Expert Finding using Markov Networks in Open Source Communities

Matthieu Vergne[1,2], Angelo Susi[1]

[1] Center for Information and Communication Technology, FBK-ICT
Via Sommarive, 18 I-38123 Povo, Trento, Italy
{vergne,susi}@fbk.eu
[2] Doctoral School in Information and Communication Technology
Via Sommarive, 5 I-38123 Povo, Trento, Italy
matthieu.vergne@unitn.it

Abstract. Expert finding aims at identifying knowledgeable people to help in decision processes, such as eliciting or analysing requirements in Requirements Engineering. Complementary approaches exist to tackle specific contexts like in forum-based communities, exploiting personal contributions, or in structured organisations like companies, where the social relationships between employees help to identify experts. In this paper, we propose an approach to tackle a hybrid context like an Open Source Software (OSS) community, which involves forums open to contributors, as well as companies providing OSS-related services. By representing and relating stakeholders, their roles, the topics discussed and the terms used, and by applying inference algorithms based on Markov networks, we are able to rank stakeholders by their inferred level of expertise in one topic or more. Two preliminary experiments are presented to illustrate the approach and to show its potential benefit.

Keywords: Expert Finding, Open Source Software, Requirements Engineering, Markov network

1 Introduction

A requirement for a system defines what this system should achieve to meet the expectations of the stakeholders who depend on it. In Requirements Engineering (RE), researchers deal with requirement-related issues such as eliciting, modelling and analysing, documenting and checking that the requirements are fulfilled [2,9]. Specific difficulties in this area are, for example, the huge amount of stakeholders to deal with, the stakeholders' heterogeneity and distribution, the difficulty to express needs and solve their conflicts [16]. Although methodologies exist to support the analyst in dealing with the full RE process, a broad mastering is generally infeasible for a single person [4,8] and makes RE processes human- and knowledge-intensive [1,12]. These difficulties show the need to support the management of the information about the requirements in an efficient and customised way. One such way is to rely on available stakeholders considered as *experts* to provide reliable information or to analyse the information provided by others.

Aiming at recommending stakeholders, two main approaches have been considered in requirements elicitation: forum-based approaches, which rely on the contribution of stakeholders in forums to evaluate their knowledge [1], and social network-based approaches, which exploit relationships between stakeholders to evaluate them relatively to the others [11]. However, while these two approaches show interesting results in their specific contexts, they are not designed to exploit the information provided by a hybrid context like an Open Source Software (OSS) project. In such a context, a large community of anonymous stakeholders provide few relations between each others and OSS-related companies can participate through some representatives only.

In this paper, we propose a novel approach exploiting concepts borrowed from both forum- and social network-based works to fill this gap. In particular, we show how they relate to two complementary perspectives to evaluate expertises, that we call *content-based* and *social-based* perspectives, which justifies their use in a unified way. While this approach could be considered in a broader scope than RE, we mainly inspire from works and build on concepts used in the RE field, justifying the scope of this paper. In our approach, we basically reuse the concept of *role* provided in social networks, the concepts of *topic* and *term* provided in forum-based works, and the concept of *stakeholder* common in both (and more broadly in

RE). By relating instances of these concepts depending on evidences extracted from available data sources, we build a new model where specific stakeholders are related to instances of the other concepts, allowing us to evaluate their expertise. We translate this model into a Markov network, which allows us to produce inferences based on the modelled expertise, in order to rank a stakeholder based on selected topics or roles for instance.

In the following, Section 2 provides an overview of the state of the art in expert finding and stakeholders recommendation in RE, focusing particularly on the two approaches mentioned previously. Then, we highlight the main limitations we want to tackle in Section 3 and describe how we do so in Section 4. We present in Section 5 two preliminary experiments, one with a small, illustrative example where our approach has been successfully applied, and another on an existing OSS project. Finally, we highlight the limits of our results and discuss how our approach can be improved in Section 6 before to conclude.

2 State of the Art

2.1 Expert and Expertise

By looking at different dictionary definitions of *expert*, we can identify a broad agreement on the concept among dictionaries like the Collins[3], Oxford[4], Cambridge[5] and Merriam-Webster[6] dictionaries. Considering the last one, more precise, you can be identified as an expert by "having or showing special skill or knowledge because of what you have been taught or what you have experienced". This definition shows two perspectives: when one *has* special skills or knowledge, whether people assess them or not, and when one *shows* special skills or knowledge, whether he actually has them or not. Ericsson [5] investigates more deeply three criteria (p. 14): a *lengthy domain-related experience*, based on evidences that one has extended knowledge ; a *reproducibly superior performance*, based on evidences that one has extended skills ; a *social* criteria, where a community agree on the status of expert that one could have. We retrieve the two former criteria in the *expertise* notion provided by Sonnentag et al. [5] (p. 375) with the *long experience* and the *high performance*. These two criteria are specialisations of the "*has*" definition, that we will call *content-based* perspective, while the social criteria represents the "*shows*" definition, that we will call *social-based* perspective.

