Using latent topics to enhance search and recommendation in Enterprise Social Software

Konstantinos Christidis a,1, Gregoris Mentzas a,*, Dimitris Apostolou b,2

a National Technical University of Athens, Zografou, 157 80 Athens, Greece
b Department of Informatics, University of Piraeus, Piraeus 185 34, Greece

ARTICLE INFO

Keywords:
Enterprise Social Software
Search
Recommender systems
Latent topic models
Latent Dirichlet Allocation

ABSTRACT

Enterprise Social Software refers to open and flexible organizational systems and tools which utilize Web 2.0 technologies to stimulate participation through informal interactions. A challenge in Enterprise Social Software is to discover and maintain over time the knowledge structure of topics found relevant to the organization. Knowledge structures, ranging in formality from ontologies to folksonomies, support user activity by enabling users to categorize and retrieve information resources. In this paper we enhance the search and recommendation functionalities of Enterprise Social Software by extending their knowledge structures with the addition of underlying hidden topics which we discover using probabilistic topic models. We employ Latent Dirichlet Allocation in order to elicit hidden topics and use the latter to assess similarities in resource and tag recommendation as well as for the expansion of query results. As an application of our approach we have extended the search and recommendation facilities of an open source Enterprise Social Software system which we have deployed and evaluated in five knowledge-intensive small and medium enterprises.

1. Introduction

Knowledge management (KM) has been recognized as a critical factor for obtaining organizational competitive advantage and has been shown to be a powerful ingredient in the success of organizations (see for example Davenport & Prusak, 1998; Desouza & Evaristo, 2003; Holsapple & Jiming, 2008; Kulkarni, Ravindran, & Freeze, 2007; Nomaka & Takeuchi, 1995). There have been many studies on the relationship between Information and Communication Technologies (ICT) and KM (Edwards, Shaw, & Collier, 2005; Holsapple, 2005; Maier, 2007; Tseng, 2008; Tsui, 2005). Most agree that ICT improve business values as an infrastructure and enabler of KM (see e.g., Alavi & Leidner, 2001; Tanriverdi, 2005). Many organizations have introduced knowledge management systems as a systematic way of applying ICT to KM. ICT have been identified as the third most important critical success factor for the adoption of KM in the SME sector (Wong & Aspinwall, 2005). New advances in ICT support innovations in KM. Enterprise Social Software, which involves using the World Wide Web as a platform for social interactions (McAfee, 2006a, 2006b), is one of the technologies driving a paradigm shift in KM (see e.g., Jung, 2009). Enterprise Social Software provides “an open and flexible environment to stimulate participation through informal interactions and aggregate these interactions into a structure that reflects the collective attitudes, dispositions and knowledge of the participants” (Eid, 2008). Such software provides innovative approaches for discovering, collecting, organizing, managing and distributing information; these approaches employ a number of tools: blogs, wikis, feeds, syndication systems, social bookmarking applications, and discussion forums.

The adoption of Enterprise Social Software leads to the emergence of additional content and information about the organization knowledge resources. This is a consequence of its key differences to contemporary technologies. Contemporary KM technologies typically predefine their employment in specific business situations (i.e., during after action reviews) and objectify knowledge (i.e., knowledge is seen as an object residing in documents and other artefacts) (Apostolou, Abecker, & Mentzas, 2007). On the contrary, Enterprise Social Software provides open, social platforms that indicate the inherently social character of knowledge, as it is constructed through sense-making episodes, involving various actors in an organization. Enterprise Social Software tools are abstracted from their practical use and are not defining their utilization in a strict and deterministic manner, while their deployment can be eventually emergent according to adapting needs, ideas, organizational practices, etc. (Patrick & Dotsika, 2007). While contemporary
KM systems formulate routine information in a structured manner with specified upfront roles, Enterprise Social Software lets structure emerge, rather than imposing it.

A prominent challenge for knowledge management systems is to discover and utilize useful content within the increased ‘knowledge base’ of the organization. To facilitate this task, categorization of resources with knowledge structures is commonly used. Knowledge structures range from the most simple and least expressive, such as folksonomies, to the most complex and precise ones, such as ontologies (McGuiness, 2003; Smith & Welty, 2001; Uschold & Gruninger, 2004). Folksonomies are bottom-up catalogues of tags. Ontologies are machine-readable specifications of domain knowledge using URLs for all data elements, properties and relationship types. Taxonomies, simple hierarchies of terms, are also used extensively by KM systems. Recently a number of techniques have emerged that can provide insight to human knowledge using unsupervised statistical analysis of data (Blei, Ng, & Jordan, 2003). These techniques can support fundamental information management processes such as searching for content and getting recommendations on relevant content.

