Evita: An Email Visualization and Tagging System using Machine Learning

1
Affiliation
Address

2
Affiliation
Address

3
Affiliation
Address

ABSTRACT

Evita (**E**mail **VI**sualizer and **TA**gger) is an email visualization and tagging application that uses a machine learning approach to alleviate the laborious and error-prone process of manually labelling emails. Evita's AdaBoost classification algorithm predicts the applicability of manually added email tags to new emails. Its Active Learning algorithm guides users by ranking the informativeness of untagged emails to the training algorithm if they were manually tagged. To help users in manually tagging emails, Evita's visualizer uses Landmark Multidimensional Scaling to display the email collection as seen by the machine learning component. Also, animation conveys the effects of tagging back to the users. The predictive aspect of Evita was favourably evaluated using standard machine learning protocols.

INTRODUCTION

One of the many problems of traditional file management systems is the need to slot files hierarchically, which requires manual classification of files into labelled folders. Such file systems create a number of issues. First, research has shown that the act of file classification is a major cognitive load, which users try to defer as long as possible [8]. Second, most information does not fall happily into neat categorization structures with simple labels, but over overlapping and fuzzy categories [11]. Third, it is almost impossible to generate category names that can be used unambiguously due to polysemy (i.e., more than one meaning for a single word). In fact, study has shown that people choose the same single word to describe a familiar object only 20% of the time [5].

Since most file management systems do not offer semantic querying of stored data, retrieval of information is predominately location-based. In order to facilitate the retrieval process, most users develop elaborate file structures to help organize their data. However, since these hierarchical file structures are static, they cannot be reconfigured to reflect changes in the data except through explicit and laborious input from the user. Also, a strict hierarchy requires file be uniquely classified.

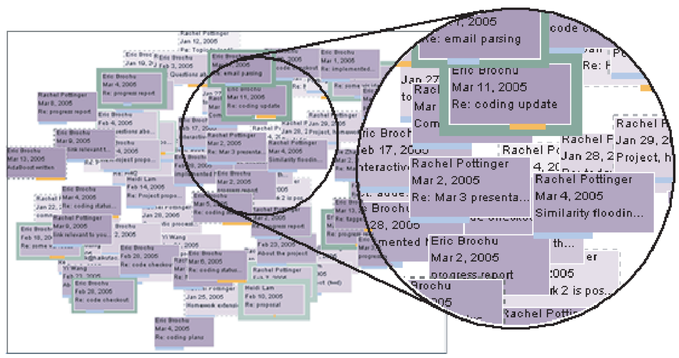

Figure 1. ***Screen capture of the main screen of Evita. Structure of the emails are displayed based on the Landmark MDS algorithm. Perceptual layering renders the time dimension of the email where more recent emails are perceived as being on a higher layer. The green box around the email icons codes the degree of desirability for the email to be tagged. The bi-color bar codes the applicability of the tag to the email as predicted by the system. Insert is a detailed view of a part of the screen.***

One possible approach in addressing these problems is to allow multiple descriptors, or tags, for each file. This permits semantic information search, and removes the need for a hierarchically-structured file system. However, there are two major costs to this approach: (1) the trade-off in information retrieval strategy where direct access (i.e., retrieval based on browsing by location) is replaced by a search-only approach (i.e., retrieval based on keyword input), and (2) the need to tag the information to allow for semantic querying.

The act of tagging, like organizing, is still a cognitive load. Also, there is no guarantee as to the consistent use of descriptors as tags. In view of this, researchers have suggested automatic machine tagging (e.g., [13]). While automatic tagging may at first seem to require the least amount of effort from users in managing their files, users can remember tags better if they assign them to the files themselves [10]. We therefore take a semi-automatic machine learning approach where users supply the tags and illustrate their use with existing emails. To assist users in the tagging process, we provide a visualization to help users work with the machine learning algorithm.

MACHINE LEARNING IN EVITA

There are two machine learning components in Evita:

1. Using the AdaBoost classification algorithm, Evita builds a model of feature/tag relationship to predict the applicability of individual tags to an email. This is done by analysing the features of previously tagged emails.
2. To minimize manual tagging, Evita uses an "active" learning algorithm that identifies the emails that would provide the most information if manually tagged.

