

Using Data Mining for Accurate Resource and Skill Demand Forecasting in Services Engagements

Jianying Hu, Moninder Singh and Aleksandra Mojsilovic

IBM T.J. Watson Research Center

1101 Kitchawan Road, Yorktown Heights, NY 10598, U.S.A.

{jyhu,moninder,aleksand}@us.ibm.com

ABSTRACT

For efficient services delivery, it is imperative that an organization be able to accurately forecast demand for various projects/skills, determine optimal staffing levels and resource allocations, etc. One way of doing so is by mining ongoing and in-pipeline project data to estimate expected demand. In order to do this accurately, each project must be accurately labeled to reflect the correct, pre-defined solution category it belongs to, since different solution categories have different staffing requirements and different cost profiles, etc. However, because of dynamic business environments and changing customer needs, solution portfolios are constantly evolving and are frequently redefined, limiting the ability of project managers to categorize projects accurately. We describe a new approach to solving this problem by formulating it in a semi-supervised clustering framework, and discuss its application in a web-based decision support system, called OnTheMark (OTM), that is being developed and deployed at IBM for demand forecasting and capacity planning as well as computing business metrics for assessing the quality of services delivery.

Categories and Subject Descriptors

H.2.8 [Database Management]: Database Applications—*Data Mining*; I.2.6 [Artificial Intelligence]: Learning; H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval—*Clustering*

General Terms

Algorithms

Keywords

Semi-supervised clustering, soft seeds, k-means, demand forecasting.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

DMBA '08, August 24, 2008, Las Vegas, Nevada, USA.
Copyright 2006 ACM 1-59593-439-1...\$5.00.

1. INTRODUCTION

A large organization, such as IBM, has multiple active service engagements at different stages of completion at any given time, with competing demands on its human resource supply. It is thus crucial for such organizations to effectively manage its end-to-end services delivery process, and have a standardized, consistent and optimized planning process that enables it to make sound business decisions on managing labor capacity. This entails the ability to process demand data accurately and forecast demand for various types of projects and associated resource skills, model different and complex (e.g., multi-skill, “what if”) gap/glut scenarios, determine optimal staffing levels to maximize profits/revenues or minimize risk of engagement loss, determine optimal resource allocations, as well as enable interlock and cross-collaboration, and support higher level strategic solution-portfolio planning. Failure to efficiently manage this process can have drastic consequences, including long and passive planning cycles, misalignment between opportunities and resources, and lack of interlock between sales and delivery, thereby leading to diminished profitability and increased risk of disruptions.

A crucial component of this forecasting and planning process is the analysis of current as well as (anticipated) future projects to estimate expected demand for various types of projects and associated skills. In order to do this accurately, each project must be accurately labeled to reflect the correct, pre-defined solution category it belongs to, since different solution categories have different staffing requirements and different cost profiles, etc. However, because of dynamic business environments and changing customer needs, solution portfolios are constantly evolving and are frequently redefined, limiting the ability of project managers to categorize projects accurately. Hence, there is a need for an automated methodology to map projects into a set of pre-defined, but highly dynamic, solution categories.

In this paper, we describe a new approach to solving this problem by formulating it in a semi-supervised clustering framework, and discuss its application in a web-based decision support system, called OnTheMark (OTM), that is being developed and deployed at IBM for demand forecasting and capacity planning as well as computing business metrics for assessing the quality of services delivery.

The rest of the paper is organized as follows. In Section

2, we give a brief description of the OnTheMark system. We describe the project categorization problem in Section 3 and discuss our proposed solution to this task in Section 4. We present an evaluation of this approach in Section 5, and discuss our future plans in Section 6.

2. SYSTEM OVERVIEW

The OnTheMark (OTM) system provides a platform for integrated workforce management, as well as acts as a decision support system for tracking critical business metrics for assessing the quality of services delivery by the Integrated Technology Services (ITS) business line of IBM Global Technology Services (GTS), one of the three business units that form IBM Global Services, the multi-billion dollar services arm of IBM that provides IT outsourcing, web hosting, and consulting and systems integration services.

