Pekka Malo
Pyry Siitari

A Context-Aware Approach to User Profiling with Interactive Preference Learning

AALTO-YLIOPISTON KAUPPAKORKEAKOULU
AALTO UNIVERSITY SCHOOL OF ECONOMICS

W-482

Pekka Malo – Pyry Siitari

A Context-Aware Approach to User Profiling with Interactive Preference Learning

Quantitative Methods in Economics and Management Science

June
2010

AALTO-YLIOPISTON KAUPPAKORKEAKOULU
AALTO UNIVERSITY SCHOOL OF ECONOMICS
WORKING PAPERS
W-482

HELSINGIN KAUPPAKORKEAKOULU
HELSINKI SCHOOL OF ECONOMICS
PL 1210
FI-00101 HELSINKI
FINLAND

Pekka Malo, Pyry Siitari and
Aalto University School of Economics

ISSN 1235-5674
(Electronic working paper)
ISBN 978-952-60-1029-8

Aalto University School of Economics -
Aalto-Print 2010

A Context-aware Approach to User Profiling with Interactive Preference Learning

Pekka Malo*† Pyry Siitari‡

April 6, 2010

Abstract

This paper proposes a context-aware method for user profiling and content retrieval based on interactive preference learning. The method uses a novel combination of ontology-driven context modeling with multi-attribute optimization, which allows the system to learn an implicit value function to represent the user's preference system. Due to the domain knowledge brought by ontologies, the system is able to account for the semantic context of the information retrieval task while constructing the user profile. Additionally, a collaborative version of the algorithm is proposed, which is useful when only little or none preference information is available on the active user. In order to demonstrate the approach, we present a personalized business news reader application. The performance of the system is evaluated using Reuters RCV1 corpus.

1 Introduction

It is forecasted that by 2011 the digital universe will be 10 times the size it was in 2006 [11]. The staggering growth rate of the world's information base is reflected as increased efforts to develop more sophisticated content filtering systems. This is a challenge especially for news service providers, who need personalized content in order to win more users.

This paper presents an interactive user profiling and content retrieval approach based on a combination of multiattribute optimization and ontology-driven context modelling. As a base scenario, we consider a personalized financial news reader application, where a large document-base needs to be filtered according to each user's preferences. To make the system usable for mobile

*Address: Aalto University School of Economics, Department of Business Technology, Runeberginkatu 22-24, P.O.Box 21220, FIN-00076 Aalto. Email pekka.malo@hse.fi, Tel. +358943131, Fax +358943138535.

†Corresponding author

‡Address: Aalto University School of Economics, Department of Business Technology, Runeberginkatu 22-24, P.O.Box 21220, FIN-00076 Aalto. Email siitari@hse.fi, Tel. +358943131, Fax +358943138535.

applications no explicit queries are considered. In order to find the relevant documents, the system profiles the active user based on his preference feedback and the content of the news items, which the user or other users with similar interests have browsed. The resulting profile is constructed such that it can be easily compared both against the content of the news items as well as profiles of the other users. This requires a common reference structure against which both users and content can be described. For this purpose, we use an ontology-based knowledge-model to establish a basis for understanding the domain's concepts and their dependencies. Instead of looking at documents as collections of keywords, we model them in terms of uniquely defined concepts which are interrelated as parts of a larger domain knowledge base. An additional benefit brought by ontologies is the recognition of concept generality or specificity, which can be used as part of relevance weighting schemes. The more specific the concept found in text, the higher its weight is likely to be when describing the profile of the given news item.

In general, an ontology can be defined as a description of a given domain, its concepts and properties in a machine-readable form [32, 34]. Although, ontology building is generally considered to be a labour-intensive task, we argue that an access to a knowledge model is necessary for capturing links between concepts and their targets. As discussed by Rajapakse and Denham [23], among others, it is unreasonable to assume that concepts could be represented solely by "keywords". For example, depending on the context, the word "bank" can be used in quite different senses ranging from a financial institution to a sea shore. If the document is entitled "financial stress tests", the probability that the word "bank" refers to a financial institution rather than "bank of a river" is very high. For human readers, such process of context identification goes without notice but achieving the same in an algorithm requires specification of what is considered as domain knowledge. Therefore, we argue that by using ontologies as a backbone for describing both users and content alike, we can take the system a step closer to the way how human brain works. Furthermore, ontology development has become considerably easier from what it used to be a decade ago due to abundance of high-quality engineering tools and collective initiatives to build common sense ontologies.

However, being context-aware is only one part of the content retrieval problem. The second part is concerned with defining what is meant by relevance. Whereas relevance is commonly used as an effectiveness measure for an IR system, there are quite different interpretations and opinions about relevance assessment; see e.g. Borlund [5], and Nyongesa and Maleki-dizaji [22]. In this paper, relevance is defined by explicit preference feedback extracted from the user. Because subjective relevance is hard to measure, we consider the use of direct preference statements as the best way to capture the inherent multidimensionality of relevance. The approach is motivated by the studies of multiattribute decision making problems, where the goal is to find the most preferred solution through interactive preference learning procedures (Zionts and Wallenius [41]; Korhonen and Laakso [17]; Roy and Wallenius [27]; Maddulapalli et al. [20]; Deb et al. [9]; Roy et al. [26]). We assume that the user is willing to indicate his

preferences for a few news items by assigning them into preference categories or by making pairwise comparisons. Based on these preference-statements, we can then estimate an implicit value function to represent the user's preference system. In case little or no preference information is available, we suggest a collaborative extension where topical similarities and profiles of other users can be used to compensate for the low preference information on the active user.

Our profiling approach is best described as topic-driven, which is motivated by the possibility of both contextual and temporal shifts in user interests. When profiling users, it is important to recognize their tendency to do multi-contextual tasking. Instead of being interested in just a single main topic, users commonly follow a variety of topics with some or no relation to each other. Further, as discussed by Jung [16], user interests generally exhibit contextual drift by varying over time from one topic to another. Rather than constructing a single generic user profile, we prefer to model each user as a collection of topic-specific profiles. By doing so, we avoid mixing profiles based on unrelated topics. In addition to optimization of the rank-order in which the system retrieves news items, the accuracy of user profiling is also important when, for example, target marketing is considered.

The evaluation of the business news reader application was done using Reuters RCV1 corpus with a subset corresponding to the TREC-11 data set [24]. The particular data set was chosen due to availability of a large collection of hand-crafted relevance judgements and topic definitions, which were supplied by the assessors of TREC-11 filtering track; see Section 8.2 for further description. The results from the experiments show that the system is able to achieve large improvements in mean average precision and recall already with the first few preference statements. Although, we find that the system reacts aggressively to first preference statements, there is also evidence of fast profile convergence. All of the experiments are carried out without using explicit queries. The content retrieval is done solely based on the relevance judgements supplied by the user, which makes the system suitable for applications where explicit queries are not feasible. We also consider it encouraging that the results were obtained using a very light-weight domain ontology with only 1500 core business concepts. This suggest that the system is able to work consistently already with small knowledge models, which reduces the need for costly ontology engineering when extending the system for new domains.

The rest of this paper is organized as follows. Section 2 gives a short review of the related work and contributions of the present paper. Section 3 provides an overview of the business news reader application used as a basis of the case study. Section 4 introduces definitions related to ontology-based context modelling. These are used in Section 5 to describe how document content profiles are build with respect to the domain ontology. Section 6 defines the concepts needed for construction of topic-specific user profiles. Section 7 presents an incremental learning approach for both individual user profiling and collaborative profiling. An experiment based on the algorithm is given in Section 8. We conclude in Section 9.

2 Related work and contributions

There exists an extensive literature on information filtering and retrieval systems where user relevance feedback is considered; see e.g. Rocchio [25], Salton and Buckley [29], Zhang and Seo [40], Nyongesa and Maleki-dizaji [22], Sulaiman et al. [36], Taghipour et al. [38], Rajapakse and Denham [23], Roy et al. [26], and their references. The studies provide valuable insights into resolving the main IR tasks of representing information and retrieving items in the order of relevance. While the most popular way to account for user preferences appears to be based on a form of reinforcement learning, there are considerable differences between the approaches which arise from the choice of content representation model. The approach suggested in this paper contributes to the existent literature in the following ways:

(i) Use of concepts instead of keywords: As discussed by Mauldin et al. [21], Nyongesa [22], and Chen et al. [7], most retrieval systems suffer from the *keyword barrier* phenomenon, which refers to the inability of information retrieval systems to convey the semantic context of documents. To alleviate this problem, we use ontologies for domain modelling. Previously concept-based modelling in information filtering with relevance feedback has been studied by Rajapakse and Denham [23], who considered the framework of Formal Concept Analysis (FCA). The approach can be viewed as an unsupervised way of deriving a document-specific concept lattice from data. However, FCA-lattice differs from an ontology in that ontologies are typically designed by domain experts with asserted concept-definitions. Because the taxonomy of an expert-designed ontology is deep and well structured, it can be used as part of concept-weighting scheme. One contribution of the paper is to extend the well known Tf-Idf weighting model by introducing an ontology-dependent component to account for generality of a given concept. Another interesting alternative is from Chen et al. [7], who proposed a semantic-enable system based on topic-maps and semantic pattern clustering and matching. The system considered in this paper, however, differs from the work of both Chen et al. [7] and Rajapakse and Denham [23] in that no explicit semantic queries are considered. Instead, the learning is done entirely based on preference-statements. The use of explicit semantic queries in the spirit of Chen et al. [7] is, nevertheless, a relevant option which will be considered in further research.

(ii) Use of implicit value function learning: Instead of commonly applied reinforcement learning algorithms, we have chosen to consider an interactive multi-attribute technique for learning an implicit user value function. The approach can be considered as a distant relative of the well-known policy iteration algorithms in reinforcement learning literature due to the underlying actor-critic architecture of the implicit value function learning algorithm. To our knowledge, similar approach has been used for information retrieval only by Roy et al. [26]. However, they use a keyword-based binary vector-space representation for documents, which is quite different from the ontology-driven model suggested in this paper. We also differ in our focus on user profiling and collaborative aspects, which are not discussed by Roy et al. [26].

