Improving Novelty in Streaming Recommendation Using a Context Model [*]

Doina Alexandra Dumitrescu Simone Santini
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Madrid, Spain
{doina.dumitrescu, simone.santini}@uam.es

ABSTRACT

In recent years there has been an increasing research interest in novelty/diversity detection in Information Retrieval and in Recommendation Systems. We propose a model that increases the novelty of recommendations using a context user profile that was created automatically using self-organizing maps. Our system was evaluated on the Reuters Corpus Volume 1 and our experiments show that filtering the recommended items using a novelty score derived from the context-based user profile provides better search results in terms of novel information that is presented to the user.

Categories and Subject Descriptors

H.3.3 [**Information Systems**]: Information Search and Retrieval

General Terms

Algorithms, Metrics, Experimentation

Keywords

Novelty, Diversity, User Model, Context

1. INTRODUCTION

The plethora of information available on the internet about virtually anything, and the duplication of this information due to the significant replication of the sources has given a great relevance to the issues of *novelty* and *diversity* in retrieval and recommendation systems [2, 5]. The definition of novelty and diversity generally accepted in information retrieval is that diversity helps the server deal with query *ambiguity*, while novelty helps the user deal with query *underspecification* [8]. Consider a query composed of the word *manhattan.* The query is ambiguous because the server doesn't know whether the user wants to know something about the neighborood in New York, about the cocktail, or about the Indian tribe: the query has several possible *interpretation*, and whoever formulated the query is interested in general in only one of them. Without further specification, the safest bet for the server is to provide a *diverse* result set, that is, one that covers all these topics, possibly in an amount proportional to the estimated *a priori* user interest (one can assume, for instance, that more people are interested in New York than in the Indian tribe). Even if the query is not ambiguous and the results are about what the user really meant (say, the neighborhood in new york), the user would want every document to be *novel*, that is, to provide information that is not present in other documents. A highly informative text about the history and the human landscape of Manhattan is probably a very good first result, but another document almost identical to the first is not an equally desirable second result: the information that it contains has already been seen. That is, every interpretation of a query contains several *aspects*, and the person who formulated it will be interested, to some degree, in all of them. These considerations highlight an important difference between diversity and novelty. Diversity is something that the user wouldn't really want: the person interested in the manhattan cocktail would be very happy to receive results only about the cocktail. Diversity is needed mainly by the server, which doesn't know enough about the user true intentions. Novelty, on the other hand, is something the user wants, since she will be interested in all the aspects of her query. Novelty is a way to avoid redundancy, while diversity is a way to deal with the server's ignorance.

If we have access to a model of the user's interests, then it is possible to reduce ambiguity and, therefore, to reduce the need for diversity, concentrating only on increasing novelty. We present a method for increasing novelty that relies on a context model built on texts previously seen by the user. The scenario that we use here to exemplify the working of our method is that of the recommendation of news, and the context is composed of past news that the user has considered interesting. This is just one possible way in which context can be gathered. Our model is very general and can be used to represent text or multimedia data. In past work, we have used, for example, the contents of the user's hard disk to represent his main interests [7]. In this paper we show and evaluate a modification to our model with the purpose of increasing the *context coverage* of the recommendations, that is, the fraction of the user context that the

[*]This work was supported by the *Ministerio de Educación y Ciencia* under the grant N. TIN2011-28538-C02, *Novelty, diversity, context and time: newdimensions in next-generation information retrieval and recommender systems.*

CARS-2012, September 9, 2012, Dublin, Ireland.

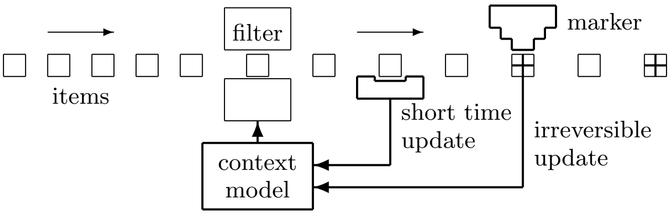

Figure 1: Schema of out stream recommendation system.

recommended items will cover, trying to avoid items that refer to parts of the context already covered by previously seen items.

