Autom Softw Eng (2016) 23:147–190
DOI 10.1007/s10515-014-0146-2

Semantic tagging and linking of software engineering social content

Ebrahim Bagheri · Faezeh Ensan

Received: 5 July 2013 / Accepted: 17 February 2014 / Published online: 12 March 2014

Abstract Social online communities and platforms play a significant role in the activities of software developers either as an integral part of the main activities or through complimentary knowledge and information sharing. As such techniques become more prevalent resulting in a wealth of shared information, the need to effectively organize and sift through the information becomes more important. Top-down approaches such as formal hierarchical directories have shown to lack scalability to be applicable to these circumstanes. Light-weight bottom-up techniques such as community tagging have shown promise for better organizing the available content. However, in more focused communities of practice, such as software engineering and development, community tagging can face some challenges such as *tag explosion, locality of tags* and *interpretation differences*, to name a few. To address these challenges, we propose a semantic tagging approach that benefits from the information available in Wikipedia to semantically ground the tagging process and provide a methodical approach for tagging social software engineering content. We have shown that our approach is able to provide high quality tags for social software engineering content that can be used not only for organizing such content but also for making meaningful and relevant content recommendation to the users both within a local community and also across multiple social online communities. We have empirically validated our approach through four main research questions. The results of our observations show that the proposed approach is quite effective in organizing social software engineering content and

E. Bagheri (✉)
Ryerson University, Toronto, ON, Canada
e-mail: bagheri@ryerson.ca

F. Ensan
Athabasca University, Athabasca, AB, Canada
e-mail: faezeh.ensan@athabascau.ca

making relevant, helpful and novel content recommendations to software developers and users of social software engineering communities.

Keywords Semantic tagging · Q&A websites · Social software engineering · Community interlinking · Web 2.0

1 Introduction

Social media have revolutionized many traditional forms of communication, collaboration and knowledge sharing. Information is now shared at unprecedented rates over the Internet reaching an order of one exabyte per day (Strandberg 2013). Information sharing through means such as blogging, microblogging and social networks form a major part of the information that is exchanged (Stieglitz and Dang-Xuan 2013). Communities of practice, such as the software engineering community, have already become engaged in using social media platforms and reaping the power of the crowds to enrich and facilitate the accomplishment of goals on all *personal, team* and *community* levels. For instance, on the personal level, websites such as Freelancer now provide the means to rapidly outsource a software development project to one or more people around the world. There are close to 8 milion professionals on this website that work individually or collaboratively on projects worth over one billion dollars. On the team level, tools such IBM Jazz (Frost 2007) and Microsoft Codebook (Begel et al. 2010) encourage communication and collaboration between developers within a team by providing means such as feeds, work item tagging and microblogging that are seamlessly integrated within the IDE. Furthermore, on the community level, Q&A websites such as StackOverflow allow developers to engage with each other and help collaboratively resolve issues.

The information that is accumulated at all the three levels results in growing repositories of software engineering knowledge that contain very useful information for the community (Hassan et al. 2010). The degree of engagement of software developers with these social platforms is an indication of their success. For instance, the contributors to StackOverflow submit five new questions and six new answers per minute to this community Q&A website while there are 2,000 visitors on the website every minute that browse through exisiting information. Several researchers have already offered insight into why and how users are motivated to become engaged and to contribute to such communities by analyzing their activities, behavior and social interaction strategies in these social websites (Singer and Schneider 2012; Zhou et al. 2012). Topic and trend analysis of the information on social websites such as StackOverflow has also been of interest, which point to the dynamic evolution of community interests through time and the siginificant amount of content that is available on various topics (Barua et al. 2012; Gómez et al. 2013).

The scale of information available on social information sharing websites such as StackOverflow, TechCrunch and ReadWriteWeb is so large that requires innovative forms of information organization, maintenance and search. Early approaches to information organization on social platforms were limited to top-down categorization schemes such as Yahoo! Directory (Gulli and Signorini 2005) and the Open

Directory Project (ODP) (Gulli and Signorini 2005). However, the major drawback of such schemes was that as the scope and amount of information grew, the hierarchical categorization of information also grew more complex and near to impossible. Alternatively, bottom-up approaches such as community tagging of information has shown to be both better manageable and more scalable. In this approach, the users are free to select a set of keywords that they think best describes their content or artifact be it a blog post, a tweet, a work item or task or a question on a Q&A website. As a result of community tagging, rich schemes have emerged that facilitate the categorization of social knowledge. The strong point of this approach is its reliance on the *wisdom of the crowd* (Kittur et al. 2007). In other words, instead of developing a predefined taxonomy from which the users can select the most appropriate category, the users are empowered to use their own terminology that would gradually accumulate into a user-generated taxonomy, often known as a *folksonomy* (Sinclair and Cardew-Hall 2008). As an example, StackOverflow has enabled its users to tag their questions with keywords they consider appropriate; as a result of which 33,484 tags have been defined and extensively used by the community for describing the posted questions.

1.1 Challenges

While community tagging provides for more scalable and manageable categorization of content, specialized social communities, such as software engineering social websites, face some challenges when deploying such an approach as we outline in the following:

- *Tag explosion* One of the main challenges of using the community tagging approach is that as the diversity of the users grows, the number of tags that are used increases as well. For instance, Barua et al. (2012) have reported that on average 1,097 tags are added to the tags available on StackOverflow every month while only a small fraction of those tags, i.e. 4 %, is employed to describe the majority of the posted questions, i.e. 90 % of the questions. This indicates that in the long-run, the number of tags will become too high to be efficiently manageable and the community will converge towards the use of a small subset of the tags, which could result in the use of generic tags for most available content defeating the main categorization role of the community tags.
- *Interpretation difference* The decision as to which tags are suitable to be used in a given situation is dependent on the semantic interpretation of the content and tags by the users. Many factors such as users' background, expertise and knowledge could impact how the users *semantically interpret* and understand a given tag. For instance, the tag *BT* can represent Bluetooth and/or BitTorrent. Therefore, while one group of users might use BT to tag content related to Bluetooth, others may use it to describe BitTorrent-related content. The difference in interpretation could be even more obscure as to how to interpret the *scope* and *coverage* of a tag. For example, the *iOS* tag could be considered to be relevant only to content that relate to the operating system per se by one group of users, while others might consider content such as mobile app development to be also related. Hence, the differences in the interpretation of the semantics and scope of tags can result in ambiguity and inaccurate organization.

- *Incomplete context* Tags are often assigned by the users once an item is created and posted. These tags are selected based on the understanding of the user of the extent of the post and its related context. However, in many cases, the user who posts some content does not necessarily have the most complete view of the context. For instance, the user reporting an issue on the bug tracking system is not necessarily aware of the full implication of the issue and hence the list of tags that may be assigned could be incomplete. The same applies to Q&A posts; the users who post a question and ask for help in a certain topic do not necessarily know the whole technology landscape to appropriately tag their question. For this reason, the tags that are assigned may not necessarily be the best choices.
- *Locality of tags* Communities often have the inclination to develop their own unique form of expression that is accompanied by the adoption of certain terminology and jargon (Guy and Tonkin 2006). This impacts the type of tags that emerge through the community effort. Therefore, each online social community will gradually develop its own set of community tags that does not necessarily match those developed by other communities for the same purpose. For instance, open-source communities often tag posts related to Microsoft products with *M$*, while other communities might use the *MS* or *Microsoft* tags for this purpose. In light of this, it is often not possible to link the information accumulated in different social websites through community tags due to their locality; hence, preventing seamless integration and connection between related content in different communities.
- *Composite tags* Some users prefer to define and employ tags that are composed of two or more words. For instance, tags such as *javascript-editor* have been used on StackOverflow. These tags are in essence two tags rather than one, i.e., one could use two tags such as *javascript* and *editor* together instead of the one composite tag. While some of these composite tags gain popularity in the community and are used frequently, they can cause problems in terms of matching with other content that has opted for singular tags. Therefore, while the two single tags of *javascript* and *editor* have the same meaning together as the one *javascript-editor* tag but in practice content with each of these could not be matched with each other; impacting the classification power of the tags.
- *Obscure similarity* In order to be able to categorize content, it is first necessary to find similarity between their corresponding tags. The simplest approach for this purpose is to consider the content that have the same set or at least a shared subset of tags as similar. The restriction of this approach is with finding items that do not necessarily have the same set of tags but rather have tags that are conceptually similar. This is quite challenging as there is no explicit information about each individual tag that would allow for the computation of similarity. One of the few ways that similarity between tags can be calculated is through the *co-occurrence of tags*. In other words, if two tags are frequently observed together then one could conclude that the two tags share some similarity. However, it should be noted that co-occurrence does not necessarily imply similarity. For instance, although the two tags *query* and *optimization* are often seen together, there is no meaningful similarity between them. Such an approach to similarity calculation may lead to inaccurate classification of content.

Websites such as StackOverflow have attempted to tackle some of these issues by providing additional mechanisms. The two main mechanisms are (*i*) *grouping of tags:* In order to avoid the *tag explosion* and *obscure similarity* problems, StackOverflow allows its users to identify tags that they consider 'the same' or 'synonyms' and to group them under one unique tag. This way, content that are annotated with any of the tags in one group can be related to each other. While this could alleviate some of the issues, our observation of the groupings shows that not all grouped tags are correct. For instance, groups such as {*PHP, php.ini,PHP-OOP, PHP-Frameworks*}, {*XCode, XCode-IDE*} and {*JQuery, JQuery-color, JQuery-chaining*} show that not all tags that are grouped together are necessarily synonyms or the same, which could further complicate the problem; (*ii*) *Content page and linking to Wikipedia:* The other feature provided by StackOverflow is the provisioning of a unique content page for each tag, which allows the users to describe each tag and provide some information about it on that specific tag page. A link to a Wikipedia can also be added to the tag page. This can address the issue of *interpretation difference*. However, the main issue is that most tags do not have their content page or Wikipedia links provided by the users. Furthermore, for grouped tags, there are sometimes links to multiple different pages on each of the tags that can add to the ambiguity of the interpretation.

1.2 Contributions

In light of the above challenges, we propose an approach that centers around *semantic grounding* of tags, i.e., explicating the semantic and conceptual denotation of each tag within the social platform as opposed to relying on the internal connotation of each tag for the individual users. In order to offer clear semantics for tags, we rely on Wikipedia for the set of tags that the users can select from, i.e., we propose to use the title Wikipedia entries as the set of possible tags that can be used to annotate content on a social software engineering platform such as StackOverflow. There are over four million pages on the English Wikipedia, which cover many different topics; therefore, it would be possible to annotate content on a social website such as StackOverflow with tags which are in effect Wikipedia entries. For instance, instead of allowing the users to tag a question on StackOverflow with just the two letters *BT*, which is ambiguous, the users would instead have the option to add one or more tags in the form of links to Wikipedia pages. The users might find cases where the concepts that they need are not yet available on Wikipedia and therefore they cannot choose a suitable tag. In such cases, a new Wikipedia entry can be created by the user, which not only resolves the issue of lack of tags but also contributes to the expansion of community knowledge on Wikipedia.

Our work in this paper is primarily based on the StackOverflow Q&A website focused on the software engineering community. Our high-level contributions can be enumerated as follows:

1. Statistical analysis techniques are developed that can automatically process content on each StackOverflow post including a question and its related answers and offer suggestions regarding the most appropriate set of semantic tags that can be assigned to that content. Each recommended tag will be in the form of a pair: Wikipedia page title and the link to that page.

2. Semantic similarity measurement methods are introduced that operate over the semantic tags in order to compute the degree of similarity between StackOverflow content such that the most appropriate form of content categorization and organization can be acheived. This also faciliates the process of finding the most relevant content, e.g., highly related questions.
3. Cross community linking is faciliated by relying on the unique shared semantic tag space provided by our approach. This allows content on different social community Websites, e.g. StackOverflow, Reddit, TechCrunch and ReadWriteWeb, to be interlinked and shared, which would have not been easily feasible otherwise mainly due to the issue of *Locality of tags*. The users will be able to have access to most relevant information on other related software engineering social Websites when browsing content on another social Website such as StackOverflow.

The rest of this paper is organized as follows: Sect. 2 describes the overall picture of our proposed approach. The details of the work and the technical minute's are discussed in Sect. 3, which is followed by a discussion on our implementation details and the tooling support for our approach. The main research questions and also the setup and structure of the experiments that were performed to validate the research questions are presented in Sect. 5, which is followed by the explanation of the obtained results in Sect. 6. Before concluding the paper, a discussion including the threats to validity and the lessons learnt, and also a review of the related work are presented. The paper is then concluded in Sect. 9.

2 Approach overview

The main objective of our work in this paper is to address the challenges of community tagging within the social software development/engineering support communities as introduced earlier. We especially focus on enhancing the practice of community tagging of content in social platforms such as StackOverflow by promoting the idea of using semantically rich tags that have clear and unambiguous denotation. To this end, we analyze social software engineering content and offer recommendations on the most suitable semantic tags that can be used to annotate and describe the content. The semantic tags that are recommended to the users can be subsquently used to not only link similar content on the same social platform, but can also facilitate the process of cross community linking. The overview of our approach is shown in Fig. 1, which consists of two processes: (*i*) offline process, and (*ii*) realtime process.