(iii) Topic-specificity and collaborative profiling: To account for the fact that users tend to work in multiple-contexts, we propose a topic-driven profiling approach where each user profile is defined as a collection of topic-specific profiles. The notion of topic-specificity is well-defined with respect to a domain ontology, which makes the topic-specific user profiles comparable to each other. This supports the use of collaborative profiling to augment single user profiles when limited amount of preference information is available. In this paper, we propose a simple extension of implicit value function learning algorithm, where information on the profiles of users with similar topics is used in conjunction with the single user profile. In the literature, topic-specific modelling has been considered mainly in web-crawling rather than content retrieval or filtering; see e.g. Fang et al. [12]. To our knowledge, none of the approaches concerned with topic-specific modelling have considered similar techniques to the ones suggested in the present paper.

3 News reader system

An overview of the news reader application's architecture is presented in Figure 1. The processing resources consist of (i) a content extraction module; (ii) a user and content profile matching module; and (iii) a news reader interface. The responsibilities of the modules can be summarized as follows:

(i) Content profiling: The incoming news items are first processed by content profiler, which contains modules for preprocessing, concept and named-entity identification, and disambiguation. The profiled content is annotated with respect to the domain ontology and weighted according to the relevance of the concepts within the document; see Sections 4-5. The knowledge base within content profile module supplies the profiler with information about domain ontology's concepts.

(ii) User profiling and content matching: The task of the value-function optimizer is to approximate the user's value function and construct a profile which can be evaluated against the content profiles to find the most preferred items. User profiling is introduced in Section 6 and the incremental learning strategy is described in Section 7. The updated user-profiles are persisted into a separate repository. The user-profile manager is also responsible for clustering similar user profiles to enable their efficient use in collaborative profiling.

(iii) News reader interface: The system considered in the experiment is designed to operate without explicit semantic queries. Instead, the system receives topic definitions from the user implicitly based on the news item which is currently being viewed or recently accessed by the user. We assume that the user is willing to also supply preference statements in the form of categorical high-medium-low judgements for a few items. The topic definitions and preference statements are then given to the user and

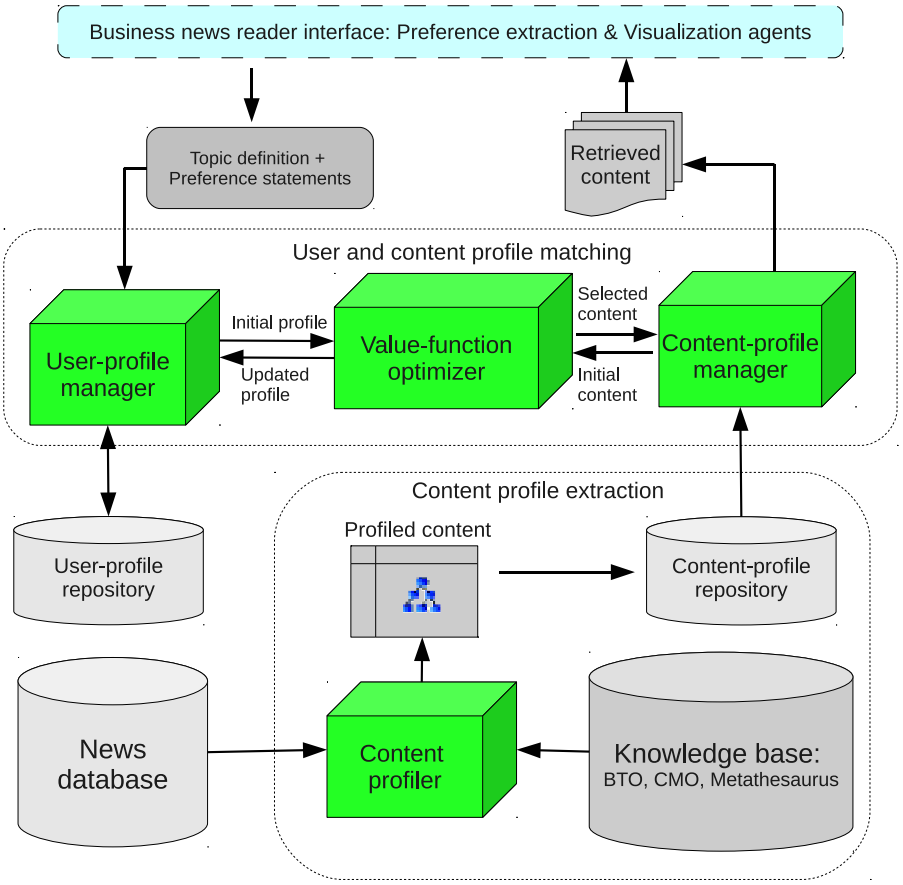

Figure 1: Business news reader

content profile matching module, which replies by returning the relevant content back to the user.

4 Context modelling with ontologies

The research on knowledge-based information filtering systems has been largely motivated by their improved ability for contextual understanding. Over the past two decades of studies on Semantic Web and Artificial Intelligence, researchers have proposed a range of different knowledge-models. A broadly adopted solution is the use of ontologies. In general, an ontology is a description of a given domain, its classes (or concepts), and properties in machine-readable form by means of an ontology language which is commonly referred to as a knowledge representation model; see e.g. Suchanek [35], Russell and Norvig [28], Staab and Studer [32]. For recent studies on the use of ontologies in financial context, see e.g. Wang et al. [39] and Shue et al. [30].

4.1 Knowledge representation model

Currently, the most popular knowledge representations are build using RDFS / OWL [32] model or their derivative products. Ontology construction is com-

monly viewed as a highly labour intensive process demanding considerable expert knowledge. Even today, the most widely adopted ontologies are still mainly hand-crafted. Examples of high-quality ontologies include the lexical database WordNet [13], medical language system UMLS [3], OpenCyc[1], UMBEL[2], and SUMO[3]. For experimental purposes, we have chosen to build a light-weight business-term ontology (BTO) which contains only the core domain concepts. This is augmented with a fact-triplet repository of relevant named-entities, such as companies, managers, products and brands.

To present the knowledge model used in this paper, we consider the RDFS extension proposed by Suchanek et al. [35, 34]. The advantage of this model is that relationships between facts and relations can be easily expressed while retaining decidability.

Definition 4.1.1 (Ontology) *An ontology over a finite set of common entities $\mathcal{C}$, a finite set of relation names $\mathcal{R}$ and a finite set of fact identifiers $\mathcal{I}$ is a reification graph over the set of nodes $\mathcal{I} \cup \mathcal{C} \cup \mathcal{R}$ and the set of labels $\mathcal{R}$, i.e. an injective mapping*

$$\mathcal{O} : \mathcal{I} \rightarrow (\mathcal{I} \cup \mathcal{C} \cup \mathcal{R}) \times \mathcal{R} \times (\mathcal{I} \cup \mathcal{C} \cup \mathcal{R}) \tag{1}$$

The basic element in the model is an entity which may refer to any abstract or concrete thing. Throughout, it is assumed that entities are discernible and we can tell whether two entities are the same. In this model, an entity is understood in a broad sense, i.e. all concepts (groups of entities with similar properties), relations, individuals and statements are considered as entities.

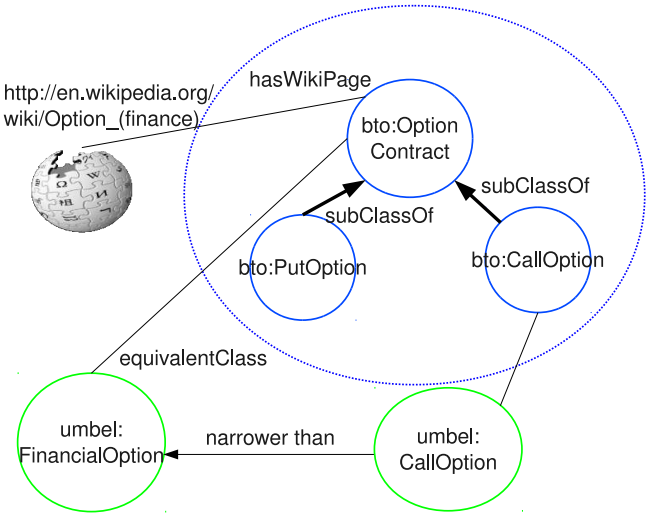

Figure 2: Part representation of OptionContract-concept and properties

The backbone of the ontology is a taxonomy of concepts which is formed by means of *subClassOf*-relation. For example, the taxonomy of our ontology

[1]http://opencyc.org/
[2]http://www.umbel.org/
[3]http://www.ontologyportal.org/

holds relations such as *subClassOf*(PutOption, OptionContract). In addition to the *subClassOf*-relation, the ontology model chosen for our application supports connections to external ontologies and folksonomies such as Wikipedia; see Figure 2. These ontology extensions will be considered in our forthcoming research.

The model used for the experiments in this paper consists of two components:

1. Hand-crafted business term ontology, $\mathcal{O}_{\mathrm{BTO}}$, which provides a taxonomic backbone for the knowledge model defined through *subClassOf*-relation, which expresses the generality vs. specificity of different concepts. In addition to subclassing relation, each concept in $\mathcal{O}_{\mathrm{BTO}}$ is equipped with mappings to a corresponding Wordnet class and a Wikipedia page defined by *hasWordnetClass* and *hasWikiPage* relations. These relations support alignment of the ontology with collectively built models, such as OpenCyc and UMBEL. The currently used version of BTO contains only 1500 core business concepts.

2. Machine-learned ontology of companies and managers, $\mathcal{O}_{\mathrm{CMO}}$, which provides the semantic structure for a medium-sized named-entity database containing approximately 0.5 million entities (companies, managers, product and brand names).