In this paper we focus on knowledge structures in Enterprise Social Software that support search and recommendation. We aim to substantiate that the use of new advances in information retrieval technologies, combined with knowledge structures, can enhance the effectiveness of search and recommendation, and in turn can facilitate knowledge work and help improve the effectiveness of Enterprise Social Software for KM. In particular, we investigate approaches that use probabilistic topic models in order to uncover hidden topics in the organizational ‘knowledge base’. We also propose a methodology for applying such approaches to Enterprise Social Software to improve search and recommendation.

The remainder of this paper is organized as follows. The following section contains a short introduction to Enterprise Social Software, how it challenges search and recommender systems as well as an overview of knowledge structures used in search and recommender systems. In Section 3 we describe probabilistic topic models, the family of information processing techniques that we employed for uncovering latent topics in corporate knowledge bases. In Section 4 we provide a detailed description of our proposal that uses latent topics in search and recommendation functionalities. Section 5 contains a description of the case study, a system walkthrough and the results of the evaluation performed. In Section 6 we provide an overview of the related work while Section 7 is dedicated to a discussion on the implications of this work. Conclusions and further work are provided in Section 8.

2. Search and recommendation for knowledge management in the era of Enterprise Social Software

2.1. Enterprise Social Software

The characteristics of the current Web 2.0 are different than that of the original Web. While the original Web was little more than a collection of semi-static pages, accessed by various users independent of each other, the current Web 2.0 is a highly social and interactive platform. This trend has found its counterpart inside organizations where it is called Enterprise 2.0 (McAfee, 2006a, 2006b). Enterprise 2.0 is based largely on fostering internal and external communities using social, participatory Web tools and aligning them with the business needs. Enterprise 2.0 is already a reality in numerous companies around the globe and many others are considering the introduction of tools from the Web 2.0 realm such as weblogs (see e.g., Kaiser, Kansy, Mueller-Seitz, & Ringlstetter, 2009).

McAfee (2006a, 2006b) analyzed the functionalities that emerged from Web 2.0 tools and proposed the SLATES concept to describe the fundamental principles of Enterprise Social Software. SLATES is an acronym for Search, Links, Authoring, Tags, Extensions and Signals. Search, including navigation elements and keyword-based search facilities, plays an essential role in retrieving previous work. Links refer to the ability for a large group of people to forge links within the system from one document to another. Authoring tools support content creation, either for individuals or for groups. This content can be complemented with additional information in the form of Tags. Tags in turn offer the opportunity to browse and retrieve content that was created (blogs, wikis) or introduced (bookmarks, images, etc.) before. Extensions refer to supporting applications such as recommender systems which estimate what users want to find and recommend it to them. Finally, Signals refer to technologies like RSS which makes it possible for users to stay on top of what is going on in the enterprise.

The SLATES principles, sometimes in combination with each other, provide a means to enhance information creation and sharing and collaboration inside organizations (McAfee, 2006a, 2006b). New tools and applications as well as a shift in collaboration culture have led to the creation of massive amounts of information. With Enterprise Social Software, employees and partners generate information either directly or by forging links to the Web. Enterprise Social Software makes a much larger volume of actionable information available within organizations and the users can benefit from this flow of information. However, as its amount can be overwhelming for human processing and evaluation, there is the growing challenge of keeping track of it and making it manageable and useful.

Working with information that is unstructured or qualitatively subjective is another challenge put forward by Enterprise Social Software. Managing unstructured information typically requires a human perspective to identify preferences, quote alternative sources, categorize, filter, and add supporting material or assessments of quality. These actions help other employees consider, consume, or apply this information to their own work. Additionally, advances in information management technologies have paved the way for unsupervised processing of unstructured information, which enables users to search for information and receive valuable recommendations about relevant information.

2.2. Search and recommendations

For KM systems, an overarching goal is to ensure that ‘the right information is available or delivered to the right people at the right time’; both push and pull information delivery strategies can accomplish this. With the emergence of Enterprise Social Software, the challenges pertaining this goal are amplified due to the expansion of available organizational content, especially unstructured one. Two technologies that support the aforementioned strategies are search and recommendation systems.

