Implicit Social Production: Utilising Socially Generated Data By-Products

Ben Jennings and Anthony Finkelstein

University College London
London, UK
b.jennings@cs.ucl.ac.uk and a.finkelstein@cs.ucl.ac.uk

Abstract. Enhancing business processes by the integration of social software is an area of active research. Once such integration has occurred, a new problem is presented - that of using social data in an effective manner. With large amounts of user generated data created, finding relevance in both data and in the people who created it as part of a business process becomes problematic. This paper frames the problem of socially generated information in the context of Open Source software development processes and of improved execution of tasks in that domain. Such social processes highlight the research area of facilitating the automatic selection of relevant data as part of a larger process. The paper introduces a novel two stage mechanism to answer such a problem. The approach is built on the concept of using the implicit social connections available from socially generated data artefacts to create a weighting model. This methodology is inherently egalitarian in nature as it uses a folksonomical strategy to construct the model. A dynamic domain specific lexicon is created to improve term weighting relevance. This weighting is then enhanced by analysing implicit proximity between participants of the socially generated production. By combining these two methods within a software framework, finding relevancy within a large corpus of socially generated data is improved. The prototype software framework built on these two approaches is constructed to provide dynamic programatic access to social data which can be incorporated as part of a larger business process to speed up the decision making process.

Key words: Workflow, Identity, Ad Hoc, Social Production

1 Introduction

Interest in utilising social software in business processes has been gaining momentum due to successful crowd sourced projects such as Wikipedia. When a business harnesses successfully social production in any numerically significant manner, a new problem presents itself, that of information overload. Such overload presents a new issue. With the abundance of information available from which to make an informed decision there is an inability for an individual to process such data. The term data artefact is used to refer to any interactions by individuals or information generated within a given domain.

Decision scaling, or lack of ability to process, presents a problem for the successful usage of such data within a business process. An exemplar of such a problem domain would be that found in Open Source software development, where in order to complete a task, significant work must be executed to find relevant support documents. Merely gathering such data artefacts is of less value if informed actions may not be informed by the artefacts. There has been much research in the information retrieval and machine

learning area on large scale information data sets. Such approaches are, in the main, seeking general purpose solutions and focus on normal language usage.

This paper introduces two approaches used together which are fundamentally built upon social interactions and the concept of weak ties [11]. Both approaches are built on social data by-products. Such a by-product may be considered as an indirect analysis of socially generated data from which implicit information may be realised. This form of analysis is inherently based on a flat structure, as no a priori hierarchical structures are being placed in relation to the relative merit of data artefacts. Domain Specific Nomenclature (DSN) seeks to address the construction of a specific dynamic lexicon to enable more relevant term weighting on socially constructed data sets. Implicit Social Proximity (ISP) is used in conjunction with the first approach and looks at clustered data and historic interactions between people adding to the business process to provide additional weighting metrics. This novel approach focuses on what is being said and to whom, rather than other forms of analysis such as method call traces [15] and commit frequency [17]. The two approaches are shown as part of a software framework. The framework presented is intended for use in the context of a business process and as such is ad hoc and lightweight in nature. The framework is able to re-adjust dynamically over time in relation to new data artefacts added to the business domain. Such a framework will speed up the decision making processes within the execution of a workflow instance and may be executed as part of the business model.

The rest of this paper is structured in four main sections. In the first section, the paper will detail how information overload is a consequence of an active social production [2] environment. The second section of the paper will look at current approaches in information analysis. Sections three will identify two specific implicit data by-products from socially generated content. The paper will then show how these data by-products can be used in tandem to improve the ability to find related data. The concluding section provides a final framing of the dual approach of using socially generated implicit data to assist in decision making within a business process.

2 Ramifications of Successful Social Production

The application of social practices to business processes has been the subject of new research [8] and social production best practices. Typically such production may be in the form of integration into business processes via a socially enabled platform such as wikis, blogs, mailing lists and ticket repositories. Such use of social techniques can, when properly managed [4], provide significant value to a business process. There is however a ramification to successful social interaction and production. This section will introduce this problem and provide a context with subsequent analysis with which to frame this paper.