The above separation is justified for maintenance reasons. Whereas the business-term ontology $\mathcal{O}_{\mathrm{BTO}}$ requires modifications practically only when new economic concepts are created, the instance database of $\mathcal{O}_{\mathrm{CMO}}$ becomes easily depracated as new managers are hired and new products are launched. Therefore, the maintenance of $\mathcal{O}_{\mathrm{CMO}}$ is done using statistical techniques rather than careful ontology engineering which is better suited for $\mathcal{O}_{\mathrm{BTO}}$.

4.2 Operationalisation of the ontology

In order to operationalize the ontology, the model must support language-specific concept expansions. Therefore, the knowledge model is augmented with a metathesaurus, which provides textual representations of each concept in the ontology.

Definition 4.2.1 (Metathesaurus) *A metathesaurus for a given language, Σ, is understood as a set-valued (one-to-many) mapping, i.e.*

$$M_\Sigma : \mathcal{C} \rightrightarrows \Sigma \tag{2}$$

where $\mathcal{C} = \mathcal{C}_{BTO} \cup \mathcal{C}_{CMO}$ denotes the set of domain concepts available in ontologies $\mathcal{O}_{BTO}$ and $\mathcal{O}_{CMO}$. For each concept $c \in \mathcal{C}$, the set $M_\Sigma(c)$ is a collection of all text strings in language Σ which represent concept c.

Whereas the ontology itself is a language independent knowledge representation, the concept-expansions obtained from metathesaurus M_Σ permit identification of concepts from written text.

The use of metathesaurus for concept identification is done by simple string-matching (e.g. using finite-state transducers) that can efficiently lookup the strings described by metathesaurus within the documents. That is, given the document D as a collection of pre-processed strings, the thesaurus-lookup process amounts to computing the inverse image of M_Σ for the collection of strings within the document, i.e.

$$M_\Sigma^{-1}(D) = \bigcup_{s \in D} M_\Sigma^{-1}(s) = \{c \in \mathcal{C} | M_\Sigma(c) \cap D \neq \emptyset\}. \quad (3)$$

The set contains references to all concepts that could be potential candidates based on the strings found both in the document and the metathesaurus. Therefore, the inverse image, $M_\Sigma^{-1}(D)$, can be considered as a noisy approximation for the set of concepts appearing in the document.

However, an access to metathesaurus is not a sufficient condition for identification, because different concepts can share the same textual representation. Therefore, the thesaurus needs to be accompanied by a disambiguation system, which accounts for the context where a particular concept candidate is detected. When potential matches for concepts have been detected in written text, the remaining problem is to ensure uniqueness of the match. The task of disambiguation step is to reduce the set $M_\Sigma^{-1}(s)$ to a single concept or an empty set, if none of the concepts in domain ontology correspond to the particular word-sense of s in its document context. The disambiguation heuristic implemented for our application can be viewed as a hybrid of the methods proposed by Simón-Cuevas et al. [31], Chen and Chang [6] and Leslie et al. [18]. To obtain information on the alternative senses of strings contained in thesaurus, we benefit from the provided mappings to Wikipedia and WordNet. A detailed treatment of the approach is, however, beyond the scope of this paper and will be described more closely in our forthcoming research.

5 Content profiling

A commonly adopted way for content profiling is the bag-of-words approach, where documents are viewed as vectors of words or terms accompanied by relevance weights computed using methods such as tf-idf. In this section, we propose a method for creating document profiles by using an ontology-based knowledge model.

5.1 Document profile

Instead of operating with word-based profiles, we use an ontology to replace words with well-defined concepts. An essential benefit of ontology-based approach is that we can use the taxonomic backbone of an ontology to create a concept weighting method that is able to account for specificity or generality of different concepts. Furthermore, we can use the ontological relations to extend document profiles with connections to related concepts.

In what follows, we define the content profiling problem as a mapping from the document space to ontology domain concepts and weight-vectors.

Definition 5.1.1 (Document profile) *Let $\mathcal{D}$ denote a collection of documents. The content profile is defined by a mapping,*

$$P_{\mathcal{D}} : \mathcal{D} \rightarrow \mathcal{C}^N \times [0,1]^N \tag{4}$$

where N is the cardinality of the domain concept set $\mathcal{C}$. For document $D \in \mathcal{D}$ the sparse matrix $P_{\mathcal{D}}(D)$ is said to be the profile of the document with respect to ontology $\mathcal{O}$.

The process of constructing the mapping $P_{\mathcal{D}}$ can be decomposed into three steps: (i) Assuming that the document collection is preprocessed[4], the first step is to use the metathesaurus supplied with the ontology to detect concept candidates amid the text strings in the document; (ii) The second step involves disambiguation of the concept candidates and pruning out irrelevant candidates not found in the domain ontology; (iii) Finally, when the set of concepts describing the document content are identified, it remains to compute each representative concept a relevance-weight within the document; see Figure 3.

5.2 Ontological concept-weighting

When characterizing document contents one of our aspirations is to provide a weighting scheme, which gives higher weights for more specific concepts[5]. However, before introducing the weighting model, we need to decide on a characterization of concept specificity.

In our domain ontology, the specificity of a concept is determined by a concept-generality function, which describes the level of generality based on the location of the concept in the taxonomy and prior appearance frequencies in an annotated domain-training corpus[6].

Definition 5.2.1 (Concept-generality function) *Let $\mathcal{D}$ denote a document collection annotated with respect to ontology $\mathcal{O}$. The concept-generality function, $N : \mathcal{C} \rightarrow \mathbb{N}$, is defined recursively as a mapping*

$$N(c) = N_{isNarrower}(c) + N_{equivalent}(c), \tag{5}$$

where

$$N_{isNarrower}(c) = \sum_{c_i \in isNarrower(c)} N(c_i) \tag{6}$$

[4]The preprocessing step includes routine tasks such as tokenization, sentence splitting, part-of-speech tagging and morphological parsing to obtain root forms of tokens. At this step, we also run a preliminary Conditional Random Field (CRF) based named-entity identifier to detect candidates for organizations, people, and locations. See e.g. Finkel et al. [14] and Dingare et al. [10] for development and training of CRF and ME models.

[5]For example, the concept *AsianOption* should have a higher weight than a more generic concept *FinancialInstrument*.

[6]When considering the choice of training corpus $\mathcal{D}$, it must be required that the corpus is well-balanced enough to provide a good coverage of the domain of the given ontology $\mathcal{O}$. Otherwise, the taxonomy based generality becomes over-weighted.

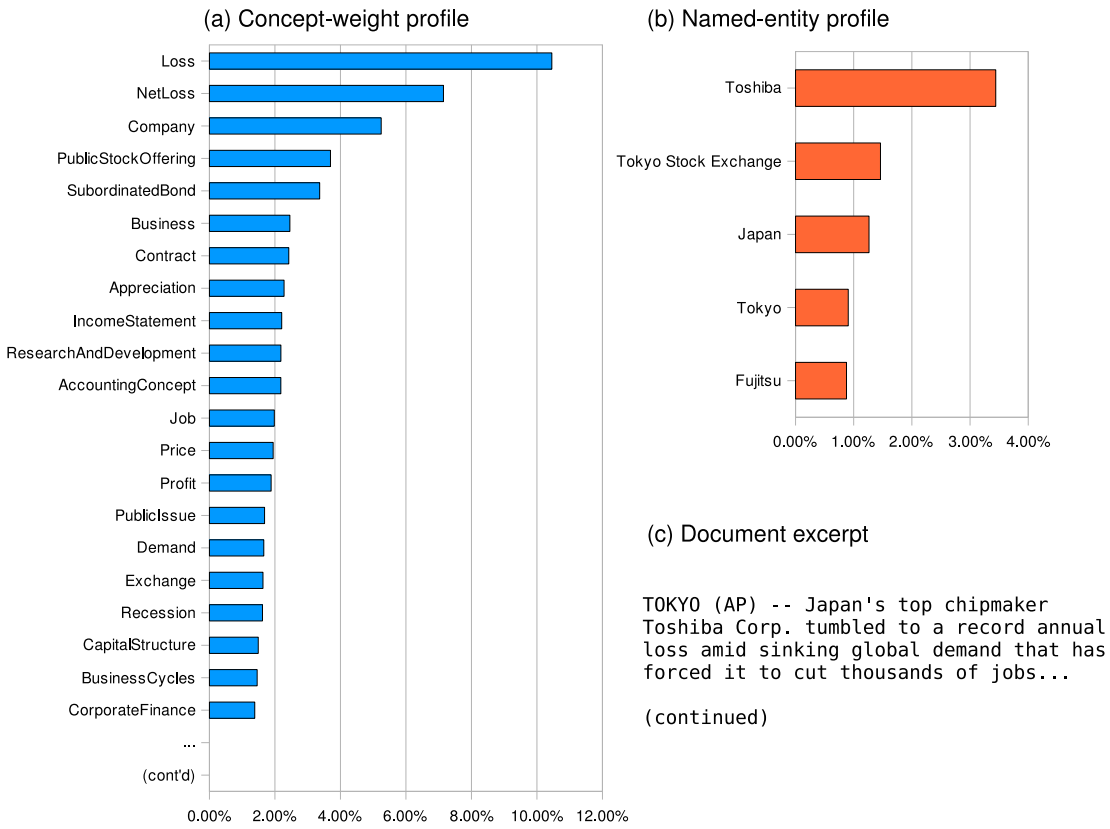

Figure 3: The figure shows a sample profile $P_{\mathcal{D}}(D)$ of a newsitem on Toshiba taken from Yahoo! Finance. For convenience the weights for (a) concepts and (b) named-entities are plotted separately.

and $N_{equivalent}(c)$ is the number documents in $\mathcal{D}$ featuring concept c or its equivalent class. We assume also the following boundary condition: if $c \in \mathcal{C}$ is a leaf-concept, i.e. isNarrower(c) $= \emptyset$, then $N(c) \geq 1$. Here, isNarrower(c) denotes the set of concepts which have semantically narrower sense than c or are subclasses of c.

The higher values of concept generality function imply broader and more general concepts; if $N(c_1) \geq N(c_2)$, we say that c_1 is at least as broad or general concept as c_2.