Search refers to “finding material (usually documents) of an unstructured nature (usually text) that satisfies an information need from within large collections (usually stored in computers)” (Manning, Raghavan, & Schütze, 2008). Search refers to any interface that allows a number of user-provoked information retrieval tasks. In the past fifty years, search has evolved from data search to information search, to syntax search, and lately to semantic search (Yates & Ribeiro-Neto, 1999). A typical search task is that of query-based searching, but search also includes browsing information resources and narrowing them down in the process. As more and more content becomes available, more effort is placed so that algorithms can effectively evaluate the relevance of results
as well as the importance of the source, especially on the web. Lately semantic search has gained ground in the race of combining structured with unstructured information in order to provide extended results to end users.

Recommender systems are addressing, in the most general view, the problem of estimating the utility or the ratings of items that have not yet been seen by the user. To address this problem different types of recommender systems have been researched, content-based or collaborative, utilizing models or heuristics (Adomavicius & Tuzhilin, 2005). Business interest in this area started from early electronic businesses techniques, such as the collaborative filtering recommendations of Amazon.com. In addition to e-business, recommender systems are used by organizational KM systems too, where the problem is redefined as estimating the relevance and usefulness of a previously unseen information resource (e.g., a document, an expert profile, a discussion thread). Recommender systems have become an indispensable part of KM systems (Maier, 2007).

Search and Recommender systems can be seen as two sides of the same coin, utilizing similar technologies. With search, we usually refer to non-personalized content retrieval based on a collection of keywords. With recommender systems, we imply that the user did not ask for help (did not act on the system). The recommender system acts as a search engine in the sense that it looks for things that can be of interest to the user; nevertheless, recommendations usually take into account the user profile, working context, and other aspects that enable intelligent processing and delivery of information.

2.3. Knowledge structures in search and recommender systems

Search and recommendation technologies are not expected to work solely with unstructured information. The usage of knowledge structures provides some amount of categorization and structure. Both the choice of the software platform and the approach adopted by the organization determine the degree of formality in knowledge structures.

Search and recommendations are similar in their underlying technologies. They both involve a query to a content base and employ information retrieval algorithms in order to estimate the relevance of the results to the query. For example, both a query-based search and a content-based recommendation can be seen as a text-classification problem in which text-based content is mapped to classes of information. User-based and other, more complex recommender systems require more specialized event logging and processing. Knowledge structures can enable the processes of both search and recommenders. In recommenders, elements such as concepts or tags can be used for calculating relevance; in search, the existence of a concept hierarchy can enable searching for broader/narrower or similar terms.

Classes of information resources are modeled using various types of knowledge structures. Traditionally, e.g., in contemporary KMSs, the problem of generating domain knowledge structures has been tackled with the use of corporate taxonomies (Gilchrist, 2001), i.e., classification schemes organizing domain entities in hierarchical tree structures. Taxonomies are typically engineered in a top-down approach where the subject area is divided into increasingly narrower and more detailed systematically enumerated categories. However, Enterprise Social Software is based primarily on the use of folksonomies, i.e., collaborative, user-generated metadata that offer an informal way of information categorization. Folksonomies are created bottom-up, in a way where the subject area is divided into individual concepts which can be composed to construct complex subjects via appropriate sets of rules (Dotsika, 2009). Various extensions of the aforementioned basic types of knowledge structures have been proposed in the literature such as ontology-based knowledge maps (Mansingh, Osei-Bryson, & Reichgelt, 2009), hybrid taxonomy–folksonomy classifications (Kiu & Tsui, 2010) and social semantic tag clouds (Kim, Breslin, Kim, & Choi, 2010).

A challenge for corporate taxonomies is to maintain them and keep them up to date. Corporate taxonomies may become easily obsolete due to the rapidly changing conditions in the business environment. On the other hand folksonomies, which are by definition arbitrary and evolving, may not be able to adequately reflect the relevant topics underlying the organizational domain of interest.

Limitations and challenges in the application of knowledge structures combined with the increasing volume of user generated information in Enterprise Social Software can lead to an inferior quality of search and recommendation results (Agichtein, Castillo, Donato, Gionis, & Mishne, 2008). We therefore need new ways to reveal the ever changing knowledge of the domain and to map it to the knowledge structures.