When human agents interact with a business process in a successful social manner, such agents will generate significant data artefacts. Human agent is a broad general term used in this work for a person within a bounded environment, i.e. a software developer, or contributor. This human agent role will have a varying skill level, from novice to core expert developer. Such an agent will be capable of adding input to the project, via such mechanisms as email, filing a bug ticket or adding software code. The data artefacts generated by such human agents as part of a business process, when taken in aggregate, can be substantive.

Business processes, in the main, represent a restricted problem space for socially generated information artefacts. When analysing such data, rather than a web scale

problem, such as Google which deals with a potentially infinite range of quantitative kinds of data, this paper considers a much narrower domain boundary. In this paper, the domain boundary under consideration, or bounded domain, looks at all socially generated content by human agents which contribute to a business process. Restricting the problem domain to focus on bounded domains allows for differing approaches to information overload.

2.1 Information Overload

To contextualise the problem of information overload, this subsection will introduce an experimental analysis using data from Open Source software developments. This domain was chosen as such projects are inherently social since they are dependant on people interacting and adding data artefacts. These data artefacts are both communication and product driven. The communication, via email lists and ticket repositories, may be viewed as socially orientated process co-ordination. To demonstrate that excessive socially generated information has a causal relationship with a growing social population, the instantiation period of an Open Source software development project was chosen.

SourceForge was selected as the common source code repository which would be used as the basis for analysis. SourceForge was started in November 1999, has two million registered users and twenty three thousand projects. For this analysis, five projects from the top popular and active projects were selected. From SourceForge's documentation, popular is defined as top downloads for all time and active as largest number of interactions of all time. This selection was limited by some of the top projects starting before SourceForge existed and so the mailing list archive was incomplete.

This Open Source Analysis is concerned with the initial period of a project as such work gains adoption of developers and users. In order to measure this, two metrics were considered; email frequency and distinct authors. Both metrics were examined over a period of at least two years to study the trending patterns in a time series analysis. In order to normalise fluctuations in the set, the slices of time used were six month periods. A variety of differing software development project genres were used to see whether any common patterns could be observed.

Figure 1 shows such an initial period from one of the projects analysed. This figure is typical of all of the projects analysed. The X axis in the graph shown in figure 1 represents periods of six months, i.e. p0 represents a six month period as does p1. The project in figure 1 is shown with two graphs, Email and Authors. The two graphs represent the two key metrics with which this analysis is concerned. The Y axis in the case of the Email graphs shows absolute volume of emails within each project for each discrete month. In the Author graphs, the Y axis shows the absolute distinct authors for any given six month period.

The key empirical observation which may be drawn from the information presented in the figure above follows that of the intuitive response when considering a software development project gaining popularity. The intuition would be that, over time, as a project grows more developers and users would be attracted to the development process which would, as a consequence, lead to a significant increase in volume of communications.

As can be seen clearly in the graphs in figure 1, the growth of email frequency and number of human agents grow together. This analysis was repeated over numerous Open Source software development projects, both client and server side technologies, and similar patterns may be observed across their initial phases. A time series analysis

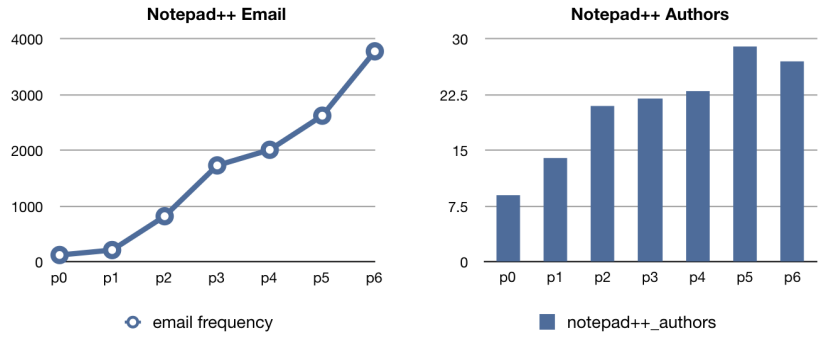

Fig. 1: Open Source Project: NotePad++

is critical to this examination in order to gain perspective on the nature of information density in these projects. If such an analysis were only to observe the end result or a discrete period in time, the correlation between a higher number of human agents directly leading to an increase in multiple agent communication would be missed.