2.1 Offline process

The key idea behind our approach is to exploit encylopedic information from Wikipedia to semantically annotate content on social software engineering community websites such as StackOverflow. Each Wikipedia page is in essence an entry in a large-scale encyclopedia that is collectively maintained and consists of the definition and description of a significant concept. During the tagging process and instead of adding ad hoc tags to their content, users can use Wikipedia page titles as their tags. So for instance, instead of adding the words *query* and *optimization* as tags, a user can add *query_optimization* (http://en.wikipedia.org/wiki/Query_optimization) as the tag for

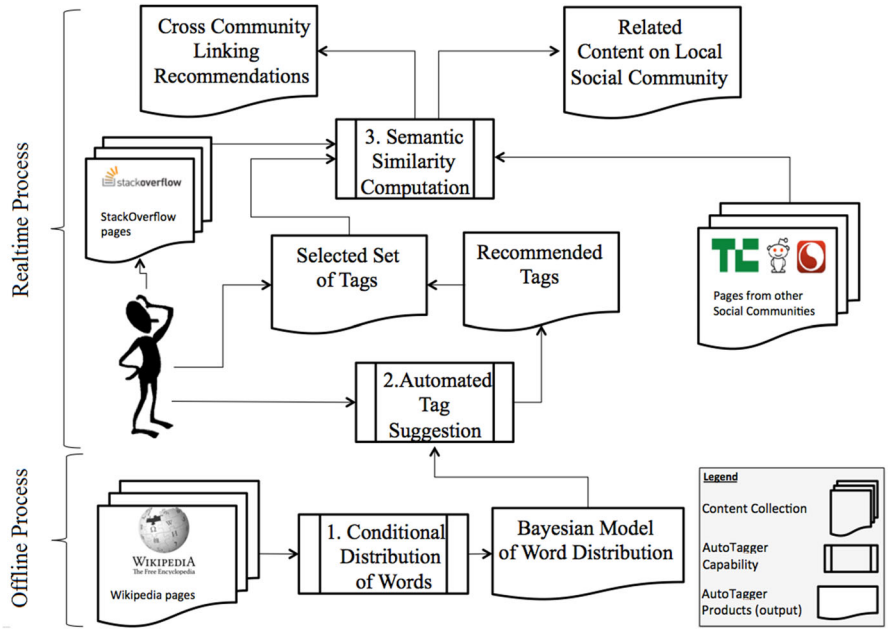

Fig. 1 Overview of our approach

that content. This way and among other benefits, anyone not familiar with the tag can look it up instantaneously.

One of the advantages of traditional community tagging is that users can quickly and inexpensively add several tags to their content without having to check additional sources. However, given the large size of Wikipedia, finding the best set of pages whose titles could serve as suitable tags can be a cumbersome process and could complicate the process of tagging. For this reason, we additionally support the users in finding the most suitable Wikipedia pages that can serve as semantic tags for specific content. To this end, we build a model that is able to contextualize user's content with Wikipedia entries. We build such a model in the offline process of our approach by measuring the frequency of occurence of words within Wikipedia pages. In other words, we analyze Wikipedia content in order to find out for each Wikipedia page, how frequently certain words or phrases occur within the body of that page. Now, with this conditional distribution of words in Wikipedia pages, we are able to predict, given a new post by the user, what are the most related Wikipedia pages to that post. The final outcome of the offline process is a statistical model that can find the most relevant Wikipedia pages to the content that has been posted by a user by looking at the similarity of their word distributions. The titles of these Wikipedia pages are used for tagging purposes.

2.2 Realtime process

The objective of the *realtime process* is to analyze each individual post submitted by a user and find the most suitable semantic tags that can be assigned to it. In addi-

tion, this process also attempts to find related content to a new post both on the same social platform and other community websites. To acheive these goals, the content of each post is automatically analyzed in the context of the statistical model of word distributions in Wikipedia that has been prepared in the offline process. In order to be able to make tag suggestions, the word distribution of the content submitted by the user is compared with the word distribution of the content in different Wikipedia pages. Those Wikipedia pages that have the highest distribution similarity with the user's post will be considered as possible candidates to be used in the tagging process. Once the candidate Wikipedia pages are identified, their titles will be used as the possible semantic tags that will be suggested to the users. It should be noted that in Q&A websites such as StackOverflow, the content of a question alone might not be inclusive of all the topics addressed by the issue in question. For instance, the user asking the question might not be aware of special topics that need to be considered in order to resolve the question; therefore, tags suggested solely based on the question content might not be comprehensive. For this reason, we provide the means for the users to revise their tags once one or more answers to the question have been submitted by other users. As answers are posted, a revised list of suitable tags will be suggested to the user by considering the content of the question and the content of the answers. This way, we can ensure that all aspects of the question and its answers have been considered in the tagging process. In any case, the users have the option to review the set of recommended tags and to revise and finalize a list of tags for their post.

The tags that are assigned to social content as a result of the proposed tag suggestion process have clear semantic grounding. The semantics of the tags are due to their strict connection with Wikipedia pages. Simply stated, each tag used in our proposed community tagging process is a representative of a well defined concept in Wikipedia; therefore, issues of *ambiguity* and *intepretation difference* are avoided. Furthermore, it is possible to compute the degree of similarity between each pair of tags based on the similarity of their corresponding Wikipedia pages. This is specially important in light of the fact that it is not straightforward to compute the semantic similarity of tags in the traditional community tagging approaches and only measures such as frequency of co-occurrence could be calculated that do not necessarily denote similarity. Now, given the fact that the similarity between two tags can be semantically measured through Wikipedia, it is possible to compute the similarity between the content of two posts by computing the similarity between their set of tags. The calculation of similarity between content would allow for providing recommendation of related content that may be of interest to those users who are browsing or reading a given post. A typical scenario would be a situation when other possibly related questions are recommended to users as they are browsing a given question on StackOverflow.

In addition to recommending content from the same local social community, the use of semantic tags, as proposed in our approach, allows for the semantic linking of information from two or more communities. The main reason for this is due to the fact that the tags used in our approach do not suffer from the *locality* issue as they are uniformly taken from Wikipedia. Therefore, content collected from any social source and annotated with tags grounded in Wikipedia semantics can be easily interlinked by measuring the similarity between the content sources based on their semantic tags.

An interesting scenario that we describe in our experiments is to recommend related articles from Reddit while a user is reading a given question on StackOverflow.

3 Approach technicalities

We now present the technical underpinings of each of the parts of our proposed approach. The details of each activity shown in Fig. 1 are given in the following.

3.1 Content preprocessing

Prior to any activity, we clean the textual content that is processed in our work through *text normalization*. To acheive this, we follow the steps proposed by Barua et al. (2012). In the first step, we remove any code snippets that might be present in the content, i.e., we remove any content that is surrounded by `<code><code>` tags. This is in essence due to the fact that source code syntax can introduce noise into our analysis. Then, we remove any HTML tags as they often only provide rendering information and do not necessarily add any semantic value to the content that we are processing. It should be noted that we only remove the HTML tags themselves and not the content embedded in them. For instance, `<p>how to display a<i> thumbnail <i>of an image</p>` is preprocessed in a way that `how to display a thumbnail of an image` will remain as a result. As the next step, we remove any common English language stop words from the textual content. We use the stop word list used by MySQL FullText feature to identify and remove such words. Finally, we perform stemming on the remaining words, through the use of Porter Stemmer in order to convert inflected words into their roots.

3.2 Statistical model of Wikipedia content

The main objective of the offline process of our approach is to have a formal understanding of how words are distributed in Wikipedia pages. This is especially important as it would allow us to reason about the similarity or closeness of user content with Wikipedia pages. As an example, lets assume that a user has submitted a question on StackOverflow regarding MySQL result sets. In order to find the most related Wikipedia pages to this question, one can first enumerate the words that have appeared in the question and try to find Wikipedia pages that contain those words. So in this case, the two http://en.wikipedia.org/wiki/Result_set and http://en.wikipedia.org/wiki/Mysql pages could be the most related given they are highly likely to contain the same words as the user's question.

We build the statistical model of Wikipedia content by analyzing word distributions in Wikipedia pages. In other words, the frequency of occurence of words in Wikipedia pages is employed as a represenation model for those pages. More specifically, we use Term Frequency-Inverse Document Frequency (TF-IDF) (Salton et al. 1975) to show the significance of words in different Wikipedia pages. The reason we employ TF-IDF is that it emphasizes on words that are highly frequent within one specific page and less frequent in other pages; therefore, those words that can be considered central to the theme of each Wikipedia page would receive higher significance in the representation

of that page as compared to other more general words. Thus each Wikipedia page would be mainly represented by those words that have higher significance for that page only and not so much relevance for other pages. Such a strategy for representing Wikipedia pages provides a strong *discriminatory power* for distinguishing between different pages. The discriminatory power of TF-IDF undermines the words that frequently occur in many different Wikipedia pages and instead strengthens the words that are quite unique for one page.

The TF-IDF scheme is composed of two independent statistics: Term Frequency and Inverse Document Frequency. The first statistic, TF, measures how frequently a word has appeared in a given document. The simplest form of TF calculation is to count the number of times a word is seen in a document. However, more complex variations of TF exist such as the *augmented term frequency* (Singhal et al. 1996), which prevents bias towards longer documents. Let us assume that w represents a Wikipedia page and t is a word (term), the TF value of t in w can be computed as follows:

$$TF(t, w) = 0.5 + \frac{0.5 \times f(t, w)}{max\{f(t_i, w) : t_i \in w\}}. \tag{1}$$

where $f(t, w)$ denotes the simple count for word t in Wikipedia page w.

The second statistic, IDF, computes whether a word is frequent or infrequent among the collection of all pages. It is often computed by considering the number of documents that contain a given word over the total of number of available documents. Let us assume that W is the set of all pages in Wikipedia, the value of IDF can be calculated as:

$$IDF(t, W) = \log \frac{|W|}{1 + |w \in W : t \in w|}. \tag{2}$$

The TF-IDF scheme is calculated by multiplying its two constituent statistics:

$$\text{TF-IDF}(t, w, W) = TF(t, w) \times IDF(t, W). \tag{3}$$

Now, it is possible to build a statistical model of word distributions for Wikipedia pages. The model can be formulated as a *term-document matrix*, which describes the weight of words in each Wikipedia page. Assuming that T denotes all possible words accross Wikipedia pages, our representation of the term-document matrix for words in Wikipedia pages is shown by $M_{|T|\times|W|}$ where $M[i, j] = \text{TF-IDF}(t_i, w_j, W)$. Each cell of M represents the TF-IDF value of a word in a given Wikipedia page. The columns of this matrix are the statistical word distributions within a Wikipedia page.

It is important to note that the term-document matrix M is built only once during the offline process and can be repeatedly used during the realtime process without having to be recomputed each time. In our development process we built the term-document matrix using the Wikipedia dumps that are openly accessible through the Wikimedia Foundation.

3.3 Automated tag suggestion

Automated tag suggestion is the first step in the realtime process of our approach and is concerned with recommending suitable tags from Wikipedia to the users based on the content that they have submitted. Our goal is to recommend the most suitable tags for the user's content in such a way that all aspects of the posted content are covered. The main strategy for acheiving this is to compare the distribution of words within user content with word distributions of Wikipedia pages. Those Wikipedia pages that have the closest distribution of words to that of the user content would be considered to be the most similar. The title of these Wikipedia pages will be recommended to the users as potential words or phrases that they can use to tag their content.

In order to be able to find similar Wikipedia pages, the distribution of words in user's content need to be calculated as well. We employ the same strategy introduced in the previous subsection to compute the statistical word distribution within user's content. However, given the calculation of TF-IDF is dependent on a set of documents W and the user's content is only an individual document, we use the same W that was used in the previous subsection, i.e., W will be equivalent to the set of all Wikipedia pages. This way, the IDF statistic will be computed in light of the frequency of words in Wikipedia pages. With this approach and assuming that the user content is denoted as uc, we will be able to build a vector $U_{|T|}$ where $U[i] = \text{TF-IDF}(t_i, uc, W)$. Each cell of the vector U represents the TF-IDF of a word in the user content.

With the vector representation of the user's content, it is possible to compare the word distributions in the user content and different Wikipedia pages. A common approach to compare documents in the *vector space model* is to compute the *cosine similarity* between two vectors (Lee et al. 1997). The cosine similarity measure is able to compute the similarity of two documents by computing the inner product of the vectors of the two documents divided by the product of the vector lengths. The major benefit of this similarity measure is that it avoids the bias caused by different document lengths, which is quite important in our case. Lets assume that uc is the content that has been posted by a user, e.g. a question on StackOverflow, and w_k is a Wikipedia page, then it is possible to compute the degree of similarity between uc and w_k using the cosine similarity as follows:

$$cos(uc, w_k) = \frac{\vec{V}(uc) \cdot \vec{V}(w_k)}{|\vec{V}(uc)||\mathbf{V}(w_k)|}. \tag{4}$$

The vectors required for uc and w_k have already been calculated and are available as U and $M[:, k]$ (the k^{th} column of M), respectively. Hence, the cosine similarity between the two vectors are calculated as:

$$cos(uc, w_k) = \frac{\sum_{i=1}^{N} U[i]M[i,k]}{\sqrt{\sum_{i=1}^{|uc|} U[i]^2}\sqrt{\sum_{i=1}^{|w_k|} M[i,k]^2}}. \tag{5}$$

where N is the number of shared words between w_k and uc.

The cosine similarity measure allows us to compute the similarity of the user's content with Wikipedia pages and to rank order them based on their similarity. The most similar Wikipedia page (w^*) to the user content would be the one that has the highest cosine similarity: $\{w^*|cos(uc, w^*) \geq cos(uc, w_i) : w_i \in |W|\}$. Once the degree of similarity between the user content and Wikipedia pages are computed and Wikipedia pages are rank-ordered based on their similarity to the user content, we consider the title of the *top-k* Wikipedia pages as the suitable tags that can be recommended to the user. The user will receive tag suggestions in the form of a pair where the tag label is the Wikipedia page title and the tag link is the Wikipedia page URI. It will eventually be up to the discretion of the users to decide whether they would like to accept the tag suggestions or to only accept a subset of the suggestions. The users can also additionally include other semantic tags they consider to be relevant that were not a part of the tag suggestions.