2. STREAM RECOMMENDATION

One important distinction between this work and others on novelty and diversity in recommendation is that we are dealing with a *stream* of news, that is, with a continuous–potentially infinite–interaction, while typical recommendation systems work based on *sessions*: the user chooses an item or requests a recommendation, and a finite set of results is shown. This operation may be repeated several times, as the user selects several items and receives correspondingly changing recommendations. No work, to the best of our knowledge, has considered novelty in a *session-less* situation, in which a countable stream of items arrive and a system has to decide whether they are relevant for the user at that moment. Note that, differently from the case of a closed session, we don't have access to the whole data base of items that can be recommended. We don't even have access to the items that have already been shown, since in a stream their number is unbound. All we can do, at any given time, is take a decision only considering how the past history of items has modified the user context. The system is shown schematically in figure 1. We have a stream of items (news, in our test scenario) that arrive continuously to the user. The user is described by the context model C (the details of this representation will be given in the next section) and each news is compared with the context and assigned a score. Whenever the score is above a certain threshold (or a predefined criterion is met) the item is presented to the user. At the same time, the context undergoes a first kind of change, which we call a *short term change.* This change (used to improve the novelty of the recommended items) is characterized by a time constant β. After a time, depending on β, the change provoked by any individual item disappears. This is one important difference between our model and the session based ones. Let us assume that a person is interested in politics. News about the election of the new French president will undoubtedly be of interest to her. However, she will probably not be interested in receiving many similar news about the election one after the other. The effect of the short term change will be do "deactivate", for a time, the part of the context that deals with this kind of news, so that further news about the French elections will not, *cœteris paribus*, be considered as interesting. However, this situation should not be permanent: after a while (a few hours, a day,...), further news about the French President will again be considered interesting, and should be recommended.

In addition to this short term change, the context evolves in a permanent and irreversible way, again under the influence of the documents read (as well as, possibly, based on other elements such as the documents that the user creates on her computer, or the emails that she writes). A number of strategies can be used for this evolution: the user can be asked to mark certain documents of special interest, the system can present only a summary, and all the documents whose complete text is accessed are permanently entered into the context, etc. We have considered the evolution of the context based on documents of interest elsewhere [7], and we shall not consider it here. From the point of view of this paper, therefore, the context, defined as in the following section, is fixed, and only the short term changes, needed to increase novelty, are considered.

3. THE USER MODEL

There is convincing evidence that a user model can help attain a higher precision in information retrieval and recommendation systems [1, 6]. In this paper, we are interested in using such a model to achieve novelty, that is, to obtain recommendations that span the whole gamut of the user's interests. The context model that we use was presented (without the extension to enforce novelty that we present here0 in [7], to which the reader is referred for details; here we shall only give a brief description. 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 or, in our case, by news items marked in the past. The model doesn't change depending on the source of the documents. 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 he has found relevant.

All the documents in the contexts are considered as a single "macro-document" and processed consequently. On this macro-document we perform stop-word removal and stemming. Each term t is assigned a weight w_t using tf-idf weighting [1]. We use the standard vector model representation of information retrieval in which each different term corresponds to an axis in the word space. We consider n-grams, constituted of n consecutive terms that appear in the document. If the n-gram p_k is composed of the terms $t_{u_1}, \ldots, t_{u_n}$, with weights $w_{u_1}, \ldots, w_{u_n}$, then its representation in the word space is given by the point

$$p_k = \frac{1}{\sqrt{\sum_{i=1}^{n} w_{u_i}^2}}(\ldots, w_{u_1}, \ldots, w_{w_2}, \ldots, w_{u_n}, \ldots)$$

in the T-dimensional term space. Note that all points are normalized, so they are in reality points in the manifold S^{T-1} (the unit sphere in $\mathbb{R}^T$). A context is represented by a set of these n-gram representations, what we call a *point cloud* representation. That is, a context is a finite set of points $\mathcal{C} \subseteq S^{T-1}$.