Several works already provide approaches to model expertises and retrieve experts. Pavel and Djoerd [17] exploit documents produced by people to build a language model for each person and infer to which extent this person has contributed to the document, which helps to evaluate the expertise of this person in the topics related to the document. Similarly, Mockus and Herbsleb [14] tackle the expert finding problem in collaborative software engineering, where the amount of code written in a piece of a software appears as a good evidence of his or her expertise in this piece. Taking a more social point of view, Zhang et al. [20] compare several algorithms used to retrieve experts using social networks built from forums of online communities, identifying askers and repliers and exploiting evaluations of the replies provided by participants. Finally, Karimzadehgan et al. [7] exploit at the same time the organisational relationships between employees of a company and the content of e-mails they have sent in mailing lists. This makes it an hybrid solution to expert finding and the closest work to our approach, to the best of our knowledge. All these works can be classified as taking a content-based perspective [14,17], a social-based perspective [20] or both [7].

2.2 Stakeholder Recommendation in RE

A recommendation system (RS) is a software application which provides items estimated to be valuable for a given user task in a given context. RSs have been widely used in e-commerce to provide users with personalised product, content or service recommendations (e.g. Amazon product recommendation, MovieLens movie recommendation) [12]. While a few works consider the expert finding problem in RE [18], by generalising to stakeholder recommendations one can find works and literature reviews comparing them [15]. Relying on these reviews, we can identify two main approaches that we can relate to the content-based and social-based perspectives identified so far.

[3] `http://www.collinsdictionary.com/`
[4] `http://www.oxforddictionaries.com/`
[5] `http://dictionary.cambridge.org/`
[6] `http://www.merriam-webster.com/`

A first approach comes from Castro-Herrera et al. [1], where the participation of the stakeholders in a forum is exploited to evaluate their knowledge on different topics. Since several threads can be related to the same topic or one thread can mix several topics, they cluster the messages by *topic* depending on their common *terms* (resulting in abstract topics represented as vectors of terms). Then, looking at the authors of the messages, the stakeholders are related to the topics in which they participate. The result is that other stakeholders can be recommended to participate in a new topic by identifying its similarity with already existing ones. From the expert finding point of view, we can see this approach as a way to exploit the knowledge provided by the stakeholders through their contributions to identify their topics of expertise, illustrating the *content-based* perspective.

In a second approach, Lim et al. have worked on StakeNet [11] for the aim to prioritise the requirements to implement depending on how the stakeholders rate them. For this aim, starting from a reduced set of well-identified stakeholders, each of them suggests people that he or she assumes to have some influence on the project. A *role*, like student, security guard or director, and a level of *salience*, a value on a scale between 1 and 5, are provided to describe how the suggested stakeholder influences the project. Based on these suggestions, a *social network* is built and classical measures are applied to evaluate the global influence of each stakeholder. From the expert finding point of view, we can see this approach as a way to evaluate the expertise of a stakeholder by aggregating the suggestions of other stakeholders, illustrating the *social-based* perspective.

3 Motivation and Problem

In the RE literature, we can find two approaches illustrating well the content-based and social-based perspective separately. However, due to the complementarity of these perspectives, a hybrid context needs to consider both of them. For instance, in the context of OSS projects, it is usual to use forums where the community can exchange ideas and discuss about issues or answer questions from newcomers [10], supporting the forum-based approach and its content-based perspective. However, companies providing OSS-related services (e.g. integration, adaptation and training) can be involved in these communities, with only a few members (e.g. representatives) actually participating in the forum, leading to have a whole set of stakeholders ignored. On the other way, companies can exploit the roles of their employees and the feedback from their co-workers to identify who are the relevant people to contact for specific issues [11,19], supporting the social-based perspective. But when considering the participants of a forum or a mailing list, where several thousands of anonymous people can join and leave at any time, personal suggestions of other stakeholders is unable to cover the whole picture. Especially when indirect evaluations, such as message evaluation, is not available, like in mailing lists.

Consequently, we aim at improving expert finding in RE by designing a more comprehensive approach, also integrating the content-based and social-based perspectives already developed in the current state of the art. To do so, we provide a new model which reuses the concepts exploited in the existing approaches, namely *stakeholders*, *roles*, *topics* and *terms*, and relate them based on evidences extracted from available sources of data. Then, we use *Markov networks* (MN), a technique computing probabilities based on graphical models (probabilistic models based on graphs), to infer the probability for each stakeholder to have some expertise and build corresponding rankings. Karimzadehgan et al. [7] also provides an hybrid solution which considers topics and terms in a probabilistic way, but we additionally consider *roles* in our model, while they consider social relationships only as a way to post-process the probabilities. Another main difference is that, using MNs, we can adapt the query to the needs of the user, such as considering several topics or integrate specific terms and roles, while the inference technique in their approach is based on a single topic.

4 Approach

4.1 Concepts and Relations

As we aim to recommend experts, we have to consider the people who will be recommended. In RE, the people involved in a project are usually called *stakeholders* and, because we consider that any person involved in a project is a potential expert to work with, we will use the same term in our approach. Each stakeholder can have one or several *roles*, such as being a developer or a manager in a company, but also being a contributor in the forum of an OSS (more specific ones can be considered). Each stakeholder can also know about some *topics*, such as security, community management, interface or specific features of the OSS. Going further, we can see that each stakeholder uses *terms*, whether it is in his contributions in the forum or in official documents he redacts as employee of a company involved in an OSS.