It is worth noting that a user content in the context of the StackOverflow website, which we will be mainly focusing on in our experiments, is a question submitted by a user and the subsequent answers that have been posted by other users on that question.

3.4 Semantic similarity computation

One of the significant challenges of community tagging approaches is to measure the similarity between the tags that have been assigned to user content. As mentioned earlier in the paper, the semantics of the tags that are used by the users are not externalized and only reside in the mental model that the users have from the domain. Therefore, while the users would in most cases have a shared understanding of a tag when they see it, the meaning of the tag is not explicitly stated within the social platform. Some social software engineering platforms such as StackOverflow have proposed to add a link to the most relevant Wikipedia page for each tag in order to resolve this issue. However, our observation has shown us that in practice the majority of the tags do not have a corresponding Wikipedia link. For this reason, it is quite difficult to measure the similarity of tags in the traditional tagging model. Simple approaches such as syntactical matches between tags can find words that have minor spelling differences, e.g. *scripting* and *Web script*; however, they are not able to find completely dissimilar tags that have the same meaning such as *programmer* and *software developer*.

One of the proposed approaches for finding similarity between tags, beyond syntactical techniques, is to measure the co-occurrence of tags and identify relationships between tags based on their co-occurrence pattern (Wartena et al. 2009). The idea behind this approach is that posts on social communities have one main coherent topic. For instance, a question on StackOverflow has a main theme and the user submitting that question is interested in finding out about that central theme. Based on this, it could be assumed that tags that are attached to a user content would mainly revolve around the main theme of the user content. Therefore, if certain tags co-occur on many user content, then it would be reasonable to assume that these tags are conceptually related. However, while the reasoning behind this approach is to some extent acceptable, there are many cases where the user content does have a main theme but

the tags address different aspects of that central theme. For instance, a question on StackOverflow that asks about MySQL query optimization can have both *query* and *optimization* as its tags. It would also be the case that both of these two tags frequently occur together on many tags; therefore, based on this similarity measurement approaches both *query* and *optimization* would be considered similar. It is important to note that while *optimization* appears with *query* in many cases, at the same time it carries other meanings if it is paired with other tags. So if this form of similarity is used, the users might find posts recommended to them which are related to *compiler optimization* when browsing through *MySQL query* content. Therefore, while co-occurrence could imply *relatedness*, it does not necessarily indicate *similarity*.

The advantage of grounding tags in Wikipedia content is that it enable us to semantically interpret the tags and to be able to find similarity between tags not only based on their syntactical similarity but also according to the degree of conceptual overlap that they have with each other. Therefore, this approach is able to find tags that are completely similar and have the same meaning such as *programmer* and *developer* and also tags that are not completely the same but share some conceptual similarity such as *iOS* amd *Android*, which are not the same but are both mobile operating systems and hence have some similarity. An ideal scenario would be to able to compute the degree of similarity between two tags enabled through Wikipedia in our approach.

Now, there are three sources of information within Wikipedia content and structure that can be used for building a semantic similarity measure between two Wikipedia articles; hence, between two Wikipedia-based tags. Strube and Ponzetto (2006) have shown that these three sources can form a suitable model for similarity measurement in Wikipedia. The first source of similarity measurement, known as *path-based measures* (Zesch and Gurevych 2007) compute the similarity between two words such as w_1 and w_2 based on their distance on a taxonomical structure. The closer the two words are in the structure the more similar they would be. The challenge is that Wikipedia articles are not formed in taxonomical structure and therefore, Wikipedia articles cannot be directly compared using the path-based approaches. However, each Wikipedia article is related to Wikipedia categories which are structured in the form of a hierarchical taxonomy. This relationship can be used to measure similarity based on the path between the categories of two articles in the Wikipedia category hierarchy. It is possible to compute the similarity of two Wikipedia article titles using this approach by:

$$sim_p(w_1, w_2) = \frac{length(w_1, w_2)}{2D}. \tag{6}$$

where $length(w_1, w_2)$ is the length of the shortest path between the categories of w_1 and w_2 and D is the maximum depth of the Wikipedia category hierarchy.

The second source for measuring similarity is the *information content*-based measures (Ponzetto and Strube 2007). In these approaches, similarity is computed as a the extent to which two words share information with each other. In Resnik's formulation, this is modeled by measuring the *information content* of the least common subsumer of the two words. Let us assume that the least common subsumer of w_1 and w_2 is the node n in the Wikipedia category hierarchy, then according to the information content-based model, the similarity of w_1 and w_2 can be computed as:

$$sim_{ic}(w_1, w_2) = \frac{\log(hyponym(n) + 1)}{\log(C)}. \quad (7)$$

where $hyponym(n)$ is the number of hyponyms for node n and C is the number of nodes in the category hierarchy.

The last source of comparison is to use *text overlap*-based measures (Gabrilovich and Markovitch 2007), which compute similarity based on the number of shared terms between the content of two Wikipedia pages. In other words, the more common words the two Wikipedia pages have, the more similar they would be. The similarity of two words w_1 and w_2, each representing a Wikipedia article, according to the text overlap approach can be computed as:

$$sim_{to}(w_1, w_2) = \tanh\left(\frac{overlap(w_1, w_2)}{length(w_1) + length(w_2)}\right). \quad (8)$$

where $overlap(w_1, w_2)$ is the number of shared terms between two Wikipedia articles, and $length()$ counts the number of terms in an article. the hyperbolic tangent function is employed to avoid the skewing effect of outliers.

We employ these three similarity sources to compute the similarity of two tags as described in the wikirelate paper (Strube and Ponzetto 2006). The similarity of two tags would be the length of a three-dimensional vector, whose dimensions represent one of the above similarity measures. Therefore, the similarity of two tags w_1 and w_2 would be the length of a vector represented by $\big(sim_p(w_1, w_2), sim_{ic}(w_1, w_2), sim_{to}(w_1, w_2)\big)$.

3.5 Content recommendation

One of the major objectives of community tagging is to enable content interlinking and organization. Content that are annotated with similar tags are often placed under the same category and are related with each other on social Websites. For instance, StackOverflow provides a list of all questions that have a particular tag in their tag list to the users. In our approach, we benefit from the ability of computing the semantic similarity between two tags to find the most similar user content and to recommend them to the users. To this end, we compute the similarity of two user content based on the similarity between their tags.

Lets assume that u_1 and u_2 denote two user contents, e.g. two questions on StackOverflow, and $t_{u_1} = \{t^1_{u_1}, t^2_{u_1}, ..., t^n_{u_1}\}$ and $t_{u_2} = \{t^1_{u_2}, t^2_{u_2}, ..., t^m_{u_2}\}$ represent the list of tags assigned to u_1 and u_2, we calculate the similarity between u_1 and u_2 based on the similarity between their tag lists.

$$sim(u_1, u_2) = sim(t_{u_1}, t_{u_2}). \quad (9)$$

Based on the discussion in the previous subsection, it is possible to compute the similarity between two individual tags. Lets assume $\alpha(t_1, t_2)$ computes the similarity of two tags t_1 and t_2 based on the length of the vector built from the three earlier similarity measures. For the sake of the following definition, lets assume that $m > n$, i.e., $t_{u_2} > t_{u_1}$. In order to be able to compute the similarity of t_{u_2} and t_{u_1}, we will need to find a set with n tag pairs, called Pairs, such that it satisfies the following conditions:

$$
\begin{aligned}
Pairs(t_{u_1}, t_{u_2}) = \{ & \\
& (x, y) | x \in t_{u_1}, y \in t_{u_2}, \\
& \nexists (x', y') \; s.t. \; \alpha(x', y') > \alpha(x, y) \\
& and \; (x', y') \notin Pairs, \\
& |Pairs| = n \\
& \}.
\end{aligned}
\tag{10}
$$

Now, with the above definition, $Pairs(t_{u_1}, t_{u_2})$ contains exactly n pairs of tags (x_i, y_i) that have the highest similarity from amongst the tags of the two user contents. These pairs can be used to compute the similarity of the two user contents:

$$
top(t_{u_1}, t_{u_2}) = \sum_{i=1}^{n} \alpha(x_i, y_i). \tag{11}
$$

where $(x_i, y_i) \in Pairs(t_{u_1}, t_{u_2})$.

$$
sim(t_{u_1}, t_{u_2}) = \frac{top(t_{u_1}, t_{u_2})}{|t_{u_2}|}. \tag{12}
$$

Simply put, Eqs. 11–12 find the most similar pairs of tags from the two tag lists based on the Pairs set. The similarity between each of the tag pairs is computed and then the overall average is computed. This average value is used as the similarity value between u_1 and u_2. The ability to calculate similarity between two user contents enables us to find the most related user content when a user is browsing through the social platform. For instance, it is possible to find the most related questions when a user is reading through to a specific question on StackOverflow.

As highlighted in Fig. 1, our approach is able to not only recommend related content from the same social platform, e.g. to suggest other related questions from StackOverflow to users browsing StackOverflow, but also able to recommend related content from other social platforms, e.g. to find and suggest related content from ReadWriteWeb when the user is browsing through questions on StackOverflow. This provides the means to create cross community links and allows users to more efficiently access the information that they are looking for. Our approach is able to provide this capability due to the fact that it encourages users to use semantic tags based on Wikipedia for tagging their content and hence is able to compare content from any sources based on a shared set of semantic tags. Furthermore, it is even able to compare content that have not adopted our tagging strategy by automatically tagging them with appropriate semantic tags and measuring cross content similarity based on these tags.

4 Tooling support

We have provided a practical implementation of our proposed approach, which is called LS^3 AutoTagger. Our implementation has been developed to support social software engineering users with both processes of *content dissemination* and *content finding*. We have downloaded and used the StackOverflow dump dataset as our main social

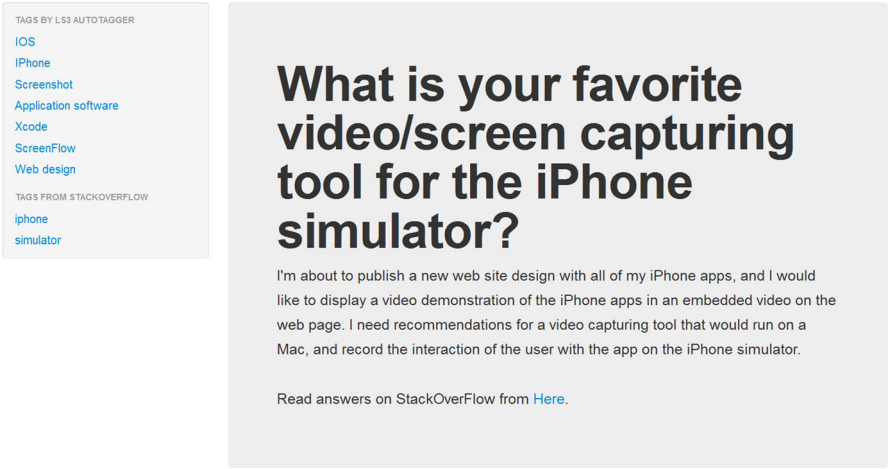

Fig. 2 The view of tag suggestion in LS^3 AutoTagger

software engineering content. The main part of the StackOverflow dump dataset that was of interest for our work was the questions and the answers that were posted by the users. Additional information such as detailed tag description that was not included as a part of the dump dataset were retrieved through StackOverflow open API version 2.1. The implementation of our tools consisted of two main components that corresponded to the two processes introduced earlier in Fig. 1. The implementation of the offline process that consisted of processing Wikipedia articles and the development of the statistical model of word distributions in Wikipedia pages was implemented using Java 1.6 and ran on a 3.4 GHz Intel Core i7 machine with 8 GB of memory. The implementation of the realtime process was done using PHP 5.0 and MySQL DBMS. The design of the Web user interface was done based on Twitter Bootstrap version 2.3.

There are three main functionalities provided to the users in LS^3 AutoTagger. The first functionality is related to the tagging process for each StackOverflow question. In our implementation, the users are able to browse through any StackOverflow question. For each question, the tool will recommend a list of suitable tags for that question. These tags are in essence in the form of links to Wikipedia pages and their label is the title of that Wikipedia page. As shown in Fig. 2, users are given the option to see both the tags that have been automatically identified and suggested by LS^3 AutoTagger and also the tags that were originally assigned to the question by the StackOverflow user who originally posted the question. Both list of tags are hyperlinked where the recommended tags by LS^3 AutoTagger refer to the underlying Wikipedia page and the tag from the users in StackOverflow are linked to the tag page on StackOverflow. In the case of the question shown in Fig. 2, the poster is interested in knowing more about screen capture software that other users employ for the iPhone simulator. The list of tags provided by LS^3 AutoTagger are iOS, iPhone, Screenshot, Application Software, Xcode, ScreenFlow and Web Design where as the tags on StackOverflow are iphone and simulator. As evident from the tags, the tags from our system proposes a more comprehensive and accurate set of tags where as the tags from StackOverflow

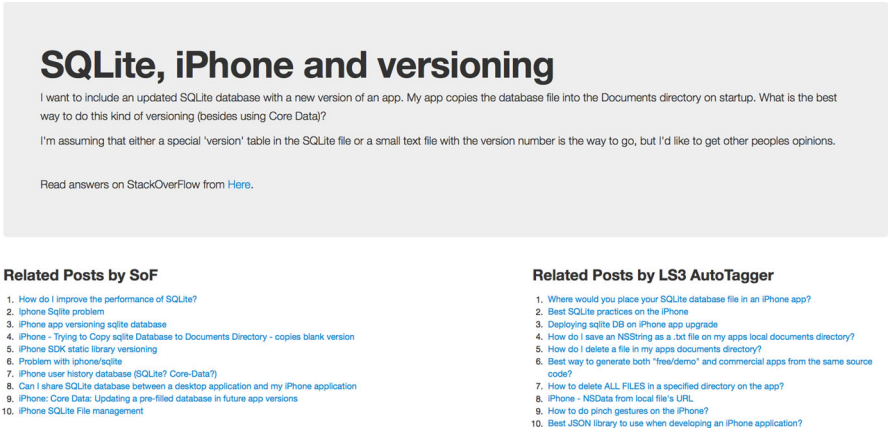

Fig. 3 The suggestion of related questions in LS^3 AutoTagger

do not even include tags to indicate that this question is related to screen capture software. An interesting observation is that among the tags proposed by LS^3 AutoTagger is ScreenFlow which is one of the widely adopted screen capture software for this purpose.