AdaBoost

The problem of predicting the applicability of a tag to an email based on manually tagged examples is one of classification. Evita uses an AdaBoost classification algorithm [4]. AdaBoost is a type of Ensemble Learning algorithm that works by combining a set of "weak" classifiers, each performing only slightly better than chance, into a "strong" classifier that is guaranteed to have better performance. Evita's weak classifiers are simple "rule-of-thumb" classifiers of the form "if condition C is false, return value c_0. If it is true, return value c_1.".

In Evita, the condition C is the presence of a term, usually a word, in the emails. The exact values of $\{C, c_0, c_1\}$ for each tag is learned from email examples where the user has indicated if the tag applies. The training algorithm is based on Schapire's BoosTexter [15].

Active Learning

Evita uses an active learning algorithm to reduce manual labelling. It is "active" as it predicts the informativeness of a manully tagged email to the training algorithm. Our algorithm is based on Schapire's, which is confidence-based and tries to greedily select emails to minimize the uncertainty of the tag predictions [16]. The users can then maximize the performance of the classifier by labelling a few "key" emails, but Evita does not constrain the order of labelling. Once an email is manually labelled, Evita updates the training set, rerun AdaBoost to find a new set of weak classifiers and weights, and the active learning algorithm updates the suggestions based on the new information.

VISUALIZATION IN EVITA

There are two major goals for the visualization: (1) to display emails in context, and (2) to convey the effect of a manual tagging event.

Displaying emails in context

While users may find it difficult to label individual emails viewed in isolation, the tagging process may be better supported if the emails are viewed in context of the email collection. Also, since the AdaBoost algorithm selects key emails based on the information the tags can provide to the system, we would like to reflect such boundaries and clusters visually to help the users in their tagging process. However, direct visualization of these boundaries and clusters can be difficult due to the large number of features used in our approach to make up the prediction value. Instead, Evita displays the underlying structure of the user's emails as seen by the classifyer. Since each email is represented by a vector of high dimension in the classifyer, we need to project the email vectors onto displayable space. In Evita, we use a dimension reduction technique called Landmark Multidimensional Scaling (LMDS) [1].

Using LMDS, Evita can display the similarity relationship between emails in 2D space, as shown in Figure 1. Individual emails are represented as rectangular email icons, and their placement is calculated using LMDS. In order to facilitate tagging, the email icons show the author, subject and first few words of the body to help users recall the represented emails. To reduce visual cluttering of the display, Evita uses perceptual layering to create visual layers, where older emails are put on lower layers to minimize distraction [17]. In addition, the visualizer conveys a "desirability" value for each email, which is our Active Learning algorithm's predictions as to the informativeness of the emails (Section). This value is encoded as the thickness of the margins of the email icons with a salient color to draw user's attention to the more "desirable" emails.

Conveying the effects of tagging

The second component of the visualization is to convey the change in tag applicability on the entire collection after the user manually tag an email. Such feedback allows the user to experiment and learn to select appropriate tags, and to motivate labelling by demonstrating progress.

Each email has a single real-valued "hypothesis" score for each tag provided by the system, encoding both the prediction and the confidence of the prediction. If the value is positive, the tag applies; if negative, it does not. The magnitude of the hypothesis is the system's confidence in this prediction. Evita uses a bi-colour bar at the bottom of each email icon to encode this information, with orange for positives, and blue for negatives. The length of the bar corresponds to the confidence. Due to the saliency of these colours, users can quickly estimate the overall applicability of the tag in the email collection at a glance.

Once an email has been tagged, the system applies that information to produce a new set of hypothesis values for the email collection. To direct the user's attention to the largest changes, Evita blinks the icons with the largest changed values, and animates the change in the confidence bar.

EVALUATION

For the purpose of evaluation, we collected a set of 78 emails communicated over a four-month period. These emails can be classified into categories of "evita project", "school administration", "database topics", and "project idea spinoffs". We tested Evita using a tag called "project".

