PSUN: A Profiling System for Usenet News

(Extended Abstract)

H. Sorensen, M. Mc Elligott,
Computer Science Department,
University College,
Cork,
Ireland.
Email: {sorensen, mike@odyssey.ucc.ie}

Abstract

Information overload has become a well-worn term. Anyone who has travelled the highways and byways of the Internet, who has received e-mail or browsed the Usenet News hierarchy is all too familiar with the term. In the following paper we outline the design of the PSUN model to filter Usenet News articles from the Internet. PSUN is an adaptive filtering model which draws on techniques from Information Retrieval, Cognitive Science, Connectionism and Artificial Life.

1 Introduction

The concept of electronic information filtering has come to the fore in recent years, primarily due to the the influence of the Internet. To address this issue, we have witnessed intensive research in the form of both commercially and freely available products promising to stem the rising tide of 'information overload' on the Internet (e.g. Newt [Sheth93], NewsWeeder [Lang94], NRT [Sanderson94], [Menczer95] and, more noticeably, SIFT [Yan94] and Harvest [Bowman95] to name but a few). The complexity of models range from straightforward rule-based approaches (e.g. MAFIA [Lutz90], Lurker [Taluskie94]) to more complex Artificial Life based approaches (e.g. Newt [Sheth93], [Menczer95]). Many problem areas have been identified and targeted, such as filtering e-mail, locating relevant research reports and information sites and filtering Usenet News. It is this latter category, that of filtering Usenet News articles, where we concentrated our attention culminating in the design of the PSUN (Profiling System for Usenet News) information filter.

As with many of the other research attempts ([Sheth93] in particular), we have modelled our filter as an interface agent which acts as a broker on behalf of a user, embodying his/her interests and goals, accepting feedback from the user on the relevance of filtered documents and adjusting the user goals based on this relevance feedback. Also, the user's interests are captured via analysis of an initially presented set of interesting documents, with no explicit specification of keywords or search terms being necessary. We differ, however, from others in that our representation method is based on a more semantically rich analysis of documents previously deemed interesting by the user.

In this paper, we shall outline the theory behind the PSUN model, we shall examine the architecture and operation of the system and shall evaluate the accuracy of the system as compared with other filtering strategies.

2 Polsemy and Word Frequency Models

Due to the similarity of information filtering to the well established field of information retrieval (see [Belkin92]), many researchers working on news article filtering systems have looked to this field for techniques to classify and categorise unrestricted textual documents. One popularly adopted approach has been the vector space model [Salton83]. Examples where this has been employed include [Sheth93], [Lang94], [Sanderson94] and [Yan94]. Although a widely respected technique, we believe there is room for improvement in its statistical analysis of text. No attention is paid to word ordering or word context in the vector, which results in its inability to deal with *polsemy* [Church91]. Suppose, for example, that a representative term vector is constructed for an article about natural language processing. To find other articles about natural language processing we construct term vectors for them and compare them to the original. All those vectors that have a sufficient similarity are deemed to represent relevant articles and are presented to the user. The problem lies in the fact that in such a vector, it is quite probable that a term such as "natural" would be highly weighted (indicating its importance as a content identifier). So articles containing this term are at a distinct advantage as regards being selected as relevant. This may seem fine until we realise that such articles could be referencing "natural yoghurt", "natural colour", or whatever instead of the desired "natural language processing".

In the design of the PSUN model we attempted to address this shortcoming of the vector space and many other word frequency models to give a more precise technique for statistically modelling document contents. We set out to construct a model which alleviated the polsemy problem by working with weighted *n-grams* (n words found to occur after each other a significantly high number of times) as opposed to single words. We thus attempted to capture the fact that, not only did certain words recur in documents of interest to a user, but that they appeared in the same context as they had in documents previously deemed interesting by the user.

3 Profile Construction

A user profile in PSUN is constructed from an initially presented set of articles, and captures the recurring words and phrases within these documents (allowing for some flexibility within the phrase structures). Each such term is structurally represented in the profile as an n-gram. To construct the n-grams, we approached the problem from a number of theoretical viewpoints**:**

3.1 Schank and Scripts

We treat the problem of recording relevant phrases as one of memorization. To solve this, we adopt Schank's theory of script formation [Schank77, Schank82]. Each word is viewed as an event and our goal is to organise these events into useful structures in the same spirit as the way that Schank proposed our own memories are organised. We thus use a connectionist model to represent 2-word sequences that occur in the user-presented documents (we call these structures *simple entities* or *S-entities*). This in itself is of little use, as it does not indicate whether a given sequence contributes to the relevance of the article. In order to determine this, we looked to the field of neural computing in search of a learning mechanism.

3.2 Hebbian Learning

Hebb's theory of neural learning is based on the fact that neurons which frequently fire together establish an inter-relationship which makes such firing even more likely in the future. This is a theory something akin (in spirit at least) to the theory of scripts which was later put forward by Schank and Abelson. In both instances we have the case where 'bonds' form between events/neurons that occur/activate together so as to structure our mind into expecting them to occur together again.

