Using Context to Get Novel Recommendation in Internet Message Streams*

Doina Alexandra Dumitrescu
Escuela Politécnica Superior, Universidad Autónoma de Madrid
Francisco Tomás y Valiente, 11
28049 Madrid, Spain
doina.dumitrescu@uam.es

Simone Santini
Escuela Politécnica Superior, Universidad Autónoma de Madrid
Francisco Tomás y Valiente, 11
28049 Madrid, Spain
simone.santini@uam.es

ABSTRACT

Novelty detection algorithms usually employ similarity measures with the previous seen and relevant documents to decide if a document is of user's interest. The problem that arises by using this approach is that the system might recommend redundant documents. Thus, it has become extremely important to be able to distinguish between "redundant" and "novel" information. To address this limitation, we apply a contextual and semantic approach by building the user profile using self-organizing maps that have the advantage to easily follow the changes in the users interests.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Algorithms

Keywords

novelty detection, self-organizing maps, semantics, content-based recommender systems

1. INTRODUCTION

The sheer amount of information available on the Internet seems to prove the old saying that there can be too much of a good thing. Being informed today seems to be no longer a matter of finding a good source - that is quite easy nowadays - as of filtering out irrelevant and duplicate data. News services, specialized sources, social networks are flooding us with an increasingly unruly stream of information that we find almost impossible to control. Some of these elements (e.g. those coming from news services) come equipped with categories, but such a fixed and brittle solution is hardly optimal. A person might, for example, have a generic interest in certain medical issues. Most newspapers publish medical news under the general heading "science", a very general category that might include irrelevant items such as the discovery of a supernova. On the other hand, political news such as the signing of the new NIH budget for stem cell research may be of interest; sport news might be interesting if they contain some information about steroids, as can be the message about his work posted by a physician friend.

In order to identify potentially relevant information, we propose the creation of an incremental user information profile using machine learning techniques to improve filtering and recommendation of online information. The model uses a self-organizing map to provide a representation of what a person finds interesting.

Relying simply on the relevance of an item for the user might create some problems in news-rich environments. For each retrieved item, there are a plethora of near identical items that provide no new information but that, being just as relevant as the first one, will also be retrieved. Information retrieval has dealt with this problem with the introduction of the concept of *novelty* [2]. In standard information retrieval, an element of the result set is *novel* if it contains information that other elements of the result set do not contain [1]. This notion is not directly applicable to filtering, since we are dealing here with a continuous stream and not with a finite result set.

In our context, a news item is *novel* if it covers a portion of the interest field of the user that no other items have covered recently. Several research efforts have been focused on using learning techniques such as self-organizing maps for recommender systems [6].

Unlike earlier approaches that use self-organizing maps, most of which do not consider the order of the words, in this work, we use sequences of words (either pairs, sentences, or paragraphs) as input features to the learning algorithm. This represents an important advantage for our problem as the semantics that derives from the co-location of the words will not be lost and will be used for word disambiguation.

To the best of our knowledge, there is no prior work that uses a self-organizing map approach to novelty detection for filtering and recommendation. A major contribution of this paper is also the consideration of the temporal aspect of the novelty, that is, after certain time, an item similar to something already seen is becoming once again interesting.

*Supportd in part by the *Ministerio de Educación y Ciencia* under the grant N. TIN2013-47090-C3-2, *VoxPopuli, Efficient reputation analysis, propagation and recommendation in social network environments.*

WWW 2015 Companion, May 18–22, 2015, Florence, Italy.
ACM 978-1-4503-3473-0/15/05.
http://dx.doi.org/10.1145/2740908.2742473.

There are various studies that investigate novelty detection based on other techniques. Most of these approaches use similarity measures such as word count or cosine metric [5]. A document is considered "novel" if its maximum similarity to all the previously seen documents is below a certain threshold.

We validate our approach by conducting various experiments on the Reuters Collection [4].

2. CONTEXT CONSTRUCTION

The basis of the model is a set of documents, which can be composed of the documents on which the user is working at the time and/or by news items marked in the past. In the following, in order to make the tests self-contained, we shall always assume that the context is based on a set of news items that supposedly the user has seen in the past and that have been found to be relevant.