When business processes are successfully integrating social mechanisms, acceleration of information density has the consequence of a decreasing ability to process generated data artefacts. Optimal usage of social practices within such processes requires a lightweight mechanism to respond adaptively to such social data. It is imperative to make good use of such social data, otherwise human agents within the bounds of the domain will lose motivation to participate. The next section of this paper will provide an outline of some of the current work in dealing with extracting meaning from large bodies of data

3 Information Analysis

Social information overload having now been established as a significant problem when dealing with an active group population, this section will look at current approaches to filtering such information. The section will first look at collaborative filtering techniques and secondly at other information retrieval processes. Both of these sections demonstrate an explicit act on the part of the human agents generating such data. This overview will provide a contrast basis for the two passive, or by-product, techniques described in the subsequent section.

3.1 Collaborative Filtering

Collaborative filtering is a well established research area [12]. Such a procedure looks to take the aggregate of data to find patterns of behaviour. Typically this form of analysis will use a large data set. The Movielens [10] research project is one whereby users will create an account and add ratings of movies. This collection of user data may be seen in commercial businesses such as netflix.com or lovefilm.com. The user can also indicate other users of the service as friends which the recommendation mechanism can use to build a social graph. The software then aggregates these ratings and friend data to look for the average response as to what would probably be a well received unseen movie.

Another application area of collaborative filtering is that of content filtering. The PHAOKS [21] UseNet recommendation engine uses this approach. In this system a subset of UseNet content is analysed for recommendation of web-sites from individual human agents of the system. These are then aggregated to look for the most popular recommendations. Akismet [19] takes the inverse of this same approach. Rather than using the population of the group to determine positive input, Akismet uses the same collaborative approach to determine blog spam.

Another application of the collaborative filtering mechanism is that of recommendation being applied to revision change management. Wikipedia is large Open Source project generating a significant amount of readily available revision data. By analysing their revision changes to documents [27], the aggregate response can be found in order to collaboratively filter the best edits. The subjective use of best is broadly defined as most accurate in such an application.

3.2 Information Retrieval

The second domain to be addressed in this section is that of information retrieval. Such a topic is too large to cover in this paper in depth [16], so a few examples which provide context to the subsequent work will be given in overview. These examples look at keyphrase identification in two approaches: assignment, where a phrase is selected from a controlled vocabulary or extraction where a keyphrase is automatically generated.

A machine learning approach has been used in the KEA ([25] and [9]) method. This uses Naive Bayes to try and build relevant key phrases and use those as a weighting metric to improve TF_IDF. For this approach to work, the training corpus must have key phrases identified a priori by an expert. Candidate words cannot be proper names and the work discusses the problem with explicit key phrase in relation to author submission.

GenEx [22] looks at a similar problem but using academic journal papers as the source for the corpus. In this approach, they treat the problem as one of supervised learning. The corpus in this work has a manually created set of keyphrase generated in an a priori manner by experts. This work does not look at synonyms and is considered a potential problem in the work.

Building on the GenEx work, [14] provides an extension to the supervised learning techniques by adding statistical and syntactical information from the document corpus as input to the machine learning algorithm. Such an approach also uses predetermined keywords. One final approach using an a priori keyphrase list is KIP [26]. This algorithm uses those keywords to improve precision and recall by assigning an automatic weighting based on that knowledge.

