CrowdIntent: Annotation of Intentions Hidden in Online Discussions

Itzel Morales-Ramirez*, Dimitra Papadimitriou† and Anna Perini*
*Software Engineering Research Unit
Fondazione Bruno Kessler, Italy
Email: {imramirez, aperini}@fbk.eu
† University of Trento, Italy
Email: papadimitriou@disi.unitn.it

***Abstract*—Stakeholders working in open-source software development use social media, emails or any available means in the Internet to communicate and express what they want or need through the use of text. The recognition of such needs or desires (that we call intentions) is usually done by a human reader, and it can require a considerable effort when the amount of messages in online discussions increases. The problem is that to support an automated recognition of the intentions hidden in the text, data are needed in the domain of software development for training classifiers. However, so far there is no data annotated with intentions that can be used for data mining purposes. In order to tackle the lack of data we have collected online discussions in the domain of software development and asked people to annotate such discussions with intentions. This collection has been performed by crowdsourcing the task of annotating sentences with their hidden intention. In this paper we report the experience of carrying out a crowdsourcing project with a heterogeneous crowd. We discuss how we applied the steps of the crowdsourcing workflow in *CrowdIntent*. Lessons learned and future work are also presented.**

I. Introduction

Communication carried out through the Internet typically rests on the exchange of emails or posts in forums, where Natural Language (NL) text is highly used, and whose production increases in number each minute[1]. Stakeholders working in open-source software development use social media, emails or any available means in the Internet to express what they want or need, to agree on decisions and to collaborate for accomplishing activities in different phases of the software development [1]. We advocate that most of the sentences of such messages can be associated to an intention. For instance, the intention of informing about an event such as an invitation for participating in a conference or that the new version of 'X' software is now available to download. Indeed, intentions are embedded in the sentences we write, either to confirm, suppose, persuade, or only to inform about certain things. However, the recognition of such needs or desires is done through the reading of the emails and implies a cognitive effort from the reader to understand the intentions behind. The problem is that to support an automated recognition of the intentions hidden in the text, it is needed data in the domain of software development for training classifiers. But, so far there is no data annotated with intentions (in the domain of software development) that can be used for data mining purposes. The data available is related to general topics discussed in telephone conversations, as reported by Novielli and Strapparava [2]. In a previous work [3] we have characterized how stakeholders communicate and express intentions through the use of mailing-list discussions. Moreover, in that work we rely on the Speech Act Theory revisited and described in the book of Bach and Harnish [4] to guide the interpretation of text in terms of intentions. Therefore, in order to tackle the lack of data we have collected online discussions (in the domain of software development) annotated with intentions. This collection has been performed by crowdsourcing the task of annotating sentences with an intention. We have designed a crowdsourcing task bearing one goal in mind, i.e. to use the derived annotated dataset to evaluate, improve and extend our previous work regarding intention detection in software development forums. In this work we have suggested a procedure for automatically characterizing sentences based on intentions that uses syntactic and grammatical rules. Details about the different options and the procedure of this characterization can be found in [5]. The identification of intentions can be a valuable tool for many text mining tasks that gather a lot of attention in the software engineering community such as the classification of user posts as bug reports, clarification or feature requests. We will explain this potential in Section VI corresponding to the future work.

The rest of the paper is structured as follows. We first give the definitions of crowdsourcing and intentions in Section II. We briefly mention the related work in Section III. In Section IV we describe the whole annotation procedure and the platform we have used. We report the experience of executing a crowdsourcing project called *CrowdIntent* with a heterogeneous crowd and how we applied the best practices of crowdsourcing. We discuss the results along with some lessons learned in Section V. And we give the conclusion in Section VII.