In the word space, we lay a self-organizing map, using a modification of WEBSOM [3]. The map is a grid of elements called *neurons*, each one of which is a point in the word space and is identified by two integer indices, that is, a neuron is given as:

$$[\mu\nu] = ([\mu\nu]_1, \ldots, [\mu\nu]_T)' \quad 1 \le \mu \le N, 1 \le \nu \le M \qquad (1)$$

The map is discrete, two-dimensional[1] with the 4-neighborhood topology. The neurons are immersed in the T-dimensional word space ($[\mu\nu] \in \mathbb{R}^T$), and in our version of WEBSOM their weights are normalized. That is, throughout the algorithm we enforce $\sum_{t=1}^{T}[[]\mu\nu]_t^2 = 1$ for all map indices $\mu\nu$. The neurons are at the same time elements of the discrete grid and points in S^{T-1}. As points in the discrete grid, the relevant distance between two neurons is their *graph distance*:

$$\delta([\zeta\xi], [\mu\nu]) = |\zeta - \mu| + |\xi - \nu| \tag{2}$$

As points in S^{T-1}, it is possible to determine the similarity between a neuron and any other point $p \in S^{T-1}$ as

$$s([\mu\nu], w) = \sum_{i=1}^{T} [\mu\nu]_i p_i \tag{3}$$

On this map we define a *neighborhood function*, $h(t, n)$, which depends on two parameters $t, n \in \mathbb{N}$; n is the graph distance between two given neurons, t is a time parameter that increases as learning proceeds. The function $h(t, n)$ represents the "degree of neighborhood-ness" of two neurons at a distance n at time t; We assume that $0 \le h(t, n) \le 1$, $h(t, 0) = 1$, and that h is monotonically decreasing in n and t. The degree to which neuron $[\zeta\xi]$ belongs to the neighborhood of neuron $[\mu\nu]$ at time t is given by $h(t, \delta([\zeta\xi], [\mu\nu]))$. In addition to the neighborhood we define a positive *learning parameter* $\alpha(t), t \in \mathbb{N}$, monotonically decreasing with t.

In order to create the map for a context $\mathcal{C}$, all the points in it are presented to the map, and the training algorithm is applied. We call the presentation of a point $p \in \mathcal{C}$ an *event* of learning, and the presentation of all the points of $\mathcal{C}$ an *epoch*. Learning consists of a number of epochs, counted by a counter t. The neurons of the map are at first spread randomly in the word space; then, for each event consisting of the presentation of the point p, the neuron with the highest similarity to p is found:

$$[*] = \arg\max_{[\mu\nu]} s(p, [\mu\nu]); \tag{4}$$

the neuron $[*]$ and all its neighbors are shifted towards p. The amount of this shift depends on the learning parameter α and on the distance from $[*]$ on the map:

$$\forall[\mu\nu] \quad [\mu\nu] \leftarrow [\mu\nu] + \alpha(t) h(t, \delta([*], [\mu\nu])) \cdot (p - [\mu\nu]) \tag{5}$$

Finally, all neurons are re-normalized in order to maintain them on the sphere S^{T-1}:

$$\forall[\mu\nu] \quad [\mu\nu] \leftarrow \frac{[\mu\nu]}{\sqrt{\sum_i [\mu\nu]_i^2}} \tag{6}$$

4. NOVELTY IN NEWS FILTERING

If we don't consider novelty, we can filter the news items as they arrive simply by representing them as a point $u \in S^{T-1}$ (considering it as a document and using the same information retrieval techniques that we have used to represent the context), and determining the maximal similarity between u and the neurons of the map

$$r = \max_{[\mu\nu]} s([\mu\nu], u) \tag{7}$$

the items that yield a value of r below a certain threshold are considered interesting and presented to the user. As we have mentioned in the introduction, the problem with this simple solution is that very similar news items can have high score because they are similar to the same neuron (that is: to the same part of the user context), although the presentation of more than one adds little to the information content of the first one. To increase the diversity of the results, we rewrite (7) adding an *interest* factor $w_{\mu\nu}$ to each neuron, and computing the relevance of the news item u as

$$r = \max_{[\mu\nu]} w_{\mu\nu} s([\mu\nu], u) \tag{8}$$

Let us call $[*]$ the "winning" neuron for a given item (the neuron closest to it, viz. the neuron for which the maximum of eq. (8) is attained). Whenever an item is recommended, the interest factor of each neuron is updated according to the equation