This section has provided a brief overview on some current approaches to utilising human agent generated data in a more meaningful manner. These approaches tend to use more explicit models and focus on english language based domains. As is appropriate in such domains, stemming is used in many approaches. These approaches all focus on a priori static analysis of a fixed corpus. The next section will introduce two implicit social forms of data, which when used together, can provide a more lightweight, egalitarian mechanism for deriving social significance.

4 Socially Generated Implicit Data

The previous section of this paper outlined some of the previous work focused on working with large amounts of user generated data. This section will present two differing

approaches fundamentally predicated on using the social nature of the data set. First a specific data set will be introduced. This will be the foundation for the two subsequent subsections of the paper. These subsections will show how implicit data may be used from socially generated production as part of a business process. Implicit connections between human agents within the business process may be viewed as weak ties and as such give a richer view on data interactions. At the end of the section some preliminary results will show how these approaches, when used in combination as part of a programmatic software framework, work to solve the problem of information overload discussed in section 2.1.

4.1 Data Acquisition

Before presenting the two specific social ad hoc applications, the experimental data set will be described. Both approaches use the same data set. Providing sufficient data from which to examine the two fold social production outlined in the previous section will enable a substantive view upon socially generated information.

A specific Open Source software development project was chosen for this analysis, Dojo Toolkit. Open Source data was chosen due to the large amount of available data and specifically of software project as this form of data has many business style workflows, such as project planning, milestone targets and product delivery. The group was selected as it had been in existence for multiple years and had reasonable adoption and diverse application usage.

In order to examine this project, a software framework was constructed to extract automatically multiple data silos. The phrase data silo is used to represent a collection of human agent generated data with no specific links to any other source of data. The three data silos in typical use within Open Source software development are an email mailing list, a commit database and a ticket (or bug tracking) system. In order to successfully complete any task within such a business process, research of such socially generated data is necessary. Access to such data would typically be presented by collated archived data and subscription to new inbound data. A two year period of data was targeted. From this time period some 22218 emails, 5941 commit messages and 23237 support tickets were extracted. Removing extraneous data from such extracted source material is an important step (as identified by [3]). Data cleaning forms part of the software framework. The extracted data is then used as the basis for the corpus of the analysis. Of this data, there were 1,505,585 term instances and 54510 distinct tokens were identified. The population of this group, after the reduction of multiple identities via the software framework, was 2792 human agents.

The system framework utilises standard textual analysis techniques to tokenise language elements within the data silos, uses stop words and creates an index. Stemming was not used in this work. As Stemming looks to take terms back to a root form, such processing would remove features in which this work is specially interested. This process is based on the "bag-of-words" assumptions. With this extraction process complete, the software framework can now be used to establish programmatically the social product of all human agents to the project across multiple data silos and different pseudonyms. Such programatic access via a software api within the framework provides a flexible mechanism to view the results of the process. This api could be integrated into existing or new workflows.

Now the specific experimental domain has been established, the next two subsections will present how specific socially produced data by-product can be used to find implicit weak ties. Those weak ties will then be used together to facilitate a higher degree of confidence in the ability to use such social data.

4.2 Folksonomies Within a Bounded Domain

The first of two social data by-product presented in this paper uses a posteriori method to establish high value features within the bounded domain. Ontological analysis, with respect to the addition of social practices, form an important part of this viewpoint. This evolution of perspective will be outlined to provide the framework for the technique presented.

Standard ontological techniques for feature analysis would take a formal approach to constructing a defined vocabulary which could be used to provide term weighting to proscribed features. A variant of such an approach was mentioned in section 3.2. By relying on a priori knowledge of features, an inherent formal hierarchy is placed on any such feature weighting. Such a hierarchy does not leverage the social benefits of crowd aggregation.

