Idea Spotter and Comment Interpreter: Sensemaking tools for Idea Management Systems

Gregorio Convertino
Xerox Research Centre Europe
6 chemin de Maupertuis
38240 Meylan, France
convertino@xrce.xerox.com

Ágnes Sándor
Xerox Research Centre Europe
6 chemin de Maupertuis
38240 Meylan, France
sandor@xrce.xerox.com

Marcos Baez
DISI, University of Trento
Via Sommarive, 14
38100 Trento, Italy
baez@disi.unitn.it

ABSTRACT

Regular contributors and facilitators using current idea management systems face the problem of information overload. With large numbers of ideas to be assessed or refined, they lack adequate support to efficiently make sense of unstructured idea descriptions and comments. We propose two sensemaking tools as enhancements of current idea management systems: the idea spotter and the comment interpreter. Both integrate interactive user interfaces that use the output of automatic linguistic analysis of ideas and comments. In this workshop paper we present the prototypes, their preliminary evaluation, and the next steps in this research.

Author Keywords

Idea Management Systems; Natural Language Processing; Mixed-Initiative; User Interfaces.

INTRODUCTION

We address the task of providing tools for handling the large quantities of unstructured content that convey the value in Idea Management Systems (IMSs). Specifically, we focus on the content of idea descriptions and the comments associated to ideas. The lack of adequate support to efficiently make sense of this content in current systems leads to two negative effects. First, at the community level, large quantities of informal content potentially remain underexploited due to information overload, and thus *the community loses useful information and makes suboptimal decisions*. Second, at the individual level, *the user has poorer experience and productivity* when making sense of many idea descriptions and comments. Our basic research question is:

What tools could future IMS offer to better manage the unstructured content within ideas and their comments?

In this paper we take the first steps to address this question. We present two prototypes (and their preliminary evaluation) that use natural language processing (NLP) to analyze free text content and then, via interactive interfaces, help the user to manage large numbers of ideas and comments in the IMS.

Specifically, to help users with managing the unstructured content in idea descriptions and comments we propose two sensemaking tools: the idea spotter and the comment interpreter. The idea spotter highlights the essential proposition(s) – idea core(s) – within the idea description to help both accelerate processing and diminish information loss. The comment interpreter categorizes the comments as different types of reaction to the idea and, for each comment interpreted, recommends the associated action by presenting the appropriate system function to perform it.

We focus our analyses on the communicative intent of the contributors: through the idea spotter we detect proposals that convey the idea core(s), as opposed to parts of the idea descriptions that convey background statements or supporting arguments. Through the comment interpreter we point out the type of the commenter's reaction to the idea, and explicitly recommend subsequent actions that might be implicit in each reaction type. Our approach is thus in line with Winograd's language/action perspective [30], suggesting that the design of collaborative tools should be based on the language/action that their functions carry out.

We implement the two prototypes as extensions of the Innovation Cockpit, an existing dashboard for facilitators [4]. We developed the NLP component with the Xerox Incremental Parser (XIP) [1]. For our preliminary evaluations, we use the data in the IdeaScale IMS, deployed within the Xerox Corporation.

In the next sections we first set our work in the context of related research, then describe the idea spotter and the comment interpreter, and report on their preliminary evaluation. We conclude with a discussion of the preliminary results, present the limitations of the actual implementation of two tools, as well as some plans for further research.

RELATED WORK

Our work is at the crossroads of research on IMSs, sense-making and NLP applications, so below we summarize relevant literature in these research areas.

IMS

IMSs are a class of web 2.0 tools for large-scale idea management and deliberation. They have become increasingly adopted in two domains: in organizations, as platforms for open innovation (Enterprise 2.0 tools), and in civic communities such as cities, as platforms for democratic participation and deliberation (Gov 2.0 tools). In the organizational domain there is a growing recognition that the collective wisdom of the community of employees (or even the customers) is an untapped resource for innovation. To respond to the increasing competitiveness of markets many companies started to use these systems to boost their innovation capabilities [5, 28]. Examples of these systems include commercial software products by Spigit, IdeaScale, Innoventive, and Imaginatik. In the civic domain, after web 2.0 tools have made it easy for networked citizens to generate, share, organize or judge information (e.g., wikis, forums, polls), there is a growing expectation of citizens to participate more directly in public affairs, analyze complex problems, and deliberate collaboratively on community matters. Examples of systems designed with this purpose include Considerit [22], Deliberatorium [14] or mIPS [26].

In current IMSs the content of the idea descriptions remains largely unstructured except for some basic template including the title, which is indicated by the authors, some system tags, which are added by the author or commenters, and dedicated fields for the idea description and comments. In their review of this class of software systems, Hrastinski and collaborators [17] point to the need for better tools to help reduce information overload.

Most platforms provide the possibility for a large number of users to enter a plethora of unstructured or semi-structured information into the systems, but little help in structuring such content. Several authors anticipate a future trend toward including stronger administrative tools to cut through the voluminous material provided by users. This is necessary in order to capture ideas and move them towards implementation. Another possibility is to use NLP for analyzing the unstructured content.

NLP for Idea Processing

To our knowledge, there is little work in NLP targeted to the semantic processing of idea descriptions in IMSs beyond indexation tools, and no application that has been aimed at spotting idea cores.

Some attempts to help reduce the amount of content of text-based posts were done in the context of forums for community discussion. For example, I-DIAG [2] uses techniques derived from text summarization [e.g. 24], which attempt to consolidate large documents or sets of documents into abstracts or shorter documents. The I-DIAG system helps to summarize the messages and documents into more succinct, durable knowledge.