II. Crowdsourcing workflow and Intentions

Let's first recall the definition of crowdsourcing (definition proposed by Estellés-Arolas et al. [6]) "crowdsourcing is a

†The author has been supported by the ERC Advanced Investigation Grant N.267856

[1] http://mashable.com/2014/04/23/data-online-every-minute/

978-1-4673-7040-0/15 $31.00 © 2015 IEEE
DOI 10.1109/CSI-SE.2015.12

type of participative online activity in which an individual, an institution, a non-profit organization, or company proposes to a group of individuals of varying knowledge, heterogeneity, and number, via a flexible open call, the voluntary undertaking of a task. The undertaking of the task, of variable complexity and modularity, and in which the crowd should participate bringing their work, money, knowledge and/or experience, always entails mutual benefit. The user will receive the satisfaction of a given type of need, be it economic, social recognition, self-esteem, or the development of individual skills, while the crowdsourcer will obtain and utilize to their advantage what the user has brought to the venture, whose form will depend on the type of activity undertaken".

By examining this definition we realize that crowdsourcing is a complex task that needs to be divided into carefully designed stages for achieving a good result. We consider the crowdsourcing workflow described by Sabou et al. in [7]. In this paper the authors discuss the best practices for carrying out a crowdsourcing project in four main stages.

- The first stage refers to the *project definition* and consists of: (a) selection of a natural language processing (NLP) problem (e.g., sentiment analysis) and crowdsourcing genre (e.g., the work can be a mechanized labor or a game with purpose); (b) decomposition of the NLP problem into tasks to make the tasks understandable by non-experts; and (c) design of the crowdsourcing task.
- The second stage concerns the *data preparation* and consists of: (a) collection and pre-processing of the corpus; (b) building or reusing a platform; and (c) running of pilot studies.
- The third stage is the *project execution* and consists of: (a) recruitment and screening of participants; (b) training of participants; and (c) management and monitoring of crowdsourcing tasks.
- The final stage is the *data evaluation and aggregation* and consists of: (a) evaluation and aggregation of annotations, and (b) evaluation of the overall corpus characteristics.

Regarding the term *intentions*, we adopt the Speech Act Theory (SAT) reviewed by Bach and Harnish [4] to define what is an intention. According to them when a person says something she/he attempts to communicate certain things to the addressee, which affect either their believes and/or their behavior. In other words, a speaker's utterance bears an intention that is aimed to affect the hearer's believe or behavior. For example in the following sentence "I have attempted to deconstruct the toucan.css file under the skins directory in the web app and then build it back together." the intention behind is *assertive*. Since this intention is defined as a speech-act that is considered as having a strong belief and intention by a sender who maintains his/her belief about something. We then have characterized the online discussions in terms of the emails' body using SAT, where each sentence expressed by a stakeholder can be assigned an intention. We have explained the characterization in a previous work [5], but for the evaluation of this work we have designed and executed *CrowdIntent* to collect data.

III. Related Work

In this section we give a brief overview of the works that have applied crowdsourcing for some of the different phases of the software development process. To the best of our knowledge there is not a vast exploration for applying crowdsourcing in the requirements engineering research community. However, there is interest towards social media as distributed, collaborative work enablers. For instance, the discovery of stakeholder communities by using concept lattices to extract hidden profiles for the set of requirements of a certain project is a research work by Azmeh et al. [8]. StakeSource [9] is a web-based tool that automates stakeholder analysis. It "crowdsources" the stakeholders for recommendations about other stakeholders and aggregates their answers using social network analysis [9]. This approach supports the stakeholder identification. Renzel et al. [10] have presented a software platform called Requirements Bazaar that supports gathering and negotiation on user feedback about software applications. In our case, the final purpose of collecting online discussions annotated with intentions is the understanding of the way the bug reports or features requests are written in terms of intentions, then supporting an automated analysis.