At this point, we have stakeholders who are related to roles, topics and terms. In our approach, we go further by exploiting the fact that knowing about a topic, like *interface*, implies generally to know some terms related to this topic, like *interface* (the name of the topic itself), *button*, *screen* and so on. In the same way, having a specific role, like *developer*, implies generally to know about some specific topics, like *interface* and *programming*, and to use specific terms. We exploit all these relations in our model to describe the expertises of each stakeholder.

Once all the concepts we consider and their relations have been presented, we have to consider the outcomes we build from them. First of all, we define an *expert* using a relative point of view: being more expert than another person means having more expertise compared to this person. This definition takes the side of *relative* experts rather than of *absolute* ones, as described by Chi [5] (chapter 2), with the latter considering people *above a threshold* to be experts even if nobody reaches this threshold in the considered community. For the notion of *expertise*, we use the definition provided by the Oxford Dictionaries by considering the expert knowledge or skill in a particular field, thus the topics she knows, the terms she uses, but also the roles she has, which supports the presence of both knowledge and skills. Consequently, someone having more expertise than someone else in a particular field is considered as more expert in this specific field, and similarly someone ranked as more expert in a field is assumed to have more expertise.

All the concepts introduced here are described in Figure 1. The relations are directed to clarify their interpretation, but we consider the relations in both directions (e.g. a term is related to a topic as well as the topic is related to the term).

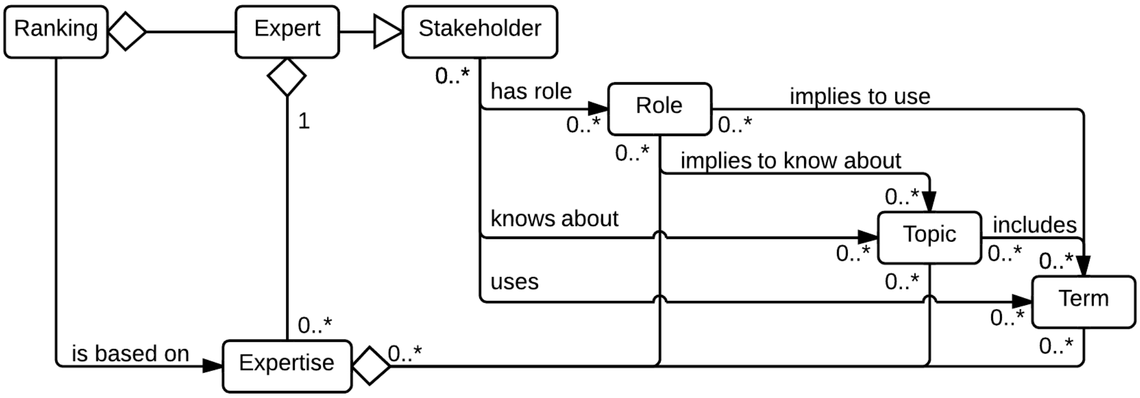

Fig. 1. UML model of the concepts and relations of our approach.

4.2 Model

In order to model the experts and their expertises, we use a *weighted graph* representing the instances of the concepts and relations previously defined. We have a set of stakeholders S, a set of roles R, a set of topics T and a set of terms C which correspond to the nodes of the graph, and a weighted edge for each relation between these nodes. Basically, each stakeholder in S is related to all elements in R, T and C, each role in R is related to all elements in S, T and C and equivalently for each topic in T and each term in C, forming a complete 4-partite graph, as shown in Figure 2. The weight of an edge represents the amount of evidences supporting the corresponding relation. For instance if we have no evidence that a stakeholder $s \in S$ knows about a topic $t \in T$, these nodes are related by an edge with a zero weight written as a tuple $\langle s, t, 0 \rangle$. Having the tuples $\langle s_1, t, 5 \rangle$ and $\langle s_2, t, 10 \rangle$ describes two relations showing that we have twice the amount of evidence that s_2 knows about the topic t compared to s_1.

The actual value of the weight depends on the interpretation of *evidence*. Lim et al. [11], in their social network, use the salience elicited from the stakeholders to weight their edges, while Castro-Herrera et al. [1] exploit the frequencies of appearance of terms and normalise them in vectors. Both these approaches

Fig. 2. Examples of models with 1 node (left) or 2 nodes (right) for each S, R, T and C, showing the different relations and the 4-partite structure (only nodes of the same type are not related).

as well as others can be exploited, with the main challenge being to have consistent weights depending on the sources used to retrieve them. Considering the technique we use, described in Subsection 4.3, the inference is independent on the scale, so having the weights 5 and 10, or respectively 1 and 2, have no influence on the results. Consequently, any arbitrary scale can be chosen for a given category of weight, and the remaining challenge is to merge weights representing different categories, like merging salience and frequencies. This is still an open problem in our approach, thus we consider that the weights are already consistent and can be simply added to have the total amount of evidences.