AdaBoost classification

We evaluated Evita's AdaBoost algorithm using *leave-one-out cross-validation* on a training set. Leave-one-out cross-validation works by training the classifier as many times as there are labelled data. Each time, one datum is withheld,

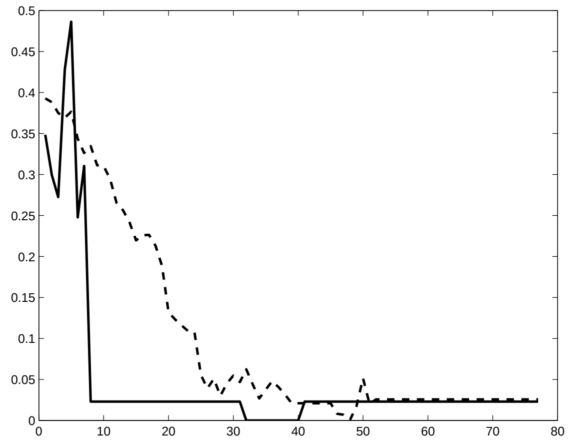

Figure 2. ***The results of experiments in evolving the AdaBoost classifier. The solid line is the result of using active learning to select the data, the dotted shows random selection. Using Active Learning, a good solution can be found with only a few emails.***

and training is done using the remaining data. The held-out datum is submitted to the system as a test datum to obtain a label prediction. If the prediction matches the label, the datum has been successfully classified. In this way, we can test on a single data set, while avoiding the problem of overfitting.

We found that only 2 of the 78 test data were misclassified. This is remarkably good result as the number of training data was quite small, as was the amount of text in the emails.

Active Learning performance

In this test, we evaluated the performance evolution of the classifier with added training data following the Active Learning evaluation model of [18]. Using our set of 78 emails with known labels, we repeatedly train AdaBoost, starting with a single positive and a single negative email, and adding a single email at instance. At each step, the emails that were not used in the training are submitted to the classifier, which predicts the labels and computes the error.

We ran the experiment twice. First, the emails were randomly added, and second, only the initial emails were randomly selected, and the Active Learning algorithm was run to find the most "desirable" email to be added to the classifier.

The results are shown in Figure 2, with the random selection as a dotted line and Active Learning as a solid line. Each plot is the average error over five trials. The random selection was slower, taking 40 emails on average to achieve the same performance level as Active Learning after 8 emails. The high initial ßpikesänd the subsequent smoothness for the Active Learning curve is due to the same emails being consistently selected in the same order by the learning algorithm for most of the trials.

RELATED WORK

A number of personal information management research projects adopt a semantic tagging approach. PARC's Placeless Document allows user-level attributes that are either added by users, or extracted by software services based on file content [2]. MyLifeBits supports multimedia files and rich annotations to allow for semantic queries [6]. MIT's Haystack allows users to select predefined or new categories to describe each document [12]. All these approaches requires users to explicitly label documents, which is known to be a difficult task. To avoid manual tagging, a number of systems have taken the automated approach. Stuff I've Seen indexes all types of information to allow for keyword-based queries [3]. While unsupervise approaches avoided the laborious manual tagging, it is unclear if the indexing/classifications by the system are appropriate and memorable to the users.

In the domain of emails, Kushmerick proposed an automated structure induction technqiue to automatically classify emails based on the user's activity [7]. Mock proposed a nearest-neigbour classifer based on existing content of user-created folders [9]. Similarly, SwiftFile adapts to user's dynamic email filing habit with an incremental learning algorithm based on a TF-IDF classifier [14]. Despite automatically slotting emails into folders, these systems still organize emails hierarchically, and emails are related by location only.

FUTURE WORK AND CONCLUSION

In working with Evita, we gained insights into both fields of Machine Learning and Information Visualization, and how they can interact to enhance the user experience.