As the identification of intentions in text, the identification of sentiments is considered a subjective activity suitable to be crowdsourced and potentially used for the triage of requirements. For instance, Mellebeek et al. [11] reported a crowdsourcing work for requesting the annotation of sentences containing user opinions in Spanish and label each opinion as positive, negative or neutral. Another crowdsourcing work is the platform PeoplePerHour[2] with a different configuration from regular platforms. In this platform people offer their services to produce a specific software product, component, or things related to the software development. These services can be exploited in different ways during the whole process of the development of the project; for instance, to outsource the design of a database, the development of prototypes, migration of databases. uTest[3] is a marketplace for software testing services, offering real-world QA services with a community of 14,000 professional testers, so this platform supports the test process.

Research work about testing using crowdsourcing is described in a paper by Pastore et al. [12], where the crowd had to assess if oracles generated automatically are correct or not with reference to the documentation provided for the task. There are other research works aiming at exploiting crowdsourcing in different phases of the software development either as the primary tool for solving a problem or as a means for collecting data for further research. Although our work is positioned in the second category, it is important to pinpoint the benefit for the research community that our collected data will bring.

[2] http://www.peopleperhour.com

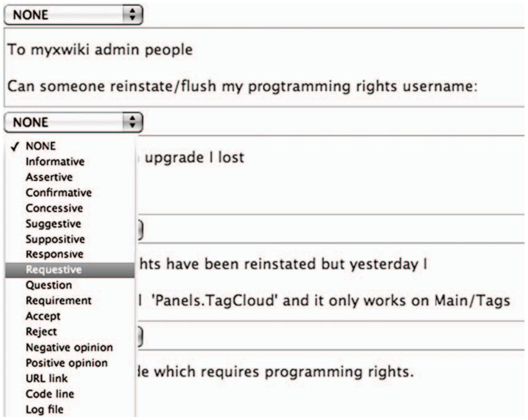
[3] http://www.utest.com/how-it-works

IV. CrowdIntent: the process, the platform and the crowd

We report the experience with our project called *CrowdIntent* in this section. We followed the best practices mentioned in Section II, though sometimes external factors lead to some deviations during the execution. Next, we detail each stage of the crowdsourcing workflow.

a) Project definition: Our motivation for performing a crowdsourcing activity was the need of counting on online discussions of stakeholders annotated with intentions. Since the task of reading and interpreting what are the intentions behind a message is a subjective task (as exemplified by Sabou et. al [7]), we opted for crowdsourcing this job.
The task was designed as follows: (0) the NLP problem was defined as *identification of intentions in online discussions* and the genre was a mechanized labour; (1) we decomposed the problem into micro tasks, this means that we decided to divide each online discussion into messages and then into sentences; (2) we displayed a task consisting of a set of sentences corresponding to one message of an online discussion; (3) the first sentences being shown were referring to the first message of a discussion; (4) the following sentences corresponded to the subsequent messages and discussions (the number of sentences per message varied in number); (5) each sentence was accompanied with a list of intentions (only one intention could be selected. Notice that this is a design constraint from the platform we decided to use. Our choice was indeed driven by the need of minimizing development and learning time, and the platform we reused has been developed, for different purposes, by one of the authors. (6) the number of planned participants was 38 and we wanted to assign three participants per task to compute the agreement among participants with the statistical measure Fleiss' Kappa [13], which is used for computing the agreement for more than two annotators; (7) the number of categories (i.e., intentions) to be displayed was 18 (see Figure 1). Notice that most of the categories of intentions are described in the book of Bach and Harnish and we have added new categories found in online discussions, such as code line, log file and URL link. (8) the rewarding could not be feasible due to a lack of budget, therefore the participation would be voluntary and altruist; (9) finally, we planned an approximated time of the overall job to be around 1 hour and 30 minutes.

b) Data preparation: We have randomly selected 20 online discussions from the archives of mailing-list discussions about the XWiki platform[4]. We have organized the threads into discussions by grouping the emails based on their Subject. The cleaning of the data has been performed by eliminating "the replies" identified in the email's body with an initial character '>' followed by a white space; then the body content has been enclosed with the characters '<' and '>' and we have removed identified signatures complying with the regular expression pattern "\n+−−\n(.*)>$"; we have performed an automatic elimination of useless sentences containing dates, considered as not significant, using again regular expressions based on patterns such as, "On 1 Feb 2012 at 10:51:59…"; we have put the key word CODE_LINE for identified lines of code (but not exhaustively); some html codes in single lines (e.g., &gt, &amp) have been removed; as well as URL links of participant's personal web site; finally we have replaced emoticons such as ":-)" and ":-(" for the key words SMILEY and SAD. We have used the tool ANNIE sentence splitter[5] to divide the discussions into sentences, resulting in 1685. We then created and populated a MySQL database.