4.3 Expert Ranking Based on Markov Networks

Before to introduce the technique we use to build expert rankings, we introduce some basic notions of the MN, also called Markov random field. A *random variable* is a variable having several possible states, each with a specific probability, such as a binary random variable x having a state in $V_x = \{\top, \bot\}$ with $P(x = \top) = 0.8$ and $P(x = \bot) = 0.2$. By representing the variables as nodes in a graph and by linking them, we can identify for each complete sub-graph contained in this graph a set of fully-connected variables called a *clique*. On each clique $g = \{x_1, ..., x_n\}$, where x_i can take any state in V_i, we can define a *potential function* $f_g : V_1 \times ... \times V_n \to \mathbb{R}^+$ which returns a value based on the state of the variables in the clique. Finally, a MN $N = (X, F)$ is defined via a set of random variables, $X = \{x_1, ..., x_n\}$, and a set of potential functions over cliques in X, $F = \{f_1, ..., f_m\}$.

Notice that we do not explicit the links between the nodes in the definitions of the network, as they are already defined through the cliques (each clique implies that we have all the possible links between the variables concerned). In the specific case where all the potential functions are defined on pairs of nodes, the MN represents a weighted graph, where the weights of the links depend on the states of the nodes. In this paper, we translate a tuple $\langle n_1, n_2, w \rangle$ in a potential function f such as $f(n_1 = \bot, n_2 = \bot) = f(n_1 = \bot, n_2 = \top) = f(n_1 = \top, n_2 = \bot) = 0$ and $f(n_1 = \top, n_2 = \top) = w$ (we consider other types of potential functions for future works). Consequently, we can represent our model as a MN, where the nodes are represented by binary random variables and the weighted edges are translated into potential functions, building a MN involving a lot of loops due to the completeness of the 4-partite graph used. The state of each node tells whether the node (stakeholder, role, topic or term) is relevant or not (respectively $\top$ or $\bot$), and, in the specific case of stakeholders, meaning that he or she is an expert or not.

The aim of MNs is to compute probabilities based on these random variables and potential functions. Considering the nodes $X = \{x_1, ..., x_n\}$, where x_i is assigned a state $v_i \in V_i$, and each clique g_i assigned to a potential function f_i, the probability to be in a specific state $\chi = \{v_1, ..., v_n\}$ is computed as $P(\chi) = \frac{\prod_{i=1}^{m} f_i(g_i)}{Z}$ where $Z = \sum_{\chi} \prod_{i=1}^{m} f_i(g_i)$ is the normalisation factor which allows to build a probability ($\sum_{\chi} P(\chi) = 1$). If we are interested in a subset of variables, it is possible to compute a partial probability by summing all the cases for the remaining variables, for instance $x = \{x_1, x_2\}, P(x_1 = \top) = P(x_1 = \top, x_2 = \top) + P(x_1 = \top, x_2 = \bot)$. Finally, assuming that some variables have a given state, we can compute a conditional probability, where the computation is done only with the configurations where the given states hold (including the normalisation factor). For instance, with the combinations having $x_2 = \top$:

$$P(x_1 = \top | x_2 = \top) = \frac{\prod_{i=1}^{m} f_i(g_i)|_{x_1=\top, x_2=\top}}{(\prod_{i=1}^{m} f_i(g_i)|_{x_1=\bot, x_2=\top}) + (\prod_{i=1}^{m} f_i(g_i)|_{x_1=\top, x_2=\top})}$$

An interesting property is its *scale independence*: if we apply a scaling factor α on the potential functions $f'_i = \alpha.f_i$ and compute the probability P' based on these functions, we can see that we get the same results:

$$P'(\chi) = \frac{\prod_{i=1}^{m} f'_i(g_i)}{Z'} = \frac{\prod_{i=1}^{m} \alpha.f_i(g_i)}{Z'} = \frac{\alpha^m \prod_{i=1}^{m} f_i(g_i)}{Z'}$$

$$Z' = \sum_{\chi} \prod_{i=1}^{m} f'_i(g_i) = \sum_{\chi} \prod_{i=1}^{m} \alpha.f_i(g_i) = \alpha^m \sum_{\chi} \prod_{i=1}^{m} f_i(g_i) = \alpha^m Z$$

$$P'(\chi) = \frac{\alpha^m \prod_{i=1}^{m} f_i(g_i)}{\alpha^m Z} = \frac{\prod_{i=1}^{m} f_i(g_i)}{Z} = P(\chi)$$

This property is of particular importance because it reduces the problem of data merging by allowing us to choose any arbitrary scale as a reference and to re-scale data extracted with different scales to this reference. The remaining problem is to consider the trust or reliability of the data, which is still an open problem in our approach.

Using this technique on our model, we build a *query* based on which kind of expert is searched, for instance someone knowing about the topics $t_{security}$ and $t_{cryptography}$. These topics provide the condition we want to match, thus the probability for a stakeholder s to be an expert is $P(s = \top | t_{security} = \top, t_{cryptography} = \top)$. By computing these probabilities for all the stakeholders, we are able to rank them from the most to the least probable expert on the corresponding topics. It is possible to combine as much topics as wanted for the query, as well as other elements like roles and terms.

