexed documents. This ensures that term occurrences with high discriminating power among all documents are reinforced. Furthermore, frequency numbers are typically normalized to eliminate the influence of document length. Although this information retrieval technique on natural language texts is quite simplistic it yields surprisingly good results compared with more complex natural language representations [21]. One of the basic assumption underlying TFIDF algorithms is that term occurrences are uncorrelated. Therefore, improvements of the algorithms take the neighborhood of terms into account and use probabilistic reasoning [22].

3 Hybrid similarity approach

Products and services offered over the Web are many times represented in a semi-structured form, therefore the computation of item similarity needs to take their full-text descriptions as well as their structured features into account. Here, we present a generic framework for the definition, maintenance and computation of similarity values between items that encompasses both technical approaches case-based similarity as well as document retrieval techniques from content-based filtering. Furthermore, we complement the approach with concepts from knowledge-based systems such as domain dependent threshold definitions and explanatory hints. We start by giving a small example.

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

3.1 Example

Our example is from the domain of accommodations, where sim_{desc} is a content-based measure for full-text descriptions while sim_{loc} and sim_{price} are functions on location and price similarity respectively.

case	description	location	price
c_1	Our farm is the ideal place for families and people looking for peace and `recreation` as well as for hikers and biking enthusiasts. We are located on a hilltop.	out in the open	28
c_2	Our farm provides `unforgettable` views on the lake and the mountains. Furthermore, we offer `organic` `produce` and relaxing `wellness` (`sauna`, solarium).	out in the open; by a mountain; near golf course; near airport	40
c_3	Spend `unforgettable` days in our emperor villa in the outskirts of Vienna. You may want to discover the city or enjoy our `wellness` and `recreational` facilities like `sauna`, jacuzzi or gym. Our restaurant serves traditional and international food as well as `organic` farm `produce`.	near town center; near golf course; near airport	85

For computing textual similarities we highlighted keywords in `typewriter` font. Note that words appearing only in a single description, e.g. jacuzzi, and those appearing in all descriptions, e.g. farm, are of no use for computing document vectors. Therefore, we get the following vector space:

keyword	c_1	c_2	c_3
recreation	1	0	1
unforgettable	0	1	1
organic	0	1	1
produce	0	1	1
wellness	0	1	1
sauna	0	1	1

We compute sim_{desc} using cosine between document vectors as given in [20]. We define similarity of location sim_{loc} using Dice coefficient [23], where twice the number of similar entries in both cases is divided by the sum of entries in each case: $sim_{loc}(c_i, c_j) = \frac{2 \times |c_i \cap c_j|}{|c_i| + |c_j|}$
Furthermore, we choose a symmetric function for sim_{price} that relates the distance of prices to the average price of both cases, i.e.
$sim_s(c_i, c_j) = 1 - \frac{2 \times \mathrm{abs}(c_i - c_j)}{(c_i + c_j)}$
The resulting similarities for sim_{desc}, sim_{loc} and sim_{price} are therefore as follows:

$$\begin{array}{lll} \mathrm{sim}_{desc}(c_1, c_2) = 0 & \mathrm{sim}_{loc}(c_1, c_2) = 0.4 & \mathrm{sim}_{price}(c_1, c_2) = 0.65 \\ \mathrm{sim}_{desc}(c_1, c_3) = 0.38 & \mathrm{sim}_{loc}(c_1, c_3) = 0 & \mathrm{sim}_{price}(c_1, c_3) = 0 \\ \mathrm{sim}_{desc}(c_2, c_3) = 0.91 & \mathrm{sim}_{loc}(c_2, c_3) = 0.57 & \mathrm{sim}_{price}(c_2, c_3) = 0.28 \end{array}$$

When we now query for a similar case for c_2 we will retrieve either c_3 or c_1 as the closest one, depending on the weighting scheme for combining the atomic similarities for the three features.

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

3.2 Knowledge-based similarity

As a first step we generalize the similarity definition for case-based recommendation given in [8] by defining the similarity between a query item q and a case c as the weighted sum of m similarity functions $sim_{d_j}(q_{d_j}, c_{d_j})$, where q_{d_j} resp. c_{d_j} are feature sets of q and c. Content-based similarity computations are integrated as specific similarity functions on full-text attributes like sketched in the example.

$$sim(q,c) = \frac{\sum_{j=1}^{m} w_j \times sim_{d_j}(q_{d_j}, c_{d_j})}{\sum_{j=1}^{m} w_j} \quad (1)$$