The second functionality of LS^3 AutoTagger is the ability to find highly relevant questions to a question that is being browsed by the user at any given time. For this purpose, LS^3 AutoTagger provides two sets of related questions for a question of interest. The first set of questions are those that are recommended by StackOverflow. It is possible to retrieve these questions using the StackOverflow open API. These questions are those that are recommended to the users when they are browsing through a question on the StackOverflow website. The second list of questions are those questions that LS^3 AutoTagger has identified as most relevant based on the similarity of its semantic tags with those of the given question. The purpose of this process is to help users find highly related questions that could potentially help the users answer their question or resolve their issue. Figure 3 shows that the top-10 most related questions are suggested to the users. In the question shown in Fig. 3, the user is looking for the best location to place her SQLite database for an app. As seen in the figure, both set of related posts are relevant to the question; however, the first related question provided by LS^3 AutoTagger addresses the current question, which is 'where would you place your SQLite database file in an iPhone app?'.

The third functionality provided by LS^3 AutoTagger is the possibility to connect information on StackOverflow with content from another social software engineering community. In our implementation we provide the means to retrieve related software engineering specific content from Reddit.com. This could potentially help developers (on StackOverflow) get engaged with another community, which is using their technology (on Reddit) and allow for the better exchange between the two related but non-overlapping communities. As shown in Fig. 4, LS^3 AutoTagger finds the most related content to the main themes of a given question on StackOverflow from Reddit and recommends such content to the users.

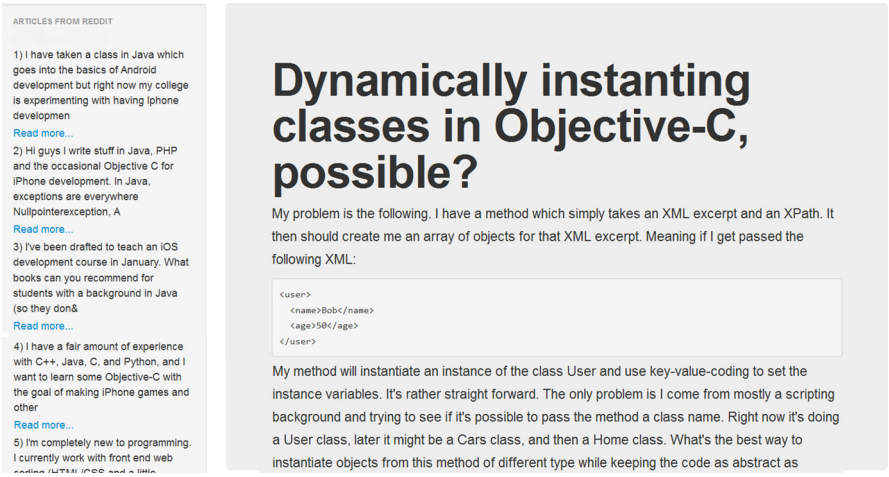

Fig. 4 The view of cross community linking by LS^3 AutoTagger

In summary, the LS^3 AutoTagger tool enables its users (*i*) to find the most suitable semantic tags for their content by suggesting tags that are semantically grounded in Wikipedia pages; (*ii*) to have access to other highly related questions on StackOverflow based on the current question they are browsing and the semantic similarity between their tags; and (*iii*) to reach relevant content from a related social community such as Reddit through cross community linking faciliated through the proposed semantic tags. It should be noted that LS^3 AutoTagger can be easily configured to perform cross community linking with other social community websites that provide open API for accessing their content. Interested readers can access LS^3 AutoTagger code from http://ls3.rnet.ryerson.ca/tmp/LS3AutoTagger.zip.

5 Experiment design

In order to evaluate our proposed approach for semantic tagging and linking of social software engineering content, we performed several experiments. The main objective of these experiments was to investigate four main research questions. In this section, we introduce these research questions and also describe the environmental setup of the experiments.

5.1 Research questions

The approach that we have taken for the evaluation of our proposed work is to empirically evaluate the impact of semantic tagging of social software engineering content on the users of software engineering social communities specifically for the case of *StackOverflow*. We have performed several experiments with the participation of software engineers who frequently use StackOverflow in their daily activities. The main *hypothesis* in our experiments was that the use of semantic tagging for annotating

social software engineering content would enhance the experience of the users in organizing, finding and relating content on a social software engineering platform. To evaluate this hypothesis, we formulate four interesting research questions and investigate how the proposed semantic tagging approach addresses each of these research questions. These four questions are as follows:

RQ1 *How does the quality of the semantic tags that are suggested to the users compare to that of the tags that were assigned by the users without intervention?*
This is an important research question given the fact that the semantic tags that we propose are based on Wikipedia content, which is not targetted for the software engineering community and can hence be considered general purpose content. We are interested to know whether the quality of semantic tags derived from such a source is acceptable by the users and how it stands in comparison with user defined tags.

RQ2 *Would the employment of semantic tags enable the recommendation of better related and more suitable content to the users?*
This research question evaluates whether the semantic-based similarity measurement and matching of content based on the proposed semantic tags improves the quality of content finding or not. In other words, would the users be more satisfied with the content recommendations provided based on the semantic tags?

RQ3 *How relevant and suitable are the cross community linking and recommendations that are derived based on the similarity of semantic tags?*
The objective of this research question is to find out whether the recommendations offered for cross community linking resonates with the users of social software engineering communities. In other words, would the users find the content recommendations from other social communities relevant and useful for the goal that they pursue?

RQ4 *Does the proposed semantic tagging approach enhance the quality of content categorization and organization?*
The main purpose of this research question is to find out whether each and every individual semantic tag assigned to a given content has significance and relevance for the content. This is because if the set of semantic tags proposed for a given content is suitable overall but there are few tags that are not too relevant, then those few tags would impact the content categorization and organization outcome.

These research questions are evaluated based on our experiments that observe the behavior and outcome of the actions of a group of social software engineering community users. We view each research question from different dimensions and report our findings.

5.2 Environment setup

5.2.1 Objects of study

The main objects of study in our experiments were the collection of question and answers posted on StackOverflow and publicly available through StackExchange blog. The information is available through a data dump shared through a creative commons

license. The data dump that we downloaded was released on September 2011.[1] In order to find appropriate semantic tags, we processed the textual content of each question along with its associated answers. The main reason for processing both the question and its answers was that there might have been some additional information in the answers that did not appear in the question. For such cases, processing the question alone would have missed some important information and hence could have resulted in an incomplete set of tags. Therefore, by analyzing both the question and its answers we attempted to cover various aspects of the content.

As mentioned earlier, in order to find suitable tags, we computed the cosine similarity between the user content, i.e. user submitted questions and answers, and the available Wikipedia pages. The value of cosine similarity can range from 0 to 1 and in our experience there are a significant number of Wikipedia pages that have an insignificant non-zero similarity value for any given content. These Wikipedia pages cannot be used as semantic tags given they are not sufficiently similar to the user content. To address this issue, we pruned all Wikipedia pages that had a similarity value of less than a specific threshold. Based on our empirical observations we found that a suitable threshold for pruning was 25 %. The remaining Wikipedia pages that had a similarity value of more than 25 % were selected and used as semantic tags for the user content. Each semantic tag was composed of a Wikipedia page title and URL.

Once the questions in the StackOverflow dump were annotated with semantic tags, the tags were used to find the most similar questions to each other. The semantic similarity measures that were introduced earlier were used to compute the similarity between two questions based on the similarity of their tags. Therefore, it was possible to provide recommendation to the users when they were browsing a specific question about other highly related questions that might be of interest.

Furthermore, for the purpose of cross community linking and recommendation, we chose Reddit as the second community of interest. Reddit is one of the most active social content sharing portals on the Web that attracts over 4 billion viewers per month. Unlike communities of practice such as StackOverflow that targets a specific community, Reddit is targetted for the general public. The reason why Reddit was chosen in our experiments was twofold: (*i*) Reddit is not solely a focused software engineering community of practice while it has substantial number of technical software engineering related content; therefore, this enabled us to explore the possibility of linking to less directly relevant content; (*ii*) Reddit provides a very fast and easy-to-use open API that faciliates the process of working with user content on that website. We employed the public Reddit API to retrieve related user content from this Website. Similar to the process for StackOverflow, we annotated each Reddit post with related semantic tags. Now, given the fact that both the content on StackOverflow and the content from Reddit were annotated based on our proposed approach with semantic tags, we were able to recommend related Reddit content for any StackOverflow question. So once a user browsed through a given question, she would receive pertinent recommendations from related content on Reddit.

[1] The dump can be accessed from StackOverflow website: http://blog.stackoverflow.com/2011/09/creative-commons-data-dump-sep-11/

5.2.2 Interacting with subjects

The main objective of our experiments was to determine how effective our proposed approach for semantic tagging and linking of content is in enhancing the experience of the users in organizing, finding and relating content on a social software engineering platforms. To this end, we performed a set of experiments with human subjects to evaluate the impact of the proposed semantic tagging approach. In our experiments, fifteen volunteers actively participated in the evaluations. The participants were all graduate students of Computer Science with the age range between 23 and 29 years. The participants all had a good knowledge of software engineering principles and were active software developers and were capable of programming in at least one programming language. The subjects were all frequent users and contributors of social software engineering communities especially StackOverflow and were therefore quite familiar with the process of finding and posting content on such websites and were already aware of the role that tags play in this process.

In the experiments, five main topic areas were selected namely, `HTML`, `iPhone`, `Java`, `MySQL` and `Android`. Each subject focused on one specific topic and three subjects were assigned to each of the topics in total. The subjects were assigned to the topics based on their background and expertise. For instance, the three subjects for the `Java` topic had all declared expertise in Java (at least 5 years of experience in using Java). The subjects were explicitly asked whether they considered themselves competent in the topic areas assigned to them and whether they had at least 5 years of experience on the topic. Subjects that declared lack of expertise in a topic would be reassigned to another topic where they had the competency. The reason for choosing five different topics was to avoid any bias as a result of inclination towards a specific topic or subject matter expertise. The subjects were asked to work individually during the experiments. Each subject received ten questions from StackOverflow randomly. The questions for each subject were in the area of the topic they had been assigned to. For each of the questions that were shown to the subjects, they received the following information: (*i*) original tags of the question from StackOverflow and the semantic tags derived based on our approach; (*ii*) recommendations on related questions offered by StackOverflow and also those suggested by our approach; and (*iii*) cross community recommendations of related content from Reddit identified by both the tags from StackOverflow and our semantic tags. The details of how the feedback was obtained is given in the next section. It should be noted that the subjects were not able to identify which results were provided to them from StackOverflow and which ones were developed by our proposed method in order to avoid any bias towards any of the two approaches.

Furthermore and in a different exercise, each subject was presented with the top-ten questions related to the five topics. Each subject would review two top-ten question lists, one list consisted of the most relevant questions for the topic according to StackOverflow and the other list included the most relevant questions to the topic according to the questions' semantic tags proposed by our approach. The purpose of this experiment was to determine the quality of content organization based on the two different tagging approaches. The more any of the two top-ten questions were related to the topic of interest, the more successful that tagging approach was in orga-

Table 1 The seven-point Likert scale used for obtaining subjects' opinion

Strongly agree	Agree	Somewhat agree	Undecided	Somewhat disagree	Disagree	Strongly disagree
(7)	(6)	(5)	(4)	(3)	(2)	(1)

nizing and categorizing user content. The users were asked to provide feedback on the suitability of both top-ten questions, the outcomes of which will be described in detail in the following sections. To avoid bias, the two lists were anonymized for the subjects.

5.2.3 Feedback and analysis

In order to obtain the participants' subjective opinion about the various aspects of our experiment, we employed the widely-used seven-point Likert scale as shown in Table 1. Once asked a questions, the subjects had the option to answer by selecting one of the linguistic terms corresponding to the point of their choice in the Likert scale. For instance as described later in the paper, a subject would be presented with a list of tags for a given StackOverflow question and be asked whether the tags are accurate or not. The subject could then answer by selecting one of the choices in the scale to show her agreement or disagreement with the accuracy of the tags for that question.

In order to analyze whether our proposed semantic tagging approach had any significant impact in the areas of the four research questions, we used the non-parametric *Wilcoxon–Mann–Whitney* (*WMW*) *test*. This statistical hypothesis test determines whether two sets of measured values are different from each other in any significant way. For experiments that are analyzed with the WMW test, we report both the p-value and the value for U. The degree of freedom is $n - 1$ in all experiments equivalent to 14, which is the number of subjects minus one.

In addition, we have also used *Chi-Square* (χ^2) *test* to find any association and the strength thereof between any of the measurements in our experiments of the four research questions. We report both χ^2 and p-value for all our measurements. Similar to the WMW test, a p-value less than 0.05 shows statistical significance.