Our motivation for the work presented in this paper is to improve the search and recommendation functionalities of Enterprise Social Software. To this end, we utilize state of the art statistical methods for supporting search and recommendation that work in tandem with both folksonomy- and taxonomy-based knowledge structures. We build a topic model of the knowledge present in the platform and utilize it in order to improve search, provide recommendations for resources and tags, and thus support the generation of more suitable knowledge structures.

3. Probabilistic topic models

In order to improve the search and recommendation functionalities of Enterprise Social Software, we focus on extending prominent knowledge structuring approaches, such as folksonomies and taxonomies with the addition of latent topics. We use probabilistic topic models as a technical background for uncovering these latent topics.

Probabilistic topic models are based upon the idea that documents are mixtures of topics, where a topic is defined as a probability distribution over words. Statistical methods can be used in order to discover a model that describes the way by which documents can be generated (Steyvers & Griffiths, 2007). Specifically, we use Latent Dirichlet Allocation (Blei et al., 2003) to elicit latent topics and use them to identify similarities; similarities are then used for information resource recommendation and for the expansion of query results.

3.1. Latent semantic analysis

Contemporary algorithms in Information Retrieval use the terms found in documents as well as their respective frequencies in order to perform analysis. An example of such an algorithm is TFIDF (Salton, Wong, & Yang, 1975), which presumes a vector space model and calculates the term frequencies and the inverse document frequencies. Latent Semantic Analysis (LSA), first applied at Bell Laboratories in the late 1980s (Deerwester et al., 1988), is a method for correlating semantically related terms in a collection of text. The method, also called latent semantic indexing, uncovers the underlying latent semantic structure in the usage of words in a body of text and proposes how to extract the meaning of the text in response to user queries. Queries against a set of documents that have undergone LSA will return results that are conceptually similar in meaning to the search criteria even if the results don’t share a specific word or words with the search criteria.

LSA maps documents and terms to a representation that is called latent semantic space. This representation is using fewer
dimensions than the number of the actual words to describe the
documents, as these dimensions are expected to capture latent
semantic aspects in the corpus. This approach has been proven to
result in more robust word processing than simple term frequency
counts used in Vector Space Models (Hofmann, 1999).

Probabilistic Latent Semantic Analysis (pLSA) is an evolution of
the previous model that incorporated a probabilistic foundation
(Hofmann, 1999). A latent class model is used to express the
word-document co-occurrence. This co-occurrence is modeled as
a mixture of conditionally independent multinomial distributions.

3.2. Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) has been proposed as an
extension of pLSA that includes prior distributions on the
generation of topics and words. Topic mixture weights are not
individually calculated for each document, but are treated as a
k-parameter hidden random variable, where k is the number of
topics. These variables that represent the word-topic and topic-
document distributions have a Dirichlet prior.

LDA is a generative method, based on the assumption that a gen-
erative model can sufficiently describe the document corpus
and this model can be discovered. The model uses latent topic variables
and probabilistic sampling techniques to generate the documents:
documents are produced by sampling topics from a distribution
of topics over documents and sampling words from a distribution
of words over each topic. To fully describe the model these distribu-
tions need to be identified. To this end, LDA is trained and distribu-
tions converge before the model can be used in applications.

Fig. 1 illustrates in plate notation the generative model: z and d
variables identify topics and documents, while \( \theta(d) \) is the distri-
bution over topics for a document d and \( \phi(z) \) is the distribution over
words for a topic z. These distributions can be used to generate
documents in the form of a collection of words (w). D is the num-
ber of documents, T is the number of topics in the corpus and N_T
the topics found in each document. Hyperparameters \( \alpha \) and \( \beta \) iden-
tify the Dirichlet priors of the above multinomial distributions
respectively. These hyperparameters can be changed in order to
control the smoothing of the distributions.

Instead of directly estimating the two required distributions, \( \theta \)
and \( \phi \), it is advisable to estimate directly the posterior distribu-
tion over z given d, that is, assignment of word tokens to topics. A Gibbs
sampling, which is a special case of a Monte Carlo Markov Chain, is used
for this approximation of \( p(z \mid d) \), which subsequently is used to esti-
mate \( \phi \) and \( \theta \). Iterative evaluation of (1), after a burn-in period,
leads to a sampling convergence to an estimate of z. Then using
(2) the topic-word and document-topic distributions can be calcu-
lated. \( C_{\text{WT}} \) and \( C_{\text{DT}} \) are matrices of counts: \( C_{\text{WT}} \) contains the number
of times word \( w \) is assigned to topic \( j \), not including the current in-
stance \( i \), and \( C_{\text{DT}} \) contains the number of times topic \( j \) is assigned to
some word token in document \( d \), not including the current in-
stance \( i \).