Prescribed taxonomies have limitations of a restricted perspective, that of the architects of the namespace, and significant upfront construction work. Applying social concepts to this problem space has led to the explosion in usage of folksonomies [23] in such applications as Flickr and Del.icio.us. Tagging is the widely used metaphor in the so called flat namespace [18]. By having no predetermined taxonomy, users of the systems are encouraged to tag a data artefact with multiple keywords, or features, which they think are indicative of the nature of the item. Such tagging can be either freeform or guided. In a guided taxonomy, as seen now in Del.icio.us, an autocomplete menu is presented to the user as they generate new tags based upon currently popular tags within their system. Such a guided taxonomy is intended to provide consensus on terminology and plurals.

Due to their nature, folksomonies are explicit in nature. The users of the system are specifically asked to define the nature of the data artefact. A further classification may be performed in a social domain by utilising the intelligence of the community, moving beyond the traditional ontological hierarchical approach and explicit tagging. Using socially implicit organically generated term features, emergent properties may be observed. By performing an analysis of all data artefacts socially generated, frequently used terms will emerge as being of significance to the group. Such an approach will only be of use in a bounded domain as the focus of production will be toward common goals.

4.3 Domain Specific Nomenclature

Linguistics has significant research into language evolution [5] and into that of slang [7] and technical jargon [6]. Domain Specific Nomenclature (DSN) refers to an aspect of language usage which is specific to a group. These terms are normally of a technical nature. DSN seeks to exploit such a facet of a bounded set of data to enable a new technique for extracting potentially high value language features.

Using the sample data as described above, all extracted terms were programmatically compared with a dictionary, using an English dictionary from the GNU iSpell project. All non-dictionary words were compiled and a subsequent frequency analysis was performed within the custom software framework.

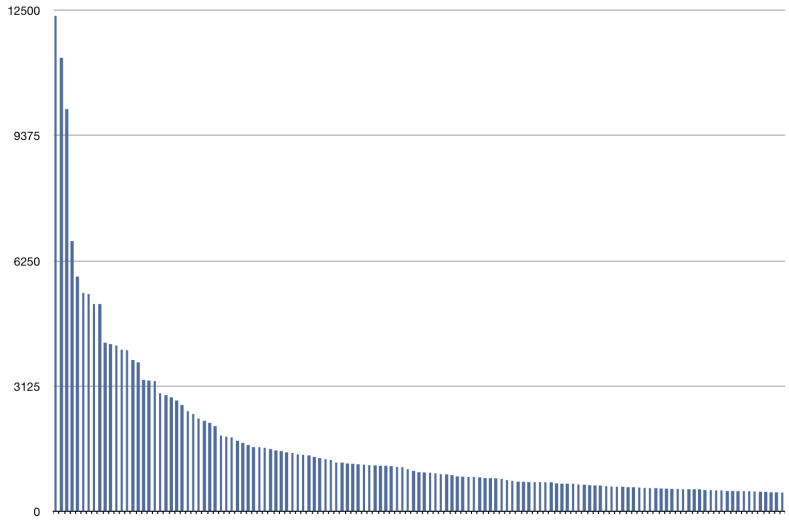

Fig. 2: Domain Specific Nomenclature Word Frequency

Figure 2 shows the result of the non-dictionary automated frequency analysis as part of the software framework. In the graph shown in figure 2, the x axis show discrete word features and the y axis show frequency of those words. The y axis data has been truncated to allow for a clearer representation. This analysis shows a clear power law curve from the Domain Specific Nomenclature terms. As standard English language terms are excluded via the dictionary reduction, all remaining terms will be specific technical terms and real names. By grouping these results by frequency it is possible to build a dynamic lexicon specific to the bounded group. This lexicon can then be used to improve term matching as these features are of specific interest to the population of the group.

As the results form a power curve distribution, not all of the derived terms will be of value. Neither the top, nor long tail [1] of the distribution are of descriptive use. For example, the most popular term, dojo, was referenced 99075 times. As this term is so frequently used, it cannot be considered a distinguishing characteristic. The long tail of the distribution is also of low value as low frequency term usage signifies little usage within the group. Therefore this approach targets the so called *"fat middle"*, looking at the eighty percent middle of the distribution. This DSN lexicon is now used in combination with the results from Term Frequency-Inverse Document Frequency (TF_IDF) to add term weightings.