4.4 Recommendation Process

In order to recommend stakeholders as experts, we need to build our model based on *sources of data*, from which we should be able to retrieve all the elements used as nodes in our graph-based models (stakeholders, roles, topics and terms) and their weighted relations. These sources can be document-based, like forums, e-mails or reports, or other models, like goal-models or social networks built from social recommendations. Based on these sources, the necessary extractors have to be designed: a *node extractor*, which retrieves the nodes, and a *relation extractor*, which retrieves the weighted relations between these nodes (e.g. algorithms 1 and 2 in Section 5). With these extractors and the sources as inputs, we first extract all the nodes using the node extractors on each source of data, before to extract all the relations using the relation extractors. We split the extraction process because we do not make any assumption on which source will provide the relevant nodes and relations and in which order they will be parsed: by extracting the nodes first, we ensure that the relation extraction step will consider all the relevant nodes. After each extraction step we aggregate the elements extracted from the different sources: a simple union for the nodes and a merging of the similar relations by summing their weights, merging from instance $\langle s, t, 2\rangle$ and $\langle s, t, 3\rangle$ into $\langle s, t, 5\rangle$.

After the nodes and relations are extracted, we can build the MN, as described in Subsection 4.3, and query it. To build the query, the properties searched for the experts to recommend (having some roles, knowing about some topics or using some terms) should be present in the network nodes. For instance, if the network contains a topic *security*, it is possible to query for an expert in this topic, possibly combining it with other topics, but also with roles and terms. If the element is not present, it cannot be queried and an equivalent need to be found, e.g. *cryptography*, which is in the network. In our approach, when we look for an expert in a topic which is not in the network, we replace this topic by the corresponding term if it exists, otherwise we ignore it. Notice that querying for an expert with a given role does not mean that only people having this role will be considered (it is not a filtering function), but that people being more related to this role (directly or indirectly, as described in the model) will be considered as more experts.

Once the MN and the query $Q = \{x_1, ...x_q\}$ are built, the probability of each stakeholder $s \in S$ to be an expert is computed as a conditional probability based on the query $P(s = \top | x_1 = \top, ..., x_q = \top)$. The stakeholders are then ranked by decreasing order of probability to infer the ranking of recommendation. The recommendation can be enriched with the probabilities to provide an evaluation of the recommendations, so that a ranking like $((s_1, 0.98), (s_3, 0.95), (s_2, 0.43), (s_4, 0.22))$ allows to select only the first two as potential experts because of their high probability. An important remark is that, as we consider relative expertise, the main information provided by our rankings is not which rank is assigned to which stakeholder, but which stakeholder is ranked higher or lower than another. In particular, the rankings (A, B, C, D) and (D, A, B, C) fully agree on (A, B, C) and disagree on the ordering of D compared to the others. By considering rank comparisons, these two rankings are completely disagreeing, which is not our interpretation here. Moreover, having a partial ordering (several stakeholders at the same rank) leads to have less informative rankings, because no order is provided between some stakeholders and the interpretation of lack of information to rank them is more natural than a strictly equal expertise.

The complete process is illustrated in Figure 3.

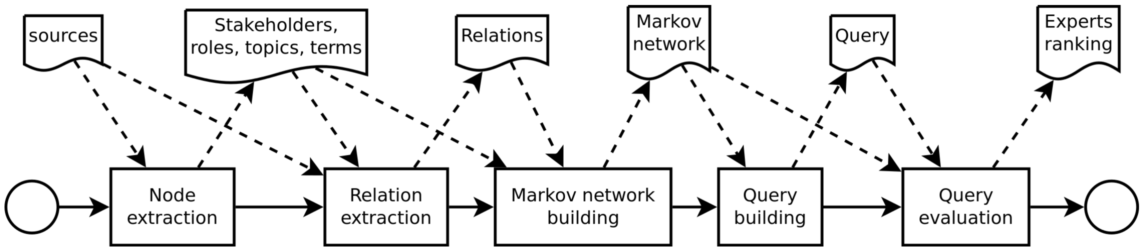

Fig. 3. Recommendation process, from left to right, with the artefacts on the top and the tasks on the bottom. The directed arrows shows the inputs and outputs of each task.

4.5 Supporting Tool

The approach has been implemented in Java and external libraries have been used to extract the data from the sources and compute the MN. We retrieve nouns to identify terms and topics using the software GATE [3], a free and open source Java software to manage text processing with natural languages. It was chosen because it appears as a reference regarding natural language processing, aggregating well known tools like Lucene and WordNet and providing a complete extraction process. The MN is built and the queries are evaluated using libDAI [6], a free and open source C++ library made to compute graphical models. It was chosen because, among all the tools or libraries able to compute graphical models like Bayesian networks and MNs, it was one of the few able to compute MNs in particular and the only one explicitly supporting loops, which is a major constraint considering our loop-intensive models described in Subsection 4.3. The global architecture of our tool is presented in Figure 4.