To myxwiki admin people

Can someone reinstate/flush my progtramming rights username:

upgrade I lost

hts have been reinstated but yesterday I

I 'Panels.TagCloud' and it only works on Main/Tags

le which requires programming rights.

Fig. 1. Platform: part of the screen showing the sentences and the list of intentions to select

We show part of the platform in Figure 1, as it can be seen, the platform displays the sentences pertaining to each message of the online discussions and the list of intentions. Besides this, in the platform there were written the three steps to perform the task, namely, Step 1: Read every sentence, Step 2: Select from the dropdown list above each sentence the label you think that represents better the sentence, Step 3: After labeling all the sentences save your work using the button below and go on with the next message. After setting up the platform and populating the database we ran a pilot execution to detect any flaws with respect to the platform and we fixed them.

c) Project execution: For recruiting the participants we sent email invitations to PhD students and Post-doc researchers mainly in the field of Computer Science, and to people working in the Information Technology industry. We informed them that the activity would be performed through an online platform and that it would require approximately 1 hour and 30 minutes to be completed. We did not specify time constraints, although we expressed our expectation to collect data after a week.

Twenty subjects accepted the invitation resulting in a heterogeneous crowd of 2 participants from Brazil, 2 from Germany, 1 from The Netherlands, 2 from Ethiopia, 1 from Mexico, 3 from China, 1 from Colombia, 1 from Paraguay, 1 from

[4] XWiki is an OSS generic platform for developing collaborative applications, see http://platform.xwiki.org/xwiki/bin/view/Main/WebHome. Archives available at: http://lists.xwiki.org/pipermail/users/.

[5] https://gate.ac.uk/sale/tao/splitch6.html#x9-1420006.4

TABLE I
Cohen's Kappa coefficient per group for the 16 and 8 categories. Gray-coloured cells are the best obtained results interpreted according to the scales of Landis and Koch (*Moderate* and *Substantial* for the G_6 with 8 categories) and the scale Green (*Fair/Good*)

#Categories	$\mathbf{G_1}$	$\mathbf{G_2}$	$\mathbf{G_3}$	$\mathbf{G_4}$	$\mathbf{G_5}$	$\mathbf{G_6}$	$\mathbf{G_7}$	$\mathbf{G_8}$	$\mathbf{G_9}$	$\mathbf{G_{10}}$
16	0.38	0.31	0.15	0.33	0.22	0.51	0.41	0.28	0.29	0.28
8 (aggregated)	0.49	0.44	0.29	0.34	0.23	0.66	0.48	0.43	0.38	0.39

Ecuador, 4 from Italy, 1 from Turkey, and 1 from Spain. All the participants have an English level required for working in the research environment, which means writing technical and/or research documents in English. Another characteristic of the crowd is that they do communicate through email to convey their needs or wants regarding daily life activities. We grouped them in pairs (labeled as $G_1 \ldots G_{10}$) and assigned them two online discussions to annotate (varying the # of sentences), but they worked individually. Members of pairs were selected randomly. We prepared a short tutorial with instructions in order to guide the participants. We sent individual emails including: a password and a URL link to access the online platform; and a PDF document containing the short guide for performing the annotation. In order to monitor and manage the progress of the participants we were querying the database to find the sentences missing the annotation with intentions.