Many representations of documents start from the frequencies of words of the document; this representation is insufficient for our problem because if we use only a word by itself, the semantics that derives from the co-location of the words will be lost, and we need co-location for word disambiguation. Rather, in the technique that we will use, the fundamental unit of representation that is extracted from the document is not the word, but a group of words, that is a *sentence*. A sentence is a n-gram defined as a sequence of n consecutive terms. From now on we will interchangeably use the terms n-gram and sentence. These n-grams are represented in the typical geometric space of the standard vector model representation, a space in which each word is an axis. Since our basis are the n-grams, the points from which the representation is derived are not points in one of the axes (as in the case of simple words: each point is a word with its weight), but points in n-dimensional sub-spaces. For example, for a n-gram, its representation in the word space is given by:

$$g_k = (w_{t_u}, w_{t_v}, \dots w_{t_l}) \tag{1}$$

where w_{t_i} are normalized weights that corresponds to each of the n consecutive terms contained by the n-gram. Each document is now represented as a bag of *n-grams*. Let $\{t_1, \dots, t_W\}$ be the set of all terms that appear in all documents, and let g_{k_i} be a n-gram formed by n consecutive words as they appear in the document D_k. Let $|g_k|_i$ be the number of times such n-gram appears in the document D_i. The n-grams are represented as points in a vector space whose axes are the words $t_1, \dots, t_W$. In this space, the n-gram g_k with weights $(\omega_u, \omega_v, \dots, \omega_l)$ of the compounding terms, is represented by the point p_{g_k}, which lies in the n-dimensional sub-space determined by the axes $\{t_u, t_v, \dots, t_l\}$

Each weight is a measure of the importance of the term in that document. Generally, the *tf-idf* measure is used [7] and we use an adaptation for n-grams based on it as we shall see shortly:

$$\tilde{\omega}_{(g_{k_i})} = pf_{(g_{k_i})} \cdot \mathrm{if}_{(g_{k_i})} \tag{2}$$

The *raw word frequency* of a n-gram (g_{k_i}) in the context R is defined as the sum of all the occurrences of the n-grams in the various documents weighted by the weight associated to the documents:

$$\mathrm{pf}_{(g_{k_i})} = \sum_i w_i |g_{k_i}|_i \tag{3}$$

In information retrieval, this weight is normalized multiplying it by a monotonically increasing function of the inverse of the fraction of the corpus documents in which the term appears. In our case, we can't use the same normalization factor since we do not have a corpus of documents that serves as a reference. What we do have is the relative frequency, e_u, with which each word, t_u, appears in the general corpus of English writing. We will approximate the n-gram frequency as the product of the frequency of the individual words, therefore we define the inverse frequency of the n-gram (g_{k_i}) as:

$$\mathrm{if}_{(g_{k_i})} = \log \frac{1}{e_u e_v \dots e_l} = -(\log e_u + \log e_v \dots \log e_l) \tag{4}$$

With this frequency, we define the *raw weight* of the n-gram (g_{k_i}) as its *pf-if* weight:

$$\begin{aligned} \tilde{\omega}_{(g_{k_i})} &= \frac{1}{C} \mathrm{pf}_{(g_{k_i})} \, \mathrm{if}_{(g_{k_i})} \\ &= -\frac{1}{C} (\log e_u + \log e_v \dots \log e_l) \sum_i w_i |g_k|_i \end{aligned} \tag{5}$$

where $C = \max\{\tilde{\omega}_{(u)}\}$ is a normalization term used to fit all weights in the unit cube of the word space.