Having proposed a measure for the generality of concepts, we can use it as a weighting element to propose an adapted $\mathcal{O}$-Tf-idf method (5.2.2). Instead of using document frequencies to weight concepts, we use the concept-generality function to replace them with an ontology-based counterpart.

Definition 5.2.2 ($\mathcal{O}$-Tf-Idf) *Let $P_{\mathcal{D}}(D) = (c, w(c))_{c \in \mathcal{C}}$ denote a document profile with respect to an ontology. In $\mathcal{O}$-Tf-Idf approach, the weight function is defined by*

$$w_D(c) = K f_D(c) \log \left(\frac{\max_{c_i \in \mathcal{C}} N(c_i)}{N(c)} \right) \tag{7}$$

where $f_D(c)$ denotes the relative frequency of concept c in document D, N is the concept-generality function, and K is a normalization constant.

The obtained weighting scheme can be decomposed into two components. The first component is the relative frequency function f_D, which determines within document weight of the concepts. The second part of the weight is defined through concept-generality function as an inverse generality weight, which is not dependendent on the given document D but only on the concept at hand. When combined together, the weighting scheme components allow more specific concepts to have higher weights while retaining otherwise similar behavior to tf-idf measures.

6 User profiling

We begin with the assumption that the interests of each user are characterized by a personal ontology which is unobservable and time-varying. The user profiling problem is then to find a mapping from the user's personal ontology to the domain ontology such that the user's interests can be compared against content profiles. In this section, we decompose user profiling task into two parts: (i) definition of active topic, which describes the user's current interest in terms of the domain ontology's concepts; and (ii) computation of a preference-weighted user-profile for the active topic.

The approach suggested in this paper is best characterized as a topic-driven user profiling method. We argue that there is generally no reason to assume existence of a stationary user profile. Instead, each user is likely to be interested in a variety of topics which are prone to drift over time as new information becomes available allowing the user to update his personal ontology. Also abrupt contextual changes can occur. For example, a news portal user might first look at financial news and then move on to sports reviews, which would be an orthogonal change of a context with weak or no relation to preceding topics. Therefore, it appears natural to model a user as a collection of topic-specific subprofiles rather than assume sufficiency of a single generic user profile.

6.1 Topic-specific user profile

Suppose that each user has an unobserved personal ontology, which has semantic overlap with the domain ontology $\mathcal{O}$. To find content which matches user's current interests, we seek to characterize them in terms of the domain ontology's concepts. Thus, we define topic T as a subset of domain ontology concepts which are semantically related in user's personal ontology and have corresponding concepts in the domain ontology, i.e. $T = (c_1, \ldots, c_n) \subset \mathcal{C}$.

Whereas each topic is assumed to be internally coherent, a single user can still have several different topics with little or nothing in common. Therefore, we propose the following definition of a user profile:

Definition 6.1.1 (Topic-specific user profile) *Let $\mathcal{U}$ and $\mathcal{T}$ denote the set of users and admissible topics, respectively. For a given user $U \in \mathcal{U}$ and topic $T \in \mathcal{T}$, a topic-specific user profile is defined by the mapping*

$$P_{\mathcal{U},\mathcal{T}} : (U, T) \mapsto (c, w_{U,T}(c))_{c \in \mathcal{C}}$$

where $w_{U,T}(\cdot)$ denotes topic-specific concept weights for user U, such that $w_{U,T}(c) \in [0, 1]$ for every $c \in T$, and $w_{U,T}(c) = 0$ for every $c \notin T$. Then the full user-profile is given by

$$P_{\mathcal{U}}(U) = (P_{\mathcal{U},\mathcal{T}}(U, T))_{T \in \mathcal{T}}$$

which leads to assume that a user profile has a natural decomposition into topic-profiles.

By specifying the user profiles as topic-specific weight vectors, we thus avoid the problem of mixing concept-weights accross weakly related topics. Unless topic specific boundaries are introduced, we would risk developing meaningless concept-weight profiles as different topics would introduce possibly contrasting weights on the concepts.

At this point, two problems remain. The first one is about how to get the user to express his interests in topic form. In the present paper, we address this by assuming that a topic definition can be extracted from the profiles of the documents currently activated or viewed by the user. Alternatively, one could consider the use of semantic queries with ontological expansion; see e.g. BaalaMithra and SominMithraa [1] and Stojanovic [33]. The second problem concerns the way how topic-specific weights $w_{U,T}(\cdot)$ should be computed for each user and a topic, and is discussed in the following section.

6.2 Preferences and value function

The approach suggested for learning the user's preferences is *progressively interactive* in the sense of Deb et al. [9]. Instead of attempting to arrive at full characterization of the user's interests in a single step, we collect the preference information in a short sequence of periodical steps in the spirit of reinforcement learning. Every time the user accesses new material he has the opportunity of making a preference statement such as "document A is better than document B" or "document A is highly preferred". Because each preference-statement is made with respect to a certain topic, we suggest the following definitions for pairwise and categorical prefence-statements.

Definition 6.2.1 (Pairwise preference-statement) *Let $D_i, D_j \in \mathcal{D}$ be a document pair (i, j) and $T \in \mathcal{T}$ a given topic. If a user $U \in \mathcal{U}$ considers document D_i preferred over D_j, we add a T-specific preference-statement $R(D_i, D_j|T) := D_i > D_j$ to the collection of all pairwise preference statements $\mathcal{R}_U$ made by the user.*

Definition 6.2.2 (Category preference-statement) *Let $F_1 < F_2 < \cdots < F_M$, where $F_i \subset \mathcal{D}$ for all $i = 1, \ldots, M$, be an ordered set of preference categories*

of documents. The set comparison $F_i < F_j$ is interpreted as a collection of pairwise statements, i.e. for every $D_k \in F_i$ it holds that $D_k < D_n$ for all $D_n \in F_j$.

Although category statements are probably less accurate than direct pairwise comparisons, we think that they have the advantage of being more user-friendly. Simply saying that a document is interesting or not interesting is usually easier than comparing two documents side-by-side and attempting to decide which one is more interesting. Furthermore, category statements should be favored due to their efficiency since each category statement generates several pairwise comparisons. Consider, for example, a three preference category system "low-medium-high"; see Figure 4. If the "low"-category contains 10 documents and the user assigns a newly presented document to "high"-category, we translate the statement into 10 pairwise statements where the new document is preferred to the previously seen 10 documents.

Figure 4: Preference extraction

In the application, we assume that the preference statements are cumulated over iterations, which allows the system to update user profile when new preference-statements arrive. If conflicting statements arrive, we replace the older statements in favor of the new ones. Once the fully updated collection of pairwise preference-statements $\mathcal{R}_U$ is available, the remaining task is to convert the preference statements into topic-specific concept weight vector $w_{U,T}$. For this purpose, we construct an optimization problem to estimate a document-value function which satisfies the given preference structure. The topic-specific concept weights $w_{U,T}$ are then obtained as parameters of the value function. Previously, similar technique has been used for linear utility functions by Zionts and Wallenius [41] and Roy et al. [26].

The value function optimization problem (VFOP) is defined as follows:

Optimization problem 6.2.3 (VFOP) *Let $V : \mathcal{D} \times \mathcal{T} \rightarrow \mathbb{R}$ denote a linear topic-specific document-value function,*

$$V(D,T) = \sum_{c \in T} w_D(c) w_{U,T}(c),$$

where $w_D(c)$ gives the concept weight in $P_{\mathcal{D}}(D)$ and $w_{U,T}(c)$ denotes the corresponding weight in user profile $P_{\mathcal{U},\mathcal{T}}(U,T)$.

Assuming that the user has submitted a non-empty collection of preference statements $\mathcal{R}_U$, the optimization problem for estimating the parameters $(w_{U,T})$ is given by

$$\begin{array}{ll} \textit{Maximize} & \varepsilon, \\ \textit{subject to} & V \quad \textit{is non-negative at every document } D_i, \\ & V(D_j,T) - V(D_i,T) \geq \varepsilon, \quad \textit{for all } R(D_j,D_i|T) \in \mathcal{R}_U. \end{array}$$

The last constraint ensures that V is consistent with the preference structure supplied by the user. Non-negativity of the value function is ensured by having $w_{U,T}(c) \in [0,1]$ for all $c \in T$.

Solving the above optimization problem gives the topic-specific user profile $P_{\mathcal{U},\mathcal{T}}(U,T) = (c, w_{U,T}(c))_{c \in T}$, where the value function parameters correspond to the user's weights for different concepts. The obtained profile is consistent with the preference structure if ε is strictly positive.

Recently, Deb et al. [9] have pointed out that the value function optimization problem generalizes well beyound linear value functions. In the present paper, we consider only linear value functions, which are fast to estimate and easy to interpret. This is also useful from collaborative perspective, because it makes user profiles comparable. In particular, we can directly compare the preferences of users who are interested in similar topics. However, a significant problem with linearity is that it implies mutual preferential independence of concepts, which is unrealistic in practice. The problem could be alleviated, for example, by including cross-product terms into the value function as discussed by Roy et al. [26]. Yet, such approach might turn out to be problematic when considering high dimensional concept-weigth vectors with large number of potential cross-product term candidates.

Therefore, instead of allowing for interaction terms in the value function, we have tried to account for such effects by extending both document profiles and topic definitions with closely related concepts. In the case of document profiles, extension of profile means that we include also concepts which are ancestors of the concepts found in the document. For example, if concepts *InvestmentBank* and *CommercialBank* appear in the text, it means that the concept *Bank* has appeared twice. In similar fashion, topic definitions can be expanded by inclusion of semantically narrower concepts as part of a topic definition. If we state that we want to read news on banks, then it implies that news, where either investment banks or commercial banks are mentioned, could be considered as potentially relevant. The use of both document profile and topic expansion generates overlap, which is helpful in content matching.

7 Incremental learning strategy

In this section, we shortly summarize the procedure for learning the user's profile from preference statements. Two separate versions of the profiling algorithm are presented. The first version considers user profiling as an isolated learning problem, in which only the newly requested preference-statements are used as a basis for constructing the user profile. The second version suggests a simple extension where collaborative profiling is considered.