There is an important restriction to this approach that is inherent within that of the *"Wisdom of Crowds"* [20]. In his work, Surowiecki states that the collection of independently-deciding individuals must be of both significant enough size and diversity. Domain Specific Nomenclature requires a significant data set with a diverse population in order to form an effective ad hoc lexicon. Synonyms are not considered a problem in this approach. Previous work 3.2 identified this as a problem to be solved

but the DSN approach specifically targets unique technical terms, or jargon. As such terms will have been created by the group, duplicate derived words are unlikely to occur. If such a fork in term usage does occur, the DSN approach will observe such a change and identify trending usage.

From this subsection, it is possible to see the progression from hierarchical (formal), to the explicit addition of social (folksomony) and then to the passive social (implicit). Such implicit data can be used in the bounds of a restricted domain to produce automatically and adaptively a Domain Specific Nomenclature lexicon. The next subsection will describe the second social production by-product, implicit social proximity and how such data will be used in conjunction with DSN to provide the ability to utilise social data more effectively.

4.4 Implicit Social Proximity

The previous subsection demonstrated how implicit social production can be used to build a lexicon of high value Domain Specific Nomenclature which can be used as the basis for term weighting. The second social by-product looks at social connections [24] between human agents within a bounded domain and how these approaches can be used with each other.

Implicit Social Proximity (ISP) looks to find connected groups within the context of grouped data in a dynamic manner, rather than for posteriori analysis [13]. Using the experimental data as described above 4.1, an initial data analysis was performed on email mailing list threads and ticket system threads. In other business processes such interactions could occur in multiple areas such as in a wiki editing revision system, or blog comment thread. When two or more human agents interact in the same thread, or common socially produced artefact, it is possible to look for previous interactions with that agent subset. A subset, in this instance, is viewed as a partial set of the complete human agent user base. An analysis tool, as part of the software framework, was constructed to look at social proximity between human agents.

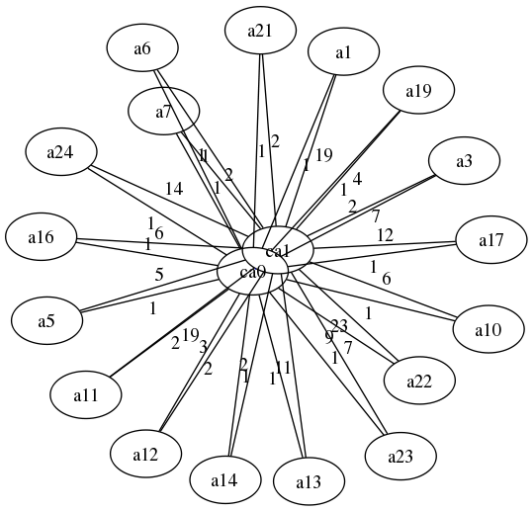

Fig. 3: Social Intersection From Weak Ties

Figure 3 shows an automatic dynamically generated intersection of two human agents within a common email thread. The nodes are representative of a discrete human agent. Nodes *ca0* and *ca1* are the two human agents under consideration of a specific email thread in which they both participate. The software framework then looks for all possible common interactions between human agents and the original two human agents. Nodes *a1* through *a24* are in the general population of the domain but which have previously interacted with both of the original human agents. The edge weighting show the frequency of interactions with the same human agents.

This automatically generated data can be used in a set of socially generated data artefacts to find human agents whose social connections can act as votes of confidence. The process of creating these proximity maps dynamically alter as new socially generated artefacts are added to the framework. Such additions happen in near real time, using social data dynamically to provide a fresher opinion predicated upon the special social zeitgeist of the social business process.

These two approaches are then used in conjunction with TF_IDF to perform within the bounded business process. Domain Specific Nomenclature uses a current in memory model based on the freshest lexicon to provide relevant feature weighting. Implicit social proximity is then used as a further weighting to push increased discoverability of highly linked human agent generated content. As such, this dual process is entirely egalitarian in nature as there is no predetermined hierarchy and uses weak ties to improve the quality of related data. With an understanding of these two approaches in place, the next subsection will show some initial early findings.