Thus, abstracting from similarity functions on the feature level to similarity functions at the level of feature sets enhances the expressivity of similarity definitions. For instance multi-valued similarity functions can be computed on flat data structures, e.g. boolean attributes that each encode the availability of a single facility, or text indexing functions can work on combined sets of textual descriptions.
In addition we introduce a **threshold value** for each similarity definition that can be understood as a knock-out criteria. This way domain experts are enabled to explicitly exclude cases from being considered similar to some query. For instance if cases whose price differs more than 50% from the compared price should be considered dissimilar the threshold value needs to be set to 1/3.
Formally, we define a knowledge-based similarity between case c and query q as follows:

$$SimKB(q,c) = \begin{cases} 0 & : sim(q,c) < t \\ 0 & : \bigvee_{j=1}^{m} (sim_{d_j}(q_{d_j}, c_{d_j}) < t_{d_j}) \\ Sim(q,c) & : \text{else} \end{cases} \quad (2)$$

If the computed similarity at any feature dimension violates threshold t_{d_j} or the overall similarity falls below threshold t the knowledge-based similarity between q and c is set to zero. This way, domain experts can explicitly define non-similarity and exclude unintended results.
Explanations are another concept from knowledge-based approaches that we add to our similarity framework. Acceptance of a system increases and users develop more trust if they are explained why an item is proposed to them.
For each feature dimension d_j, i.e. similarity function, two explanation texts pex_{d_j} and nex_{dj} can be formulated by the domain expert. The positively formulated one applies if the computed similarity value reaches a specific positive threshold $t_{pex_{d_j}}$. Consequently, the negatively formulated explanation text applies only in case the similarity on dimension d_j falls below threshold value $t_{nex_{d_j}}$.

$$PEx(q,c) = \{pex_{d_j} | sim_{d_j}(q,c) \geq t_{pex_{d_j}}\}$$
$$NEx(q,c) = \{nex_{d_j} | sim_{d_j}(q,c) \leq t_{nex_{d_j}}\}$$

Therefore, the similarity between query item q and c is explained by the

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

union of all positively and negatively formulated explanation texts that apply. If the overall similarity is computed to be zero, then no explanations need to be given.

$$Ex_{SimKB}(q,c) = \begin{cases} \emptyset : SimKB(q,c) = 0 \\ PEx(q,c) \cup NEx(q,c) : \text{else} \end{cases} \quad (3)$$

Later on, a user interface component can provide an explanation button for each proposed item that delivers explanations like *The River Inn offers facilities comparable to the Oceanside Hotel. Prices are about the same range. However, it is not that close to the beach.* Placeholders allow more lively formulations such as referring to the name of the proposed or the query item. Furthermore, the domain expert has the possibility to define an order on the different explanatory strings to make sure that they are intuitively combined.

3.3 Text similarity

In classic content-based filtering applications, where webpages or documents are recommended, similarity between items is in most cases based on preferred occurrences of keywords, i.e. if a user likes some documents, other documents are proposed that contain characteristic terms from his/her preferred documents. In application domains like email or webpage filtering a model of negative preferences is built from characterizing terms to create spam filters [24].
In the domain of product and service recommendation we propose to use both positive and negative user preferences to determine the similarity between full-text descriptions.
Descriptions of products and services must be seen more technical than news documents or Web pages, as they are typically using a limited set of domain specific terms. They deliver consistent and precise descriptions of the product or service that catch the attention of users within stringent space limitations ($\sim$ a few hundred characters). Furthermore, the assumption that what is not described is also not available holds empirically most of the time. I.e. a hotel that is situated close to a lake or offers wellness facilities will not conceal these facts to its potential customers. Therefore, if a term occurs in a document that is missing in the reference document a negative weight, i.e. a negative preference is applied.

4 Implementation

We implemented the presented techniques in a service component for B2C applications, which means it is a technical module that offers a Java API. It can be used by a shop software to realize a *show me similar items* type of functionality. Furthermore, we can employ the similarity framework to build more sophisticated recommendation services on top of it, e.g. tweaking critiquing applications or to build a user model from

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

item characteristics preferred by the user in the past. We give an architectural overview, discuss the implementation of additional similarity measures and finally sketch the editor environment, that ensures a comfortable way of maintenance.

The system's overall design is based on an extensible component based architecture using the Hivemind[1] open source framework for component registration. It is part of a to-be-commercialized bunch of service components for developing interactive selling applications. Given a query item, the API returns similar items upon request that are in addition justified by an explanatory expression.

Following the generic framework approach, the system can be easily extended with additional similarity functions as long as their implementation follows the interface definition. The analysis of full-text descriptions is based on the open-source indexing engine Lucene, an open source project implementing a high performance text indexing and search engine [2]. It is the most commonly used open source IR library with a very active user and developer community [25].