$$w_{\mu\nu} \leftarrow \begin{cases} \lambda w_{\mu\nu} & \text{if } [\mu\nu] = [*] \\ \min\{1, (1+\beta) w_{\mu\nu}\} & \text{otherwise} \end{cases} \tag{9}$$

with $0 < \lambda < 1$ and $\beta > 0$. According to this equation, the interest of the neuron closest to a recommended news item is reduced by a factor λ, reducing its contribution to (8) and therefore making it less likely to win again. This means that further news items close to the one that caused this neuron to win will be less likely to be recommended. If the neuron is no longer the winner for any item, its interest will be increased by a factor $(1 + \beta)$ with each recommended item, until the value 1 is restored (or until the neuron wins again, in which case its interest factor will again be reduced). If the neuron wins only once, its *relaxation time* (the number of items necessary for it to go back to 1) is

$$t_r = -\frac{\log \lambda}{\log(1+\beta)} \tag{10}$$

If we want to achieve effective diversity, this time should be of the order of magnitude of the number of neurons in the network, but not greater, as if neurons win too often their interest factor will decrease, on average, as a function of time. A reasonable choice is to set $t_r \approx \frac{K}{10}$, where K is the number of neurons. Expressing, say β as a function of λ and t_r, we obtain

$$\beta = \left(\frac{1}{\lambda}\right)^{\frac{1}{t_r}} - 1 \tag{11}$$

so, for the desired recovery span, we have

$$\beta = \left(\frac{1}{\lambda}\right)^{\frac{10}{K}} - 1 \tag{12}$$

In this schema, we only reduce the interest factor of the neuron $[*]$. In our tests, we also tried another schema, in which we reduce (by a progressively smaller amount) the interest factor of neurons in the neighborhood of the winner using the same function h that we have used in the training algorithm. In this case, the update of the interest factor is

[1] The map can be k dimensional; however, maps with $k > 4$ are rarely used because of the combinatorial explosion in the number of neuron which would lead to over-fitting most training sets. We have used mostly maps with $k = 2$, and we shall consider only this case here, mainly because it avoid notational complexities.

done with the function

$$w_{\mu\nu} \leftarrow \begin{cases} \lambda h(0, [\mu\nu], [*]) w_{\mu\nu} & \text{if } \delta([\mu\nu], [*]) < \tau \\ \min\{1, (1+\beta) w_{\mu\nu}\} & \text{otherwise} \end{cases} \qquad (13)$$

where h is the neighborhood function, and τ is a suitable threshold (in our experiments, we set $\tau = 2$).

5. TESTS

We consider two measures. The first one is the standard *precision* (the fraction of recommended documents that are relevant). Given a context and a set of recommendations, we expect that the use of novelty will somewhat decrease precision (if there are two very relevant documents that are almost identical, a system that considers novelty will recommend only one of them). So, this measure is a sort of quality control: we do accept a certain reduction of precision, but we want to keep it under control, to avoid losing too much result quality. The second measure is a determination of novelty. We measure, for a given number of recommended items, the number of neurons that win. If D is the number of documents recommended and, during the recommendation of these documents, n_w different neurons are winners (i.e. they are the neurons that achieve the maximum in eq. (8)), then the *coverage* is defined as $V = n_w/D$ ($0 < V \leq 1$). Note that $n_w < K$ so that if the documents arrive in a stream ($D \to \infty$) in the long run this value will tend to zero. In order to obtain a significative value, we always consider sets of recommended documents smaller than the number of neurons in the network.

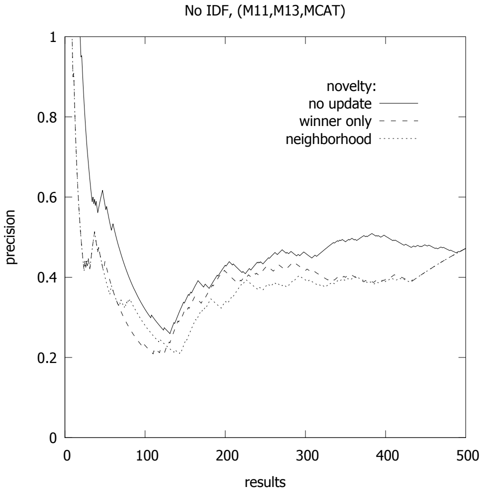

Figure 2: Precision vs. number of recommended items (simple weighting).