While these weights are quite adequate for some contexts, other contexts are marred by the presence of outliers: very relevant n-grams with high weights that "push" all the other weights close to zero, reducing considerably the representativity of the point cloud. We can't simply eliminate the outliers, because they are, after all, the most important terms in the context, but we can reduce their predominance by *balancing* the weights through a suitable non-linear transformation. In our model, we choose simply a power function with a suitable exponent $0 < \alpha < 1$, obtaining the *balanced* weights:

$$\omega_{(u)} = (\tilde{\omega}_{(u)})^\alpha = -\frac{1}{C^\alpha} \left[(\log e_u + \log e_v \dots \log e_l) \sum_i w_i |u|_i \right]^\alpha . \tag{6}$$

In the tests that we report in this paper, we used the value $\alpha = 0.7$.

At the end of this step, the context $\mathfrak{C}$, represented by the set of points $I_{\mathfrak{C}}$ in this space, is used as the training data for the construction of the self-organizing map using a modification of WEBSOM [3] The training procedure will adapt the map so that it reflects the semantic relationships between the vectors of the input space. As SOM feature, instead of the word weights we use the n-grams as mention before. Moreover, the SOM will be represented by a two dimensional grid of nodes, arranged in a rectangular form, where each node, m_i, is represented by a vector of the same dimension as the input space. Before starting the learning process, the map is initialized with random values. At each training step, a random input vector is compared with all the map nodes. The similarities of the vectors are reflected in their Euclidean distance on the map. The winning node or the best-matching unit (BMU) is the closest node to the input vector:

$$BMU = \operatorname*{argmin}_{m_i} d(x, m_i) = \operatorname*{argmin}_{m_i} \left[\sum_{i=1}^{W} (x - m_i)^2 \right]^{\frac{1}{2}} \tag{7}$$

The map has a 4 neighbourhood topology, therefore, for the neuron m_i, the BMU will be moved together with its topo-

logical neighbours in the direction of the input vector according to the adaptation rule:

$$m_i(t+1) = m_i(t) + \zeta(t) \cdot h(t,d) \cdot [x(t) - m_i(t)]$$

This rule depends on the parameters $\zeta(t)$ (the learning factor at time t) and $h(t,d)$ (the neighbourhood function that delimits the amount of the winning neuron and neighbourhood that will adapt, which in our experiment is a distance-dependent Gaussian which shrinks in time).

During the learning, $\zeta(t)$ and $h(t,d)$ will decrease monotonically in time till converge to zero. Therefore, the amount of adaptation of the weights of the nodes from the neighborhood will be lower as the algorithm comes to a solution. As a final step, the weights of all the neurons will be normalized. Once the map has been trained and the n-grams organized according to theirs similarities, we will use this representation as our user profile in the filtering algorithm as it will be described in the next section.

3. NOVELTY DETECTION

3.1 The algorithms

As in any recommender system, the creation of the user profile is an essential step. Using the semantic representation of the user profile described in the previous section, three algorithms are proposed:

Filtering Algorithm:
The user profile is compared with the incoming stream of data. A document is considered relevant if its similarity from the map is above a given threshold. To this end, a similarity value to each neuron of the map is computed:

$$\text{sim}(D, m_j) = \frac{\sum d_i \cdot m_{ji}}{(\sum d_i^2)^{1/2} \cdot (\sum m_{ji}^2)^{1/2}}$$

The neuron that has the maximum similarity value is the BMU and this similarity is considered to be the measure of relevance of that document to the user profile.

Based on the similarity values to the user profile, only the documents that are beyond a certain threshold are shown to the user. In a next step, we select the top-500 documents that represent our *Filtering Results List.*

The Novelty Algorithm:
The filtering results list is used as an input for the novelty algorithm. This is an ordered list that contains the top 500 most similar documents to the context at a certain moment in time, D_k, where $k = 1, 500$. We will compute the *novelty similarity* measure between all the neurons and the document, as defined below:

$$\text{sim}(D_k, m_j) = \lambda_{m_j} \cdot \frac{\sum d_{ki} \cdot m_{ji}}{(\sum d_{ki}^2)^{1/2} \cdot (\sum m_{ji}^2)^{1/2}}$$

where λ_{m_j}, $(0 < \lambda_{m_j} \leq 1)$, is the *interest factor* that determine the novelty that the neuron m_j is given to respect to the others neurons. Initially, all the neurons will have an interest factor equal to 1 and with each selection as BMU of a neuron, the neuron novelty factor will decrease:

$$\lambda_{m_j} = {}^{\lambda_{m_j}}/_{100} \tag{8}$$

After certain time, a news very similar to something already seen becomes novel again. The maximum novelty similarity value will determine the *novel BMU* (NBMU) and define the *novelty score*:

$$\text{novelty}_{\text{score}}(D, \xi) = \max_{j=1,n}(\text{sim}(D, \text{NBMU}_j))$$