Preliminary Usage An initial small scale evaluation was performed. This evaluation was to determine the effectiveness of the described process, not to detail a deployment within a business context. The participants in this study were programmers with at least two years javascript development experience. This choice is of significance, as the Open Source project under examination is a javascript framework.

For the test procedure, an automated web based testing framework was created. There were no direct interactions in the testing procedure with the subjects apart from an initial instruction page within the testing software. In the evaluation, the five participants were shown three discrete documents, in this instance emails from the test data. These documents were outside of the test corpus so as to not base any predictions on already known data. They were then shown three different potentially related documents per method, positioned randomly, and asked if they thought there were useful correlation. The documents for evaluation were either: selected randomly, using the Numpy random number generator, a standard TF_IDF implementation or TF_IDF enhanced by both the automatically generated social lexicon and the implicit social proximity. Table 1 shows the results of the experts' acceptance of the potentially related documents.

Table 1: Comparative Usage Table

Document	Random	TF_IDF	Social
A	7%	40%	73%
B	13%	47%	60%
C	7%	53%	87%

As expected, the random selection had very low acceptance. TF_IDF performed well in most cases but with the addition of the two social processes described above, the experienced developers noted an improvement. These results are too small in scope to provide a high level of confidence in the general application of this approach but suggests a further more in-depth study would be of value.

Domain Specific Nomenclature and Implicit Social Proximity use socially produced data artefacts to generate a mechanism to improve the ability for a business process to leverage human agent generated content. Such a mechanism would have less value if it only worked in an a priori manner as the corpus of the content will change quickly. The proposed approach processes new data artefacts in near real time. As the framework is lightweight it can respond to ad hoc changes in social usage and behaviour within the business process. The framework presents access to the data via an api and, as such, may be integrated into existing business applications without requiring retooling. This ease of integration is likely to encourage adoption.

5 Conclusions and Future Work

Bringing social applications to business processes can provide valuable input and generate work of significance. Without a mechanism to filter this content, when any substantive scale as been achieved, information overload becomes a problem which should be addressed.

This paper introduced two socially orientated approaches: Domain Specific Nomenclature and Implicit Social Proximity as part of a software framework. Both approaches use existing socially generated data artefacts to derive information. Focusing on terms specific to a domain enables ad hoc lexicons to be dynamically created based upon the most up to date information generated by the group. Such data enables weighting of terms in a dynamic folksonomic manner. The further use of social data generated by human agent interactions facilitates the promotion of content based on popular subset intersections. Both of these approaches require no alteration in human agent behaviour or adjustment to any existing workflows. From this basis, business applications which have already integrated social interactions could utilise such an approach from existing data sets via the programatic interface.

In the future, the first step is to conduct a wider study with a larger range of test documents and more domain experts to provide a higher degree of confidence in the approach. Further applications of DSN could be used, dependant on the business process. In the experimental domain considered in this paper, the whole body of terms were considered, independent of when they were created. In another, time critical news based domain, freshness of term evolution could be of a greater weighting value. Another area under consideration is to expand the silo concept to take in additional information, such as wiki revision changes and blog comment threading. By utilising implicit data by-products, it is possible to improve the ability to use the output of social production in an inherently egalitarian manner. Such enhancements will support ad hoc business processes and enable lightweight interactions.