Danes [11] also developed a system for summarizing relevant ideas resulting form surveys requesting input for new ideas. His method builds a semantic network of the most relevant concepts occurring in the answers to the survey, which are interpreted as key ideas. Paukkeri et al. [23] use NLP for clustering ideas in idea databases according to their subject matter.

NLP for Comment Processing

We have found one example of NLP application for processing comments by Westerski et. al. [29], which is integrated in Gi2MO IdeaStream, an open-source IMS http://ideastream.gi2mo.org/mine_ideas.html. The system uses sentiment analysis for mining sentiment polarities for rating ideas. Although sentiment analysis could replace part of our NLP system, our approach is not confined to leveraging the sentiment aspect of comments (see comment categories in Table 2 below). More importantly, the analysis of content is, in our approach, the means for action recommendation.

A case-study by Jouret [18] on an IMS deployed at Cisco provides some evidence suggesting the need of tools that help handling comments automatically or semi-automatically as a source of expert information. The author observed that whereas the votes (and we can add sentiments mined from comments) could favor “cool” ideas over those commercially and technically viable, some commenters showed deep subject-matter expertise and insight. According to this study voting was less useful than comments in helping the facilitators choose the 40 semifinalists among the ideas suggested.

The relevance of processing comments in IMSs is also indicated by Kain et al. [20] who propose and idea management system where the comments are clustered according to their topics. The clustering, however, is provided by system design, and not by automatic processing: the system asks the users to choose a topic for the comment before inserting it.

The recent work by Westerski and colleagues [27] is the only study we know that started correlating the comments, as informal reactions to ideas, to other metrics. They analyzed 50000 ideas from systems deployed in four organizations (Dell, Starbucks, Cisco and Canonical). Their findings, although not conclusive, confirm that such annotations can help to identify new metrics that allow a more efficient comparison of ideas.

Sensemaking

Besides IMS design, our work is related to research in sensemaking. An example of web 2.0 tools in support of sensemaking is the SparTagus system, an annotation and tagging tool for web pages [15]. While web 2.0 technologies are the basis of collaborative content processing including collaborative deliberation, their success depends on the design and tools that help the user make sense of the contents. Sense-making tools like Cohere [7] or ContextBar/ContextBook [6] allow the users to highlight, aggregate or link relevant content. Mixed-initiative sensemaking [12] – similarly to our proposition in the present paper – uses NLP to perform some preprocessing tasks that make subsequent human sensemaking easier by automatically detecting relevant content.

IDEA SPOTTER

Idea Spotter: Method

Idea descriptions in IMSs usually consist of two parts: a title and an idea body. The title is often a concise formulation of the idea and the body often contains argumentative and descriptive content besides the title. Spotting the idea core(s) in the idea body can be useful for several reasons, among others, the following:

- The titles are often not well-formulated summaries (e.g. because they are written before the body) and the idea cannot be understood on the basis of the title alone.
- The idea can have multiple cores, and just one of them is expressed in the title.
- The idea core is hidden within the idea body and if the user does not carefully read the body then misses it.

Idea Spotting is based on analyzing idea descriptions in terms of speech act theory, which was first developed by Austin [3]. According to speech act theory, speakers perform illocutionary acts by utterances using special linguistic structures. Conveying ideas is usually performed by directive speech acts, which are defined as acts that cause the interlocutor – in our case the community that can implement the proposed ideas – to take a particular action.

Speech-act theory has been mainly used in NLP applications for analyzing and generating dialogue systems [e.g. 25], since communicative intent has a major role in managing dialogue turns. Cue-word-based, pattern-based and statistical systems show good performance in detecting speech-acts, indicating that the task is feasible for state-of-the-art NLP components. For example, Reithinger and collaborators report an average of 70% recognition rate of speech-acts in dialogues.

We implemented an exploratory rule-based method that builds on the fact that speech-acts are associated with specific linguistic structures. We have to take into account, however, that there is not a one-to-one correspondence between speech-acts and linguistic structures: the same speech-act may be executed by different linguistic structures, and most linguistic structures are ambiguous with respect to the illocutionary act that they convey, i.e. the same structure can convey different illocutionary acts depending on the communicative situation. For example, a request can be executed by an imperative, a question, or a performative assertion. Inversely, a question may be an inquiry, a request or a suggestion.

We developed the idea spotter algorithm based on two existing corpora of ideas: the IdeaScale IMS at Xerox and the public content shared on http://mystarbucksidea.force.com. In order to detect idea cores, using a development corpus of 680 ideas and 100 ideas from the two corpora respectively, we set up a list of linguistic structures that convey the directive speech-acts of the idea cores. In our case, the ambiguity of the speech-act value of the linguistic structures is resolved by the set communicative situation of IMSs as a space for proposing ideas. Our rules thus identify *particular linguistic structures* that potentially convey a directive speech act, and select them as idea spots if they appear in a *context that involves relevant entities* (the idea-owner, the enterprise). In particular cases, however, (performative verbs and explicit idea assertions, see Table 1 below) no context is necessary for selecting the idea spot. The enterprise-specific domain vocabulary was compiled manually based on the development corpus.

Our exploratory rule-based method was developed as a new layer on top of the general-purpose dependency parsing module of XIP. For the first implementation, we defined about 40 syntactic rules using a small set of 50 words of domain vocabulary.

Table 1 lists the linguistic structures used by the method to detect idea cores and presents example sentences for each structure. The words in bold indicate the relevant linguistic structure involving the relevant context.

linguistic structure	example
performative verb	**I suggest/propose/recommend**
idea assertion	**My proposal** is to establish system links with the vendors …
imperative	**Develop** a PC **application** (or other type of support) to allow small lenders to perform their own origination and servicing of private loans.
conditional	So the SBU **could implement** an Opex machine within their processes
question	So **why should we invest** in a professional development program **?**
need assertion	**We need** a resource hotline .