d) Data evaluation and aggregation: Once all the participants finished the annotation task, we first calculated an approximation of the time spent by participant. We took the time registered in MySQL for the first annotation of sentences and the final annotation, eliminating spans higher than 30 minutes, which we interpreted as pauses. Considering the estimated time, it is interesting to observe how participants of the same group differ in time consumed for annotating the same amount of sentences. We can see the time per participant in Table II. The table reads as follows, the first column (Gpo) represents the group identifier; the second and last columns (Annotator) show the codes assigned to the participants for accessing the online platform that belong to the same group; the third and fifth columns (Min) refer to the time spent by participant, in minutes; and the column (#Sent) shows the number of sentences that each participant was annotating.

TABLE II
Time spent on annotating sentences by participant

Gpo	**Annotator**	**Min**	**#Sent**	**Min**	**Annotator**
G_1	annot1AT	18	84	31	annot2TB
G_2	annot3CU	77	192	102	annot4UD
G_3	annot5EV	25	103	19	annot6VF
G_4	annot7GW	19	104	17	annot8WH
G_5	annot9IX	36	100	7	annot10XJ
G_6	annot11KY	53	187	70	annot12YL
G_7	annot13MZ	146	190	93	annot14ZN
G_8	annot15OA	62	202	54	annot16AP
G_9	annot17QB	123	248	92	annot18BR
G_{10}	annot19SC	74	275	85	annot20CT

On one side, we can highlight for instance that the time between participants of G_5 highly differs, i.e. annotator annot9IX performed the task of annotating 100 sentences in 36 minutes while for annot10XJ it took 7 minutes. We can also mention that annot10XJ is doing a PhD related to software testing topics while the other annotator is a software developer. On the other side, annotators of the groups G_3, G_4, G_8, G_{10} seem to share a similar time effort with small differences of 6, 2, 8 and 11 minutes, respectively. As previously said, the heterogeneity of the participants' background gave us contrasting results that we are still analyzing[6].

Another analysis that we performed was the computation of the Cohen's Kappa coefficient (k value) in order to obtain the percentage of agreement per group. We decided that for the data evaluation we would discard two intentions from the 18 categories presented during the task, ending up with 16 categories. We decided this because the default intention *Informative* as well as the extra label *NONE* do not contribute to the final purpose of our research. We then computed the k value with the 16 categories for the 10 groups and after this we grouped the categories in order to compute the k value with 8 categories for the 10 groups as well. These 8 categories have been derived from the 16 categories by grouping some intentions in abstract classes, considering a similarity of interpretation based on the literature [4]. For instance, the intentions Assertive, Confirmative and Concessive grouped as *Assertive*; Question, Requestive and Requirement grouped as *Requestive*; and URL link, Code line and Log file as *Attach*. Other intentions such as Accept, and Negative opinion were used normally. We interpreted the results of the k values against two scales: the Landis and Koch [14], and the Green [15] scale, respectively, to understand how is the quality of the data in terms of participants' agreement. We show the results in Table I.

The interpretation of the results with respect to the scales are the following: Landis and Koch's scale classify the k value in Slight from 0.0 to 0.2, Fair from 0.2 to 0.4, Moderate from 0.4 to 0.6, Substantial from 0.6 to 0.8 and Perfect from 0.8 to 1.0. For Green's scale k value can be classified in Low from 0.0 to 0.4, Fair/Good from 0.4 to 0.75 and High from 0.75 to 1.0. Based on the results shown in Table I we can observe that only few groups reach a moderate/ fair-good agreement with 16 categories (i.e., G_6 and G_7), but this is improved when we grouped those categories into 8, thus we obtained moderate, substantial and fair-good for groups G_1, G_2, G_6, G_7 and G_8.