OTM combines top-down revenue projections and bottom-up planning from the backlog of ongoing engagements as well as the opportunity pipeline to perform demand forecasting and skills capacity planning on a rolling, four-quarter basis. It also utilizes a statistical forecasting methodology to analyze the opportunity pipeline data and predict how many of these opportunities will turn into reality. Currently deployed in the United States by ITS, OTM is proving to be critical for ensuring that skills demand is properly estimated, appropriate investments are made, and optimal channels are utilized, and has led to considerable reduction in planning cycle time, and improved accuracy and insight into the business as well as the potential for improved profitability. As per GTS request, OTM will become the GTS worldwide capacity planning standard, and is being gradually deployed worldwide throughout 2008.

Three kinds of data from strategic and operational business systems is used by OTM to drive the creation of the demand and capacity plans. These are (i) demand data from sales backlog as well as existing project pipeline, (ii) supply information based on the role and skill of each practitioner within each SPL¹ in the solution portfolio, and (iii) the work breakdown structures (staffing models) for each solution within every SPL. The work breakdown structure provides the list of resources, skills and time needed to execute a standard solution. The capacity plan by SPL is the final deliverable from the ITS capacity planning process. The capacity plan relies on the data described above to determine the gap/ glut of resources by skill within each SPL. Delivery executives, working with the ITS skills team and senior management, then apply their knowledge of the business to complete the capacity plan for their respective SPL.

The importance of accurate categorization of the projects to the solution portfolios becomes apparent by looking at the forecasting process in detail. As shown in Figure 1, demand forecasting in OTM considers three types of engagements: (A) Ongoing engagements, (B) Pipeline opportunities that are engagements that are still in different stages of the sales process and could potentially become signed engage-

¹IBM ITS offerings are arranged as a set of solution categories called service-product lines or SPLs, each consisting of multiple solutions spanning a specific services area, such as business continuity and resiliency services, middleware services, etc.

ments during the forecasting quarter and thus incur demand on the workforce, and (C) Wedge engagements that are engagements that currently don't fall in (A) or (B) at forecasting time, and account for the revenue differential between the revenue targets and the sum of the expected revenues from the ongoing engagements and opportunities. To forecast demand for ongoing engagements, the system first calculates the expected revenue from each engagement based on the revenue generated from the engagement to date as well revenue generated from prior engagements dealing with the same SPL solution as that project. The estimated revenue for each engagement is then converted into work hours using the revenue rate specific to the SPL solution associated with that engagement. Finally, the work hours are converted into a demand/capacity plan for that engagement using the staffing models associated with the SPL solution for that engagement. Each step of the process, thus, depends upon the correct categorization of the engagement to the appropriate SPL solution. As discussed earlier, correct categorization is made even more critical by the constantly changing solution portfolios (due to dynamic business environments and changing customer needs) that limits human categorization and necessitates automated mapping. The demand for opportunities is similarly calculated by first estimating revenue for each engagement using a statistical revenue forecasting model, based on the probability of actually winning the opportunity, estimated duration, revenue profiles and the SPL solution assigned to that project by the project manager. Once the revenue estimate is available, demand is calculated as in the case of ongoing engagements. Demand for the third kind of engagements (wedge) is similarly calculated by estimating revenue (difference of revenue targets and expected revenue from ongoing and pipeline engagements) and then applying SPL solution specific revenue rates and staffing models to generate the demand statement. Note that for both opportunities as well as wedge demand forecasting, the accuracy of the demand statement depends upon the accuracy of the SPL solution specific revenue rates and staffing models, which in turn depend upon the correct categorization of completed and ongoing engagements to appropriate solution categories.