7.1 Single user learning strategy

Of its main parts the structure of the heuristic resembles that of the *actor-critic architecture* of reinforcement learning literature, where it is often referred to as the policy iteration algorithm, see e.g. Sutton et al. [37], Barto et al. [2] and Haykin [15]. There the idea is to alternate between two modes of operation: the role of an actor, which is responsible for updating the user profile in order to improve document retrieval; and the role of a critic, which is responsible for collecting the feedback from the user.

Thus, assuming that the active user is interested in topic T, the user-profiling algorithm described in Algorithm (1) can be understood as an iteration of

(i) an evaluation-step (lines 8-11), in which the currently retrieved document set $\mathcal{D}_t$ is presented to the user, and the user is then asked to state his preferences for the documents, i.e. update the cumulated collection of preference-statements $\mathcal{R}_U$; followed by

(ii) an improvement-step (lines 12-14), in which the current user profile is updated in order to improve the collection of documents retrieved by the system.

The policy used to implement the document retrieval function *Select-Documents-to-Show* is chosen to be ε-greedy. Once we know the updated value function parameters $w_{U,T}$, the collection of documents is ranked in the order implied by the value function. While majority of the documents are chosen among the highest ranking items, there is an ε probability of choosing lower-ranked documents. This allows us to benefit from both exploitation and exploration of search behavior.

The most intensive part of the algorithm is the optimization step where the value function estimator *Solve-VFOP* is called. However, instead of solving the VFOP-problem 6.2.3 from scratch every time the method is envoked, we use the solution obtained from a previous iteration to provide the solver with a warm start. The use of warm starts works rather well as long as the preference-statements supplied by the user are not in conflict with an existing set of preferences from the previous iterations. When conflicting statements arrive, the current collection of statements $\mathcal{R}_U$ needs to be modified by systematically eliminating older statements, which are inconsistent with the most recent one.

Algorithm 1: User-profiling heuristic

Input: Document collection $\mathcal{D}$, topic definition T
Output: Concept-weights in user profile $w_{U,T}$

```
1  begin
2      t ← 0;
3      continue←true;
4      D_t ←Create-Initial-DocumentSet (D,T);
5      w_{U,T} ←null;
6      R_U ← ∅;
7      while continue do
8          Present-Documents (D_t);
9          continue ← Termination-Check;
10         if continue then
11             R_U ←Request-Preference-Update (R_U);
12             w_{U,T} ←Solve-VFOP (D,R_U, w_{U,T});
13             t ← t + 1;
14             D_t ←Select-Documents-to-Show (D, T, w_{U,T});
15     return w_{U,T};
16 end
```

7.2 Collaborative profile initialization

When a large database of users is available, we can use the prior information about similar topic profiles to reduce the number of preference request calls when new users arrive or new topics are created. In this section, we propose a simple extension to the above profiling algorithm, where the final user profile is given as a dynamically weighted average of the profile learned using the isolated profiling algorithm 1 and a collaborative component.

7.2.1 Consensus profiles and topics

We begin by assuming that the profile database has been clustered to reduce the size of the profiling problem. Let $\mathcal{U}_{\text{db}}$ and $\mathcal{T}_{\text{db}}$ be the set of users and topics saved to the application database. Let $P_{\text{db}} = \{P_{\mathcal{U},\mathcal{T}}(U,T) \mid U \in \mathcal{U}_{\text{db}}, T \in \mathcal{T}_{\text{db}}\} = \bigcup_{U\in\mathcal{U}_{\text{db}}} P_{\mathcal{U}}(U)$ denote the set of topic-specific profiles which are already available in the database. To speed up the profiling process, we commonly apply a clustering algorithm to group similar topic-specific profiles together. The resulting partitioning into K clusters is denoted by $P_{\text{db}} = \bigsqcup_{i=1}^{K} P_{\text{db}}^{(i)}$.

For each cluster $i = 1, \ldots, K$, a consensus profile $\bar{P}_{\text{db}}^{(i)}$ is computed based on the non-conflicting preference statements that are shared by the user profiles within the cluster. Thus, the profile weights are obtained by solving the VFOP problem (6.2.3) with cluster-specific collection of preference-statements. In the

same vein, we define a concensus topic $\bar{T}_{\text{db}}^{(i)}$ corresponding to each cluster by setting

$$\bar{T}_{\text{db}}^{(i)} = \{c \in \mathcal{C} \mid w_{i,c} \geq \alpha, \bar{P}_{\text{db}}^{(i)} = (c, w_{i,c})_{c \in \mathcal{C}}\} \tag{8}$$

where $\alpha \in [0, 1]$ is a threshold parameter controlling the minimum weight of a concept allowed to enter the concensus topic vector. Thus, from this point onwards we will work with the reduced description of the profile database $P_{\text{db}} \approx \{\bar{P}_{\text{db}}^{(1)}, \ldots, \bar{P}_{\text{db}}^{(K)}\}$.

7.2.2 Collaborative profile initialization

Let U denote an active user and T denote an active topic. To initialize the user profile, we need to find the consensus profile (cluster) which is closest to the active topic T, i.e.

$$k = \underset{i \in \{1,\ldots,K\}}{\text{argmax}} \ \text{rel}(T, \bar{T}_{db}^{(i)})$$

where relatedness measure is defined by

$$\text{rel}(T, \bar{T}_{db}^{(i)}) = \frac{1}{|\bar{T}_{db}^{(i)}|} \sum_{c \in T} \sum_{c^\star \in \bar{T}_{db}^{(i)}} \text{score}(c, c^\star)$$

with concept-pair scores computed in the spirit of the Normalized Google Distance [8] based on a training corpus $\mathcal{D}_t$:

$$\text{score}(c, c^\star) = \frac{\log(\max(|\mathcal{D}_t(c)|, |\mathcal{D}_t(c^\star)|)) - \log(|\mathcal{D}_t(c) \cap \mathcal{D}_t(c^\star)|)}{\log(|\mathcal{D}_t|) - \log(\min(|\mathcal{D}_t(c)|, |\mathcal{D}_t(c^\star)|))}$$

where $\mathcal{D}_t(c)$ denotes the set of news stories featuring concept c.

Once the closest cluster k is known, the user profile is initialized by augmenting the set of preference statements associated with the cluster consensus profile to the user's initial preference set $\mathcal{R}_U$.

8 Implementation and experiments

To evaluate the performance of the system, we consider two types of tests: (i) document rank-order learning tests (Section 8.3); and (ii) topic-definition based document retrieval tests (Section 8.4). The first test type evaluates the system's ability to learn the correct preference-based document ranking as compared to the ranking implied by known value functions. The second test set evaluates precision-recall aspects of topic-specific document retrieval tasks using a corpus with relevance-judgements made by TREC assessors. The weight of the experiment is on the second test type, which allows more general comparison of results. All tests are carried out using the Reuter's RCV1 collection, which is described in Section 8.2.

8.1 System implementation

The system used in the experiment was implemented as a combination of Java and C++ software. The infrastructure is built on the GATE [4] platform supported by The University of Sheffield, which provides a generalized Java architecture for developing text engineering components. The platform comes with a comprehensive collection of integrated and third party NLP tools for preprocessing tasks. As a solver for value-function optimization problem (6.2.3), we have used the simplex algorithm supplied in COIN-OR CLP-package [19], which was integrated to the system through Java Native Interface (JNI). For automated ontology engineering tasks, we used Sesame[7] with MySQL repository. The manual engineering tasks related to BTO-ontology were done in Protégé framework[8].

The setup was deployed on two Linux 64-bit boxes with 4GB of memory. The need for capacity was highest during the disambiguation step of corpus processing, which was done as an offline operation before starting the actual experiment. However, the optimization part itself was expectedly fast and suitable for online use.

8.2 Reuters Collection

As a dataset for system evaluation we consider a subset of the Reuter's RCV1 corpus[9] used in TREC-11 filtering track[10]. The corpus consists of about 800,000 news stories from years 1996-1997. The document set is partitioned into a training set (items dated between 1996-08-20 to 1996-09-30) and a test set (remainder of the collection). The training and test set are further divided into 100 topic-specific subsets, which are augmented with the relevance judgements made by the assessors of TREC-11. Thus, although the data set has been originally intended for testing filtering applications, it is reasonably suitable for evaluating systems that consume explicit user preference feedback such as the batch-adaptive processing approach described in this paper.

Due to the wide range of topics in the corpus, we listed about 40 business related topics out of which 35 were selected for our tests. The topic identifiers used for the study are available on request. The final screening was done to exclude topics with complicated narratives, which could not be handled without defining an explicit query to account for the required constraints[11]. The selected topics correspond to a collection of about 18,000 business news items ranging from economic espionage to trade unions and commodity trading; see Figure 5 for a sample topic definition. Following the rules of TREC-11, the topic profiles were constructed based on the predefined small training sets with binary rele-

[7] http://www.openrdf.org

[8] http://protege.stanford.edu

[9] Reuters corpus volume 1. http://about.reuters.com/researchandstandards/corpus/

[10] TREC 2002 Filtering Track Collections. http://trec.nist.gov/data/

[11] For example, the topic R113 entitled "Ford foreign ventures" was excluded because its narrative imposed a constraint requiring that only intact ventures are relevant and that the entity involved must be foreign, which could hardly be fully accounted through learning only.

```
<top>
<num> Number: R135
<title> WTO trade debates
<desc> Description:
The WTO has had an impact upon world trade. What are the current trade
issues being debated by the WTO?
<narr> Narrative:
Relevant documents will contain information pertaining to an issue
between two or more members of the WTO such as tariff rates imposed by
one entity against others for a specific commodity.
</top>
```

Figure 5: Sample TREC-11 topic definition.

vance statements. The total size of a separate training corpus corresponding to the topics is about 1700 documents[12].