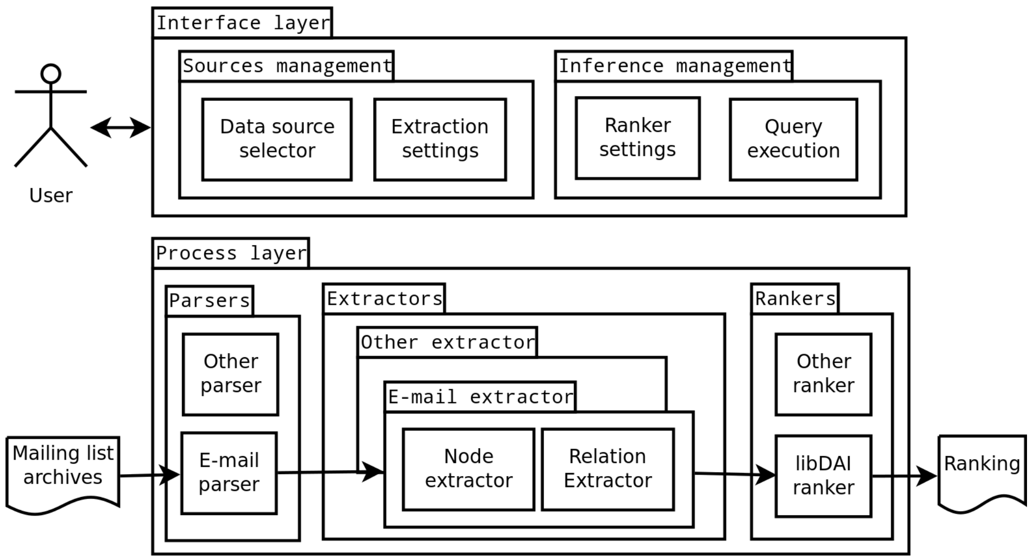

Fig. 4. Architecture of the implemented tool.

5 Preliminary Experiments

5.1 Illustrative Example

In order to stress our approach in a controlled situation, we have lead an experiment with 3 participants Alice, Bob and Carla, discussing via e-mail about 2 cooking-related threads : Asian food and European dessert (the names and topics have been renamed to preserve anonymity). The experiment has started just after its presentation to the participants and has lasted 2 days during which 30 messages were exchanged, with 8 contributions from Alice (4 for Asian food, 4 for European dessert), 9 from Bob (4 for Asian food, 5 for European dessert) and 13 from Carla (6 for Asian food, 7 for European dessert). To build a gold standard, the participants were asked to fill a form for each discussion after the experiment, asking for their level of knowledge in the discussion (newbie, advanced, expert) and the most knowledgeable participant from their point of view. Bob and Alice were identified as experts in European dessert while Carla was the expert in Asian food.

In order to build our model, we have designed a *node extractor* and a *relation extractor* which fit to our source of data: the set of e-mails exchanged. The node extractor uses Algorithm 1 to consider an authors as a *stakeholder*, the nouns in the subject of the e-mails as *topics*, and the nouns in the body as *terms*. We did not consider *roles* in this experiment, but the MN does not differentiate the types of nodes: using roles in place of topics for instance, assuming the weights are the same, leads to the exactly same result. Thus, while they should be considered for a proper validation, we claim that it is not critical for our preliminary experiments. The relation extractor uses Algorithm 2 to relate the terms in the body and the topics in the subject to the author, as well as the terms and topics together.

Algorithm 1 Node extractor for e-mails.

Require: $mail$: Natural language e-mail
Ensure: S, R, T, C: Extracted stakeholders, roles, topics and terms

$S \leftarrow \{stakeholder(authorOf(mail))\}$
$R \leftarrow \emptyset$
$T \leftarrow \{topic(x) | x \in nounsOf(subjectOf(mail))\}$
$C \leftarrow \{term(x) | x \in nounsOf(bodyOf(mail))\}$

Algorithm 2 Relation extractor for e-mails.

Require: $mail$: Natural language e-mail
Require: S, R, T, C: Stakeholders, roles, topics and terms
Ensure: L: Weighted relations

$L \leftarrow \emptyset$
$a \leftarrow author(mail)$
if $stakeholder(a) \in S$ **then**
  for all $t \in termsOf(bodyOf(mail))$ **do**
    if $term(t) \in C$ **then**
      $L \leftarrow merge(L, \{\langle stakeholder(a), term(t), 1\rangle\})$
    end if
  end for
end if
for all $topic \in T$ **do**
  if $nounOf(topic) \in nounsOf(subjectOf(mail))$ **then**
    $L \leftarrow merge(L, \{\langle stakeholder(a), topic, 1\rangle\})$
    for all $t \in nounsOf(bodyOf(mail))$ **do**
      if $term(t) \in C$ **then**
        $L \leftarrow merge(L, \{\langle topic, term(t), 1\rangle\})$
      end if
    end for
  end if
end for

The extraction process has identified 3 stakeholders, 4 topics, 293 terms and 2063 relations, building a MN of 300 nodes and 2063 functions. The 4 topics include the 2 more than the expected ones because the participants were asked to launch the discussions by themselves, letting them formulate the discussion subjects. The results, displayed in Table 1, consider an empty query aiming to ask for expert without specifying any constraint (all the nodes in the MN can have any state), while the query with "European dessert" or "Asian food" implies to ask for experts in the corresponding topics (restricting these topic nodes to the state $\top$). Alice and Bob appear as more expert than Carla on European dessert, while it is the opposite on Asian food, as shown by our gold standard.

Stakeholder	$Q = \emptyset$	Rank	GS	Q =European dessert	Rank	GS	Q =Asian food	Rank	GS
Carla	0.50088	1	-	0.49941	3	2	0.49978	1	1
Bob	0.49959	2	-	0.49969	2	1	0.49946	3	2
Alice	0.49908	3	-	0.50106	1	1	0.49975	2	2

Table 1. Results of the experiment. The rank can be compared to the gold standard (GS) to check the fitness.