3. PROBLEM DESCRIPTION

As described previously, the project categorization problem boils down to the mapping of each project to an appropriate category, where each category is defined by a category name along with one or more category descriptions specifying the scope of that category. Thus, for example, one such category could be named "Server Product Services for Microsoft" while the associated category descriptions may include "MS Application Development and Integration Services", "MS Evaluation and Planning for On Demand", etc. On the other hand, each project is associated with a set of basic features comprising of a skill allocation vector that is computed from the actual hours billed for resources of various skills on that project [7]. Thus, each basic feature corresponds to a particular skill and the value of the feature in that vector is the proportion of time that was billed for that skill on that project. For example, consider a project that required 30% of Project Manager (PM) time, 30% of Software Architect (SA) time and 40% of Software Developer (SD) time and 0% for all other skills. The basic feature set would comprise of a feature corresponding to each skill,

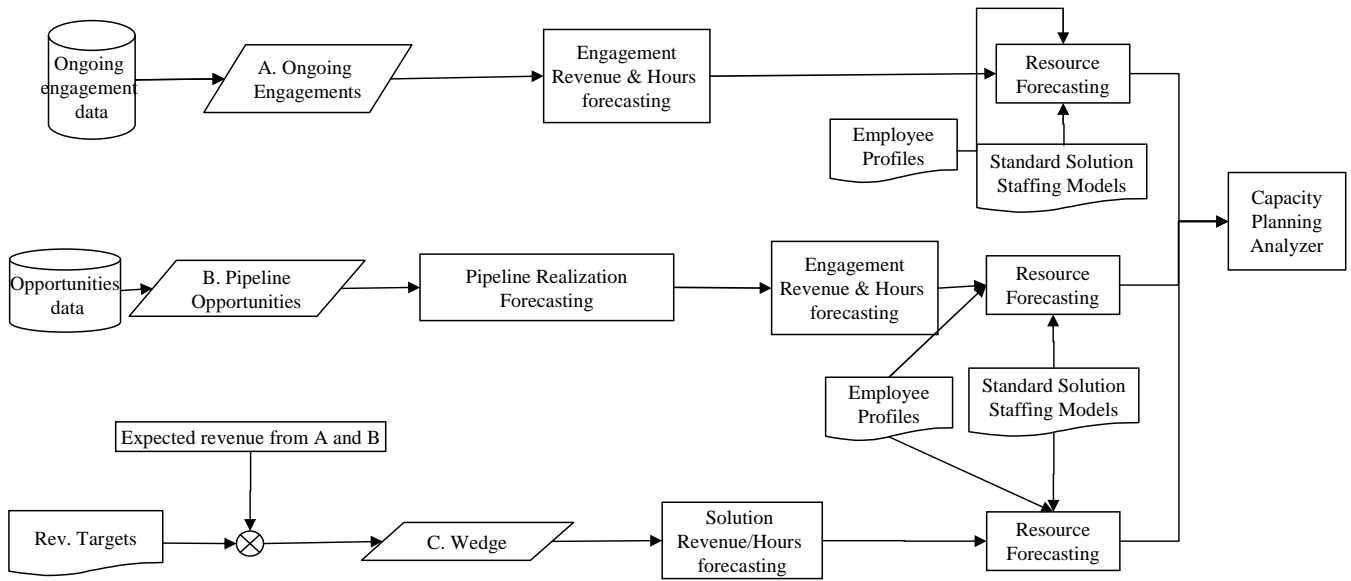


Figure 1: Flowchart of the OTM demand forecasting process.

with values corresponding to the proportional time billed for each, such as 0.3 for PM, 0.4 for SD, etc. Other attributes such as size (in terms of revenue), duration and profit margin could also be included in the basic feature set. In addition, each project has an *optional* description field containing noisy, unstructured text descriptions specifying (potentially) the nature of the project, typically entered by the project manager at the beginning of the project. These descriptions, if present, are limited by a fixed field-width constraint, and are thus fairly short (four-five words or less). Since the project managers are not required to adhere to a common taxonomy or fixed conventions or terminology, the descriptions vary widely in the quality of information they contain. Sometimes, the description is fairly informative, such as “iSeries consult svcs”, “zOS planning & migration”, “perform review system i”, “GDPS SW, Implementation”, etc. At other times, it may be too general, such as “hourly svcs” or “application infrastruct”, or totally non-informative, such as “P001”, “PCR#2” or “prime bidder”. This is in sharp contrast to the category names and descriptions (as described previously) that are much more cleaner, longer and descriptive.