6 Observations and evaluation

The goal of our evaluations is to validate our hypothesis, i.e., our proposed semantic tagging approach has a positive impact on organizing, finding and relating content on a social software engineering platform. For this purpose, we report and analyze our observations with regards to the four research questions.

6.1 RQ1. quality of semantic tags

The first research question intends to analyze the quality of the tags that have been proposed by our semantic tagging approach compared to the traditional form of tags that were attached to user content on StackOverflow. From our point of view, there are two major restrictions that can impact the quality of the semantic tags: (*i*) Wikipedia is

an open encyclopedia that is not written for a specific audience; therefore, its content is not necessarily the most technical information source for the software engineering community. For this reason, one concern could be that with focus on Wikipedia as the source for semantic tags, the tags would not be quite informative or accurate. (*ii*) One could argue that not all aspects and concepts of software development especially emerging technologies is covered in Wikipedia; therefore, limiting the tags to only information available on Wikipedia could be a limitation.

To analyze this research question, we further broke down the quality of tags into four dimensions, namely:

1. *Informativeness* are the tags assigned to a user content descriptive and representative of the main concepts that are addressed in the content?
2. *Comprehensiveness* have each and every aspect of the user content been covered by the tags that have been assigned to the user content?
3. *Accuracy* are the chosen tags for the given user content appropriate?
4. *Consistency* do the set of tags for a user content enjoy a degree of harmony and relatedness between themselves or in other words are there any out of context tags among them?

As mentioned earlier, in our experiments, the subjects were presented with ten random questions from StackOverflow. For each question, the subjects were provided with two sets of tags: one from StackOverflow and the other proposed by LS^3 AutoTagger. The subjects were then asked to provide their subjective opinion about each one of these sets of tags in each of the four dimensions of quality using the seven-point Likert scale shown in Table 1. Therefore, for each of the questions, a subject's feedback would consist of her subjective judgement of the tags from the perspective of *informativeness, comprehensiveness, accuracy* and *consistency*. It is important to point out the the subjects did not know which tags were from StackOverflow and which ones were proposed by LS^3 AutoTagger. The information on LS^3 AutoTagger user interface was annonymized for this purpose.

We collected a total of 1,200 data points for this research question (15 participants evaluating 2 tag sets based on 4 dimensions for 10 different questions: $15 \times 2 \times 4 \times 10$). In order to determine whether our proposed semantic tagging approach has been able to improve the quality of tags from the subjects' perspective, we compared the feedback provided by the subjects on each of the four dimensions. Our comparison was based on the WMW test method where the feedback points for StackOverflow and our proposed approach were compared together. The two sets of feedbacks were compared to see if any significant statistical difference was observed. Table 2 shows the results of our observations.

In Table 2, the second column shows the median of subjective values assigned to the tags available on StackOverflow. The third column shows how frequently the subjective value assigned to tags on StackOverflow were better than those assigned to the tags from LS^3 AutoTagger. Columns 4 and 5 show the same in the case of LS^3 AutoTagger. Columns 6 and 7 report the result of the WMW test between the subjective values assigned by the subjects. It can be seen that according to the p-value, the quality of the tags from both *informativeness* and *comprehensiveness* dimensions have improved as a result of our proposed approach. Based on the p, both of these

Table 2 The comparative analysis of the quality of tags

	Median SOF	Frequency SOF	Median AutoTagger	Frequency AutoTagger	U	*p* value
Informativeness	5	2	6	9	167.5	0.02
Comprehensiveness	4	0	5	9	173	0.01
Accuracy	5	5	5	5	132	0.43
Consistency	5	5	4	4	138.5	0.28

can be considered to be statistically significant (p-value < 0.05) and can therefore be taken as a sign of the positive impact of our proposed approach in improving the quality of tags in terms of being informative and comprehensive. This means that the tags that are proposed by our approach (*i*) are more descriptive and representative of the main concepts presented in the content; and (*ii*) provide a better coverage of the various aspects of the user content.

However based on our observations, the other two dimensions of tag quality were not improved as a result of the proposed semantic tagging approach. The quality of tags in terms of accuracy and consistency did not have too much difference between the tags from StackOverflow and the tags proposed by our approach. One explanation for this could be that the users who post content on StackOverflow are often very clearly aware of at least some of the important aspects of their question. Therefore, the tags that they assign are quite accurate and consistent/focused. So, while their tags might not cover all of the aspects of the questions as shown in comparison to the comprehensiveness of our proposed tags or may not be as informative as seen in the informativeness dimension, the ones that are provided by the users are accurate and consistent. This is inline with the general observation of community tagging practices where community tags are quite accurate for classification and organization but not so much informative for the general community (Chi and Mytkowicz 2008).

We further analyzed the relationship between the four quality dimensions using the Chi Square test. The feedback provided by the subjects on each of the dimensions was used to investigate whether any two dimensions showed any degree of correlation. The results of our analysis are shown in Tables 3 and 4. The first table is the correlation between the quality dimension based on the feedback of the subjects on the tags from StackOverflow and the second table is the same only on feedback on tags from LS^3 AutoTagger. Both tables confirm that there is no positive significant correlation between the three dimension of our research question and shows that the four dimensions could be measured and evaluated independently.

In summary, it was observed that our proposed approach is able to significantly improve the quality of tags in terms of informativeness and comprehensiveness while not negatively impacting the other two dimensions namely, accuracy and consistency. So the concerns regarding the applicability of Wikipedia as the source for the semantic tags in a social software engineering platform does not seem to have an impact on the quality of the tags in our experiments. In fact, the tags from Wikipedia show to have a higher quality compared to the community driven tags from StackOverflow.

Table 3 The correlation between quality dimensions based on StackOverflow

	Informativeness	Comprehensiveness	Accuracy	Consistency
Informativeness				
χ^2		0.5	1.9	1.8
p value		0.9	0.9	0.9
Comprehensiveness				
χ^2			1.5	2.9
p value			0.9	0.9
Accuracy				
χ^2				2.1
p value				0.9
Consistency				
χ^2				
p value				

Table 4 The correlation between quality dimensions based on LS^3 AutoTagger

	Informativeness	Comprehensiveness	Accuracy	Consistency
Informativeness				
χ^2		0.4	0.9	1.63
p value		0.9	0.9	0.9
Comprehensiveness				
χ^2			1.3	1.5
p value			0.9	0.9
Accuracy				
χ^2				2
p value				0.9
Consistency				
χ^2				
p value				

6.2 RQ2. local content recommendation

The second research question is mainly concerned with analyzing whether the use of semantic tags proposed by our approach improves the relevance and suitability of the content recommendations offered to the user. The StackOverflow website provides a recommended list of related questions to its users when the users are browsing through a question. To the extent of our knowledge, the details on how the related questions are suggested has not been released by StackOverflow. However, the list of recommended related questions is accessible through StackOverflow's open API. Therefore, we compare the two list of recommended related questions to see which one

Table 5 The comparative analysis of the suitability of the recommendations

	Median SOF	Frequency SOF	Median AutoTagger	Frequency AutoTagger	U	*p* value
Relevance	5	5	6	7	136.5	0.32
Helpfulness	4	2	5	8	169.5	0.01
Taintedness	7	7	6	2	155	0.08

is more suitable from the perspective of the subjects participating in our experiments. In order to measure suitability of recommendations, we considered three aspects:

1. *Relevance* how relevant are the recommended questions to the main theme of the question on hand?
2. *Helpfulness* would any of the recommended questions help resolve the problem addressed in the current question?
3. *Taintedness* what is the extent to which undesirable or unrelated questions are included in the recommendations?

Similar to the first research question, the subjects were presented with ten random questions in the subject's topic area and for each of these questions, the subjects received two sets of ten highly related questions, one from StackOverflow and the other based on the semantic tags of our approach. The subjects were then asked to evaluate the suitability of the recommendations by providing their subjective judgment with regards to the three dimensions introduced above. The subjective feedback was obtained through the use of the seven-point Likert scale shown in Table 1. As a result of this process, we collected 900 data points in total (15 participants evaluating 2 set of recommendations based on 3 dimensions for 10 different questions: $15 \times 2 \times 3 \times 10$), which were used to determine whether the proposed semantic tags had any positive influence on the recommendations or not.

We employed the WMW test in order to analyze whether the recommendations provided through the use of the semantic tags had any significance over those provided by StackOverflow. The subjective feedback provided by each subject for the recommendations of StackOverflow and LS^3 AutoTagger in each dimension were compared. The outcome is shown in Table 5. As shown in the table, the feedback from the subjects showed that the recommendations from both StackOverflow and LS^3 AutoTagger were quite relevant, i.e., the recommendations addressed the same topic areas as the question that was being analyzed. Furthermore, both recommendation approaches included only very few tainted recommendation, i.e. those recommendations that the subjects thought should not have been included in the list at all. The performance of both approaches is not statistically different. We would like to point out that while the details of the StackOverflow recommendation algorithm is not publicly available, we think that other additional information such as users browsing patterns are most likely considered in formulating the recommendations. In light of this, we believe that our recommendation model which is only reliant on the semantic tags for finding the most related questions performs exceptionally well given it does not have access to additional user behavior information which is available to StackOverflow.

Table 6 The correlation between recommendation usefulness dimensions based on StackOverflow

	Relevance	Helpfulness	Taintedness
Relevance			
χ^2		0.3	0.6
p value		0.9	0.9
Helpfulness			
χ^2			0.4
p value			0.9
Taintedness			
χ^2			
p value			

Table 7 The correlation between recommendation usefulness dimensions based on LS^3 AutoTagger

	Relevance	Helpful	Taintedness
Relevance			
χ^2		0.6	0.8
p value		0.9	0.9
Helpfulness			
χ^2			1.1
p value			0.9
Taintedness			
χ^2			
p value			

While the quality of the recommendations for StackOverflow and LS^3 AutoTagger are not statistically different in terms of *relevance* and *taintedness*, we observed that LS^3 AutoTagger outperforms StackOverflow in terms of *helpfulness*. This means that while both approaches provide relevant recommendations, the recommendations provided by LS^3 AutoTagger are more focused and are targetted towards the problem that is being addressed in the current question. The helpfulness of recommendations by LS^3 AutoTagger is statistically significant compared to the recommendations by StackOverflow (p-value < 0.05). This is very important given many users on StackOverflow are most likely browsing this website to find the solution to a problem that they are facing.

In addition, we investigated the relationship between the three dimensions of recommendation suitability using the Chi Square test. The results of the correlation are shown in Tables 6 and 7. According to the p-values, there is no significant correlation between the three dimensions that we have defined for recommendation suitability. This means that the observations made by the subjects of our experiments were independent for each dimension. In summary, our observations showed that while recommendations based on LS^3 AutoTagger and StackOverflow are not statistically different

in terms of *recommendation relevance* and *recommendation taintedness*, they significantly outperform StackOverflow's recommendations in terms of *helpfulness*. This is an indication that recommendations based on our proposed semantic tags can help the users find a solution to their problem in a shorter amount of time.

6.3 RQ3. Cross community recommendation

The main purpose of this research question is to explore the quality of the recommendations that are made across different social communities. As mentioned earlier in the paper, linking different communities can be beneficial for the members of the online community in several ways such as expanding their perspective by presenting content they would not otherwise encounter, engaging them with the other relevant communities that they were not aware of, engaging them to contribute to other areas where their expertise could be needed, just to name a few. One of the suitable methods for connecting communities would be to consider community tags to interlink content. The reason is that the use of information retrieval techniques where full content similarity is used would most often result in the retrieval of content from other communities that are fully similar to the content in the current community. Therefore, such a linking process would not enable the users to expand their viewpoint and become aware of other related but not necessarily exactly similar content.

By using community tags, it is possible to find related content that are in the same topic areas but are not necessarily exactly the same. Therefore, while staying within the topic area of interest to the user, more diverse content can be retrieved from other social communities and presented to the user. The main challenge pertains to the differences between the community tags that are gradually accumulated in different communities. For instance, the tags used in StackOverflow do not necessarily match those that are employed in ReadWriteWeb. Therefore with the differences in the tags and without a clear semantic grounding for each of the tags, finding similarities between content across different communities becomes a difficult task. This research question intends to investigate whether the use of semantic tags improves the quality of cross community recommendation or not.

In order to obtain the opinion of the subjects regarding the cross community recommendations, we considered three aspects:

1. *Relevance* how related the cross community recommendations are to the content that the user is viewing on the primary social community?
2. *Novelty* do the subjects show any interest in reviewing the cross community content and whether they find the content novel in that they would not have encountered it if they only considered information from the single community?
3. *Taintedness* what is the extent to which undesirable or unrelated content are recommended to the user from the other community?

A similar experimental procedure to the other two research questions was setup for RQ3. The subjects were provided with ten random questions from StackOverflow from their topic area of interest. For each of the questions, two lists consisting of ten related content from Reddit were provided to the subjects. The first list was based on the matches between the tags on the StackOverflow question and the tags of the

Table 8 The comparative analysis of the suitability of the cross community recommendations

	Median SOF	Frequency SOF	Median AutoTagger	Frequency AutoTagger	U	*p* value
Relevance	4	2	5	11	186	0.001
Taintedness	4	1	6	12	197	0.0002
Novelty	3	2	4	9	175	0.008

content from Reddit. However, for cases when no tags were similar in the two social community websites, the tags on StackOverflow were used to search Reddit and find related content. The second list of recommendations was derived according to the semantic tags that were assigned to the StackOverflow content as proposed by our approach based on their semantic similarity with content on Reddit. The subjects were asked to provide their subjective evaluation of the two lists for all ten questions according to the above three aspects without being aware of how each of the two lists were developed. We collected 900 data points from the seven-point Likert scale (15 participants evaluating 2 set of recommendations based on 3 dimensions for 10 different questions: $15 \times 2 \times 3 \times 10$).