For reasons of runtime efficiency, similarities between cases are precomputed on a regular basis. At runtime, values are only retrieved from the database and no computation needs to take place. Weights between similarity functions are dynamically adapted based on user preferences. Therefore also similarities between feature sets, i.e. the results of the different similarity functions, can be stored.

Each similarity function implementation applies a scale of evaluated data, i.e. nominal, ordinal, numeric or full-text. Similarity between nominally and ordinal scaled features must be modeled explicitly by a domain expert, i.e. he/she can enter appropriate similarities into a matrix of possible feature values. For numeric features several functions, e.g. fuzzy measures or step functions are implemented. Full-text similarity values are computed using positive and negative preferences.

Knowledge acquisition and maintenance of a knowledge-based system are crucial for its effective use. Our similarity framework is part of a comprehensive suite for interactive selling applications. Therefore, the definition and configuration of similarity functions is fully supported by an editor environment based on the Eclipse Rich Client Platform. The domain expert can graphically select a set of features of a data object and associate them with a similarity function. Furthermore, threshold values and explanatory texts can be edited and a default weighting scheme defined. Within a separate testing screen, the domain expert can choose a case and query for similar items and analyze the result for plausibility.

5 Evaluation

Although a hybrid item similarity concept is advantageous from the point of expressivity alone, we also empirically evaluated two of the presented

[1] See http://jakarta.apache.org/hivemind for reference.
[2] see http://lucene.apache.org

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items. Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web), Springer 2006 (LNCS, 4082), pp. 21-30.

concepts by instrumenting experiments with data from the tourism domain. The data set consisted of 2127 hotels, guesthouses, farms and B&B accommodation opportunities.

In the **first experiment**, we compared the effectiveness of using positive and negative preferences for the computation of textual similarities between hotel descriptions. In order to evaluate the effectiveness of positive and negative preferences in text similarity, we created an online questionnaire with one random query document D_q and three reference documents $D_1, \ldots, D_3$ containing the full-text description of an accommodation. Each of the three reference items was computed using a different algorithm:

- positive and negative preferences $alg_{pos/neg}$, where terms occurring in both documents were multiplied with a standard factor 1 and terms that did not occur in the query document were punished with a weighting factor of -0.3.
- only positive preferences alg_{pos},
- random selection algorithm alg_{random} within the same accommodation category.

63 persons participated in the experiment and had to rate which of the three reference items in their opinion matched best with the query item. We also stored timestamp information with each submitted answer, so we could eliminate answers that did not allow themselves time for reading the accommodation descriptions. So we considered the answers of 55 participants for final evaluation: Around 51% of all answers indicated that the document retrieved by $alg_{pos/neg}$ matches the query document best. 35% rated the resulting documents from alg_{pos} highest and remaining 14% supported alg_{random}. Due to the nature of the experiment, i.e. the collection of the subjective opinion of the users, the preference random selection algorithm can be understood. Concluding, $alg_{pos/neg}$ was recommending items that had a more concise textual description of approximatively the same size as the query item while alg_{pos} has a tendency for longer documents that have a higher chance of containing most of the queried keywords. Although the experiment has a rather small sample size it nevertheless suggests that $alg_{pos/neg}$ significantly improves the retrieval results for product and service descriptions.

In the **second experiment**, we measured the performance of a hybrid similarity concept vs. pure content-based text similarity and structured item similarity.

The online questionnaire contained a semi-structured query item and again three reference items, that where computed according to the following three algorithms:

- hybrid similarity definition alg_{hybrid}, where text similarity was weighted 30% and all structured features like category, price or facilities 70%
- alone text similarity with positive and negative preferences $alg_{pos/neg}$
- pure similarity of structured features alg_{struct}.

This time, the 75 participants had to rank the three alternatives. The results of the second experiment confirmed our hypothesis that a hybrid similarity concept matches best to the concept of similarity users have in 50% of all answers rated the proposals of alg_{hybrid} best, while 18% preferred $alg_{pos/neg}$ and 31% supported alg_{struct}.

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

6 Conclusions

Computing the similarity between items is an essential functionality for recommendation applications in e-commerce, either to show similar items to users upon their explicit request or as an underlying capability for implementing tweaking critiquing systems as well as for building up user models. In many domains (e.g. e-tourism, consumer electronics or real estate) the offered products and services are represented by semi-structured information, i.e. a set of features and full-text descriptions. Up to now recommender systems have been either using document retrieval techniques or computed weighted sums of similarity functions on a structured feature representation. Our contribution is a hybrid framework for computing item similarity, that encompasses existing work on case-based recommendation and content-based filtering systems. Furthermore, we innovated weighted sum measures by adding knowledge concepts like threshold values, explanations and maintenance facilities. We conducted an experimental evaluation and implemented our framework as part of a suite of services for developing interactive selling applications.