References

1. C. Anderson. *The Long Tail: Why the Future of Business Is Selling Less of More*. Hyperion, July 2006.

2. Y. Benkler. *The Wealth of Networks: How Social Production Transforms Markets and Freedom*. Yale University Press, October 2007.
3. N. Bettenburg, E. Shihab, and A. E. Hassan. An empirical study on the risks of using off-the-shelf techniques for processing mailing list data. In *ICSM'09: Proceedings of the 25th IEEE International Conference on Software Maintenance*, pages 539–542, 2009.
4. J. Bosman. Chevy tries a write-your-own-ad approach, and the potshots fly. *The New York Times*, Jan 2006.
5. M. Christiansen and S. Kirby. Language evolution: Consensus and controversies. *Trends in Cognitive Sciences*, 7(7):300–307, 2003.
6. R. Coombs, S. Chopra, D. Schenk, and E. Yutan. Medical slang and its functions. *Social Science & Medicine*, Jan 1993.
7. B. Dumas and J. Lighter. Is slang a word for linguists? *American Speech*, Jan 1978.
8. S. Erol, M. Granitzer, S. Happ, S. Jantunen, B. Jennings, A. Koschmider, S. Nurcan, D. Rossi, and R. Schmidt. Combining bpm and social software: Contradiction or chance? *Software Process Improvement and Practice Journal Special Issue on BPM 2008 selected workshop papers 2009*, 9999(Special Issue on BPM 2008), 2009.
9. E. Frank, G. W. Paynter, I. H. Witten, C. Gutwin, and C. G. Nevill-Manning. Domain-specific keyphrase extraction. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI1999)*, pages 668–673, 1999.
10. N. Good, J. Schafer, J. Konstan, and A. Borchers. Combining collaborative filtering with personal agents for better recommendations. *Proceedings of AAAI*, Jan 1999.
11. M. Granovetter. The strength of weak ties. *ajs*, 78(6):1360, 1973.
12. J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Trans. Inf. Syst.*, 22(1):5–53, January 2004.
13. L. Hossain and D. Zhu. Social networks and coordination performance of distributed software development teams. *The Journal of High Technology Management Research*, Jan 2009.
14. A. Hulth. Improved automatic keyword extraction given more linguistic knowledge. *Proceedings of the 2003 conference on Empirical methods in natural language processing*, 10:216–223, 2003.
15. H. Kagdi and D. Poshyvanyk. Who can help me with this change request? *Proceedings of 17th IEEE International Conference on Program Comprehension*, 9:273–277, May 2010.
16. R. Kosala and H. Blockeel. Web mining research: A survey. *ACM SIGKDD Explorations Newsletter*, Jan 2000.
17. D. Ma, D. Schuler, T. Zimmermann, and J. Sillito. Expert recommendation with usage expertise. In *Proceedings of the 25th IEEE International Conference on Software Maintenance*, September 2009.
18. A. Mathes. Folksonomies-cooperative classification and communication through shared metadata. *Computer Mediated Communication*, Jan 2004.
19. M. Mullenweg. Akismet, 2007. `http://akismet.com/faq/`.
20. J. Surowiecki. *The Wisdom of Crowds*. Anchor, August 2005.
21. L. Terveen, W. Hill, B. Amento, D. Mcdonald, and J. Creter. Phoaks: A system for sharing recommendations. *Communications of the ACM*, 40(3):59–62, 1997.
22. P. Turney. Learning algorithms for keyphrase extraction. *Information Retrieval*, Jan 2000.
23. T. Vanderwal. Folksonomy, 2007. `http://www.vanderwal.net/folksonomy.html`.
24. S. Wasserman and K. Faust. *Social Network Analysis: Methods and Applications (Structural Analysis in the Social Sciences)*. Cambridge University Press, January 1995.
25. I. H. Witten, G. W. Paynter, E. Frank, C. Gutwin, and Craig. Kea: Practical automatic keyphrase extraction. In *ACM DL*, pages 254–255, 1999.
26. Y.-F. B. Wu, Q. Li, R. S. Bot, and X. Chen. Domain-specific keyphrase extraction. In *CIKM '05: Proceedings of the 14th ACM international conference on Information and knowledge management*, pages 283–284, New York, NY, USA, 2005. ACM Press.
27. H. Zeng, M. Alhossaini, L. Ding, R. Fikes, D. McGuinness, et al. Computing trust from revision history. *Intl. Conf. on Privacy, Security and Trust*, 2006.