The subjective feedback obtained from the participants was analyzed using the WMW test, the results of which are reported in Table 8. As shown in this table, the cross community recommendations that were identified based on the semantic tags of our proposed approach have a statistically significant, higher and better results in all three aspects. The first column of the table shows the median of the subjective opinion of the participants with regards to the cross community recommendations based on the StackOverflow community tags. The median for this column on the first two rows, i.e. *relevance* and *novelty* is 4. Putting the median value in the context of the second column which shows the number of times that recommendations based on StackOverflow were preferred over the recommendations based on LS^3 AutoTagger (2 vs 11 and 1 vs 12), it is apparent that the subjects clearly preferred the cross community recommendation provided by LS^3 AutoTagger. Furthermore, the findings show that recommendations based on the semantic tags are able to address one of the main objectives of cross community recommendation, which is to offer novel content to the users. It should also be noted that while LS^3 AutoTagger provides more relevant content to the users, its recommendations are also less tainted with content that are considered to be undesirable by the subjects in our experiments. In the comparison shown in Table 8, the taintedness of the recommendations by LS^3 AutoTagger is statistically less than those recommendations based on the StackOverflow community tags.

Our investigation of the interdependence between the three aspects of cross community evaluation also shows that these three aspects are not correlated with each other (Tables 9, 10). Similar to previous analysis, we have measured the correlation between these three aspects namely *relevance, novelty* and *taintedness* through the Chi Square test. The lack of correlation between these three aspects shows that the results of each of them can be independently considered and taken into account. This means that while the subjects of the experiments did not necessarily reveal any sim-

Table 9 The correlation between cross community recommendation aspects based on StackOverflow

	Relevance	Taintedness	Novelty
Relevance			
χ^2		0.3	1.3
p value		0.9	0.9
Taintedness			
χ^2			1.2
p value			0.9
Novelty			
χ^2			
p value			

Table 10 The correlation between cross community recommendation aspects based on LS^3 AutoTagger

	Relevance	Taintedness	Novelty
Relevance			
χ^2		0.8	0.5
p value		0.9	0.9
Taintedness			
χ^2			0.4
p value			0.9
Novelty			
χ^2			
p value			

ilar response patterns over these three aspects, they believed that in all three aspects the cross community recommendations were significantly better than those identified based on the StackOverflow tags.

In summary, our observations show that the quality of recommendations based on semantic tags are superior to those based on the StackOverflow tags. An additional observation can also be made by contrasting the results of Tables 5 and 8. It can be seen that the relevance of the recommendations by both approaches when the recommendations were made based on the same community content, i.e. StackOverflow, did not show too much difference. However, a significant difference is observed when those tags are used to enable cross community recommendation. In this case quality of recommendations based on the local community tags is much less than our proposed approach. One can conclude that while the community tags are able to provide relevant recommendations in the same community, they cannot be effectively used to connect two social communities. Furthermore, even within the same social community, such as StackOverflow, the helpfulness of the results based on our approach is significantly better.

6.4 RQ4. content organization

One of the benefits of using community tags is the ability to organize and categorize content based on these tags. The organization of content can be acheived by categorizing content that have the same tag in the same group. For instance, it is possible to find and organize all of the questions that have the `iPhone` tag attached to them in StackOverflow. This way if the users are interested to browse through questions that are related to `iPhone`, the questions can easily be categorized by showing only those questions that have this tag. The use of community tags is specially important as it does not require the content to be classified by experts and catgorization can happen based on any of the available tags that are attached to the content by the users. Our fourth research question analyzes whether the employment of semantic tags as proposed by our approach improves the organization of content especially in the context of StackOverflow.

In order to evaluate and compare the content organization capability of StackOverflow community tags and the semantic tags proposed by our approach, the subjects were presented with questions related to five different tags. For each tag, two question lists were presented to the subjects, each of which included the top-ten questions one based on StackOverflow community tags and the other based on the proposed semantic tags. The subjects were then asked to provide their subjective opinion about each of the two question lists according to the following aspects:

1. *Relevance* how relevant are the questions that are provided as the top-ten questions for the tag in question?
2. *Taintedness* what is the extent of undesirable questions included in the list of relevant questions for the tag under examination?

The subjective opinions of the participants were obtained based on the seven-point Likert scale. There were a total of 300 data points collected from the subjects (15 participants evaluating 2 set of questions based on 2 dimensions for 5 different tags: $15 \times 2 \times 2 \times 5$). We employed the WMW test to evaluate whether the employment of our proposed semantic tags improves the organization of questions under the various tags. The results of the WMW test is reported in Table 11. As seen in this table, the results obtained from both approaches are not tainted. This means that both approaches perform quite well in terms of filtering out completely irrelevant questions from the list. In other words, the number of *false positive* questions in the list of related questions is quite low. The difference of the two approaches in terms of degree of taintedness is negligible and the reported p-value shows that no statistically significant difference could be observed.

Table 11 The comparative analysis of the content organization

	Median SOF	Frequency SOF	Median AutoTagger	Frequency AutoTagger	U	p value
Relevance	5	1	6	9	168.5	0.01
Taintedness	6	2	6	5	130	0.48

On the other hand, the subjective feedback of the participants show that the degree of relevance of the question retrieved based on the semantic tags can be considered statistically significant compared to the community tags from StackOverflow. In other words, while taintedness is quite low meaning that irrelevant questions were not included in any of the two lists, the higher relevance for the questions based on the semantic tags is an indication that our approach is more efficient in finding highly relevant content.

Furthermore, similar to the other three research questions, we investigated the correlation between the two aspects that were measured in this research question. The obtained results from the Chi Square test show that there are no statistical correlation between *relevance* and *taintedness* in the context of content organization. Our χ^2 analysis showed p-values of over 0.9 that indicate that no significant correlation was observed between these two aspects. Therefore, the findings of the two aspects in Table 11 can be interpreted independently. In summary, our analysis of the feedback from the subjects of our experiments showed that while both approaches are able to filter the unrelated content out of the organization of content for each tag, our proposed model is significantly better in finding and ranking highly relevant content per tag.

6.5 Summary of findings

Our main hypothesis in our work was that our proposed semantic tagging approach is able to have a positive impact on organizing, finding and relating content on and across multiple social software engineering platforms. The experiments to validate the four research questions related to this hypothesis revealed the following observations about our approach:

- Our proposed approach has been effective in improving the quality of the tags from the perspectievs of *informativeness* and *comprehensiveness* while keeping in par with the quality of community tags from StackOverflow in terms of *accuracy* and *consistency*. Hence, the employment of our approach would result in higher quality tags from an overall perspective.
- Both our proposed approach for semantic tagging and the community tags from StackOverflow received the same degree of acceptance from our experiment subjects in terms of *relevance* and *taintedness* of content recommendations. However, our approach is superior in terms of providing *helpful* recommendations. Given the users on websites such as StackOverflow are often looking for a solution to an existing issue, higher degree of *helpfulness* in the offered recommendations is a favorable aspect of our proposed approach.
- In terms of cross community recommendation, our proposed approach has a better performance in all the three aspects that were considered by the subjects of our experiments, namely *relevance*, *novelty* and *taintedness*. Therefore, our approach would be highly preferred for the purpose of community linking.
- From the perspective of content organization, both approaches were able to effectively filter out irrelevant content; however, the subjects believed that the content retrieved by our approach had a higher degree of relevance.

Based on our observations, we believe that our proposed semantic tagging approach is able to provide for a more effective platform for content tagging within the social software engineering communities.

7 Discussions

In this section, we provide (*i*) our analysis of the threats to the validity of our experimentations; (*ii*) some insight into the lessons learned from the experiments and (*iii*) a discussion on how our approach is able to address the challenges that were outlined in the outset of the paper.

7.1 Threats to validity

Empirical evaluation is always subject to different threats that can influence the validity of the results. Here, we will discuss the threats to conclusion, construct, internal and external validity. We will specifically refer to the aspects of our experiments that may have been affected by these threats.

Conclusion validity Conclusion validity is the extent to which the conclusions about the presence of a statistically significant relationship between the treatments and the outcomes are valid. In our experiments, a limited number of data points (3,300 in total) were collected due to our restricted resources. In addition, we only focused on social software engineering content from StackOverflow and Reddit. Although these settings are comparable to similar studies in the area of empirical software engineering (Genero et al. 2008; Serrano et al. 2008) and can provide a basis for understanding the behavior of our proposed approach, they may pose threats to the drawn conclusions. For instance, the use of a wider range of social software engineering communities and more subjects in our experiments could impact the coverage and accuracy of our observations and findings. The reason why content from a wider array of social software engineering communities was not used was due to the fact that their content was not publicly accessible either through readily available dumps or open API.

Construct validity Construct validity is concerned with the extent that the independent and dependent variables provide accurate measurements of what they are intended to measure. The dependent variables in our experiments were measured using the subjective opinion of the participants. The threat posed by using subjective measurement mechanisms is that different participants may have different attitudes towards the evaluation of the dependent variables. For instance, some participants may be reluctant to provide high subjective values to the options that they are evaluating, whereas the others may have quite a different mindset. In spite of this fact, this subjective measurement does capture what it claims to measure, which is the value of each of the dimensions presented per research question from the perspective of software developers who are users of social media.

Another important issue that may impact construct validity is the use of a 7-point Likert scale to gather the subjective opinions of the participants. As the Likert scale is ordinal and therefore, provides only limited number of options to the participants, it may not provide the participants with the capacity to express their opinions in as

precise manner as they would like. However, despite this threat, empirical research has shown that the best number of options for Likert scale is between 4 and 7 (Lozano et al. 2008). This is because although more than 7 options will give better *psychometric properties*, they are in many cases likely to exceed the *discriminative capacity* of the participants. We used seven points in our Likert scale setup, which is within the recommended 4–7 range. Furthermore, the participants of our experiments did not report any issues or challenges in working with the 7-point Likert scale. There are also additional threats to construct validity as a result of using the Likert scale, which is due to the fact that Likert scale is an ordinal scale. According to Hubbard and Evans (2010), there are three important threats caused by ordinal scales: (1) the ordinal labels employed in ordinal scales such as the Likert scale could be inconsistently interpreted by different users or even by the same user under different circumstances; (2) many users treat ordinal scales as if they had the properties of ratio scales and hence could provide unreliable information; and (3) the distance between the different labels of an ordinal scale is not deterministically known and therefore clear comparison between the significance of various ordinal labels could be hard for an ordinary user to do. For these reasons, it is important to notice that the use of the ordinal Likert scale could have impacted the construct validity of our experimentations. Finally we would like to point out that while in the experiments the likert scale of 4 was represented as 'undecided', it would have been more appropriate to name it as 'neutral'. This is because subjects may have decided to be 'neutral' on an issue, which is not the same as being 'undecided'. In future experiments, we will consider capturing additional information from the subjects as to why they chose to be 'undecided' or 'neutral'.

Internal validity Internal validity is the degree that conclusions can be made about the causal effect of independent variables on dependent variables. An internally invalid experiment can lead to results that are not necessarily inferred from a causal relationship. For this reason, we investigate the following issues:

- *Difference between subjects* In the study reported in this paper, there was no significant difference between the expertise of the participants, due to the fact that all of them were Computer Science graduate students that had taken at least one advanced software engineering course. Therefore, error variance from difference between subjects is kept to a possible minimum.
- *Maturation effects* The participants did not become involved in any other training or experimentation process during our experiments in order to minimize the maturation/learning effect.
- *Fatigue effects* The sessions held by the experimenters with each individual participant was limited to 90 min. Since the participants are all Computer Science students and a usual lecture time is 90 min, the fatigue effect is not significant in this study.
- *Persistence effects* The study was performed on subjects that had no prior experience in participating in similar studies. Therefore, persistence effects were avoided.
- *Subject motivation* The participants were all graduate students and were therefore quite aware of the importance and impact of empirical evaluations for our research. Furthermore, the experiments were done on a volunteer basis and it is fair to assume that the participants were quite motivated to take part in the study.

Plagiarism and influence were controlled by explicitly asking the subjects not share any information about the study with each other until after the experiments were concluded. Furthermore, the evaluation sessions were held individually for each participant with the presence of an experimenter.

External validity External validity is the extent to which the obtained results of a study can be generalized to the setting under study and other relevant research scenarios. The results of a completely externally valid study can be generalized and applied safely to the software engineering practice and be recommended as blueprints. The following two issues were considered for external validity:

- *Materials and tasks used* In our experiments, we have used the complete data dump from StackOverflow as well as content from Reddit. The experiments also focused on five main topic areas where the subjects had expertise in. However, other social software engineering communities or other subject areas need to be further empirically investigated as the content becomes publicly available to verify the generalizability of the observations.
- *Subjects* Due to the difficulty of getting professional software engineers' participation in our experiments, we used Computer Science graduate students in our studies. In our experiments, the participants do not particularly need a high level of industrial experience to be able to complete the experiment. Furthermore, authors such as Basili et al. (1999) believe that in many cases, the use of students in experiments has little impact on the validity of the results. However, further experiments involving industrial professional subjects are needed to ensure the external validity of our experiments.