8.3 Experiment 1: Learning document rank-order

We begin the experiment by an initial inspection of how well the system can learn the document ordering implied by the user's value function. However, because the preference statements available in TREC-11 material are binary assignments into relevant vs. irrelevant categories, we cannot use them to evaluate the system's ability to order documents according to their values. Therefore, we opted to use our own test approach for evaluating document ordering based on virtual users with known value-functions. The result comparisons are then based on simple rank-order correlations.

The experiment was done as follows. (1) First, we defined 10 virtual users with a set of topics and value-functions. The weights within value functions were fixed manually according to the selected topic definitions. (2) Next, the document collections corresponding to each topic were arranged according to the true value functions. The ordered collections were classified into high (top 10%), low (last 45%), and medium (remaining 45%) preference categories. (3) Then the user profiles given to the system were initialized by giving one document from the high category and one from the low category as first preference statements. (4) Having initialized the profiles, the actual experiment was carried out by running the profiling algorithm 25 rounds. During each round, the documents were ordered according to the estimated value function. Then, a window of 50 documents was constructed with majority taken from the set of highest ranking documents. The greedy probability for random selection from the remaining document set was $\varepsilon = 10\%$. Given the current document window, the virtual user then assigns one best document and one worst document (which has not been rated before) to an appropriate category according to the true value function.

To evaluate correspondence between the estimated document orderings and

[12]The training data set corresponds to TREC-11 batch training set.

the ordering implied by the true value function, we recorded rank-order correlation statistics for each round. Figure 6 shows the rank-order correlations for the 25 rounds as averaged accross the set of virtual users. For comparison, a corresponding trace is shown for the well-known Rocchio retrieval algorithm. Default parameters for Rocchio are used: the weight of the initial profile is 1, relevant document weight is 0.8, and irrelevant document weight is 0.1. Both algorithms use the same ontology-based document profiles and preference-statements. Total number of documents involved in this experiment was about 7000. Although, rank-correlation is a rather harsh measure in a large document set, we find that this initial illustration shows encouraging results for the optimization based algorithm due to its faster learning ability. It seems that the results produced by Rocchio stabilize faster, whereas the widening of the performance gap suggests that the optimizer algorithm is able to extract further information from also later arrived preference statements.

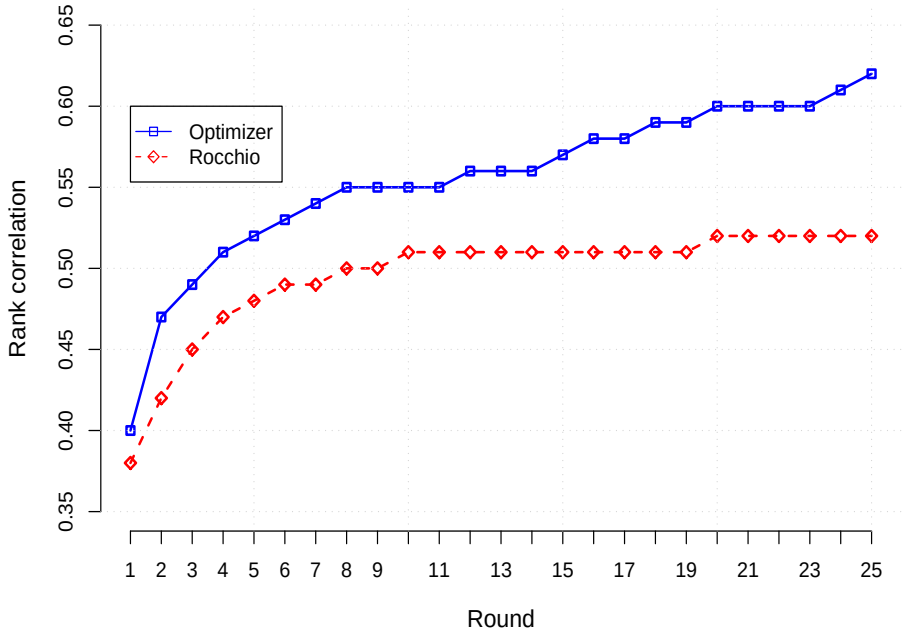

Figure 6: Rank order correlations for Optimizer and Rocchio.

8.4 Experiment 2: TREC business news retrieval

The main test of the system is focused on evaluating the precision and recall aspects of business news retrieval. The system starts with the selected collection of 35 topics and a set of training documents for each topic which have been labelled by assessors of TREC as relevant or irrelevant. The number of training samples ranged from 18 to 80 documents depending on the topic. The task is then to use this information for building a user profile which allows ordering of

the test document sets according to document relevance.

To study the learning rate of the system, we split the topic-specific training material into two subsets. The first random subset, where at least one positive and one negative sample must be found, is used to define a collaboratively initialized user profile. The remainder of the training material is chosen to represent the preference statements given by the active user which are fed to the system in random order with only one preference statement per round. The system was prohibited from using relevance judgements for documents which were not in the topic-specific training material indicated by TREC-11.

The evaluation is based on three measures capturing different performance aspects: (i) Mean average precision (MAP); (ii) Mean scaled utility (T11SU) [24]; and (iii) Precision-Recall (P-R) curves. See Appendix for definitions. In order to summarize results, all reported means are computed as macro-averages accross topics.

8.4.1 Result overview

Figure 7 shows the performance of the system in terms of MAP for the first 10 learning rounds, where each round corresponds to providing the system a new relevance judgement. That is, at the end of the ten rounds the user has given 10 relevance judgements on top of the initial profile. For comparison, we have plotted a corresponding curve for Rocchio's algorithm [25]. The parameters used for Rocchio are the same as in Experiment 1. Both algorithms operated using exactly the same preference information and document profiles. Performance with respect to T11SU utility measure is given in Figure 8.

Comparison of the results suggests at least three main observations. First of all, we find that Optimizer outperforms Rocchio by an average margin greater than 10 percent when measured with MAP. The performance gap is smaller when T11SU utility measure, which considers also recall aspects, is used.

The second observation concerns the shape of learning curves. Recalling that the initial profiles (round 0) are the same for both Optimizer and Rocchio, it is worthwhile to notice that the rounds 1-3 account for majority of the performance gap. This suggests a fundamental difference in the workings of the two algorithms. Whereas Rocchio produces a steadily increasing learning curve, as expected from an algorithm which resembles moving average to a certain degree, the Optimizer shows much more aggressive reaction toward the first received relevance judgements. The same qualitative conclusions apply to T11SU based comparison as well.

The third observation is about the speed of convergence. The profiles produced by Optimizer appear to converge reasonably fast. The figures suggest that already 10 rounds are enough on average to stabilize the learning curve. Assuming that the user is willing to supply only very few preference statements, it is essential for the system to be able to react swiftly, which turns the sensitivity to first preference statements into an advantage of the Optimizer. Figures 9 and 10 illustrate how the sensitivity to preference-statements declines over profiling rounds. The figures show MAP and T11SU records for a single topic

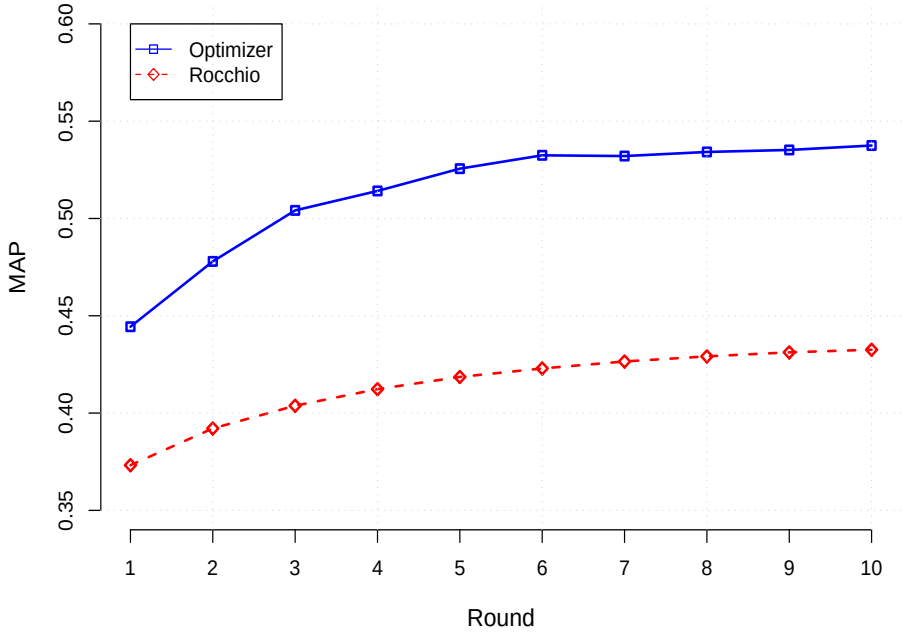

Figure 7: Mean Average Precision (MAP) for Optimizer and Rocchio.

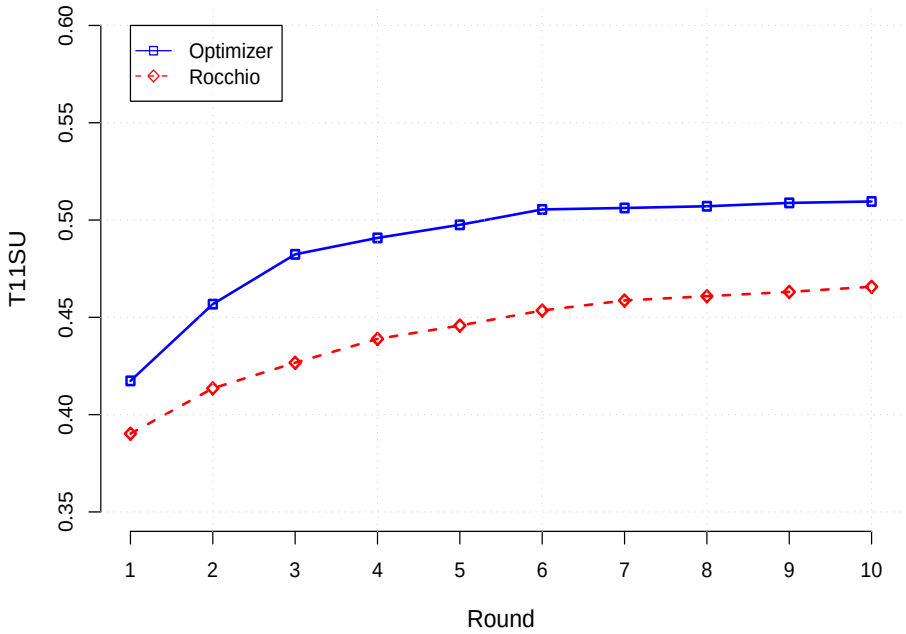

Figure 8: T11SU scaled utility for Optimizer and Rocchio.

profiling task when the preference statements (one for each round) are randomly resampled from the training set for the given topic.