\[
P(z_i = j \mid z_{-i}, w_i, d_{-i}) \propto \frac{C_{\text{WT}}^{wij} + \beta}{\sum_{w=1}^{W} C_{\text{WT}}^{wij} + W \beta} \frac{C_{\text{DT}}^{dij} + \alpha}{\sum_{j=1}^{J} C_{\text{DT}}^{dij} + T \alpha}
\]

(1)

\[
\phi_{ij}^{(t)} = \frac{C_{\text{WT}}^{wij} + \beta}{\sum_{w=1}^{W} C_{\text{WT}}^{wij} + W \beta} \frac{C_{\text{DT}}^{dij} + \alpha}{\sum_{j=1}^{J} C_{\text{DT}}^{dij} + T \alpha}
\]

(2)

LDA exhibits qualities and drawbacks. One of the main advanta-
ges of the method is that it is naturally generalized to new doc-
ments. After the topics have been trained, it is possible to infer the
distribution that could have generated a new, previously un-
seen, item. Additionally the parameters needed are not growing
with the size of the training corpus. The topics generated by this
method are not epistemologically claimed to be more than latent
multinomial variables, nevertheless are capturing probability dis-
btribution of words based on the co-occurrence. Moreover, these
distributions are exchangeable, i.e. after the document re-training
no assumptions can be made to relate topics from the previous
with topics from the current model.

4. Using LDA to enhance search and recommendation

Our approach focuses on combining structured taxonomies and
folksonomies with latent topics (Fig. 2). By enabling periodic pro-
cessing of resources and unsupervised discovering of latent topics,
discovered topics are used as a basis for enhancing search and rec-
ommendation functionalities (Christidis & Mentzas, 2010).

4.1. Search

Searching within an organizational information repository is a
demanding application. A major challenge is that the words used
in the queries do not necessarily appear in the content deemed rel-
levant by the user. Proposals to address this problem include
semantic search and query expansion techniques which try to
make use of synonyms, idiosyncratic terms or semantically similar
terms in order to be able to retrieve results that are relevant but do
not contain the exact words used in the query (see Mangold, 2007).
Such approaches require the existence of an explicit knowledge
structure to assist in resolving term similarities.

Probabilistic topic models can be used as an unsupervised tech-
nique for document categorization, and then as a basis for query
expansion. In this approach issues like word relations, synonyms
and word ambiguity are addressed by statistically discovering rela-
tions between words. There is no need for maintaining an explicit
knowledge structure; this is derived implicitly by the topic model.
A challenge here is that repeatedly calculating probabilistic rela-
tions in every query can be inefficient in terms of processing
power.

In this work, similarities derived by the probabilistic topics are
stored in a word index in order to improve the speed of the query
expansion. Our approach implements a thesaurus-based solution

![Fig. 1. LDA plate notation (Blei et al., 2003).](image-url)
in which words found together in a latent topic are considered similar and are stored together with their similarity measure, after a threshold and pruning process is followed. This way, when the user executes a query including a number of terms, her query is expanded by some of the highly related terms.

To compute each document’s score we use (3) as described in Park and Ramamohanarao (2009)

\[ S_d(Q) = \mu S_d(E) + (1 - \mu) S_d(Q) \]  

(3)

In (3), \( Q \) is the query terms and \( E \) is the expansion based on the topic model thesaurus. \( S_d \) is the document scoring results as received by the underlying social software and \( \mu \) (0 < \( \mu \) < 1) is the mixing parameter.

**4.2. Resource recommendation**

LDA, when trained, produces two probabilistic distributions: topics over words and documents over topics which are approximated as illustrated in Section 3. The documents over topics distributions are used as similarity measures for relevant resource suggestion. The resources that were in the initial training corpus are already described in topics. In the case of a new, previously unseen resource, the model infers which of the latent topics are related to the resource.

To describe the resources, the distribution over topics is used in the form of a vector. This vector has as many dimensions as the topics found in the corpus. When a new resource is added to the system, its topic distribution is inferred based on the words it contains. To calculate similarity between resources, we calculate their cosine similarity using as a vector the topic distribution (Table 1).