5.2 OSS-Based Experiment: XWiki

While the previous experiment allows to have a better control on the data, another experiment has been run on an OSS project to fit better to the contexts targeted by this approach. Our tool has been applied on the mailing list of the XWiki[7] OSS community, which involves also a company selling support and training on this OSS. We have used the mailing list archives[8] from January to May 2013, retrieving 805 e-mails in 255 threads. We did not use *roles* in this preliminary experiment, but we plan to do so in future works by exploiting some data available from the XWiki community, as discussed in Section 6.

In order to build our model, we have used the Algorithm 1 and Algorithm 2 to extract the nodes and relations from the e-mails. An additional effort has been made to clean the data, especially to identify unique authors by aggregating different e-mail addresses for similar names of author, and to remove noise in the body of the e-mails like quotations. However, this process still need to be improved because some noise, like huge source code excerpts, is removed manually by forbidding special terms which appear in this noise. In order to make the computation tractable, we systematically remove the nodes having the smallest total weight (i.e. summing all the weights of its relations) and their relations. We do so in an iterative way until we reach a targeted configuration (e.g. 10 stakeholders, 10 topics and 100 terms).

The extraction process has identified 120 stakeholders, 216 topics, 4854 terms and 75470 relations, and different reduction policies and potential functions have been applied to build the MN. Some preliminary experiments have shown that we are able to build a ranking from this dataset and we were able to assess the coherency of some results by having obvious experts like main contributors generally highly ranked, and participants of specific discussions highly ranked for the topics of their discussions. We did not build a proper gold standard, but we plan as future work to build one based on the rankings provided by some requirement analysts. For instance, by reducing the network to 5 stakeholders, 10 topics and 100 terms, we come with two committers, *committer 1* (a main committer) and *committer 2*, and three other forum participants, *participant 1*, *participant 2* and *participant 3*. During the time span considered, *committer 1*, *committer 2* and *participant 1* have discussed about data migration issues during an XWiki upgrade on the forum. By querying the MN on "Data migration", which are topics available after the reduction process, these 3 stakeholders are generally ranked as the top 3. More investigation is needed to improve and assess the validity of our approach in this context, but these preliminary experiments provide a good support.

6 Discussion

The illustrative example and the OSS-based experiment suffer a lot of limitations due to, respectively, their restricted context and preliminary state. However, the aim of this paper is to present our approach and to illustrate it and show potential benefits by exploiting the results we were able to get from these

[7] `http://dev.xwiki.org`
[8] `http://lists.xwiki.org/pipermail/users/`

experiments, saving the proper validation for future work. In particular, the OSS-based experiment provides a relevant context for our purpose but, due to the amount of noise (e.g. systematic quotations, huge source code excerpts) and the difficulty to remove it systematically, we need to use more advanced techniques to improve our results. Moreover, while the presented experiments do not use any role, the OSS-based experiment provide some sources of data that we can use to retrieve and relate them, such as a Hall of Fame which describes specific types of committers and contributors, and organisational models which describe the different types of actors involved.

Looking at the approach, several points can be discussed. First of all, a MN computation is not scalable due to the computation of the full graph. We can see it particularly well in the OSS experiment where the reduction of the network was mandatory to use the libDAI library. Another way to compute our model could be to compute part of the MN in a smarter way, using optimisation techniques, or to use other techniques which are more local like social network measures (e.g. PageRank, degree centrality) or search based techniques (e.g. hill climbing, genetic algorithms). Another point is the lack of relation between two nodes of the same type in our model: we could consider for instance that topics are co-related, such as "security" and "cryptography", or two stakeholders working in the same office or on the same OSS module are related. We can also discuss the querying process, where asking for an expert in a specific role does not mean that only people having this role will be ranked due to the probabilistic property of our approach. Such filtering behaviour could be considered to improve its adaptivity, for instance weighting the elements of the query to give them some importance. Finally, we can discuss the interpretation of the probabilities used to infer the ranking, especially their proximity leading to a probable lack of robustness, or the inability to identify a clear threshold to differentiate actual experts from novices. We are currently looking at other potential functions and information to exploit in our data to tackle this problem.

We can also consider related works to improve or extend our approach. For instance, Massa and Avesani [13] describe trust metrics for recommendations based on collaborative filtering, which could inspire us for the merging of the weights coming from different sources. In particular, the frequency of the term in a forum, which can be above thousand, compared to roles provided by official documents, which can be one or two evidences, implies to use some normalisation methods and trust metrics. Yarosh et al [19] provide a taxonomy which includes roles and topics, but also other concepts and classifies them as selection criteria or tasks to achieve, which could be interesting to extend the expressiveness of our approach. Finally, we consider that the scope of this approach can also be discussed because, although we focus on RE works to inspire us, we could imagine to use it in other domains or to use more restrictive assumptions which holds in RE to improve the performance of our approach.

7 Conclusion

This paper focuses on expert finding to improve the support of RE processes in large and dynamic contexts like OSS projects. We show how current approaches in RE, based on forums and social networks, relate to two complementary perspectives to evaluate expertises, namely content-based and social-based perspectives. We provide a novel approach by combining and enriching concepts from these works to build a model that we translate into a Markov network to infer the probability that stakeholders have some searched expertises. We show in an illustrative example how this approach can be successfully applied and present a preliminary experiment in a real OSS case, before to discuss the results and limitations of our approach.