This leads to a particular class of categorization problems that can be characterized as follows: (i) there is a dataset with underlying *basic features* attached to every point; 2) there is a set of categories, each accompanied with *category descriptions*; and 3) some or all of the data points also have textual *data descriptions*, generated independently of the category descriptions (e.g. outdated labels, or labels assigned without any predefined taxonomy in mind).

While matching of objects to predefined categories can often be carried out via supervised classification, in our application it is extremely difficult to generate training examples for all categories. This is because the predefined solution taxonomy is highly dynamic and constantly evolving because of changing market requirements in the consulting business. This makes it impossible to rely on manual labeling or sam-

ple selection, as they would have to be conducted whenever the taxonomy changes. We therefore needed to come up with a “training sample free” solution that can be used to assist project categorization.

4. TECHNICAL SOLUTION

A careful study of the data reveals that the project descriptions, when available, some times provide insight as to which category the project likely belongs to that is independent from what one can learn from basic features alone. Such descriptions are difficult to incorporate directly as additional features since this is an optional field that is sparsely filled with varying qualities. On the other hand, the insight gained from these descriptions can be captured and used to guide the clustering process based on the dense basic features.

We thus cast this problem as a semi-supervised clustering one, and developed a solution wherein text-based matching between category and data descriptions is used to generate “soft” seeds that are subsequently used to guide clustering in the basic feature space. We introduce a novel *seed re-assignment penalty* measure to effectively make use of the text matching results with varying degrees of confidence.

Past research has proposed various methods to incorporate “constraint violation penalty” in clustering with pair-wise constraints [11, 3], but not for cases using seeds with varying confidence levels in a k-means setting. While semi-supervised clustering has been the focus of several recent projects [11, 1, 3], most of the prior studies were carried out using constraints that were artificially generated from true labels. A few real world applications using pair-wise constraints have been described in the past [11, 8] in applications including GPS-based map refinement [11] and landscape detection [8]. However, to our knowledge, the current paper is the first to demonstrate the benefit of semi-supervised clustering using seeding in a real world application.

Our solution consists of two steps: (i) soft-seed generation, followed by (ii) soft-seeded semi-supervised clustering.

4.1 Soft-Seed Generation

In the first step, seeds are generated on the basis of the similarity between the available project descriptions and the category descriptions, measured using the standard TF*IDF (Term Frequency - Inverse Document Frequency) method [10]. Using the similarity score as a measure of the confidence in the mapping (with identical strings having a score of 1 and totally different strings having a score of 0), a seed set can then be generated based on a chosen threshold, T , on the score. Thus, for each project, if the maximum similarity score is greater than T , then the category corresponding to that score is assigned to the project instance; otherwise, the project is not assigned to any category. The set of projects labeled in this way (along with the corresponding similarity scores, referred to henceforth as seed scores) provide the “soft seeds” for clustering. We refer to these as “soft” seeds for two reasons: (i) The labels based on textual matching are not guaranteed to be accurate. Therefore they are not viewed as hard constraints on cluster membership, but rather soft ones with varying degrees of strength tied to the confidence score, and (ii) the seeds do not necessarily provide complete coverage of all categories. Note that in the latter case, semi-supervised clustering cannot be used to achieve completely automatic categorization (nor can any other learning scheme). However it still provides great value by reducing the amount of manual labeling required since now one only needs to map the “un-covered” clusters to the “un-covered” categories, instead of having to label individual data points,

Clearly, the choice of T determines the trade-off between the coverage and quality of the seed set: the higher the threshold, the “cleaner” the seed set, but it also provides less coverage. As shown in the experiments later, when the seed labels are used as hard constraints the clustering outcome is highly sensitive to the selection of T . This sensitivity can be greatly reduced through the introduction of soft seeds. In fact, our experiments indicate that using the proposed soft seeded k-means algorithm one can achieve significant benefit from seeding even when no threshold is used (i.e., when results from text matching are included), thus largely eliminating the need of choosing the hard threshold T in practice.