Our concern had been that emails were too brief, and had too few features to create a valid content-based model. On the contrary, there are a few attributes that makes email particularly good candidate for content-based classifiers like AdaBoost: (1) the content is usually designed to be clear and concise with fairly little noise; (2) email replies regularly quote the entire body of the original text allows accumulation of content. Indeed, we can take further advantage of the unique features of emails such as threads in to improve Evita's performance.

Evita's visualizer acts as the interface between the user and its machine learning component. This role is different from that of most visualizations where the purpose is to help user manage email by exploring and examining the entire collection. Instead, Evita directs the user's attention to the relevant results provided by its machine learning component, and avoids the need for active exploration of all the data. Further, by conveying the effect of their tagging actions back to the user in an interesting and information manner, Evita may encourage the users to spend more time training the system, and thus improving the performance of the machine learning component and the informativeness of the visualizer.

REFERENCES

1. de Silva, V. and Tenenbaum, J.B. Global versus local methods in nonlinear dimensionality reduction.

Advances in Neural Information Processing Systems 15, p721-728, 2003.

2. Dourish, P., Edwards, W.K., LaMarca, A., Lamping, J., Petersen, K., Salisbury, M., and Terry, D.B. Extending document management systems with user-specific active properties, *ACM Transactions on Information Systems (TOIS) 18*(2), p.140-170, 2000.
3. Dumais, S., Cutrell, E., Cadiz, J. J., Jancke, G., Sarin, R., and Robbins, D. C. Stuff I've seen: a system for personal information retrieval and re-use, In *Proc. 26th International ACM SIGIR Conference on Research and Development in Information Retrieval*, p72-79, 2003.
4. Freund, Y. and Schapire, R.E. Experiments with a New Boosting Algorithm. In *Proc ICML*, p148-156, 1996.
5. Furnas, G.W., Landauer, T.K., Gomez, L.M., and Dumais, S.T. Statistical semantics: analysis of the potential performance of key-word information systems. *Bell Systems Technical Journal 62*(6), p1753-1806, 1983.
6. Gemmell, J., Bell, G., Lueder, R., Drucker, S., and Wong, C. MyLifeBits: Filfilling the Memex vision. In *Proc. of ACM Multimedia'02*, p.235-238, 2002.
7. Kushmeric, N. and Lau, T. Automated Email Activity Management: An Unsupervised Learning Approach. To appear in *IUT'05*, 2005.
8. Malone, T. How do people organize their desks? implications for the design of office information systems. *ACM Trans. Office. Info. Syst. 1*, p99-112, 1983.
9. Mock, K. An experimental framework for email categorization and management. In *Proc. International Conference of Research and Development in Information Retrieval*, p. 392-393, 2001.
10. Lansdale, M. On the memorability of icons in an information retrieval task. *Behaviour and Information Technology 7*, p131-151, 1988.
11. Lansdale, M. The psychology of personal information management, *Applied Ergonomics 19*, p55-66, 1988.
12. Quan, D., Bakshi, K., Huynh, D., and Karger, D. R. User interfaces for supporting multiple categorization. In *Proc. Interact 2003*, p228-235, 2003.
13. Ravasio, P., Schar, S. G., Krueger, H. In pursuit of desktop evolution: user problems and practices with modern desktop systems. *TOCHI 11* (2), 156-180, 2004.
14. Segal, R. and J. Kephart. Incremental learning in SwiftFile. In *Proc. International Conference on Machine Learning*, p863-870, 2000.
15. Schapire, R.E. BoosTexter: A Boosting-Based System for Text Categorization. *Machine Learning 39* (2), p135-168, 2000.
16. Schapire, R.E. Advances in Boosting. In *Proc 18th Conference on Uncertainty in Artificial Intelligence*, p446-452, 2002.
17. Van Laar, D. L. Psychological and cartographic principles for the production of visual layering effects in computer displays. *Displays 22*, 125-135, 2001.
18. Zhu, L. and Ghahramani, Z. Combining Active Learning and Semi-Supervised Learning using Gaussian Fields and Harmonic Functions. In *Proc of the 2003 ICML Workshop on the Continuum from Labeled to Unlabeled Data in Machine Learning*, 2003.