For the NBMU detected, the interest factor will be decrease, so that for future selection can represent the "true" novelty value. This step is repeated for all the following documents from the filtering results list and as a result is obtained a list that will be sorted based on the novelty similarity values and presented to the user. This list is called the *Novelty List.*

The Extended Novelty Algorithm
is similar to the Novelty Algorithm, the main difference is that the decreasing of the interest factor of the BMU will be applied not only to the winning neuron but also to its neighborhood. Based on this, the *Extended Novelty List* is obtained.

3.2 Novelty evaluation metrics

In this paper, we proposed a novelty measure that considers the *context coverage* of the set of recommended documents:

$$\text{coverage} = \frac{n_{\text{NBMU}}}{n_D} \tag{9}$$

where n_{NBMU} is the number of different neurons that win when that document was chosen to be recommended, and, n_D represents the total number of documents recommended.

4. EXPERIMENTS

4.1 Test collection

To evaluate the performance of the model described in Section 2, we conducted experiments using the Reuters Corpus Volume 1 [4]. This is a large collection of 806,791 news stories in NewsML format created by Reuters journalists during a period of a year. For each document, category codes for topic, region and industry sector are identified and assigned as corresponding meta data. The topics codes and the news stories are used as a basis for our experiments. The collection contains a total of 126 topics, distributed in four top level hierarchies: **CCAT** (*Corporate/Industrial*), **ECAT** (*Economics*), **GCAT** (*Government/Social*), and **MCAT** (*Markets*). Each main category includes sub-topics organized as a tree and for each news story zero or more topics are assigned. Only 103 topics are used, as the remaining 23 are not associated to any news document.

4.2 Context creation methodology

We generated four contexts, where each corresponds to a collection main topic: **MCAT**, **GCAT**, **CCAT** and **ECAT**. We do not present results for **ECAT** and **GCAT**: the former represents only a small fraction of the data set, and the latter gives virtually the same results as **MCAT**. For each main topic we have randomly chosen two sub-topics and used 5% of arbitrary news documents of each one.

Table 1 shows the context confusion matrices. Each row of the table 1 represents a context topic, while each column represents the percentage of common words with that context topic. This demonstrates the difficulty of the problem: it is very hard to separate the topics based on word distribution only.

Table 1: The context confusion matrix: The common words percentage

	MCAT	CCAT
MCAT	100	57.6
CCAT	47.7	100

4.3 Experimental novelty detection results

The experiments are conducted using Reuters Corpus Volume 1. For each context generated, we created the *Filtering*, *Novelty* and *Extended Novelty* results lists as described in Section 3. The first list is considered our baseline, meanwhile the last two were compared to it. For each list, we perform the evaluation by computing the mean average precision and the mean novelty score at a depth of 500 (see Figures 1 and 2). For each context, the relevant documents are the ones that belong to the corresponding context topics. From the results, we observed that by using the *Novelty* and *Extended Novelty* algorithms, there was a significant improvement in terms of novelty, nevertheless, precision does not have a significant variation. This suggest that the cosine measure is an effective tool to get relevant information to the user context, but not sufficient to get novel information also (for that, we use the novelty algorithm).