4.2 Soft-Seeded K-means

In the second step, the soft seeds generated in the first step, along with the corresponding correspondence scores, are used to cluster the projects by incorporating these into a semi-supervised clustering framework. For the clustering exercise, we chose to use a k-means type algorithm since it has been successfully used in a variety of application domains, including image segmentation, information retrieval, text categorization, and business analytics [9, 2, 4, 7]. While the original algorithm was designed to operate in a completely unsupervised manner, several semi-supervised variations have been proposed recently to take into consideration “side information” in the form of both class labels and pairwise constraints [11, 1, 3]. The new variation we have developed, **Soft Seeded k-means**, is most closely related to the Seeded k-means and Constrained k-means algorithms pro-

posed by Basu *et. al.* [1]. The differences are that in Seeded k-means and Constrained k-means the seeds are either used only for initialization, or treated as hard constraints. Furthermore, both algorithms assume that there is at least one seed for each cluster. Our algorithm treats the seeds as “soft” constraints through the introduction of a seed re-assignment penalty, and allows incomplete coverage of clusters by seeds.

To describe the Soft Seeded k-means algorithm, we first define some notations. Let $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$, $\mathbf{x}_i \in \mathbb{R}^d$ denote the data set, $\mathbf{L} = \{l_1, l_2, \dots, l_n\}$ denote the seed label vector where $l_i = 0$ if \mathbf{x}_i is unlabeled, and $l_i = j, j \in [1, \dots, k]$, if \mathbf{x}_i is a seed assigned to cluster j . Furthermore, $\mathbf{S} = \{s_1, s_2, \dots, s_n\}$ denotes the seed score vector where $s_i \in [0, 1]$ and $s_i = 0$ if \mathbf{x}_i is not a seed. Finally, let k be the total number of clusters, and m be the number of clusters covered by seeds. Without loss of generality, we assume that the first m clusters are covered. The algorithm can then be described as follows:

1. Iterate $m=1:M$
 - (a) Initialize centroids ξ_1, \dots, ξ_m using labeled data points.
 - (b) Initialize centroids of the remaining clusters through random sampling of unlabeled data points.
 - (c) For each data point x_i , assign it to the cluster j^* that minimizes: $\mathcal{D}(x_i, \xi_j) + \mathcal{P}(j, l_i, s_i)$.
 - (d) For each cluster C_i , update its centroid using current assignments
 - (e) Iterate between (c) and (d) until convergence (i.e., no change of membership take place)
 - (f) Store resulting partition its *model fitness score* \mathcal{F}_m .
2. Return the partition with the highest score \mathcal{F}_m .

Here $\mathcal{D}(x, y)$ is a pairwise distance measure defined over \mathbb{R}^d , and $\mathcal{P}(j, l_i, s_i)$ is the seed re-assignment penalty measure defined as following:

$$\mathcal{P}(j, l_i, s_i) = \begin{cases} 0 & \text{if } j = 0 \text{ or } j = l_i \\ \frac{\gamma}{1 + e^{-(s_i - \beta)}} & \text{otherwise} \end{cases} \quad (1)$$

In other words, the penalty is 0 if a data point is either not a seed, or it is a seed assigned to the same cluster as indicated by the seed label. Otherwise, the soft seeding constraint is violated and the penalty incurred is a sigmoid function of the seed score s_i (described in Section 4.1). The sigmoid implements a soft stepping function with values in the range $(0, \gamma)$. The middle point of the soft step is defined by β and the slope of the step is controlled by α . In general, γ should equal the maximum pairwise distance, and a reasonable value for β is the mean of the seed confidence scores.

As in any k-mean style algorithm, the inner loop of the algorithm only converges to a local minimum, and thus depends heavily on centroid initialization. For clusters covered by seeds, the initial centroids are estimated using the seeds. For the remaining clusters, no such side knowledge exists, and we are faced with the same initialization challenge as in unsupervised k-means clustering. It has been shown that one of the most effective methods of getting better initialization in this case is to perform multiple runs with different random initializations, followed by model selection [5]. This is the strategy adopted by our algorithm.