Table 1. Idea Spotter. Linguistic structures conveying idea cores and example sentences. The words in bold constitute the indicator pattern.

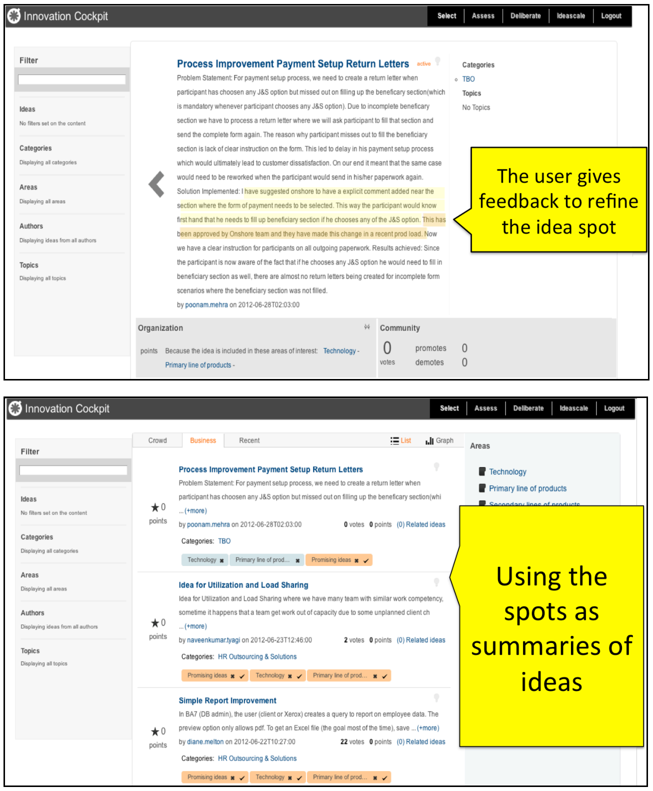

Figure 1. Idea Spotter. The system indicates the spot detected and the user refines it (top). The system uses the spot as summaries for ideas (bottom)

Idea Spotter: Prototype

The software prototype of the idea spotter is a new tool that extends an existing dashboard for facilitators, the Innovation Cockpit.

In the dashboard, the idea spotter extends the "Select" section or module, which allows facilitators to find and organize the promising ideas. The facilitator is presented with a list of idea previews ranked according to various criteria. Without the idea spotter the idea preview included only the first sentence. With the idea spotter, instead, the facilitator can directly read the core(s) of the ideas at a glance. In this case idea cores are more effective idea summaries as shown in Figure 1 (bottom). This represents an advantage with respect to either previewing the first sentences or showing the entire content of the idea, which would more likely impose higher information overload. The idea spotter might also help in the process of identifying relations between ideas, such as duplicates or complementary ideas, by restricting the comparison of the semantic similarities among the ideas to the snippets. This suggests synergy of combining machine learning to cluster ideas, linguistic analysis to highlight the central arguments of an idea, and the ability for the human to validate the results via a simple interface.

Once the user has identified an idea, he or she can access its profile and refine the core: The interface includes a highlighting "pencil" for extending, reducing, removing, or creating idea spots within the idea body, or description and contextual actions when hovering the fragments to edit or remove them (Figure 1).

Idea Spotter: Preliminary Evaluation

We run, in parallel, the evaluation of the user interface design and the evaluation of the NLP algorithm. For the evaluation of the user interface, six professionals with prior experience using IMSs were given a low-fidelity prototype of the user interface - mockups presented on paper and via interactive slides -, as well as an introduction to the typical scenarios of use. After a short presentation on the prototype and the scenario of use, the feedback was collected through a survey that gave each professional the opportunity to write down her/his evaluation of the specific features. The survey allowed us to validate the usefulness of the proposed design and learn about unaddressed needs. Four of the six professionals found clearly useful the features of highlighting the spot in the idea description (Figure 1, top) and using the spots as summaries for ideas (Figure 1, bottom). One professional wrote: “It is quite informative”; another wrote: “yes it is useful; for an alternative design perhaps try to abstract the text”. The other two professionals found the features “possibly useful” or were not sure. On the second feature one of them wrote: “this would help me decide whether to click or not on the idea” but about the first feature he wrote “[the highlighting is] maybe useful; [but] I may want to understand also the arguments leading to this idea”. The evaluation of the interactive version of this second prototype, combining the algorithm and the user interface is still ongoing (see Discussion section).

We also tested the performance of the idea spotting algorithm on 55 idea descriptions. The ideas in this test sample were selected randomly and had not been used for informing the development of the rules.

The quality of results of the idea spotting algorithm was evaluated by two independent human coders, two of the authors. They followed a written coding protocol that they had agreed on. The protocol provided a definition of what an idea core is and is not, a list of exemplars of idea cores and a coding procedure. Each idea description could contain none, one, or more idea cores. After coding the 55 idea descriptions independently, the coders reviewed the results, computed human-human inter-coder agreement[2], and then agreed upon the gold-standard idea cores out of all the idea cores that were identified by at least by one of the two coders. The resulting 55 ideas, coded by the humans, were then used as the ground truth to which the idea spotting algorithm was compared. The ground truth

[2] Agreement was considered in cases where the two segments of text coded by the two humans overlapped for at least 50% of the number of words of the union of these

contained 62 segments[3] of text from the 55 idea descriptions. Finally, the human-machine inter-coder agreement[4] was measured between the ground truth and the automatically coded ideas. Below we report the agreement scores (as kappa, or as percentages when we could not find a well-justified measure of agreement by chance for kappa):

- *Human-human coding agreement*. Considering the 55 ideas, the coders agreed in 73% of them on at least one idea core per idea or on the absence of an idea core (3 ideas, 4.5%); the corresponding kappa is 0.706. Alternatively, considering the 61 segments of text from the 55 ideas, the two coders agreed in 67.2% of them (2 out of 3).
- *Human-Machine coding agreement*. Considering the 55 ideas, the coders agreed in 55% of them on at least one idea core per idea or on the absence of an idea core (2 ideas, 3.6%). Alternatively, considering the 61 segments of text from the 55 ideas, the two coders agreed for 50.8% of them (1 out of 2). The algorithm missed 47.5% of the segments.