We now have the basis of a method whereby we can permit a collection of entities to organise themselves autonomously into varying degrees of attraction to each other based on external stimuli presented to them [Mc Elligott94]. In other words, we now have a system where we can enable our S-entities to organise themselves into phrase-like structures of varying degrees of certainty, using a weighting scheme.

We actually employ two separate weighting schemes for S-entities. A *weight* is used to estimate the likelohood that one term will appear after another (i.e. for some word A, it measures the likelihood that other words C, D, etc will appear with it in interesting articles). The second measure used is a *strength* and is used to measure the relative importance of one S-entity as compared with all others appearing in the profile. The weight and strength of each S-entity in a profile are adjusted using unconstrained and constrained Hebbian Learning, respectively. This adjustment occurs during the initial profile construction phase and as a result of ongoing relevance feedback.

Two problems now arise. One is to restrict our interest to those term sequences that are good enough to become useful scripts and the second is how to introduce a mechanism whereby we can form scripts of longer length than two. The solution to both these problems was inspired by a theory put forward by Minsky known as *K-lines*.

3.3 Minds and Minsky

In "The Society Of Mind" [Minsky87], Minsky proposed that the mind worked as a vast society of interacting agents which co-operated and competed with each other to produce intelligent behaviour. To co-ordinate these agents, he put forward a scheme not altogether different from the way events are organised by Schank's scripts. He proposed a theory of K-lines to govern memory and learning. This mechanism to allow the storage and recall of 'good ideas' inspired a similar technique that we ourselves have employed to store and recall 'good scripts'. When a sequence is formed that satisfies both the local and global script criteria outlined in the previous section, we create what we will term a *supervisor*. These supervisors, between them, make up the article's representation. They represent a collection of phrases taken from the article which have been judged to be of potential use in the future categorisation of articles - and, hence, will migrate into the user's profile. These supervisors could be said to work in a reverse manner to K-lines. Instead of activating a 'mental state' similar to the one we were in when we got the good idea, it is now the re-occurrence of this previous 'mental state' (or, more accurately, the previous good sequence of words) which activates the supervisor.

Each supervisor, S_i, is capable of making a contribution C_i to the rating of incoming articles. The total contribution by all the supervisors is limited to a set maximum.. The contribution that each supervisor may make depends on four variables: the *strength* of the sequence it was constructed for, the *weight* of the sequence and finally, the frequencies of occurrence of the two S-entities (based on the work of [Luhn58]).

To construct longer scripts, we proceed in a process of *supervisor chaining*. A second pass is made of the article to determine if any of the supervisors occurred in sequence. If they did, a link is recorded in the originating supervisor in much the same way as links are recorded between S-entities. For example, if there was a supervisor S1 formed for "object oriented" and another, S2, formed for "oriented programming" and, on doing a second pass, it was noticed that the term "object oriented programming" appeared in the article, then a link is formed between the two supervisors going from S1 to S2. This link is allocated a value based on a percentage of the parent's contribution value. We now have what could be termed a *virtual supervisor* complete with its own ability to contribute to the rating of an article. By taking some of the parent's contribution value and giving it to a link we are essentially imposing a higher level of specificity. If this or other links occur sufficiently, the power of the parent supervisor to contribute on its own to the rating of an article may be totally diminished with all it's initially allocated contribution value having been distributed among links to other supervisors.

A similar technique is used to monitor for the formation of longer scripts. In theory, using this method a script could grow to be any length although early results indicate it to be rare for a script with more than four terms/events to form. A script formed from *n* supervisors ($S_1......S_n$) draws rating potential from the first *(n - 1)* supervisors involved in the script. For example, for a script "in natural language processing", this would draw a standard percentage rating potential from each of the supervisors "in natural" and "natural language". So we can see a bias being introduced in favour of the contributions that longer scripts can make.

4 Article Rating

When incoming articles are received from the newsfeed, after some textual preprocessing (stemming, etc), the supervisors act as autonomous agents which search for occurrences of relevant phrases. If found, then the supervisors contribute an amount to the overall rating of the article proportional to their own previously computed contribution value. This is a relatively straightforward task, and the final ratings are used for ranking the received articles before presentation to the user.

5 Relevance Feedback

Relevance feedback has been shown to play a beneficial role in IR systems (e.g. the SMART system [Salton83]) and it is a technique that we take full advantage of in the PSUN system. A user may classify a received document by awarding it differing degrees of relevance. This feedback may result in the weight and strength values of S-entities being increased/decreased and, consequently, in the contribution of different supervisors being adjusted. In this way, each rated article is integrated into the profile in a manner which reflects its rating.

6 Survival of the Fittest

While our goal with PSUN was to maximise the figures of recall and precision, it was to do so in the context of each user and not from the standpoint of seeking global averages. An automated technique was required for profile/user matching and was found in the shape of a Genetic Algorithm [Holland75]. PSUN is 'controlled' by the settings of 13 different parameters. These parameters control everything from window size [Church91] to the rate of strength allocation among links between S-entities. Altering the values of these parameters can effect changes in the *specificity* of the profile construction process. For example, altering the window size from the default of 1 to n results in not only links between two successively activated S-entities being recorded but also between all S-entities being activated within n activations of each other. This can result in a higher number of links, a higher number of scripts and thus a higher number of supervisors in a user's profile. The more supervisors there are, the more articles will get a rating.