To validate the proposed approach, we conducted some experiments using 2.5 gigabytes of uncompressed news stories from Reuters Corpus Volume 1 (RCV1-v1) [4]. The Reuters's is a collection of 806,791 news stories, each one associated with some metadata. In our experiments, we used the text and the topics categories each news document belongs to. The RCV1-v1 test collection contains 117 topics in a three-level hierarchy with, at the top, four categories: CCAT (Corporate/Industrial), ECAT (Economics), GCAT (Government/Social), MCAT (Market). Fourteen of these categories contain no document, and in our experiments we used only the 103 that do.

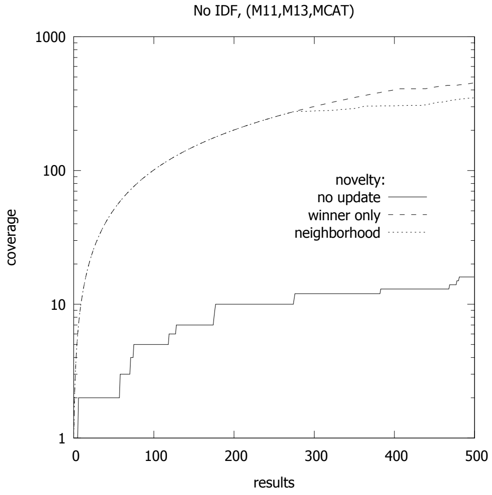

Figure 3: Coverage vs. number of recommended items (simple weighting)

We began by creating a context. To this end, we used 5% of the news items contained in the sub-categories (of MCAT) *M11* (EQUITY MARKETS) and *M13* (MONEY MARKETS), for a total of about 5000 documents. Once the context is in place, we create a stream of news with all the elements in the Reuters data base and use our method for recommending items that are close to the context. As a ground truth, we consider as relevant all the items that are in the context categories (M11 and M13) and in their parent category (MCAT), but not those that are in the sibling categories of M11 and M13. We stop the stream when 500 items have ben recommended (this is necessary in order to get significant coverage values, as we mentioned before). Each item is analyzed and represented as a point $u \in S^{T-1}$ (see section 4). The weights of the terms that compose each item are determined according to two different schemas: *Simple weighting* (without an inverse document frequency (idf) term; the weight of a word is calculated using the normalized term frequency (tf), and *tf-idf* (the normalized frequency of a word is divided by the logarithm of its frequency in the British National Corpus (BNC) word frequency list[2]). Figures 2–5 show the results of our tests. Each figure contains three curves: the first is obtained by filtering the items without taking into account novelty (having the interest factor of each neuron always equal to 1). For the second, we used the adaptation of (7), with $\lambda = 0.1$ and a relaxation time long enough so that once a neuron has won, its interest factor will remain low for the duration of the test; the third is obtained using the neighborhood adaptation (9), with $\tau = 2$. The values of λ and τ are not critical, and some preliminary tests on limited data sets allowed us to select reasonable values. Figures 2 and 3 show the precision and the coverage V for filtering done with simple weighting (without an idf term). The precision drops only marginally when considering novelty, but the number of neurons that "win" is increased by more than an order of magnitude, confirming that the results are relative to a broader portion of the user context.

[2] Available at http://www.natcorp.ox.ac.uk/

Figures 4 and 5 show the same results using a weighting scheme with the idf term for the news items. The results are qualitatively the same, although we observe a general drop in precision. We have not investigated the causes of this drop, which we suspect are related to the small number of words in the items. Note that in the range 200-500 results, the system with novelty actually achieves a better precision that the simple system. We haven't thoroughly investigated this phenomenon, but it is possible that the simple system get "stuck" in groups of similar documents from other categories with a similar (and relatively high) value of measred relevance. Introducing novelty in this case would reduce the presence of these groups.