It is worth noting that in two cases the algorithm found an idea core that the humans had overlooked: in the first case the humans had not found any core, and in the second case they had found a suboptimal one. This shows that idea spotting is a difficult task for even humans, and that the automatic coding, although imperfect, can sometimes outperform humans.

COMMENT INTERPRETER

Comment Interpreter: Method

The Comment Interpreter automatically categorizes the comments as different types of reaction to the idea, maps them onto the intended action types that are implicit in each comment type, and proactively presents to the user the appropriate system function required to perform such inferred action types. We describe below the two components of the prototype: the *comment categorizer* and the *action recommender*.

Comment Categorizer

We consider three basic comment types or categories. These are defined based on three distinctive aspects of the idea, which is the main target of the commenter's reaction:

- Reaction related to the *content* of the idea.
- Reaction related to the *value* of the idea: expression of the commenter's attitude or judgment about the idea value.
- Reaction related to the *status* of the idea within the IM process.

The comments can be classified in more than one category.

In our NLP component we define the reaction categories in terms of patterns involving linguistic structures and/or associated lexical items/expressions as shown in the table below.[5] The patterns have been identified on the basis of a development corpus of about 280 comments. Our set of linguistic structures as well as lists of lexical items are expectedly not complete, nevertheless in our experiment the proposed patterns accounted for a large part of all the reactions (see preliminary evaluation). It is important to specify that we apply these patterns on the first sentences of the comments only. Only for one case, PROS-CONS (see in Table 2 below), the analysis can conditionally be extended to the second sentence as well. The reason for this is that, based on analysis of development corpus, we found that the first sentence conveys the reaction type in 96% of the comments. This restriction helps resolve the ambiguity of the communicative value of our patterns.

As shown in Table 2, each reaction type is further categorized into fine-grained categories, which may be specific to the community and purpose of the IMS. The fine-grained categories included in our list in Table 2 apply to an enterprise IMS used to promote open innovation. Describing the complete list of the linguistic structures is beyond the scope of this article.

comment category	fine-grained type	example
reaction relative to the content of the idea	PRIOR ART	This **solution exists** within BPS.
	ADDITIONAL INFORMATION	I **wonder if** it is something that could be centralized.
expression of the commenter's attitude or judgment about the idea value	AGREE	**True**, **I support** this idea.
	DISAGREE	**I don't agree**, …
	POSITIVE ATTITUDE	**Great idea**, Kelly!
	NEGATIVE ATTITUDE	**This is not** the kind of savings / revenue idea that we are searching for.
	PROS-CONS	**Yes** , we can use WebEx or Live Meeting, **but** it only displays our computer screen.
reaction relative to the idea status or process	PROCESS	**Ranking - Hold** until 2010

Table 2. Comment categories and example sentences. The words in bold constitute the indicator pattern.

Similarly to the idea spotter, the comment categorizer has been developed on top of the general-purpose dependency parsing module of XIP. Based on the analysis of the development corpus, we defined about 200 syntactic rules and about 160 lexical items.

[3] Five segments from two ideas were excluded because part of bulleted lists of which only the first item was considered.

[4] Agreement consisted in cases where the automatic core contained the union of the human cores.

[5] The expression of the commenter's attitude or judgment about the idea value could be detected by plugging into the system a sentiment analysis component.

Action Recommender

The second component of the prototype logically maps the three basic comment types onto three basic actions types:

- Detecting the reaction to the idea content leads to recommending actions aimed at content generation or management.
- Detecting the commenter's attitude towards the idea value leads to recommending actions aimed at voting or deliberating on which ideas are worth implementing.
- Detecting reaction about the process leads to recommending actions aimed at managing the idea state or process with respect to an agreed workflow.

When interpreting a comment, the action recommender proactively presents to the user the appropriate system function required to perform the action types. The expected benefit of this new functionality is a reduction in the number of interaction steps for the users. Note that this gain in performance may pertain to one or more users and can be measured over period of asynchronous interaction.

In the next section we give three specific examples, one for each of the action types listed above.

Comment Interpreter: Prototype

As for the idea spotter, the comment interpreter extends the Innovation Cockpit, the dashboard for facilitators. As support for sensemaking, this tool offers two functionalities: it gives an overview of the types of comments around an idea; when possible, it interprets the comment and recommends a relevant action to the users who submit or receive a comment.

The functionality of categorizing comments is represented in Figure 2 as a bar chart giving an overview of the 50 comments made to the same idea by type.

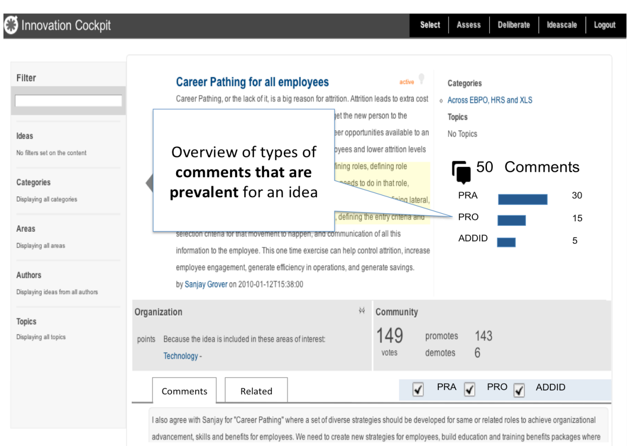

Figure 2. Comment Interpreter. Overview of the categorized comments for an idea (bar chart).