V. Discussion and Lessons learned

As clearly observed the reduction of categories in the aggregation stage has improved the agreement of some groups. This

[6] Part of the analysis has been submitted in another paper (under revision), but a technical report can be read at http://selab.fbk.eu/imramirez/TR_CAiSEDec2014.pdf

observation is aligned with the crowdsourcing best practices since one recommendation is to present no more than 10 categories (preferable 7 categories) to the participants [7]. However, we wanted to take the risk of proposing all the categories of speech acts found in the literature [4] and make the comparison during the aggregation step. The results indicate that there is not a strong agreement overall, implying a low quality, thus this leads us to replicate our experiment with an improved design. In the new design we are considering the reduction of categories, a group of people doing the activity not in a distributed setting but in the same room, in this way we can perform a controlled experiment having the people focused only on the task to do. As we saw in the table of time per participant (Table II) there are groups with big differences in time that could be explained as that in a distributed setting, some of the annotators may be less committed and not completely focused on a cognitive task, which may imply a decrement in the quality of the data. We also think about a session of training and Q&A before the annotation task and increasing the number of participants per task, i.e., having at least three annotators giving a judgment on any task.

The observations concerning the time spent during the annotation activity indicate that participants might experience a learning process when they are asked to annotate several sentences in few time. Indeed, in crowdsourcing there is the suggestion of designing a pre task to prepare the participants and make them feel more familiar with the activity to be performed. However, this pre task is costly due to the data used should be different from the original dataset in order to not repeat probable wrong answers. Regarding the platform and the constraints it posed on the design of the annotation task, we have started to explore crowdsourcing platforms such as crowdflower[7] and crowdcrafting[8]. The first platform allows researchers to perform a project with up 30,000 data rows per month. The only money that is paid goes directly to the contributors that work on the tasks and a 20% markup to cover our costs. And the second platform is totally designed for research purposes, which means is free but lacks of implemented features such as a pre filtering of participants that must be implemented by the crowdsourcer. We also observed that the data must be improved from the point of view of generation of sentences. We have tried other sentence parsers such as the Stanford parser[9], OpenNLP[10] and LingPipeline[11], but so far Stanford parser outperforms the ANNIE sentence splitter used in our experiment. The threads to validity have been considered, for instance, the internal validity has been addressed through the explanation of the crowdsourcing task with the short tutorial, by giving some examples without strongly influencing the participants on their decisions for annotating. Regarding the *conclusion validity* by using the results obtained from statistical evidence we cannot draw a strong conclusion but we have only indicators encourage to replicate the crowdsourcing experiment. *External validity* regards the generalization of our observations in other context. We believe that our task design could be easily be adapted to online discussions in other domains provided they use speech-act based natural language.

VI. Future Work: Towards the Discovery of Bug Reports or Feature Requests through Intentions

Having a first annotated corpus consisting of discussions from a software development domain, allow us to move on with the exploitation of the dataset. We use the annotated dataset described in previous sections[12] to feed automatic classifiers and see whether both the quality of the data and the subjectivity of the task will allow us to predict successfully the intentions of sentences of unknown data, i.e., the automatic classification of new sentences into a set of selected intentions: (a) Assertive, (b) Requestive, (c) Responsive, and (d) Attach. The automatic detection of the intentions hidden in the sentences of online discussions would be a great step towards the understanding and computerized management of messages either in forums or mailing-list discussions about software development. One of the most important tasks that we can benefit from this, as we explain next, is the classification of the messages into categories such as bugs or features.

We train three very well-known machine learning algorithms namely: SMO (Support Vector Classifier with sequential minimal optimization), NaiveBayes and J48 (i.e., decision tree). We use the suite WEKA[13] for the task. Each one of the sentences with its label, i.e., intention, was considered as an instance. The intention label is the feature to be predicted (class feature) while the sentences are preprocessed (e.g., upper case to lower-case transformation), tokenized into words and transformed into a vector of n-grams of size from 1 to 5 words, weighted by its TF-IDF (Term Frequency-Inverse Document Frequency) value.