**4.3. Tag recommendation**

Tagging is the assignment of keywords to a resource. Users that want to assign metadata to a resource can either enter their own keywords, in the case of folksonomies, select them from a pre-existing knowledge structure. Enterprise Social Software typically allows for free tagging, which leads to the formation of folksonomies. Still, many knowledge management systems make use some kind of taxonomy or catalogue of terms which should not be overlooked. Process and analysis of folksonomies is a field that has been intensely researched as in (Hotho, Jäschke, Schmitz, & Stumme, 2006). Additionally a number of techniques have been examined in order to combine unsupervised probabilistic latent semantic analysis with folksonomies. Specifically, LDA has been evaluated for tag recommendation in folksonomies, outperforming state of knowledge structures and latent topics.

**Fig. 2.** Knowledge structures and latent topics.

\[ \text{Similarity}(A, B) = \cos(\theta_A, \theta_B) = \frac{\theta_A \cdot \theta_B}{\|\theta_A\| \cdot \|\theta_B\|} \]  

(4)

In the example shown in Fig. 3, three resources are already present in the system: a blog post, a bookmark and an article. Their topic distributions have already been calculated and can be seen on the arrows that connect them to the topics. When a new resource is added, its topic distribution is inferred according to the words it contains.

In this example, the resource contains words such as book and finance, and is therefore linked to Topic 2 that relates to education and Topic 3 that relates to economy. The degree of the similarity of a document to a specific topic depends on how many topic words are found in it. To calculate similarity between resources, we calculate their cosine similarity using as a vector the topic distribution (Table 1).
The first tag recommendation technique we use in our approach is based on the similarity measures calculated in resource recommendation. The system infers the related resources from the topic distribution, and then proposes the tags already assigned to highly relevant items. This approach reuses highly used tags (or terms in case of explicit knowledge structures such as taxonomies), therefore reinforcing the model of terms used before in the system. In (5), the vector containing the relevance of each tag to document A is calculated by multiplying the matrix containing the tagging of each document (TD), with the complete document topic distribution in the dataset (H), and in turn with the vector describing the topic distribution of the specific document (A).

$$\text{tagSim}_A = \text{TD} \cdot \Theta \cdot \Omega$$

The second technique is based on the fact that topics that emerge from the analysis are represented by distributions of words. In the context of tag recommendations, these distributions can become the keyword pools. The dominant words of a topic can be used in order to categorize resources that are in some degree generated by this topic. For example, as in Fig. 3, a latent topic that we can call ‘nature’, is modeled by the system in the form of a list: ['green', 'tree', 'flower', 'field', ...], where the words are placed in order of probabilistic importance. If a resource is closely related to this topic, then these dominant words are suggested as possible annotation terms. This technique can work both with folksonomies and taxonomies. This calculation is expressed in (6) where the vector describing the topic distribution of the specific document (A), is multiplied by the word topic distribution matrix in the dataset.

$$\text{tagSim}_A = \Omega_A \cdot \Phi$$

Moreover, these two techniques can be combined to propose the list of tags using a weighted combination approach. This hybrid recommender can both support an existing taxonomy and suggest new keywords based on dominant words in an evolving knowledge base.

5. Case study

5.1. System and deployment

We have tested our approach by integrating it within an open source Enterprise Social Software, the OrganiK system. OrganiK provides a collection of social applications for the corporate environment and supports both taxonomies and folksonomies for annotating information resources (Christidis, Mentzas, & Apostolou, 2011), while it builds on and extends the prominent open source content management system Drupal.

OrganiK has been deployed in and used by five small and medium enterprises including a translation/localization services company, two information technology service companies, a content provider and a maritime consulting company. The companies had an active role in the development, deployment and the eventual use of the OrganiK system. Employees were present in the early discussions on how the system would work. Additionally during the development process, the evolving system was available to users, loosely following the perpetual beta paradigm. After the system reached a stable state, the users started using it in their daily work.

5.2. System walkthrough

All information resources entered in OrganiK are recurrently analyzed and hidden topics are extracted. The topic distributions are saved in the system database, as well as the model than can be used in order to infer topics from a previously unseen document. A screenshot from the maritime consulting installation can be found in Fig. 4. Assume an employee finds a new resource in the World Wide Web that is of interest to the enterprise. He inserts it together with his personal comments in the enterprise collaborative platform in the text area of a blog (Region 1 of Fig. 4). The text content of the resource is analyzed in the background and relations to the existing topics are inferred. The user can then locate related
resources, which are suggested based on the latent topics. Sugges-
tions appear in Region 2 of Fig. 4 and evolve as the user types in
new information. The user can enhance the content of his resource
by adding information from other relevant items or linking to
them.