The functionality of recommending actions includes the following types of action recommendations:

1. reaction to the content of the idea (RCI)
2. reaction to the expressed commenter's attitude related to the idea value
3. reaction to the idea state or process

Type 1. Reaction to the content of the idea (RCI)

A comment that reacts to the content of the idea is typically aimed at refining it by proposing additional content (e.g., a new application of the same idea) or changing or clarifying part of the existing content. This is the sub-type that, in Table 2, we call ADDITIONAL INFORMATION. Another similar sub-type is the comment where the commenter appends a reference to prior work or ideas. We call this sub-type PRIOR ART. For both of these two sub-types of RCI, as the system interprets the comment, it recommends to the commenter the inception or full execution of the corresponding action: propose a modification of the content of the idea.

An action is recommended if an ADDITIONAL INFORMATION (ADDID) comment is detected by the system. In the example below, when the commenter submits the comment with the title "Automation of creating and sending files to Vendor/Trust", the system classifies the comment as an ADDID, and this comment triggers the recommendation that in Figure 3 is visually represented by a blue dialog box with two buttons. The recommended action is to insert the content of the comment to the text or description of the idea (see Figure 3).

We envision an extension of this first type of action recommendation in future versions of the prototype. If the user accepts the recommendation illustrated in Figure 3, then s/he is given the recommendation of where the content of the recognized ADDID comment could be added within the description of the idea. I.e., after accepting the recommendation to insert the content, the user can also be recommended 'where' to insert it. This can be done based on semantic similarity between the content of the comments and the content of each paragraph in the description, as in Mail2Wiki [16], our prior work. Finally, as the edit or addition is made to the idea content, the system sends a request to the author of the idea (and/or the moderator in the system) to either accept or reject the edit or addition proposed. Alternative ways to request confirmation of the proposed change can be implemented (e.g., through a version manager such as in the github.com code management system).

A similar procedure can be used for the sub-type PRIOR ART. If the comment mentions a related idea, the commenter can be given the options of either inserting the reference to the related ideas in the description of the idea, as for the ADDID sub-type described above, or as an inter-document reference (e.g., an hypertext link) to a prior idea stored in the system. The recommendation of the related ideas can be done based on conventional methods that use semantic similarity between the content of the comment and the content of the prior ideas stored in the system or in other repositories that could be indexed and compared.

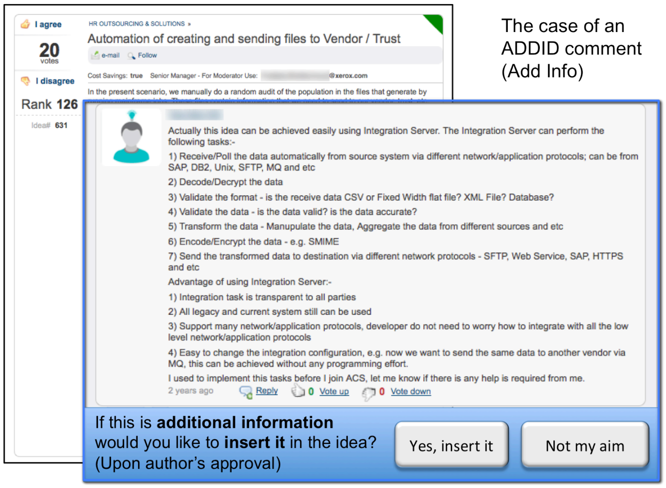

Figure 3. Comment Interpreter: Type 1. The user is recommended the action of inserting the content of the recognized ADDID comment to description of the idea.

Type 2. Reaction to the idea value

For POSITIVE ATTITUDE or AGREE comments the commenter can be given the option of giving a Vote-Up to the idea and for NEGATIVE ATTITUDE a Vote-Down.

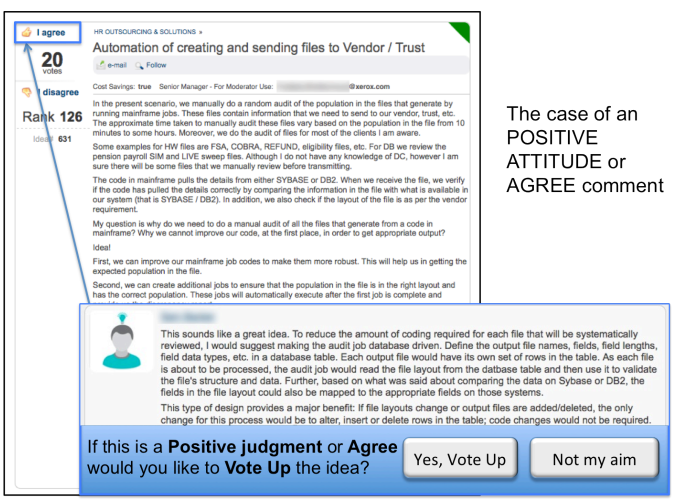

Figure 4. Comment Interpreter: Type 2. The user is recommended the action of voting up the idea based on the recognized POSITIVE ATTITUDE comment.