References

1. Burke, R.D., Hammond, K.J., Young, B.C.: The findme approach to assisted browsing. IEEE Expert **July/Aug.** (1997) 32–40
2. Shimazu, H.: Expert clerk: Navigating shoppers‘ buying process with the combination of asking and proposing. In: 17^{th} International Joint Conference on Artificial Intelligence (IJCAI). (2001) 1443–1448
3. McCarthy, K., Reilly, J., McGinty, L., Smyth, B.: Experiments in dynamic critiquing. In: International Conference on Intelligent User Interfaces (IUI). (2005) 175–182
4. Sarwar, B., Karypis, G., Konstan, J., Riedl, J.: Analysis of recommendation algorithms for e-commerce. In: ACM Conference on e-Commerce (EC). (2000) 158–167
5. Burke, R.: Hybrid recommender systems: Survey and experiments. User Modeling and User-Adapted Interaction **12(4)** (2002) 331–370
6. Adamavicius, G., Tuzhilin, A.: Towards the next generation of recommender systems: A survey of the state-of-the-art and possible extensions. IEEE Transactions on Knowledge and Data Engineering **17(6)** (2005)
7. Balabanovic, M., Shoham, Y.: Fab: Content-based, collaborative recommendation. Communications of the ACM **40(3)** (1997) 66–72
8. O‘Sullivan, D., Smyth, B., Wilson, D.: Understanding case-based recommendation: A similarity knowledge perspective. International Journal of Artificial Intelligence Tools (2005)
9. Ricci, F., Werthner, H.: Case base querying for travel planning recommendation. Information Technology and Tourism **3** (2002) 215–266
10. Burke, R.: Knowledge-based recommender systems. Encyclopedia of Library and Information Systems **69** (2000)

Zanker M., Gordea S., Jessenitschnig M., Schnabl M.: A Hybrid Similarity Concept for Browsing Semistructured Product Items.
Proceedings of 7th International Conference on Electronic Commerce and Web Technologies (EC-Web),
Springer 2006 (LNCS, 4082), pp. 21-30.

11. Jannach, D.: Advisor suite - a knowledge-based sales advisory system. In: European Conference on Artificial Intelligence - ECAI 2004. (2004)
12. Osborne, H., Bridge, D.: A case base similarity framework. In: 3^{rd} European Workshop on Case Based Reasoning (EWCBR). (1996)
13. Osborne, H., Bridge, D.: Similarity metrics: A formal unification of cardinal and non-cardinal similarity measures. In: 2^{nd} International Conference on Case-based Reasoning (ICCBR). (1997)
14. Adah, S., Bonatti, P., Sapino, M., Subrahmanian, V.: A multi-similarity algebra. In: ACM SIGMOD international conference on Management of data. (1998) 402–413
15. McGinty, L., Smyth, B.: Tweaking critiquing. In: Workshop on Personalisation and Web Techniques at International Joint Conference on Artificial Intelligence (IJCAI). (2003)
16. McGinty, L., Smyth, B.: The role of diversity in conversational systems. In: 5^{th} International Conference on Case-Based Reasoning (ICCBR). (2003)
17. Stahl, A.: Combining case-based and similarity-based product recommendation. In: 6^{th} European Conference on Case-Based Reasoning (ECCBR). (2006)
18. Burke, R.D.: The wasabi personal shopper: A case-based recommender system. In: 11^{th} International Conference on Applications of Artificial Intelligence (IAAI). (1999) 844–849
19. Mladenic, D.: Text-learning and related intelligent agents: A survey. In: IEEE Intelligent Systems. Volume 14. (1999) 44–54
20. Salton, G., Buckley, C.: Weighting approaches in automatic text retrieval. Information Processing and Management **24(5)** (1988) 513–523
21. Lewis, D.D., Jones, K.S.: Natural language processing for information retrieval. Communications of the ACM **39(1)** (1996) 92–100
22. Joachims, T.: A probabilistic analysis of the rocchio algorithm with tfidf for text categorization. In: International Conference on Machine Learning (ICML). (1997)
23. Frakes, W.B., Baeza-Yates, R., eds.: Information Retrieval, Data Structure and Algorithms. Prentice Hall (1992)
24. Resnick, P., Miller, J.: Pics: Internet access controls. Communications of the ACM **39(10)** (1996) 87–93
25. Gospodnetic, O., Hatcher, E.: Lucene in Action. Manning, Greenwich, CT (2005)