As the user types in text in the page, tags are suggested (Region
3 of Fig. 4). The recommender is assisting the employee to catego-
rize this resource. Tags are derived from both the corporate taxon-
omy and the latent topic wording. A few minutes later, another
user looks for hazard. This word is not contained in the page that
the previous user wrote, however using latent topics for query
expansion, related articles emerge – including the previous one.
The query is expanded to cover related documents by using closely
related words based on the latent topics.

5.3. Evaluation

We monitored system usage in the five SMEs. The system was
used by a total of 32 users during a period of six months. In this pa-
er we present the evaluation of the system in just one of the five
companies for reasons of brevity.

Users were asked to evaluate the performance of the search and
recommendation functionalities as well as of specific Enterprise
Social Software features. To this end, two types of questionnaires
were used: one containing specific questions about the system fea-
tures to which users provided their responses on a Likert scale; the
second contained open ended questions related to the usefulness
and effectiveness of the system features.

Search was positively evaluated as they were positive about the
system’s ability to locate the resources they were looking for (74% of
the respondents answered positively, i.e. agree or strongly
agree) and retrieve relevant results (83% of the respondents an-
swered positively), see Fig. 5.

Recommendations were in general positively accepted as well
(Fig. 6). Seventy-eight percent of respondents indicated that the
system helped them notice all possible tags for annotating their
documents while 73% found that the suggested tags were suitable
for annotating their documents. With regards to resource recom-
mendations, 78% of respondents indicated that the system helped
them locate all resources they were looking for and 80% found that
the suggested resources were relevant to their needs.

Users were asked to evaluate how the system would affect their
personal and the company’s performance (Fig. 7). On a personal
level, users seemed to be doubtful on how the system could help
their visibility (40% disagreed) or reputation (37.5% disagreed).
However, the results at the company level questions imply that
the users expect a major positive impact if the system is used exten-
sively inside the company. This positive opinion is attributed to in-
creased knowledge sharing and reuse (85%), identification of new
business opportunities (69%) and efficient collaboration (78%).

Employee responses to the open-ended questions were in agree-
ment with the aforementioned results. Users found that the system
could help them be better informed and aware about activities in
their organization concerning their work than before. They, never-
theless, mentioned that “People have to use it regularly, though”.
When asking the employees on the social connection resulting from
the system the employees found it useful but not very important
(since “we are a small organization”) and already have direct means
to connect to each other. A user mentioned: “OrganiK definitely
helps me be more connected with my colleagues but being a small
organization, I think that OrganiK’s main benefits are with knowl-
edge sharing and collaboration rather than social connection”.

The respondents found that the system could help capturing and
organizing the knowledge and expertise in the organization; “Peo-
ple have to be actively encouraged to use it, though”. Finally, con-
cerning work activities, the overall message is that the system is a
very good platform for collaboration. Employees found that espe-
cially in teams, workers “on a regular basis could share far more
information and discuss work in forums, etc.”.

6. Related work

Search and recommender systems in social environments, both
inside and outside the enterprise, have recently been a subject of
intensive research, see a recent survey by Park, Kim, Choi and
Kim (2012). (Dmitriev, Eiron, Fontoura, & Shekita, 2006) have sug-
gested the use of implicit and explicit annotations as user feedback
for improving the enterprise search, while (Amity et al., 2009)
have proposed ways to combine heterogeneous information in or-
der to augment search functionality. (Dugan et al., 2007) propose a
social bookmark recommendation system that takes the form of a
game. Kim, Alkhaldi, Saddik, & Jo (2011) discover relevant and
irrelevant topics for users and use them to enriches an individual
user model with collaboration from other similar users and (Guy,
Zwerdling, Ronen, Carmel, & Uziel, 2010) examine the recomme-
dnent of social media resources in the enterprise based on people,
tags, and their aggregate relationships. Anderson and Mohan
(2011) studied four knowledge-intensive firms and their use of social networking for knowledge management; Guy, Ur, Ronen, Perer, and Jacovi (2011) have designed and evaluated a recommendation system for recommending strangers to employees. Zheng &
Li, (2011) investigate the importance and usefulness of tag and time information when predicting users’ preference and how to exploit such information to build an effective resource-recommendation model in social tagging systems. Furthermore, semantically enhanced ways to search and recommendation have been proposed in the SemSLATES approach of Passant, Laublet, Breslin, and Decker (2009). These approaches are mostly using heterogeneous user generated content, social graph information and semantic technologies. In this work we additionally utilize the results of latent topic analysis in order to capture hidden semantics in content.