In Figure 4, when the commenter submits a comment, which the system classifies as POSITIVE ATTITUDE or AGREE, a recommendation is triggered, which is presented in a blue dialog box with two buttons in Figure 4. If the user clicks on the button "Yes, Vote Up", then the system will directly execute the "Vote Up" function (i.e., "I agree" button on the top left of Figure 4) and give feedback as the user is submitting the comment. Note that the "Vote Up" action happens without the need for the user to carry out the extra steps required for finding or moving to the voting tool and activating it.

Type 3. Reaction to the idea state or process

As the ideas are refined and judged by the community, some of the comments are directed at suggesting changes in the state of the idea. For example, in the comment presented in Figure 5, the commenter writes "This idea should move to 'Completed' status and has been implemented […]". Therefore, for a PROCESS comment the commenter could be given the option of giving a Vote-Up to the idea and for NEGATIVE ATTITUDE a Vote-Down.

In Figure 5, when the commenter submits a comment, which the system classifies as PROCESS, the recommendation appears in a blue dialog box with two buttons. If the user clicks the button "Yes", then the system will let the commenter change the status of the idea as described by the pop-up menu function illustrated in Figure 5 (bottom), where the user changes the status from "Acknowledged" to "Implemented". Alternatively, if the commenter does not have the rights to change the state, the system can asynchronously recommend the same action to the moderator, who can execute it when (s)he is available.

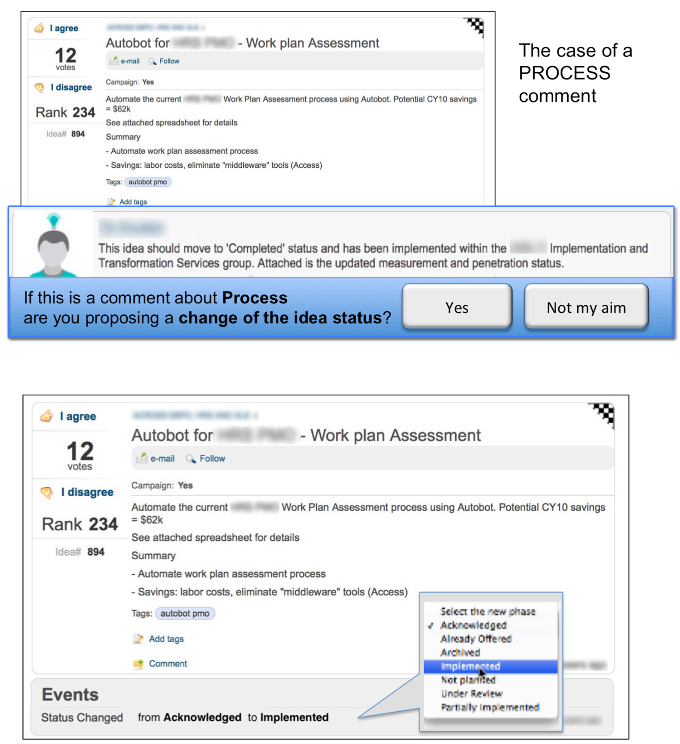

Figure 5 Comment Interpreter: Type 3. The user is recommended the action to propose a change in the idea status (top). If the user accepts the recommendation, s/he is prompted to change the status of the idea (bottom).

Comment Interpreter: Preliminary Evaluation

Similarly to the evaluation of the idea spotter, six professionals with some experience on IMSs evaluated a low-fidelity prototype of the user interface (on paper and interactive slides). Their feedback was then used to refine the design of the software prototype which is still being evaluated.

In parallel, we evaluated the performance of the algorithm. The algorithm was tested using a random set of 69 comments. It provided at least one category for 70% (48) of the comments, and 10.4%(5) were assigned multiple categories. As a simple evaluation of the quality, we compared the automatic classification of the 48 comments with the classification by two human coders. We presented the labels independently to 2 human coders to assess inter-annotator agreement. They fully agreed in 81% of the cases, partially agreed in 12.5% of the cases and disagreed in 6.3%. The Kappa was .81. After agreement between the two human annotators we found agreement with the automatic classification in 87% of the cases, both partial agreement and disagreement in 6.3%. The Kappa was .87.

ARCHITECTURE

The idea spotter and the comment interpreter were integrated into the *Innovation Cockpit* interface and backend infrastructure. As illustrated in Figure 2, the architecture of the Innovation Cockpit was designed to provide a set of core services, which allows developers to build novel tools for managing any type of IMS through a common set of APIs. We exploited this aspect to bring the idea spotter and interpreter to a wider range of IMSs.

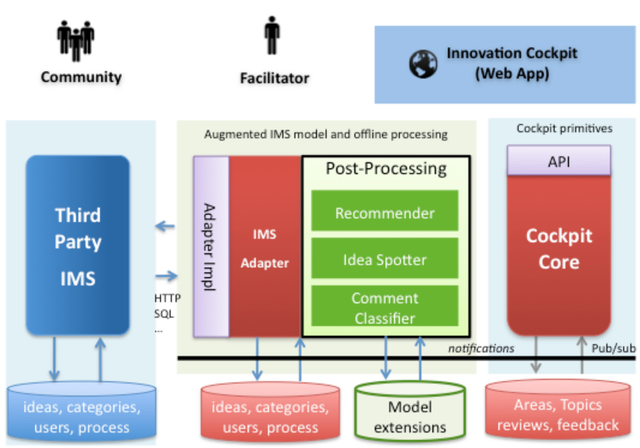

Figure 2. Innovation Cockpit and the post-processing module

In the implementation, the tools rely on a set of backend services built as extensions to the Innovation Cockpit. To this end, the basic IMS model composed of ideas, categories, relations, users and processes, was extended with two types of annotations:

- Idea fragment: This annotation on the idea represents the area(s) of the idea content considered as a core
- Comment type: This annotation on the comment represents the type or category it belongs to.