7.2 Lessons learnt

As a result of undertaking the experiments to validate our four research questions, we encountered some interesting and important issues pertaining to social software engineering content. The first issue was related to the content provided through Wikipedia. Given Wikipedia is in essence an open source of encyclopedic knowledge, it is only intended to give an overview of the selected but significant topics in various areas. Therefore, one would not consider Wikipedia as an ultimate reference point for learning technology or becoming familiar with the in-depth details of a subject matter. Some communities gather and continuously maintain an up-to-date knowledge repository for their field alongside Wikipedia such as Stanford Encyclopedia of Philosophy. Therefore, the expectation from Wikipedia is not to serve as the conclusive source of information especially in a fast-changing area such as software development and engineering. The first concern that we had was given Wikipedia mainly focuses on *variety* as opposed to *specificity* that the tags drawn from Wikipedia would not be as useful or specific as possible. However, our observations from the subjects of our experiments showed us that suitable tags for social software engineering content are those that strike a right balance. In other words, desirable tags are those that are not too specific and fine grained as to cause sparse content categorization and at the same time not too general as to result in excessively populated content categories. For instance, if the tags are too fine-grained then there is a high likelihood that not too many social

content could be found that share the same tags and on the other hand, if the tags are too general then too many social content with the same tags will be found. Our observation was that Wikipedia pages seem to respect this desirable balance and hence can be viewed as suitable tags for content annotation.

Our second observation was based on the information reported in Table 2. The subjects of our experiments reported that the tags suggested by our proposed approach are significantly better in terms of *informativeness* and *comprehensiveness*; however, they are on par with the community tags from StackOverflow in terms of *accuracy* and *consistency*. It seems that while community users are capable of finding very accurate and to the point tags for their content, their main challenge is to select tags that would be explanatory and covering of all aspects of the content. Therefore, one possible approach would be to ask the users to interactively tag their content by considering the tags that are proposed to them based on Wikipedia content. For instance, the users could be presented with a list of potentially suitable semantic tags from which they could select the ones that are desirable and discard the ones that are not considered to be relevant. This would enable the users to become familiar with potentially useful semantic tags that are recommended to them; hence, increasing the chances of assigning more informative and comprehensive tags for their content.

The other important issue that we encountered pertained to the problem solving nature of social community websites such as StackOverflow. One of the major goals of such websites is to enable their users to rapidly solve an issue they are facing by either posting a new question in hopes of relevant answers from other community contributors or to look at existing questions and the answers that are provided. In the latter case, users looking for solutions to their issue would be interested in receiving pertinent recommendation from the social platform in a way that would directly help them solve their problem. In such situations, it would not be sufficient for the recommendations to be relevant as they need to be helpful and novel as well. Based on the observations reported in Tables 2, 5 and 8, it seems that given the fact that semantic tags are more informative and comprehensive that recommendations from the semantic tags are in turn more accurate and helpful. For this reason, it is possible to not only find relevant content but those that are novel and helpful in the process. The use of Wikipedia content also facilitates this, as its entries are neither too specific nor too overly general.

The final point that we would like to point out is *open* versus *closed* nature of social software engineering content. While the content on many online social communities are collectively aggregated through the individual contributions from the users, many of such community websites do not offer open and/or free access to their information either as a data dump or open API. Based on the results reported in Table 8, it seems that linking two or more communities can result in the identification and introduction of novel information to the communities, which could lead to effective knowledge sharing, opportunities for collaboration and broadening the users perspective, to name a few. Therefore, releasing these data to the public domain either through data dumps or through an easy-to-use API will provide the means for extensive cross community sharing and linking of information.

7.3 Treatment of challenges

In Sect. 1.1, we highlighted a set of major challenges that specialized social communities faced when deploying a community tagging strategy. Here, those challenges are revisited in order to analyze how well LS^3 AutoTagger has been able to address them:

- *Tag explosion* The main reason for tag explosion is due to the fact that users are free to add any set of tags to their content without consideration for already existing tags. Therefore resulting in an abundance of tags that are only sparsely used. In our approach, this challenge is addressed because of two mechanisms: (i) our approach automatically provides tag recommendations for the user content and gives the user the ability to adjust the tag list as desired. The recommended list provides insight in terms of the most suitable semantic tags; therefore, the users are less likely to use less relevant and sparsely used tags for their content; and (ii) in the proposed approach the users are bound to select tags that have corresponding Wikipedia pages; therefore, this will eliminate the chance of creating duplicate and redundant tags.
- *Interpretation difference* Based on the users' background, domain of expertise and knowledge, a single tag can be interpreted differently. For instance, as mentioned earlier, BT could be interpreted as representing Bluetooth or BitTorrent. Our approach eliminates the interpretation difference problem by semantically grounding tags in Wikipedia entries. This way, BT will either be represented as BT (http://en.wikipedia.org/wiki/BitTorrent) or BT (http://en.wikipedia.org/wiki/Bluetooth), which are distinguishable tags.
- *Incomplete context* Community tagging approaches often require the user who posted the user content to annotate the content with suitable tags. However, if the user is unaware of all the different aspects pertaining to the topic, the set of tags would not be descriptive of the whole picture (e.g., the poster of a question on StackOverflow would not know the whole landscape of her question). Our approach addresses this issue by not only considering the initial user content (the question) but also the additional content provided by the other users (answers to the question) when proposing the semantic tags. Therefore, the recommended tags will cover various aspects of the content.
- *Locality of tags* As shown in the literature Guy and Tonkin (2006), communities have the tendency to develop their own unique collection of tags that is not necessarily similar or related to the tags of another community. As shown in our experiments, our approach addresses the locality of tags challenge by grounding tags in Wikipedia articles. Therefore, the collection of tags for different communities would be all based on Wikipedia pages; therefore, content from different social communities can easily be integrated.
- *Composite tags* This challenge is a direct result of the employment of inappropriate tags by the users. More specifically in this case, the users use tags that should have been defined using two separate tags (e.g. javascript-editor). Our approach is able to effectively address this challenge by only allowing the employment of tags that have corresponding Wikipedia pages. In other words, a user can only use a tag if there is a Wikipedia page for that tag. Therefore, such cases would be avoided. Our

approach additionally support composite tags that do in fact have corresponding Wikipedia entries. For instance, Garbage collection (http://en.wikipedia.org/wiki/Garbage_collection_(computer_science)) is an important concept that has its own Wikipedia entry; therefore, can be used as a tag. Our approach would only prevent the use of arbitrary composite tags.

- *Obscure similarity* It is quite hard to find the similarity of the tags that are used by different users within a social community. We already discussed that approaches based on co-occurrence of tags do not necessarily address similarity. In our work, given the fact that tags are semantically grounded in Wikipedia articles, we are able to compute the semantic similarity of the tags using three different techniques, namely path-based, information content-based and overlap-based methods. Therefore, even if the tags do not co-occur in the historical data or they are syntactically dissimilar, it is still possible within our proposed work to compute their semantic similarity by considering contextual information from Wikipedia.

8 Related work

The software engineering community has become increasingly interested in exploiting the benefits of social media for further enhancing the quality of its work (Begel et al. 2010; Storey et al. 2010). Software developers frequently use social media for various purposes such as collaborative code editing and review, bug and issue tracking, crowdsourcing and information sharing through forums, blogs and Q&A websites. The adoption of social media in software engineering practices has attracted researchers to invest in understanding and analyzing social content that is shared at either team or community levels. The use of natural language processing techniques has gained some attention as they allow for the systematic processing and analysis of social media and software engineering content (Bagheri et al. 2012; Barua et al. 2012; Pollock 2012). For instance, Pagano and Maalej (2012) process the blogging behavior of over a thousand software developers to understand the main topics of interest and developers interaction patterns. In a similar study, Tian et al. (2012) analyze the content of microblogs posted by software developers to find out what are the significant topics and categories covered in the tweets. Similarly, Achananuparp et al. (2012) provide a Twitter analytics platform that allows for the identification of trending software engineering topics in software developers' tweets. Other work such as the work by Zhong and coworkers (2012) mine specifications from natural language API documentation and also from repositories such as Google code. Other tools such as Reverb and coworkers (2013) are able to find and suggest relevant Web pages that could assist the developers in solving the challenges they face through monitoring their coding practice in the IDE and relating it to software engineering Web pages.

Our focal point in this paper has been the use of community tagging for organizing and accessing software engineering content. Treude and Storey (2012, 2010) have performed extensive empirical studies on the use of tagging approaches in software engineering. Their work has focused on how tags can be used for managing work items within a software team through tools such as IBM Jazz. They find tags to be a suitable light-weight mechanism for bridging the gap between technical and social aspects of

software development. Hale et al. (2011) have found that community tags can be an effective source for enhancing traceability between the various artifacts of the software development lifecycle. Al-Kofahi et al. (2010) have also analyzed software developer work items in IBM Jazz and observed that the employed tags for each work item are often incomplete or not fully precise. For this reason, they proposed an approach, called TagRec, which uses fuzzy set theory to find the best matching tags for the work item. In their work, they analyze the textual content of a work item and find similar work items that are already tagged. Based on similarities between the two work items, the tags from the previous work item is suggested to be used in the new work item. While these approaches address the issue of tagging software engineering content, they are solely focused on work item tagging in a team setting whereas our work in this paper is focused on tagging open social software engineering content and knowledge such as those shared on StackOverflow. A more recent work by Xia et al. (2013) has focused on recommending suitable tags for software engineering information sites primarily based on historical data. They propose a sample-based approach to combine three tag recommendation strategies. In their work, called TagCombine, they integrate the following three strategies: a multi-label ranking strategy which considers each tag as a label, a similarity-based ranking method that suggests tags by comparing the object in question with similar already tagged objects, and tag-term approach where affinity of tags and terms are determined based on historical data. The major differentiating factor between Xia et al and our work is that we propose semantic tags grounded in Wikipedia content whereas the work in Xia et al. (2013) only recommends tags based on prior observation of user assigned tags. Therefore, the possible advantage of our work is the possibility to semantically tag content in LS^3 AutoTagger that would allow the semantic content similarity computation and cross community linking that would not be possible in Xia et al's work.

Closely related to the theme of our work in this paper, Gottipati et al. (2011) propose a bayesian approach for classification of social sofware engineering content. The objective of their work is to semantically tag social software engineering content for the purposes of finding relevant content on software forums. However, their definition of semantic tags is different from ours. In their approach, the authors define seven main tags, namely *question, answer, clarifying question, clarifying answer, positive feedback* and *junk*. These tags are referred to as the semantic tags, which are more of a form of content classification rather than the typical form of content tagging. However, in our approach, the semantic tags that are recommended for each social content carry significant semantic value. Similar to our observation, Wang et al. (2012) have identified the issue of terminological inconsistency in tagging practices. In other words, different users employ dissimilar words or phrases that might have overlapping semantics to describe similar content. To address this issue, the authors have proposed an approach to build a hierarchical taxonomy from existing community tags, which could be later used to measure similarity between tags. The work by Wang et al. (2012) focuses on one of the key challenges that we address in this paper by mining co-occurrence relationship and content overlap between posts to build the tag taxonomy. However, as mentioned earlier the underlying assumption that considers co-occurrence as a measure of similarity is not too accurate. Furthermore, Wang et al. do not clarify how tag evolution will be addressed in their approach and how often the tag taxonomy

would need to be revised and how that would impact their similarity measurements. These issues do not impact our proposed work as we semantically ground our tags in Wikipedia content and not on social software engineering content which are being tagged themselves.

Our work in this paper has primarily been based on content from the social Q&A StackOverflow website. In light of the fact that StackOverflow openly provides its content to the community through data dumps various researchers have already embarked on the analysis of such social software engineering content. For instance, Barua et al. (2012) have processed the content in StackOverflow data dump to understand what developers have been talking about over time and how topical trends have formed and gradually changed in this social online community. In a different study over StackOverflow content, Parnin et al. (2012) have explored the quality of API discussions formed through crowd documentation and whether those can be considered suitable points of reference for understanding the correct usage of different APIs. Furthermore, Treude et al. (2011) have analyzed question and answer content on StackOverflow to understand how developers respond to questions on social open communities, i.e., which questions receive the highest quality answers and which ones remain unanswered. More interesting work on analyzing the content of StackOverflow can be found in the related literature (Pal et al. 2012; Ponzanelli et al. 2013; Posnett et al. 2012).

We would like to note that our work is not only related to tagging within software engineering social content and can be viewed within the scope of the broader textual content tagging literature. As pointed out by Lops et al. (2013), tag recommendation can be performed either through content analysis or through collaborative filtering. Collaborative filtering based techniques focus on the authors' social tagging behavior, in other words, they rely on the relationship between the authors' social interactions to recommend tags to the content that they produce. On the other hand, content-based approaches analyze the content such as word frequencies, semantic connotations and significant words to find similarities between already tagged content and non-tagged content. Our work in this paper falls within the realm of content-based technqiues as it relies on the analysis of word frequency distributions in user content and its comparison to that of Wikipedia articles. There have been other related work that focus on the use of Wikipedia content for labeling. For instance, the work by Carmel et al. (2009) labels clusters of documents by using Wikipedia page categories and titles. The main difference between our work and the work by Carmel is that we compare similarity between user content and Wikipedia articles by comparing word distributions whereas the work by these authors find related Wikipedia pages by simply search Wikipedia using important words from the user content. We encourage interested readers to explore significant related work in the information retrieval literature such as Sigurbjörnsson and van Zwol (2008), Wang et al. (2012) and Zangerle et al. (2011).

9 Concluding remarks

The scale of information available through social software engineering communities is growing as new software development technologies emerge. Software engineers and

developers take effective advantage of the social media to communicate, collaborate and share information at unprecedented rates. This wealth of information requires appropriate mechanisms that would enable software developers to organize, search, and find the most appropriate content based on their needs in the shortest amount of time. Traditional forms of top-down content organization have shown to lack scalability and agility required for large amounts of social content. Light-weight bottom-up approaches such as community tagging have shown to be scalable for large social content. However, the use of open-ended community tags incurs new challenges as mentioned earlier in this paper.