For evaluating each algorithm we apply the standard 10-fold cross-validation technique. According to this technique, the initial dataset is divided into 10 subsets (10 folds), 9 of which are fed to the classification algorithm so as to generate the model and the 10th is used to test the accuracy of the model. We show in Table III an excerpt of the results of classifying the sentences into the four intentions: Assertive, Requestive, Responsive and Attach. The table shows the F-measure (F-M) for each algorithm, a standard evaluation metric for classification that combines both precision, i.e., the number of sentences that have been correctly classified to an intention over the total number of classified sentences into this intention, and recall, i.e., the number of sentences that have been classified in this intention over the number of sentences that belong to this intention according to the annotators.

[7] http://www.crowdflower.com/

[8] http://crowdcrafting.org/

[9] http://nlp.stanford.edu/software/tokenizer.shtml

[10] https://opennlp.apache.org/

[11] http://alias-i.com/lingpipe/demos/tutorial/sentences/read-me.html

[12] The dataset annotated with intentions by the 20 participants is available online at http://selab.fbk.eu/imramirez/PeopleAnnotatedFilesSep2014/files.zip

[13] http://www.cs.waikato.ac.nz/ml/weka/

TABLE III
RESULTS OF THE CLASSIFICATION OF INTENTIONS (F-MEASURE).

Intention	SMO	NaiveBayes	J48
	F-M	F-M	F-M
Assertive	0.607	0.509	0.53
Requestive	0.644	0.568	0.606
Responsive	0.568	0.442	0.501
Attach	0.692	0.526	0.646

The algorithm that performs best the classification is SMO, which is a variant of the support vector machine algorithm. For all intentions its accuracy varies from 0.607 to 0.692. Moreover, all classifiers give acceptable F-Measure values with a range from 0.442 to 0.692. These results can be considered as a positive insight that allows us to go ahead with our research.

The ultimate purpose of using the trained classifier model is to apply it with a different dataset of online discussions whose thread of messages is already classified as a bug report or feature request. What we intend to do with this is to suggest a new technique for identifying messages reporting bugs or features but through the discovery of certain intentions. We believe this new technique can be used altogether with other data mining techniques such as sentiment analysis or topic modeling, for building an enriched classification model for categorizing messages as bug reports, feature or clarification requests. The other dataset that we are working on concerns the discussions of stakeholders of Apache OpenOffice that have been crawled from the bugzilla platform[14]. These discussions have been already classified into categories such as bug reports and feature requests by the stakeholders that have submitted the first comment. However, other experienced stakeholders and community members by reading the threads of messages discover that some threads should be reassigned to another category. This observation leverages our motivation to explore the combination of specific intentions written by stakeholders and to discover which are the intentions or combinations of intentions that are found in bug reports and how they differ from the intentions found in feature requests. The expected outcome of this work is an intention-based model of communication representing the combination of intentions frequently expressed by stakeholders when discussing about feature requests or bug reports.

VII. CONCLUSION

In this paper we described the experience of applying crowdsourcing best practices in our project called *CrowdIntent*. The aim of *CrowdIntent* was the collection of online discussions annotated with intentions. Such discussions are from stakeholders who work in software development context, specifically in the open-source software XWiki. We discussed briefly the first insights from the annotated dataset we obtained so far through crowdsourcing and how we are planning to use the data to build a model for classifying the messages and threads as bug reports, feature or other requests. Our intuition is that intentions frequently expressed in bug reports are different from the intentions expressed in feature or other requests and by exploiting this kind of information we can achieve better classification results.

REFERENCES

[1] P. Laurent and J. Cleland-Huang, "Lessons learned from open source projects for facilitating online requirements processes," in *REFSQ 2009, June 8-9*, ser. LNCS, vol. 5512. Springer, 2009, pp. 240–255. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-02050-6_21

[2] N. Novielli and C. Strapparava, "Dialogue act classification exploiting lexical semantics," in *Conversational Agents and Natural Language Interaction: Techniques and Effective Practices*. IGI Global, 2011, pp. 80–106.