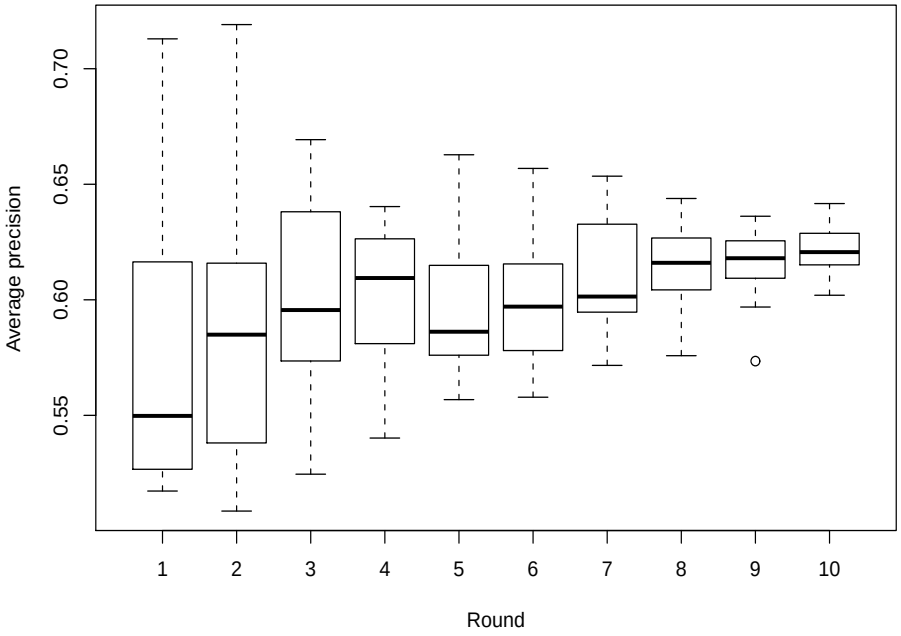

Figure 9: MAP for topic R135 with randomly resampled preference statements.

To evaluate the general performance of the system, Figure 11 shows Precision-Recall (P-R) curves for different learning rounds. The plots are constructed from macro-averages accross topics, which provide more evidence of performance gains achieved during the first few learning rounds. The P-R curves also confirm the fast convergence observed above. The learning effects appear to be highest for the lower levels of recall, which is reflected as more negative slopes of P-R curves.

8.4.2 Learning with BTO and CMO ontologies

The previous experiments were carried out using full ontology with both BTO and CMO components. In this section, we study the contributions of the two ontologies to precision-recall performance. During the development of the system, it became clear that the performance is considerably affected by the ability of the ontology to recognize specific instances of classes / named-entities. For this purpose, we introduced a complementary CMO ontology. Whereas the original BTO ontology was designed to account for the general business concepts, which are relatively time-invariant, we used CMO ontology to describe how more time-sensitive concepts such as companies, managers, products and brands are connected.

To evaluate the effect of using CMO, Figure 12 shows P-R curves for the

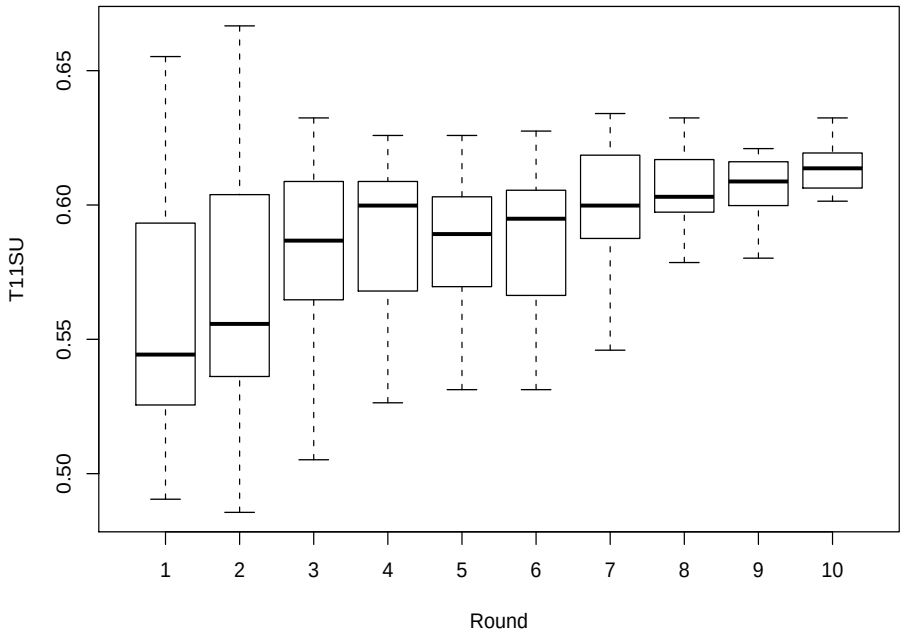

Figure 10: T11SU for topic R135 with randomly resampled preference statements.

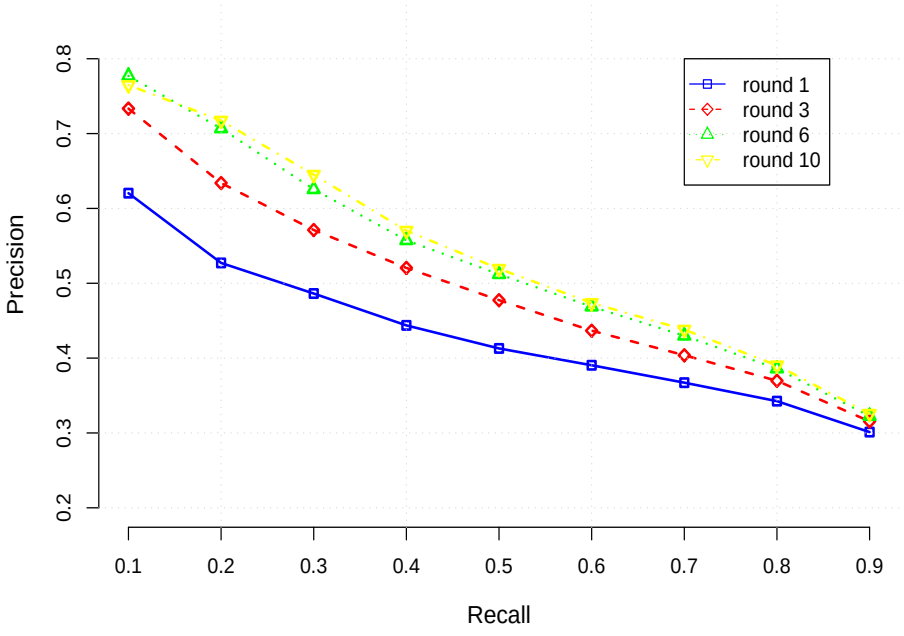

Figure 11: Precision-Recall curves at different learning rounds.

Optimizer with BTO only and the full system (BTO and CMO). We find that

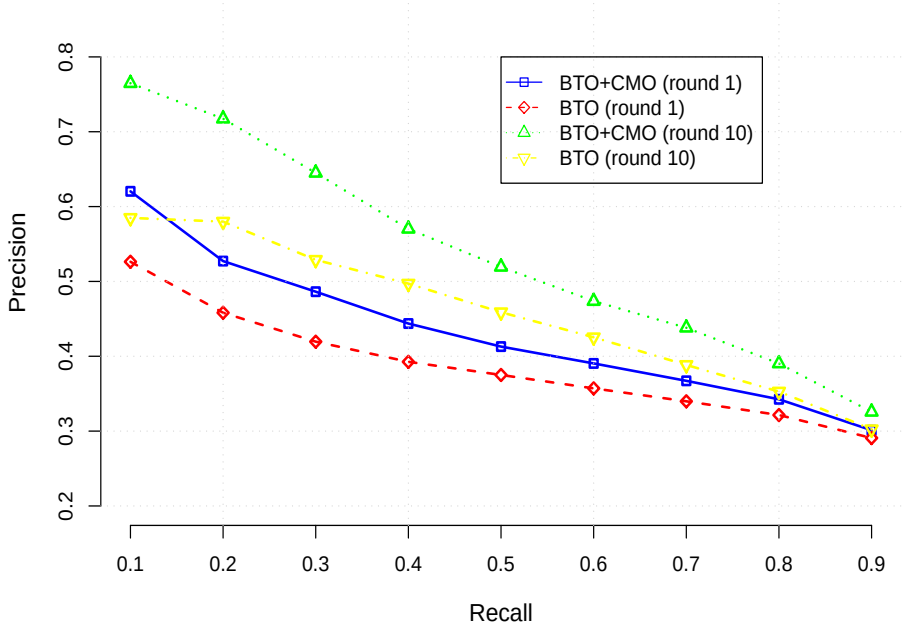

Figure 12: Precision-Recall curves for Optimizer with BTO+CMO ontologies and BTO ontology only. To describe the learning effect the curves are shown for rounds 1 and 10 for both setups.

the use of CMO brings a considerable improvement in precisions for all recall levels. A comparison of the results for rounds 1 and 10 suggests that the use of CMO not only shifts the starting level but also increases the learning effect, which is reflected as a wider gap between round 1 and round 10 curves for the system with CMO.