The above *model extensions* are provided by the *post-processing module* of the IMS adapter. This module incorporates the post-processing services of the idea spotter and the comment interpreter, and coordinates the data processing and augmentation. In doing so, whenever the IMS adapter performs a synchronization operation with the target IMS (e.g., updating ideas or comments from IdeaScale), this post-processing module is able to efficiently update the accompanying idea and comment annotations while also ensuring data consistency.

At the implementation level the post-processing extensions presented act as a wrapper of the XIP NLP tool.

DISCUSSION

This research responds to the need for better tools for processing unstructured content in idea descriptions and comments in IMSs. The solution proposed includes two tools that enhance current IMSs: the idea spotter and the comment interpreter. The two tools implement the same design approach of providing mixed-initiative support for sensemaking by integrating automatic linguistic analysis of text with simple and interactive user interfaces.

Our preliminary evaluations pointed to some limitations of the present prototype. We discuss them below, and mention some possible future steps to address them.

Limitations and Future Work

A first general limitation pertains to the type of IMS deployments that can benefit from the proposed solution. We originally intended to develop rule-based algorithms based on our analysis of unstructured content from two IMSs: ideas from a stable community of employees in an enterprise (Xerox Corporation) and ideas from the large crowd of customers of a large coffee-shop company (http://mystarbucksidea.force.com). We observed that the homogeneity of the user community is an important factor for the development of the NLP component. Specifically, when evaluating the two datasets for the development of linguistic rules we found that while the comments in the employee IMS could easily be categorized according to linguistic patterns, those from the community of customers (http://mystarbucksidea.force.com) showed such diversity of expressions that we could not allocate the effort of extracting patterns. The two idea-spotting data-sets, however, lent themselves well to the same categorization.

An important implication that could be validated in future research is that the use of some of the semi-automatic techniques that combine natural language processing techniques and interactive user interfaces are applicable and useful for IMSs used by stable communities of practice and not to the same extent for crowds of individuals that have little knowledge and protocols in common. Indeed, prior research on language has shown that stable communities establish their own linguistic conventions. This is consistent with Clark's (1996) concept of "communal common ground", which includes the conventions around language. I.e. since communities in organizations share common ground (e.g., corporate culture, rules, roles, workflows) and goals (e.g., innovation), well-determined types of actions are performed within IMSs, and their expression tends to be consistent across the community members.

Our preliminary evaluations showed that the tasks of detecting ideas spots and categorizing comments were difficult even for the human coders and that the results of the NLP components in our prototypes approached human performance. At the same time, we found a set of limitations pertaining to the robustness of our current implementation of the NLP components.

- The present idea spotter ignores structure within the idea description (e.g., explicit lists, headings). This shortcoming is responsible for 6 missed idea cores. In an improved system we plan to account for structure present in the idea descriptions.
- The idea spotter does not take into account the terms in the title. We have found that the gold-standard idea cores often explain a term present in the title in more details like in the following example:

  > Title: "Soft **VOIP Phones**"
  >
  > Gold-standard idea spot: "Instead of spending the Revenue and man-hours on manual configuration and maintenance of the AVAYA **VOIP phones** we can provide the users with the **VOIP phone** applications on their machines."

  In our future work, we plan to analyze regularities in the relationship between the idea core and the title, which could be exploited as a cue for the idea spotting algorithm.

Future implementations could also make more informed decisions about the scope of the idea core, e.g. by including or not including an additional sentence that clarifies it. E.g. consider the following chain of sentences:

> The Solution: ACS would offer a compliance solution independent of the print/mail vendor. Such solution would be via a third party provider with whom ACS has an existing relationship.

The second sentence explains the first sentence, the idea core. So it could be included in it provided the system had some rules based on discourse analysis. Future versions of the system could also integrate a co-reference resolution module.

Finally, an inherent and general limitation of the categorization component of unrestricted text is that the finite set of linguistic structures considered cannot fully cover the variety of linguistic structures that people use in natural language. Our research prototypes demonstrate the feasibility of classifying ideas and comments in IMSs according to linguistic patterns that allow a fair coverage of such variety, but we do not claim to have covered all possible expressions.

CONCLUSION

We have proposed two enhancements of current IMSs:

- Current IMSs support the understanding and organization of informal content through headings and user-entered topical annotations such as tags or keywords. As metadata allowed by the system, these are attached to the unstructured body of text describing the idea. However, current systems do allow sorting out relevant content from within the unstructured text describing the idea. Our system provides a new way for the IMS platforms to highlight the core proposition(s) contained within the text describing the idea.
- Current IMSs support end-users in providing their reaction to existing suggestions or ideas through structured and unstructured annotations. Examples of structured annotations are up-votes or down-votes, and examples of unstructured annotations are comments. While the structured annotations can be easily aggregated and exploited by analytics tools when evaluating the annotated suggestions or ideas, the content of unstructured annotations remains underexploited. There is no understanding from the part of the system of the type of operation(s) that the commenter intends to perform on the idea. As a result, the IMS cannot proactively connect different actions on the same idea (e.g., a positive comment and an up-vote) and, when relevant, proactively recommend desirable next actions (e.g., if a comment aims to extend the content of an idea current IMSs do not make this next action directly possible). Our prototype assists in using the comments for expanding ideas, managing the process, or complete the voting mechanism by showing positive or negative comments, and thus yields to the user a unified commenting experience.

The preliminary evaluations provided evidence that the proposed methods and design approach are effective: the first prototype of the NLP component shows satisfactory performance, which suggests the appropriateness of the methodology. The preliminary evaluation of the design by prospect users was also positive. To continue addressing the question of "what tools can augment IMS to better manage the unstructured content within ideas and their comments", we intend to improve the robustness of the NLP components and carry on studies with users to summatively evaluate the interactive versions of the two prototypes proposed in this paper.