To address such challenges, we have proposed a tagging approach that systematically grounds community tags into the semantics embedded in Wikipedia pages. In other words, we propose that instead of allowing the users to select any keyword they desire for the purpose of tagging, they can select from the topics covered in Wikipedia to tag their social content. This way, the tags that are used by the users of different social software communities would have the same semantic interpretation across users and communities. In order to facilitate the process of semantic tagging of content on social software engineering communities such as StackOverflow, we have proposed a statistical approach that automatically finds the most appropriate semantic tags for the content posted by the users. The users have the option to consider the suggested semantic tags and decide on the final list of tags that they would like to assign to their content. The major benefits of our approach are: (*i*) it provides a uniform way for tagging social software engineering content with semantically grounded community tags; (*ii*) the semantic tags can be effectively compared and even in cases where syntactical similarity between the tags are non-existent, the degree of semantic similarity between the tags can be computed based on the grounding of the tags in Wikipedia; (*iii*) the similarity between the semantic tags can be used to pinpoint relevant and related content that might be helpful or interesting to the users on the social software engineering community; and (*iv*) pertinent content from two or more social communities can be interlinked and connected through the use of the semantic tags and their degree of similarity, which would not have been otherwise possible using the traditional community tags.

In light of the fact that each of the tags in our proposed approach has a clear semantic interpretation based on Wikipedia content, the main challenges that were mentioned in the paper for traditional community tags can be overcome by our approach, namely: *tag explosion* would be addressed as only tags from Wikipedia pages can be employed; hence avoiding the possibility of using semantically-replicated but syntactically different tags, *interpretation difference* would be avoided as each tag has a clear corresponding Wikipedia entry that disambiguates the intended meaning of that tag, *locality of tags* would not be an issue given the semantic tags can be used across different communities while preserving their semantics, *composite tags* will not be assigned to content in light of the fact that the semantic distinction between concepts has already occurred at the level Wikipedia pages, *obsecure similarity* will be resolved as both syntactic and semantic similarity between the proposed tags can be formally calculated and *incomplete context* is addressed by considering the complete picture of a user submitted content when recommending the most suitable set of semantic tags for that content.

We have also empirically validated our proposed semantic tagging approach through four main research questions. As a result of our observations, it is evident that our proposed approach is effective in proposing high quality tags for user content on social software engineering communities and is able to make helpful and novel recommendations regarding relevant content for the user both within and across multiple online communities.

There are several interesting research challenges that we would like to explore as a part of our future work: The first area that we would like to explore pertains to the use of Wikipedia pages for tagging software engineering content. As mentioned earlier, there might be cases when a Wikipedia page does not exist for a given concept that would be highly relevant to be used as a tag for a user content. Our assumption in this paper has been that the user would take it upon herself to create a new Wikipedia entry so that it can be used as a tag for the newly generated user content and to also beneficially serve the community. However, we would like to empirically investigate this assumption in order to see whether users would in fact engage in this form of activity or not. The second aspect that we would like to further study is the use of thesauri and ontologies in our semantic similarity measurement. In the current form, our model measures similarity based on the comparison of the distributions of the StackOverflow content and that of Wikipedia articles. However, given content can be written with different writing patterns such as synonymy or polysemy, we would like to explore whether additional sources of information that would help contextualize the calculation of word distribution be able to improve LS^3 AutoTagger.

References

Achananuparp, P., Lubis, I.N., Tian, Y., Lo, D., Lim, E.-P.: Observatory of trends in software related microblogs. In: Proceedings of the 27th IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 334–337 (2012)

Al-Kofahi, J.M., Tamrawi, A., Nguyen, T.T., Nguyen, H.A., Nguyen, T.N.: Fuzzy set approach for automatic tagging in evolving software. In: IEEE International Conference on Software Maintenance (ICSM), pp. 1–10 (2010)

Bagheri, E., Ensan, F., Gasevic, D.: Decision support for the software product line domain engineering lifecycle. Autom. Softw. Eng. **19**(3), 335–377 (2012)

Barua, A., Thomas, S.W., Hassan, A.E.: What are developers talking about? an analysis of topics and trends in stack overflow. Empirical Softw. Eng. **14**, 1–36 (2012)

Basili, V., Shull, F., Lanubile, F.: Building knowledge through families of experiments. IEEE Trans. Softw. Eng. **25**(4), 456–473 (1999)

Begel, A., DeLine, R., Zimmermann, T.: Social media for software engineering. In: Proceedings of the FSE/SDP Workshop on Future of Software Engineering research, ACM, pp. 33–38 (2010)

Begel, A., Khoo, Y.P., Zimmermann, T.: Codebook: discovering and exploiting relationships in software repositories. In: ACM/IEEE 32nd International Conference on Software Engineering, vol. 1, pp. 125–134 (2010)

Carmel, D., Roitman, H., Zwerdling, N.: Enhancing cluster labeling using wikipedia. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR '09, ACM, pp. 139–146. New York, NY (2009)

Chi, E.H., Mytkowicz, T.: Understanding the efficiency of social tagging systems using information theory. In: Proceedings of the Nineteenth ACM Conference on Hypertext and Hypermedia, ACM, pp. 81–88 (2008)

Frost, R.: Jazz and the eclipse way of collaboration. Softw. IEEE **24**(6), 114–117 (2007)

Gabrilovich, E., Markovitch, S.: Computing semantic relatedness using wikipedia-based explicit semantic analysis. IJCAI **7**, 1606–1611 (2007)
Genero, M., Poels, G., Piattini, M.: Defining and validating metrics for assessing the understandability of entity-relationship diagrams. Data Knowl. Eng. **64**(3), 534–557 (2008)
Gómez, C., Cleary, B., Singer, L.: A study of innovation diffusion through link sharing on stack overflow. In Proceedings of the Tenth International Workshop on Mining Software Repositories, IEEE Press (2013)
Gottipati, S., Lo, D., Jiang, J.: Finding relevant answers in software forums. In: Proceedings of the 2011 26th IEEE/ACM International Conference on Automated Software Engineering, IEEE Computer Society, pp. 323–332 (2011)
Gulli, A., Signorini, A.: The indexable web is more than 11.5 billion pages. In: Special Interest Tracks and Posters of the 14th International Conference on World Wide Web, ACM, pp. 902–903 (2005)
Guy, M., Tonkin, E.: Tidying up tags. D-Lib Mag **12**(1), 1082–9873 (2006)
Hale, M., Jorgenson, N., Gamble, R.: Analyzing the role of tags as lightweight traceability links. In: Proceedings of the 6th International Workshop on Traceability in Emerging Forms of Software Engineering, ACM, pp. 71–74 (2011)
Hassan, A.E., Xie, T.: Software intelligence: the future of mining software engineering data. In: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, ACM, pp. 161–166 (2010)
Hubbard, D., Evans, D.: Problems with scoring methods and ordinal scales in risk assessment. IBM J. Res. Dev. **54**(3), 2–10 (2010)
Kittur, A., Chi, E., Pendleton, B.A., Suh, B., Mytkowicz, T.: Power of the few vs. wisdom of the crowd: Wikipedia and the rise of the bourgeoisie. World Wide Web **1**(2), 19 (2007)
Lee, D.L., Chuang, H., Seamons, K.: Document ranking and the vector-space model. Softw. IEEE **14**(2), 67–75 (1997)
Lops, P., de Gemmis, M., Semeraro, G., Musto, C., Narducci, F.: Content-based and collaborative techniques for tag recommendation: an empirical evaluation. J. Intell. Inf. Syst. **40**(1), 41–61 (2013)
Lozano, L.M., García-Cueto, E., Mu niz, J.: Effect of the number of response categories on the reliability and validity of rating scales. Methodol. Eur. J. Res. Methods Behav. Social Sci. **4**(2), 73–79 (2008)
Pagano, D., Maalej, W.: How do open source communities blog? Empirical Softw. Eng. **15**, 1–35 (2012)
Pal, A., Harper, F.M., Konstan, J.A.: Exploring question selection bias to identify experts and potential experts in community question answering. ACM Trans. Inf. Syst. **30**(2), 10:1–10:28 (2012)
Pandita, R., Xiao, X., Zhong, H., Xie, T., Oney, S., Paradkar, A.: Inferring method specifications from natural language api descriptions. In: Proceedings of the 2012 International Conference on Software Engineering, IEEE Press, pp. 815–825 (2012)
Parnin, C., Treude, C., Grammel, L., Storey, M.-A.: Crowd documentation: Exploring the coverage and the dynamics of api discussions on stack overflow. Technical Report, Georgia Institute of Technology (2012)
Pollock, L.: Leveraging natural language analysis of software: Achievements, challenges, and opportunities. In: 28th IEEE International Conference on Software Maintenance (ICSM), IEEE, pp. 4–4 (2012)
Ponzanelli, L., Bacchelli, A., Lanza, M.: Seahawk: stack overflow in the ide. In: Proceedings of ICSE, pp. 1295–1298 (2013)
Ponzetto, S.P., Strube, M.: Knowledge derived from wikipedia for computing semantic relatedness. J. Artif. Intell. Res. (JAIR) **30**, 181–212 (2007)
Posnett, D., Warburg, E., Devanbu, P.T., Filkov, V.: Mining stack exchange: Expertise is evident from initial contributions. In: International Conference on Social Informatics, pp. 199–204 (2012)
Salton, G., Wong, A., Yang, C.-S.: A vector space model for automatic indexing. Commun. ACM **18**(11), 613–620 (1975)
Sawadsky, N., Murphy, G.C., Jiresal, R.: Reverb: recommending code-related web pages. In: Proceedings of the 2013 International Conference on Software Engineering, IEEE Press, pp. 812–821 (2013)
Serrano, M.A., Calero, C., Sahraoui, H.A., Piattini, M.: Empirical studies to assess the understandability of data warehouse schemas using structural metrics. Softw. Qual. J. **16**(1), 79–106 (2008)
Sigurbjörnsson, B., van Zwol, R.: Flickr tag recommendation based on collective knowledge. In: Proceedings of the 17th International Conference on World Wide Web, WWW '08, ACM, pp. 327–336. New York, NY (2008)
Sinclair, J., Cardew-Hall, M.: The folksonomy tag cloud: when is it useful? J. Inf. Sci. **34**(1), 15–29 (2008)
Singer, L., Schneider, K.: Influencing the adoption of software engineering methods using social software. In: 34th International Conference on Software Engineering (ICSE), IEEE, pp. 1325–1328 (2012)

Singhal, A., Buckley, C., Mitra, M.: Pivoted document length normalization. In: Proceedings of the 19th annual international ACM SIGIR Conference on Research and Development in Information Retrieval, ACM, pp. 21–29 (1996)

Stieglitz, S., Dang-Xuan, L.: Emotions and information diffusion in social media-sentiment of microblogs and sharing behavior. J. Manage. Inf. Syst. **29**(4), 217–248 (2013)

Storey, M.-A., Treude, C., van Deursen, A., Cheng, L.-T.: The impact of social media on software engineering practices and tools. In: Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, ACM, pp. 359–364 (2010)

Strandberg, K.: A social media revolution or just a case of history repeating itself? the use of social media in the 2011 Finnish parliamentary elections. New Media & Society (2013)

Strube, M., Ponzetto, S.P.: Wikirelate! computing semantic relatedness using wikipedia. AAAI **6**, 1419–1424 (2006)

Tian, Y., Achananuparp, P., Lubis, I.N., Lo, D., Lim, E.-P.: What does software engineering community microblog about? In: 9th IEEE Working Conference on Mining Software Repositories (MSR), IEEE, pp. 247–250 (2012)

Treude, C., Barzilay, O., Storey, M.-A.: How do programmers ask and answer questions on the web?: Nier track. In: 33rd International Conference on Software Engineering (ICSE), IEEE, pp. 804–807 (2011)

Treude, C., Storey, M.-A.: Work item tagging: communicating concerns in collaborative software development. IEEE Trans. Softw. Eng. **38**(1), 19–34 (2012)

Treude, C., Storey, M.-A.D.: Bridging lightweight and heavyweight task organization: the role of tags in adopting new task categories. ICSE **2**, 231–234 (2010)

Wang, M., Ni, B., Hua, X.-S., Chua, T.-S.: Assistive tagging: a survey of multimedia tagging with human–computer joint exploration. ACM Comput. Surv. **44**(4), 25:1–25:24 (2012)

Wang, S., Lo, D., Jiang, L.: Inferring semantically related software terms and their taxonomy by leveraging collaborative tagging. In: 28th IEEE International Conference on Software Maintenance (ICSM), IEEE, pp. 604–607 (2012)

Wartena, C., Brussee, R., Wibbels, M.: Using tag co-occurrence for recommendation. In: Ninth International Conference on Intelligent Systems Design and Applications, ISDA'09, IEEE, pp. 273–278 (2009)

Xia, X., Lo, D., Wang, X., Zhou, B.: Tag recommendation in software information sites. In: Proceedings of the 10th Working Conference on Mining Software Repositories, MSR '13, pp. 287–296 (2013)

Zangerle, E., Gassler, W., Specht, G. Using tag recommendations to homogenize folksonomies in microblogging environments. In: Proceedings of the Third International Conference on Social Informatics, SocInfo'11, pp. 113–126. Springer-Verlag, Berlin, Heidelberg (2011)

Zesch, T., Gurevych, I.: Analysis of the wikipedia category graph for nlp applications. In: Proceedings of the TextGraphs-2 Workshop (NAACL-HLT 2007), pp. 1–8 (2007)

Zhou, A., Qian, W., Ma, H.: Social media data analysis for revealing collective behaviors. In: Proceedings of the 18th ACM SIGKDD international conference on Knowledge discovery and data mining, ACM, pp. 1402–1402 (2012)