Table 1: Seed Coverage and Accuracy at Different Confidence Levels

Conf.	0.8	0.7	0.6	0.5	0.4	0.3	0.2	0.1
% of total	9	10	31	32	33	40	43	59
Coverage	1	2	4	5	5	7	7	7
Accuracy	100	100	100	99	98	96	89	78

5. EVALUATION

Evaluation of the proposed project categorization methodology is constrained by the very nature of the problem: the project to project-category mappings available in the project data (done by project managers) are often invalid due to the changing categories, and the manual relabeling of the projects by domain experts is very difficult due to the scale and tediousness of the process. So, we focussed our experiments on a representative and significant SPL from IBM ITS for which the project mappings were validated/re-mapped based on domain experts' input. The data set contains 302 projects from the Server Services Product Line, belonging to 8 predefined categories. Each project is associated with a 67 dimensional skill allocation vector. The pair-wise vector distortion is computed using L_1 distance. For results reported in this paper, $\alpha = 10.0$, $\beta = 0.4$, and $\gamma = 2.0$.

For each category, the category descriptions were first processed for punctuation and stop-word removal, stemming, abbreviation expansion (e.g. VOIP is expanded to Voice over IP, IPT is expanded to IP Telephony, etc.) and domain specific term mapping (e.g. AS400 is mapped to iSeries, System 390 is mapped to zSeries, etc.). The resulting descriptions were then tokenized into word-tokens using white space as delimiters. Finally, for each project description, the TF*IDF similarity score was computed between that description and each of the category descriptions, and used as a measure of the confidence in the mapping.

We generated 8 different sets of seeds by setting the threshold at levels equally spaced between 0.8 and 0.1, which covers the range of all non-zero text matching scores on this data set. Table 1 shows the coverage and accuracy of the different sets of seeds. At the highest threshold, only 9% of the data points are seeds, covering just one cluster, with 100% seed accuracy. As the threshold is lowered the seed percentage and coverage increases, and the seed accuracy decreases. Note that in this data set, even when the threshold is set at the lowest level, i.e., all information from text matching is included, the seeds still do not provide complete coverage of the clusters.

The Soft-Seeded k-means algorithm (SSk-mean) was then ran and compared against standard k-means (k-means), Seeded k-means (Sk-means) and Constrained k-means (Ck-means) [1], for all seed sets. To evaluate the outcome of each clustering algorithm, we used the pairwise F-Measure [3], defined as:

$$Precision = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsPredictedInSameCluster}$$

$$Recall = \frac{\#PairsCorrectlyPredictedInSameCluster}{\#TotalPairsInSameCluster}$$

$$F - Measure = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

Each algorithm was run 10 times at each seed setting, and the average F-Measure is reported. For all algorithms the number of random initializations was set to 10. In the case of Seeded and Constrained k-means, since the original algorithm assumed complete coverage, to make it applicable here we used random initialization for the uncovered clusters, followed by standard model selection using overall distortion, which appears to be consistent with the setting used for the experiments reported in [1].

Results are shown in Fig. 2. As seen in the plot, the proposed Soft Seeded K-means algorithm significantly outperforms both Seeded k-means and Constrained k-means. One can also observe that the new algorithm is better at addressing the trade off between the benefit of including more seeds and thus incorporating more side knowledge, and the drawback of allowing more noise in the seeds. In general, when the seed confidence levels are relatively high, and thus the seed accuracies are high, including more seeds naturally increases the clustering performance. However as the seed accuracy gets much lower, the performance tends to decrease again. Constrained k-means outperforms Seeded k-means at high confidence levels, however its performance drops sharply as the seed accuracy gets below 95%. On the other hand, the Soft Seeded k-means algorithm is able to achieve better performance at all levels. In some sense, the soft seeding mechanism allows it to combine the best of the previous algorithms such that it is robust against seed noise while at the same time makes most effective use of the constraints introduced by the seeds.