[3] I. Morales-Ramirez, M. Vergne, M. Morandini, A. Siena, A. Perini, and A. Susi, "Who is the expert? combining intention and knowledge of online discussants in collaborative RE tasks," in *ICSE '14, May 31 - June 07*. ACM, 2014, pp. 452–455. [Online]. Available: http://doi.acm.org/10.1145/2591062.2591103

[4] K. Bach and R. M. Harnish, *Linguistic Communication and Speech Acts*. Cambridge, MA: MIT Press, 1979.

[5] I. Morales-Ramirez and A. Perini, "Discovering speech acts in online discussions: A tool-supported method," in *Joint Proceedings of the CAiSE 2014 Forum, Thessaloniki, Greece, June 18-20, 2014.*, ser. CEUR Workshop Proceedings, S. Nurcan, E. Pimenidis, O. Pastor, and Y. Vassiliou, Eds., vol. 1164. CEUR-WS.org, 2014, pp. 137–144. [Online]. Available: http://ceur-ws.org/Vol-1164/PaperDemo01.pdf

[6] E. Estellés-Arolas and F. González-Ladrón-de Guevara, "Towards an integrated crowdsourcing definition," *Journal of Information Science*, vol. 38, no. 2, pp. 189–200, 2012. [Online]. Available: http://jis.sagepub.com/content/38/2/189.abstract

[7] M. Sabou, K. Bontcheva, L. Derczynski, and A. Scharl, "Corpus annotation through crowdsourcing: Towards best practice guidelines," in *LREC 2014, May 26-31*. European Language Resources Association (ELRA), 2014, pp. 859–866. [Online]. Available: http://www.lrec-conf.org/proceedings/lrec2014/summaries/497.html

[8] Z. Azmeh, I. Mirbel, and P. Crescenzo, "Highlighting stakeholder communities to support requirements decision-making," in *REFSQ 2013, April 8-11*, ser. LNCS. Springer, 2013, vol. 7830, pp. 190–205. [Online]. Available: http://dx.doi.org/10.1007/978-3-642-37422-7_14

[9] S. L. Lim, D. Quercia, and A. Finkelstein, "Stakesource: harnessing the power of crowdsourcing and social networks in stakeholder analysis," in *ICSE (2)*, J. Kramer, J. Bishop, P. T. Devanbu, and S. Uchitel, Eds. ACM, 2010, pp. 239–242.

[10] D. Renzel, M. Behrendt, R. Klamma, and M. Jarke, "Requirements bazaar: Social requirements engineering for community-driven innovation," in *RE 2013, July 15-19*. IEEE, 2013, pp. 326–327. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/RE.2013.6636738

[11] B. Mellebeek, F. Benavent, J. Grivolla, J. Codina, M. R. Costa-jussá, and R. Banchs, "Opinion mining of spanish customer comments with non-expert annotations on mechanical turk," in *Proceedings of the NAACL HLT 2010 Workshop on Creating Speech and Language Data with Amazon's Mechanical Turk*, ser. CSLDAMT '10. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 114–121. [Online]. Available: http://dl.acm.org/citation.cfm?id=1866696.1866714

[12] F. Pastore, L. Mariani, and G. Fraser, "Crowdoracles: Can the crowd solve the oracle problem?" in *Software Testing, Verification and Validation (ICST), 2013 IEEE Sixth International Conference on*, March 2013, pp. 342–351.

[13] J. L. Fleiss, B. Levin, and M. C. Paik, "The measurement of interrater agreement," *Statistical methods for rates and proportions*, vol. 2, pp. 212–236, 1981.

[14] J. R. Landis and G. G. Koch, "The measurement of observer agreement for categorical data," *biometrics*, pp. 159–174, 1977.

[15] A. M. Green, "Kappa statistics for multiple raters using categorical classifications," in *Proceedings of the Twenty-Second Annual SAS Users Group International Conference (online)*, San Diego, CA, March 1997.

[14] https://issues.apache.org/ooo/