Probabilistic topic models have already been used as a way to support recommender systems and search. In Jin, Zhou, and Mobasher (2004), the authors used probabilistic latent semantic analysis for supporting the mining of usage patterns. Chen et al. (2009) and Haruechaiyasak and Damrongrat (2008) applied topic models for recommending resources within communities and articles respectively. In Krestel et al. (2009) and Diaz-Aviles, Georgescu, Stewart, and Nejdl (2010), the authors propose methodologies for suggesting annotations to users while Tsai (2011) used latent topics to mine blogs and recommend relevant tags for given blog posts. In the enterprise environment, Zhao et al. (2010) proposed the use of latent topics in order to identify and visualize latent communities of interest. In Schirru (2010), an approach of using latent topics has been proposed in order to capture interests of the employers and recommend documents. To the best of our knowledge, our approach is the first that not only uses a latent topic model for both search and recommendation in an enterprise environment, but also utilizes it in order to support the generation and refinement of knowledge structures.

7. Discussion

Our system utilizes LDA-based document analysis for query expansion and for recommending resources to users. When an item, e.g., a document, is read by the user, the system either retrieves the topic distribution or, in the case of a new item, infers the underlying topic distribution.

Latent topic detection for content recommendation and search is unsupervised and provides a number of benefits, compared to other supervised and model-based methods. It is not depending on an explicit knowledge structure such as a taxonomy or ontology and does not require effort by the user in order to categorize the resources. Additionally, it can scale to accommodate evolving organizational knowledge bases and is less dependent on the wording of the text than other text-analysis techniques, since words with multiple meanings can be found in different topics.

Tag Recommendation can influence the creation of knowledge structures in the Enterprise Social Software; the recommendation of tags for the model can help the organizational knowledge structure evolve to describe emerging topics. As new documents are inserted into the system, new latent topics are detected in them as the probabilistic topic model is reassessed to match the document corpus. Dominant words in these topics are constantly suggested as keywords for the new related resources, therefore proposing refinements and extensions in the underlying knowledge structure to better describe the discovered topics.

The qualities of the statistical approach put forward in this paper become evident in an enterprise social platform, established on a combination of knowledge structures of varying formality. Probabilistic topic analysis can improve flexibility and stability, as the corporate knowledge structure and the arbitrarily dynamic refinements and extensions are complemented with the latent topics.

Our techniques for search and recommendation which are based on probabilistic topic models are available as open source components, integrated into Organik, which is available as open source.\footnote{http://organik.opendfki.de.} Practitioners can take advantage of them, re-deploy them in other knowledge management systems or further extend them to provide additional functionalities such as recommending experts. Moreover, researchers can investigate how probabilistic topic models can benefit knowledge management systems that make use of other knowledge structures than taxonomies and folksonomies examined in this study. For instance, research is needed to develop other techniques with which probabilistic topic models can work in tandem with ontology-based knowledge structures.

8. Conclusions and further research

In this work we have demonstrated how probabilistic topic models as an integral part of Enterprise Social Software can
enhance recommendation applications and improve the efficiency of search functionality. Our approach addresses problems in query expansion and can recommend relevant resources and tags which in turn can leverage the creation and evolution of knowledge structures including taxonomies and folksonomies. It also provides a sound basis for item-to-item collaborative and content-based recommendations. Our approach does not require significant effort from users as documents evolve to cover diversifying subjects and is easily scalable to a large number of documents.

As future work, we can use alternative probabilistic topic modeling methods, beyond Latent Dirichlet Allocation. These methods have emerged in the last years and can provide insight into additional qualities in the document topics. For instance, Blei and Lafferty (2005) capture correlation between topics while Blei, Griffiths, Jordan, and Tenenbaum (2004) can capture topic hierarchies. Furthermore the processing of user generated content such as text, terms and comments and user behavior remains to be further explored and enhanced. User generated content can take a number of different forms (blogs, articles, bookmarks, and micrologs) and each form can be handled in a different way for topic extraction, see for example, Zhao et al. (2011) for microblog topic analysis.

Acknowledgement

Acknowledgements Research reported in this paper has been partially financed by the European Commission in the OrganIK project (FP7: Research for the Benefit of SMEs, 222225).

References