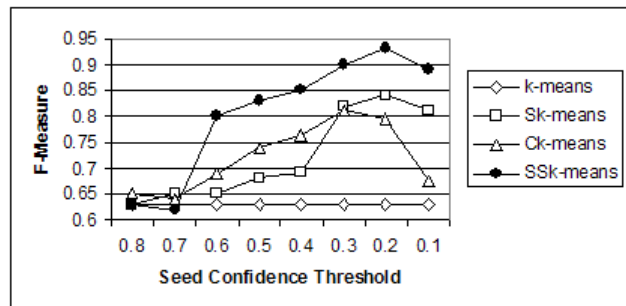


Figure 2: Performance Evaluation Results

6. FUTURE PLANS AND CONCLUSION

We have described a system, OnTheMark, that is being developed and deployed across IBM ITS to help manage services delivery and compute business metrics that assess the quality of service. One of the critical functionalities of OTM is the ability to accurately forecast demand for various projects/skills, which in turn needs projects to be correctly mapped to appropriate project categories. However, given the dynamic nature of business and changing customer needs, the project categories themselves are constantly evolving and frequently getting redefined. This leads to the need for an automated method for categorizing projects to such dynamic categories. We have presented a novel so-

lution to this class of categorization problems using a semi-supervised clustering approach, by introducing a novel variation of the k-means algorithm called Soft Seeded k-means, which can make effective use of the side information provided by seeds with a wide range of confidence levels, even when they do not provide complete coverage of the pre-defined categories. This large degree of flexibility, which is key in making semi-supervised clustering work in practice, is achieved through the introduction of a novel seed re-assignment penalty measure. Experiments using real world ITS data demonstrate significant improvements over previous algorithms. While this new methodology is being currently applied off-line to the project data for use in the OTM forecasting process, we are currently in the process of fully integrating it with the web-based OTM tool that is being gradually rolled out to IBM ITS worldwide in 2008.

We are also exploring additional ways of improving the performance of the clustering algorithm, both in terms of the quality of the seeds generated as well as the quality of the clusters discovered. While we currently use simple techniques, such as stop word removal and abbreviation expansion, for matching project descriptions with category descriptions, it is likely that the matching process can be greatly improved by the use of lexical databases, such as WordNet [6], that provide a way of enhancing similarity calculations through the use of synonyms, sense determination, morphological analysis, part of speech determination and identification of different word forms, etc., instead of just straightforward token similarity provided by vanilla string similarity methods. Hence, we are currently integrating WordNet into the seed generation step of the algorithm. Another enhancement we are exploring is in terms of better model initialization or selection metrics for situations where the seed coverage is incomplete (i.e., the seeds do not cover all categories), since the quality of the clusters formed are heavily dependent on the initial centroids.

7. REFERENCES

- [1] S. Basu, A. Banerjee, and R. Mooney. Semi-supervised clustering by seeding. In *Procs. 19th ICML*, pages 19–26, 2002.
- [2] P. Bellot and M. El-Beze. *A clustering method for information retrieval (Technical Report IR-0199)*. Laboratoire d’Informatique d’Avignon, France, 1999.
- [3] M. Bilenko, S. Basu, and R. Mooney. Integrating constraints and metric learning in semi-supervised clustering. In *Procs. 21st ICML*, Banff, Canada, 2004.
- [4] I. S. Dhillon and D. S. Modha. Concept decompositions for large sparse text data using clustering. *Machine Learning*, (42):143–175, 2001.
- [5] U. Fayyad, C. Reina, and P. Bradley. Initialization of iterative refinement clustering algorithms. In *Procs. 4th KDD Conference*, 1998.
- [6] C. Fellbaum. *Wordnet: An Electronic Lexical Database*. MIT Press, 1998.
- [7] J. Hu, B. Ray, and M. Singh. Statistical methods for automated generation of service engagement staffing plans. *IBM Journal of R&D*, (3), March 2007.
- [8] Z. Lu and T. Lean. Semi-supervised learning with penalized probabilistic clustering. In *Proc. NIPS*, 2004.
- [9] J. Marroquin and F. Girosi. *Some extensions of the k-means algorithm for image segmentation and pattern recognition (AI Memo 1390)*. Massachusetts Institute of Technology, Cambridge, MA, 1993.
- [10] G. Salton and C. Buckley. Term weighting approaches in automatic text retrieval. Technical Report 87-881, Comp. Sci. Dept, Cornell University, 1987.
- [11] K. Wagstaff, C. Cardie, S. Rogers, and S. Schroedl. Constrained k-means clustering with background knowledge. In *Procs 18th ICML*, pages 577–584, 2001.