As future work, we plan to improve our approach by considering normalisation and trust metrics to manage heterogeneous sources of information as well as to optimise the MN computation or to use more local techniques to improve the scalability. The query expressiveness should be also improved to allow a better control of the inference process and adapt the results to the needs of the user of our approach. Other potential functions are investigated in order to improve the confidence and robustness of our rankings. Finally, we plan to further exploit the data provided by the OSS case to exploit roles and to design a complete case study.

Acknowledgements. The authors are grateful to Itzel Morales-Ramirez for her help in exploiting the tool GATE for the nouns extraction. This work is a result of the RISCOSS project, funded by the EC 7th Framework Programme FP7/2007-2013, agreement number 318249.

References

1. Carlos Castro-Herrera and Jane Cleland-Huang. Utilizing recommender systems to support software requirements elicitation. In *Proc. of the 2nd International Workshop on RSSE*, pages 6–10, New York, USA, 2010. ACM.
2. Betty H. C. Cheng and Joanne M. Atlee. Current and future research directions in requirements engineering. In *Design Requirements Engineering: A Ten-Year Perspective*, volume 14 of *LNBIP*, pages 11–43. Springer Berlin Heidelberg, 2009.
3. Hamish Cunningham, Kalina Bončeva, and Diana Maynard. *Text Processing with GATE.* University of Sheffield Dept. of Computer Science, Sheffield, 2011.
4. D. Damian, L. Izquierdo, J. Singer, and I. Kwan. Awareness in the wild: Why communication breakdowns occur. In *Second IEEE ICGSE, 2007*, pages 81 –90, August 2007.
5. K. Anders Ericsson. *The Cambridge Handbook of Expertise and Expert Performance*. Cambridge University Press, June 2006.
6. Joris M. Mooij. libDAI: a free and open source C++ library for discrete approximate inference in graphical models. *JMLR*, 11:2169–2173, August 2010.
7. Maryam Karimzadehgan, Ryen W. White, and Matthew Richardson. Enhancing expert finding using organizational hierarchies. In *Advances in Information Retrieval*, number 5478 in LNCS, pages 177–188. Springer Berlin Heidelberg, January 2009.
8. I. Kwan and D. Damian. The hidden experts in software-engineering communication (NIER track). In *Proc. of the 33rd ICSE*, pages 800–803, New York, USA, 2011. ACM.
9. A. van Lamsweerde. *Requirements engineering : from system goals to UML models and software specifications*. Wiley ; John Wiley [distributor], Hoboken, N.J.; Chichester, 2007.
10. Paula Laurent and Jane Cleland-Huang. Lessons learned from open source projects for facilitating online requirements processes. In *Requirements Engineering: Foundation for Software Quality*, number 5512 in LNCS, pages 240–255. Springer Berlin Heidelberg, January 2009.
11. Soo Ling Lim, Daniele Quercia, and Anthony Finkelstein. StakeNet: using social networks to analyse the stakeholders of large-scale software projects. In *Proc. of the 32nd ACM/IEEE ICSE*, volume 1, pages 295–304, New York, USA, 2010. ACM.
12. W. Maalej and A.K. Thurimella. Towards a research agenda for recommendation systems in requirements engineering. In *2nd International Workshop on MARK*, pages 32 –39, September 2009.
13. Paolo Massa and Paolo Avesani. Trust metrics in recommender systems. In *Computing with Social Trust*, HCI Series, pages 259–285. Springer London, January 2009.
14. A. Mockus and J. D. Herbsleb. Expertise browser: a quantitative approach to identifying expertise. In *Proc. of the 24th ICSE*, page 503–512, New York, USA, 2002. ACM.
15. J.G. Mohebzada, G. Ruhe, and A. Eberlein. Systematic mapping of recommendation systems for requirements engineering. In *ICSSP*, pages 200 –209, June 2012.
16. B. Nuseibeh and S. Easterbrook. Requirements engineering: a roadmap. In *Proc. of the Conference on The Future of SE*, ICSE '00, pages 35–46, New York, USA, 2000. ACM.
17. Pavel Serdyukov and Djoerd Hiemstra. Modeling documents as mixtures of persons for expert finding. In *Proc. of the IR Research, 30th ECIR*, page 309–320, Berlin, Heidelberg, 2008. Springer-Verlag.
18. M. Vergne, I. Morales-Ramirez, M. Morandini, A. Susi, and A. Perini. Analysing user feedback and finding experts: Can goal-orientation help? In *iStar 2013 Workshop*, 2013.
19. Svetlana Yarosh, Tara Matthews, Michelle Zhou, and Kate Ehrlich. I need someone to help!: a taxonomy of helper-finding activities in the enterprise. In *Proc. of the Conference on CSCW*, page 1375–1386, New York, USA, 2013. ACM.
20. Jun Zhang, Mark S. Ackerman, and Lada Adamic. Expertise networks in online communities: structure and algorithms. In *Proc. of the 16th international conference on WWW*, page 221–230, New York, USA, 2007. ACM.