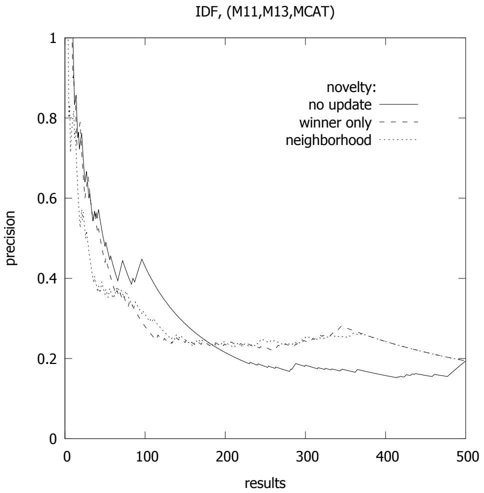

Figure 4: Precision vs. number of recommended items (tf-idf weighting)

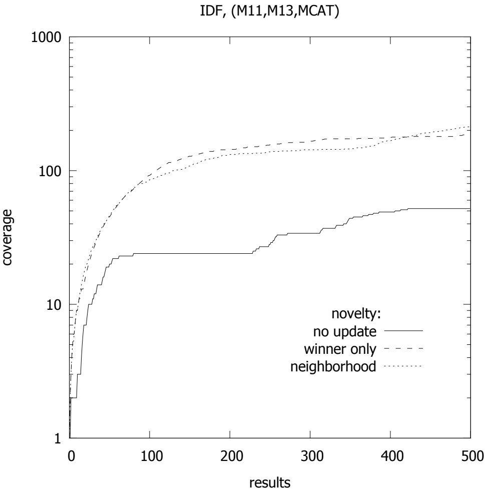

Figure 5: Coverage vs. number of recommended items (tf-idf weighting)

We must also remark that the MCAT category is quite difficult to filter, as many of the significant terms that appear in it are part of the economics jargon, and appear in other categories as well, such as ECAT. This explains the relatively low values of precision that we obtain, independently of the use of the method for increasing novelty. This might also explain why in some cases the novelty system attains a higher precision than the one based only on relevance: there are probably parts of the context more ambiguous than others sine they are sensitive to words that appear in categories other than MCAT. With the novelty system, once an item has activated these part, they become "desensitized," and cease to produce irrelevant results.

6. CONCLUSION

In this paper, we have presented a method for increasing the novelty of a recommendation system that receives a *stream*, as opposed to the session-based recommender systems common in the literature. We have shown that a model of the user context can be useful to increase the novelty of the recommended results without having to receive results irrelevant for the user. The fact that we receive a potentially infinite stream of items poses new challenges for novelty increasing systems, as redundancy, which is taken to be a constant in session based system, "fades away" in the case of stream: a result very similar to one already received will be considered redundant in the short term, but may become interesting again in the long term. To deal with this problem, we have introduced an interest factor in our model that is reduced when a certain area of the context is activated but that, if the area is no longer activated, recovers in a time that can be controlled by design.

7. REFERENCES

[1] J. e. a. Allan. Challenges in information retrieval and language modeling: report of a workshop held at the center for intelligent information retrieval. *SIGIR Forum*, 37(1):31–47, Apr. 2003.

[2] E. Gabrilovich, S. Dumais, and E. Horvitz. Newsjunkie: providing personalized newsfeeds via analysis of information novelty. In *WWW '04: Proceedings of the 13th international conference on World Wide Web*, pages 482–490, New York, NY, USA, 2004. ACM.

[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), 1997.

[4] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *J. Mach. Learn. Res.*, 5:361–397, Dec. 2004.

[5] X. Li and B. W. Croft. An information-pattern-based approach to novelty detection. *Information Processing & Management*, 44(3):1159–1188, May 2008.

[6] G. Pasi. Contextual search: Issues and challenges. In A. Holzinger and K.-M. Simonic, editors, *Information Quality in e-Health*, volume 7058 of *Lecture Notes in Computer Science*, pages 23–30. Springer Berlin / Heidelberg, 2011. $10.1007/978 - 3 - 642 - 25364 - 5_3$.

[7] S. Santini and A. Dumitrescu. Context as a non-ontological determinant of semantics. In *Proceedings of the 3rd International conference on Semantics and digital media technologies*, 2008.

[8] R. Song, Z. Luo, J.-Y. Nie, Y. Yu, and H.-W. Hon. Identification of ambiguous queries in web search. *Information Processing & Management*, 45(2):216 – 229, 2009.