REFERENCES

1. Aït-Mokhtar, S. , Chanod, J-P., Roux, C. (2002). Robustness beyond shallowness: incremental dependency parsing. Natural Language Engineering. 8(2/3), 121-144.
2. Ackerman M.S., Swenson A., Cotterill S., and DeMaagd K. 2003. I-DIAG: from community discussion to knowledge distillation. In Communities and technologies, Huysman M. et al. (Eds.). 307-325.
3. Austin J.L.: How to Do Things With Words. Cambridge (Mass.) 1962.
4. Baez M. and Convertino G. (2012) Innovation Cockpit: A Dashboard for Facilitators in Idea Management. In the proceedings of ACM CSCW 2012, Seattle, USA.
5. Bailey, B., and Horvitz, E. What's your idea?: a case study of a grassroots innovation pipeline within a large software company. *Proc. CHI 2010*, ACM, 2065–2074.
6. Bier, E. A., Billman, Dorrit, Dent, Kyle and Card, Stuart K. (2009): Collaborative Sensemaking Tools for Task Forces. In: Proc. of the HFES 2009. 439-443.
7. Buckingham Shum S. (2008). Cohere: Towards Web 2.0 Argumentation. In Proc. COMMA'08. France.
8. Chesbrough, H., Vanhaverbeke, Wim, and Joel West, eds., 2006. Open Innovation: Researching a New Paradigm, Oxford: Oxford University Press.
9. Clark, H. H., 1996. Using Language. Cambridge University Press.
10. Dahlander L., Wallin M.W. A man on the inside: unlocking communities as complementary assets, Research Policy 35 (8) (2006).
11. Danes J. Y. Application of Text (Idea) Mining to Internet Surveys: Electronic Capture of the Structure of Ideas and Semantic Concepts. Proc. of the Second International Conference on Electronic Business. 2002.
12. De Liddo A., Sándor A., and Shum S. B. 2012. Contested Collective Intelligence: Rationale, Technologies, and a Human-Machine Annotation Study. Comput. Supported Coop. Work 21, 4-5, 417-448.
13. Gangi, P. M. D. & Wasko, M. (2009), Steal my idea! Organizational adoption of user innovations from a user innovation community: A case study of Dell IdeaStorm, Decision Support Systems 48 (1), 303-312.
14. Gurkan A., Iandoli L., Klein M., Zollo G. 2010. Mediating debate through on-line large-scale argumentation: Evidence from the field. *Inf. Sci.* 180, 19 (October 2010), 3686-3702.
15. Hong L., H. Chi, Budiu R. , Pirolli P., Nelson L. (2008) SparTag.us: A low cost tagging system for foraging of web content. In Proc. AVI 2008. ACM Press.
16. Hanrahan B., Bouchard G., Convertino G., Weksteen, T., Kong N., Archambeau C., Chi, E.H. Mail2Wiki (2011): Mail2Wiki: Low-cost Sharing and Early Curation from Email to Wikis. In Proc. C&T 2011.
17. Hrastinski S., Kviselius N.Z., Ozan H., and Edenius M. 2010. A Review of Technologies for Open Innovation: Characteristics and Future Trends. In Proceedings HICSS '10. IEEE Computer Society, USA, 1-10.
18. Jouret G.. 2009. Inside Cisco's Search for the Next Big Idea. Harvard Business Review.
19. Jurafsky D., Shriberg E., Fox B., and Curl T.: Lexical, Prosodic, and Syntactic Cues for Dialog Acts. In Proc. of ACL/COLING-98 Workshop on Discourse Relations and Discourse Markers. 1998. 114-120
20. Kain A.S., Kirschner R.J., Lang A. and Lindemann U.: Facing the open innovation dilemma - structuring input at the company's border. 2011. Proc. ICED11, 487-498.
21. Klein, M. Enabling large-scale deliberation using attention-mediation metrics. Computer Supported Cooperative Work (CSCW) 21 (2012), 449–473.
22. Kriplean T. Freelon D., Borning A., Bennett A. (2012) Supporting reflective public thought with considerit. In Proc. of CSCW '12. ACM, NY, USA, 265-274.
23. Paukkeri M-S. and Kotro T.: Framework for Analyzing and Clustering Short Message Database of Ideas. Proc. of I-KNOW '09 and I-SEMANTICS '09. 2009
24. Radev, D.R. and Eduard Hovy (1999): Intelligent text summarization. AI Magazine, 20 (3).
25. Reithinger N. and Klesen M.: Dialogue Act Classification Using Language Models, In Proceedings of EuroSpeech-97, 1997, pages 2235-2238
26. Vivacqua A.S., Expedito C., Galuzzo F., Borges, M.R.S., da Silva, S.T.F. (2010), Moving from Ideas to Proposals. CSCW10 Worksh. on Collective Intelligence In Organizations: Toward a Research Agenda.
27. Westerski A., Dalamagas T., and Iglesias C. A., (2013) Classifying and comparing community innovation in Idea Management Systems, Decision Support Systems.
28. Westerski, A., Iglesias, C. A., Nagle, T. The road from community ideas to organisational innovation: a life cycle survey of idea management systems. IJWBC 7, 4 (2011), 493–506.
29. Westerski, A. and Iglesias, C. A. (2012) Mining Sentiments in Idea Management Systems as a Tool for Rating Ideas. Large-Scale Idea Management and Deliberation workshop. COOP2012, France, 2012.
30. Winograd, T. (1987). A language/action perspective on the design of cooperative work. Human Computer Interaction, 3(1), 3-30.