The improvement brought by CMO can be understood when considering identification of relevant news from a set of conceptually similar documents. It is often the case that two conceptually otherwise similar news items, e.g. company financial reports, can only be distinguished from each other by comparing their named-entities. Therefore, we argue the inclusion of named-entities into a document profile is almost equally important as modelling the actual domain concepts. Furthermore, the fact that performance of the system did not suffer any slowdowns due to considerably increased document profile sizes suggests that the optimizer's performance is relatively robust for even large dimensional spaces.

9 Concluding remarks

In the present paper, we have shown how a combination of ontology-based knowledge modelling and multiobjective-optimization techniques can be used to create document retrieval systems with personalized content. The proposed method differs from traditional keyword-based approaches in at least two respects. First, both content and users are described in terms of uniquely identified concepts. Second, the preference-based profile model allows accurate content retrieval without use of explicit queries. This makes it suitable for mobile applications where the profiling needs to be done using the content recently accessed by the user and a limited number of simple preference-statements.

To evaluate the performance of the proposed technology we have considered a business news reader application using a subset of Reuters RCV1 collection with TREC-11 relevance-judgements as a test-bed. The results from the initial experiments suggest that already the first few preference statements are sufficient to achieve relatively good precision and recall levels. The profiles produced by the system also showed fast convergence. Therefore, we believe that the system is successful in setups where the preference-statements supplied by the user are consistent or when clusters of similar user-topic profiles can be identified.

During the development of the system, several ideas to improve system performance were identified. We are currently investigating how off-domain concepts can be accounted in modified weighting techniques with the aid of well-established folksonomies such as Wikipedia. This is accompanied by development of more accurate disambiguation techniques that can be adapted also for off-domain concepts. Also experiments are done to extend the system such that it is suitable for pure filtering tasks in addition to the batch-adaptive processing considered here. Our forthcoming research suggest that further improvements can be achieved by using explicit queries with ontology-based expansions.

Acknowledgements

This work is supported by a project grant from Emil Aaltonen Foundation. We are also grateful for the research grant from Jenny and Antti Wihuri Foundation.

Appendix: Performance measures

(i) Precision at rank n:

$$P@n = \frac{\#\text{ of relevant items in top } n \text{ results}}{n},$$

which measures the relevance of the top n results of the retrieved news items list with respect to the given topic.

(ii) Average precision:

$$\text{AP} = \frac{\sum_{n=1}^{N} P@n \cdot \text{rel}(n)}{R}$$

where R is the total number of relevant documents and rel$(\cdot)$ is a binomial indicator for relevant document ranks.

(iii) T11SU utility described in the report for TREC-11 results [24]:

$$\text{T11SU} = \frac{\max(\text{T11NU,MinNU}) - \text{MinNU}}{1 - \text{MinNU}},$$

where the lower limit of negative normalised utility is MinNU $= -0.5$, and normalised utility T11NU is given by T11NU $= (2R^+ - N^+)/(2R)$, where R^+ is the number of relevant documents retrieved and N^+ is the number of non-relevant documents retrieved.

References

[1] S.M. BaalaMithra and S.M. SominMithraa. Context-aware automatic query refinement using indian-logic based ontology. In *Third International Conference on Advances in Semantic Processing*. IEEE, 2009.

[2] A.G. Barto, R.S. Sutton, and C.W. Anderson. Neuronlike adaptive elements that can solve difficult learning control problems. *IEEE Transactions on Systems, Man, and Cybernetics*, 13:834–846, 1983.

[3] O. Bodenreider. The unified medical language system (UMLS): integrating biomedical terminology. *Nucleic Acids Research*, 32:267–270, 2004.

[4] K. Bontcheva, V. Tablan, D. Maynard, and H. Cunningham. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10:349—373, 2004.

[5] P. Borlund. The concept of relevance in IR. *Journal of American Society for Information Science and Technology*, 54:913–925, 2003.

[6] J. Chen and J. Chang. A concept-based adaptive approach to word sense disambiguation. Technical report, National Tsing Hua University, 1998.

[7] M-Y. Chen, H-C. Chu, and Y-M. Chen. Developing a semantic-enable information retrieval mechanism. *Expert Systems with Applications*, 37:322–340, 2010.

[8] R. Cilibrasi and P. Vitanyi. The google similarity distance. *IEEE Transactions on Knowledge and Data Engineering*, 19:370–383, 2007.

[9] K. Deb, A. Sinha, P. Korhonen, and J. Wallenius. An interactive evolutionary multi-objective optimization method based on progressively approximated value functions. Technical report, Indian Institute of Technology Kanpur, 2009.

[10] S. Dingare, M. Nissim, J. Finkel, C. Grover, and C.D. Manning. A system for identifying named entities in biomedical text: How results from two evaluations reflect on both the system and the evaluations. *Comparative and Functional Genomics*, 6:77–85, 2004.

[11] Chute et al. The diverse and exploding digital universe. An IDC White Paper, 2008.

[12] W. Fang, Z. Cui, and P. Zhao. Ontology-based focused crawling of deep web sources. In *Knowledge Science, Engineering and Management*, pages 514–519. Springer, 2007.

[13] C. Fellbaum. *WordNet: An Electronic Lexical Database*. MIT Press, 1998.

[14] J. Finkel, T. Grenader, and C.D. Manning. Incorporating non-local information into information extraction systems by gibbs sampling. In *Proceedings of the 43nd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 363–370, 2005.

[15] S. Haykin. *Neural Networks and Learning Machines*. Pearson Education, 3 edition, 2008.

[16] J. Jung. Ontology-based context synchronization for ad hoc social collaborations. *Knowledge-Based Systems*, 21:573–580, 2008.

[17] P. Korhonen and J. Laakso. A visual interactive method for solving the multiple criteria problem. *European Journal of Operational Research*, 34:152–159, 1986.

[18] L. Leslie, T-S. Chua, and R. Jain. Annotation of paintings with high-level semantic concepts using transductive inference and ontology-based concept disambiguation. In *MM*, 2007.

[19] R. Lougee-Heimer. The Common Optimization INterface for Operations Research. *IBM Journal of Research and Development*, 47:57–66, 2003.

[20] A.K. Maddulapalli, S. Azarm, and A. Boyars. Sensitivity analysis for product design selection with an implicit value function. *European Journal of Operational Research*, 180:1245–1259, 2007.

[21] M. Mauldin, J. Carbonell, and R. Thomason. Beyond the keyword barrier: Knowledge-based information retrieval. *Information Services and Use*, 7:103–117, 1987.

[22] H.O. Nyongesa and S. Maleki-dizaji. User modelling using evolutionary interactive reinforcement learning. *Information Retrieval*, 9:343–355, 2006.

[23] R. Rajapakse and M. Denham. Text retrieval with more realistic concept matching and reinforcement learning. *Information Processing and Management*, 42:1260–1275, 2006.

[24] S. Robertson and I. Soboroff. The TREC 2002 filtering track report. In *Proceedings of the 11th Text REtrieval Conference (TREC-11)*, 2002.

[25] J.J. Rocchio. Relevance feedback in information retrieval. In *The SMART Retrieval System*, pages 313–323. Prentice Hall, 1971.

[26] A. Roy, P. Mackin, J. Wallenius, J. Corner, M. Keith, G. Schmick, and H. Arora. An interactive search method based on user preferences. *Decision Analysis*, 5:203–229, 2009.

[27] A. Roy and J. Wallenius. Nonlinear and unconstrained multiple-objective optimization. *Naval Research Logistics*, 38:623–635, 1991.

[28] S. Russell and P. Norvig. *Artificial Intelligence: a Modern Approach.* Prentice Hall, 2002.

[29] G. Salton and C. Buckley. Improving retrieval performance by relevance feedback. *Journal of the American Society for Information Science*, 41:288–297, 1990.

[30] L. Shue, C. Chen, and W. Shiue. The development of an ontology-based expert system for corporate financial rating. *Expert Systems with Applications*, 36:2130–2142, 2009.

[31] A. Simón-Cuevas, L. Ceccaroni, A. Suádrez-Rodriguez, and M. Campos. A concept sense disambiguation algorithm for concept maps. In *Proc. of the Third Int. Conference on Concept Mapping*, 2008.

[32] S. Staab and R. Studer. *Handbook on Ontologies.* Springer, 2004.

[33] N. Stojanovic. An approach for ontology-enhanced query refinement in information portals. In *International Conference on Tools with Artificial Intelligence*, pages 346–357. IEEE, November 2004.

[34] F. Suchanek. *Automated Construction and Growth of a Large Ontology.* PhD thesis, Saarland University, Saarbrücken, Germany, 2009.

[35] F. Suchanek, G. Kasneci, and G. Weikum. Yago: A large ontology from wikipedia and wordnet. *Elsevier Journal of Web Semantics*, 2008.

[36] M. Sulaiman, Y. Al Murtadha, and A. Mustapha. Improved reinforcement-based profile learning for document filtering. *IJCSNS International Journal of Computer Science and Network Security*, pages 133–139, 2008.

[37] R.S. Sutton and A.G. Barto. *Reinforcement Learning: An Introduction.* MIT Press, Cambridge, MA, 1998.

[38] N. Taghipour, A. Kardan, and S.S. Ghidary. Usage-based web recommendations: a reinforcement learning approach. In *RecSys '07: Proceedings of the 2007 ACM conference on Recommender systems*, pages 113–120, New York, NY, USA, 2007. ACM.

[39] S. Wang, Z. Zhe, Y. Kang, H. Wang, and X. Chen. An ontology for causal relationships between news and financial instruments. *Expert Systems with Applications*, 35:569–580, 2008.

[40] B-T. Zhang and Y-W. Seo. Personalized web-document filtering using reinforcement learning. *Applied Artificial Intelligence*, 15:665–685, 2001.

[41] S. Zionts and J. Wallenius. An interactive programming method for solving the multiple criteria problem. *Management Science*, 22:656–663, 1976.