With PSUN, we initially begin with a family of profiles for each distinct user interest. We produce new generations of profiles using crossover and mutation strategies employing information derived from user feedback, our goal being to design a new profile which matches the user's ratings over the course of this feedback base as closely as possible.

In doing so, we hope to have adapted a profile to the *individual* precision and recall desires that a user has exhibited for a specific profile. Our problem then, is one of searching the space of possible profile configurations to find the must suitable in a particular instance.

In our design of a genetic algorithm to maximise two functions simultaneously, we used a technique known as *disassortive mating* [Ryan94] in which two populations of individuals (in our case sets of parameter values) each attempt to maximise one of the functions. One population attempts to find that combination of parameter values which produces the biggest number of supervisors (i.e. attempts to maximise recall) while the other population attempts to find the combination that produces the closest ratings to those given by the user (i.e. attempts to maximise precision). We then proceed by trying to find that individual that most satisfies both populations and, in doing so, most satisfies the recall and precision requirements of the user.

7 Results

We have compared the results of PSUN with those of a conventional vector-space model. These results have been favourable as regards this new model. We have assessed the strengths and weaknesses of the PSUN profiling model both from a precision/recall viewpoint and from the perspective of users of the system. We do believe that, on the whole, PSUN embodies a new and important approach to that of information filtering, and that the approach of using an autonomous user agent is fully justified.

References

[Belkin92] Nicholas J. Belkin and W. Bruce Croft. *Information Filtering and Information Retrieval: Two Sides of the Same Coin ?* Communications of the ACM, December 1992, Vol. 35, No. 12.

[Bowman95] C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, Michael F. Schwartz and Duane P. Wessels. *Harvest: A Scalable, Customizable Discovery and Access System.* Technical Report CUCS-732-94. Department of Computer Science, Boulder Colorado. March 12, 1995.

[Church91] K. Church, W. Gale, P. Hanks and D. Hindle. *Lexical Acquisition (Using Statistics in Lexical Analysis)*. Lawrence Erlbaum, 1991.

[Holland75] John H. Holland. *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: University of Michigan Press 1975.

[Lang94] Ken Lang. *NewsWeeder: An Adaptive Multi-User Text Filter.* Research Summary. August 1994. Available at http://anther.learning.cs.cmu.edu/ifhome.html

[Luhn58] H. P. Luhn. *The Automatic Creation of Literature Abstracts*. IBM Journal of Research and Development. Volume 2, Part 2, 1958.

[Lutz90] Ernst Lutz, Hans V. Kliest-Retzow and Karl Hoernig. *MAFIA - An Active Mail-Filter-Agent For an Intelligent Document Processing*

Support. Multi-user Interfaces and Applications. S. Gibbs and A. A. Verrijn-Stuart (Editors), Elsevier Science Publishers B. V. (North Holland), 1990.

[Mc Elligott94] Michael Mc Elligott and Humphrey Sorensen. *An Evolutionary Connectionist Approach To Personal Information Filtering.* Proceedings of the Irish Neural Networks Conference. University College Dublin, September 12-13, 1994.

[Menczer95] Filippo Menczer and Richard K. Belew. *Artificial Life Applied to Adaptive Information Agents.* 1995 Spring Symposium on Information Gathering From Heterogeneous, Distributed Databases.

[Minsky87] Marvin Minsky. *The Society of Mind.* Simon & Schuster, New York, 1987.

[Ryan94] Conor Ryan. *Pygmies and Civil Servants.* Advances in Genetic Programming. Kenneth E. Kinnear (ed). MIT Press. ISBN 0-26211188-8.

[Salton83] Gerard Salton and Michael J. Mc Gill. *Introduction to Modern Information Retrieval*. Mc Graw Hill International, Auckland, 1983. ISBN 0070544840.

[Sanderson94] Mark Sanderson and Keith van Rijsbergen. *NRT (News Retrieval Tool)*. Electronic Publishing, Origination and Dissemination, Volume 18, 1994.

[Schank77] Roger Schank and Harold Abelson. *Scripts, Plans, Goals and Understanding*. Lawrence Erlbaum, 1977.

[Schank82] Roger Schank. *Dynamic Memory: A Theory of Reminding and Learning in Computers and People.* Cambridge U. P. London, 1982.

[Sheth93] Beerud Sheth and Pattie Maes. *Evolving Agents For Personalized Information Filtering.* Proceedings of the Ninth IEEE Conference on Artificial Intelligence for Applications, 1993.

[Stevens92] Curt Stevens. *Knowledge-Based Assistance for Accessing Large, Poorly Structured Information Spaces.* PhD thesis, University of Colorado, Boulder, 1992.

[Taluskie94] Vince Taluskie. *Lurker.*
Available at ftp://ftp.ph.utexas.edu/pub/perl/lurker-1.1.tar

[Yan94] Tak W. Yan and Hector Garcia-Molina. *SIFT - A Tool for Wide Area Information Dissemination.*
Available at http://sift.stanford.edu/