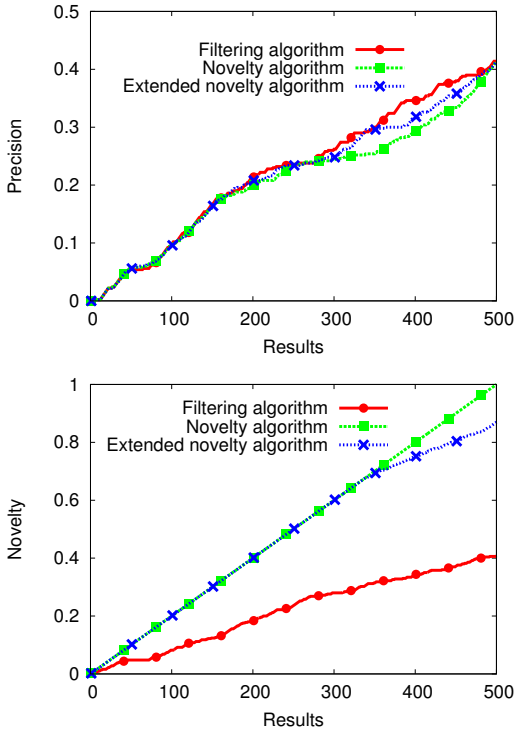

Figure 1: Precision of the filtering results and novelty score of the MCAT context.

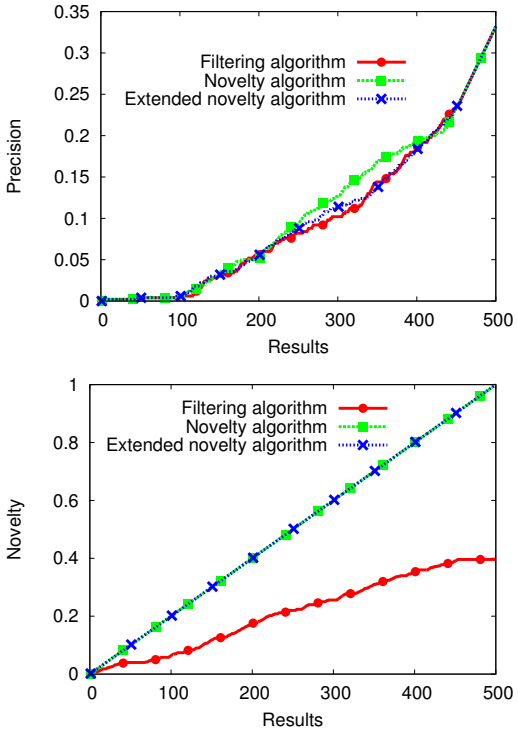

Figure 2: Precision of the filtering results and novelty score of the CCAT context.

5. REFERENCES

[1] C. L. Clarke, M. Kolla, G. V. Cormack, O. Vechtomova, A. Ashkan, S. Büttcher, and I. MacKinnon. Novelty and diversity in information retrieval evaluation. In *Proceedings of the 31st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '08, pages 659–666, New York, NY, USA, 2008. ACM.

[2] K. Goldberg, T. Roeder, D. Gupta, and C. Perkins. Eigentaste: A constant time collaborative filtering algorithm. *Inf. Retr.*, 4(2):133–151, July 2001.

[3] S. Kaski. Computationally efficient approximation of a probabilistic model for document representation in the websom full-text analysis method. *Neural Processing Letters*, 5(2):69–81, 1997.

[4] D. D. Lewis, Y. Yang, T. G. Rose, F. Li, G. Dietterich, and F. Li. Rcv1: A new benchmark collection for text categorization research. *Journal of Machine Learning Research*, 5:361–397, 2004.

[5] Y.-l. Lin and P. Brusilovsky. Towards open corpus adaptive hypermedia: A study of novelty detection approaches. In J. Konstan, R. Conejo, J. Marzo, and N. Oliver, editors, *User Modeling, Adaption and Personalization*, volume 6787 of *Lecture Notes in Computer Science*, pages 353–358. Springer Berlin Heidelberg, 2011.

[6] M. Phanich, P. Pholkul, and S. Phimoltares. Food recommendation system using clustering analysis for diabetic patients. In *Information Science and Applications (ICISA), 2010 International Conference on*, pages 1–8, April 2010.

[7] G. Salton and C. Buckley. Term-weighting approaches in automatic text retrieval. *Inf. Process. Manage.*, 24(5):513–523, Aug. 1988.