Using the reader's context to customize news streams

Alexandra Dumitrescu and Simone Santini

Escuela Politécnica Superior, Universidad Autónoma de Madrid

Abstract. Many people today subscribe to streaming news services, and their number can be expected to grow together the number of web sites that offer a streaming service using protocols such as *Real Simple Syndication*.
In this paper we present a method, and the relative algorithms, for ranking the news for relevance based on the *working context* of the reader. We use the contents of the reader's computer as an indication of his or her interests, and build with them a suitable context representation. We then use it to filter the incoming stream of news.

1 Introduction

At the end of the XVIII century, Kant wrote that "reading the newspaper is the morning prayer of the modern man". We don't know whether he pushed this analogy (quite daring as it was, for a devout Pietist as Kant) to the point of identifying the newsstand with the modern Church, but he might as well have. If he had, and if he could see us today, he would recognize that while the nature of his morning prayer hasn't changed that much (except, maybe, for becoming shorter and more superficial), the way we get to it has. Quite a bit. Keeping with the zeitgeist of this age that looks with suspicion at most forms of community, the reader today receive their prayers directly in the privacy of their own home.

The habit has become quite common, among internet users, to subscribe to news services using protocols such as *Real (sic) Simple Syndication* (RSS) that allow a news reading program, residing on the computer of the reader, to inquire periodically the availability of news and receive them in the form of an XML file with the headline, various keywords, a short synopsis of the news item, and a link to a related web page.

The RSS protocol has become so popular that selecting the right news sources and the right categories has become somewhat of a new art. Be too lax in your criteria and you will soon be flooded by an overwhelming amount of irrelevant (to you) trivia; be too strict, and you will miss the crucial story that you were waiting to hear, thereby committing embarrassing social *faux pas* when everybody around the dinner table talks of something you know nothing about. To this we might add that by being strict and rigorous one might lose an important element of serendipity. You are, say, a computer addict but find fishing excruciatingly boring, therefore you would not touch any issue of *Rifle & Pole* with a ten foot pole (pun intended). But, in so doing, you risk of losing the terrific article, to appear on the September issue, on artificial intelligence application for smart baits in fly fishing, or any article with a connection to *A river runs through it* that, in spite of your dislike for fishing, happens to be your favorite Redford film.

The problem, the way we see it, is that, as good as the classification that come with the news can be, as cleverly chosen as the keywords can be, they are not *your* classes, they are not *your* keywords. The people who choose the keywords and classify the new may be good, but they are not you: they don't have your special interests, your particular way of deciding what is interesting and what is not. And the problem is, even you are not completely you. Not always, not at any time. Your individuality (and your interests with it) are not an abstract a-temporal quality. You exist historically, as a lived *now*, and the individuality who is interested in certain things today is not the same as the individuality who was interested in other things five years ago [2]. You change, and the things you like change with you. This change is continuous, subtle, and it can's be just schematized by saying that five years ago you liked wrestling and today you like transcendental metaphysics. There is (in general, at least) no specific point in time, in the last five years, at which you stopped liking wrestling and you started liking transcendental metaphysics, just a continuous shift of interests from one thing to the other and, at the same time, from a hundred other things to a hundred more. You are a complicated persons, and simple classification won't capture you.

Of course, one might consider a simple solution: every user will explicitly declare the things he or she is interested in, choose a suitable set of keywords to cover the areas of interest, so that the news reader can match these keywords against the pre-defined categories that come attached to every news story. This solution works very well from a computing perspective, but ignores completely important aspects of human psychology. For one thing, people are very good at recognizing interesting things, but they are not so good at describing explicitly what they are interested in. They invariably restrict too much (thereby losing interesting news) or too little (thereby being overwhelmed by irrelevant news). People, quite simply, don't know what they are interested in until they see it. Second, matching against pre-defined categories presents the same problem as we have mentioned before. The solution works only if the pre-defined categorization is done along the same lines as the categorization you have (implicitly) in mind, and this is practically never the case. We are led to two considerations:

i) The pre-defined categories attached to news are often meaningless; the only relevant thing, the only basis on which a decision can be made in the text itself of the news, nothing less.

ii) People can't explicitly declare what they are interested in, but they do declare it implicitly through their *activities*: the things people do and say determine what they are interested in, not the explicit declaration of interest that they can make.

We can't, of course, encode and represent the whole activity universe of a person, since most of it takes place without being registered in any way, but there is an important component of one's activities to which we do have access: those that take place with the help of digital devices. The advantage here is double: on the one hand, digital devices have more memory capacity and become more "expressive," so to speak, therefore allowing a better recording of the activities that take place with their help. On the other hand, they become more ubiquitous, so that the fraction of one's general life activities that are recorded on digital devices increases.

We are led again to the basic assumption that we made in [8]: we can use the content of one's computer (or cell phone, or MP3 player, or camera... you name it) to get a representation of one's activities and, therefore, of one's interests. In [8] we used this context to direct one's searches on the internet, in this paper we will use to filter the news that one receives.

2 Context representation

As we mentioned in the introduction, we represent the context by analyzing the contents of one's computer and by developing a suitable representation for it. The technique that we use here is quite similar to one that we used in a system for context-based query rewriting for internet search, called *Cœus*, that we presented two years ago in this very series of conferences [8]. The method of construction of the context representation is not specific to this application, and is an improvement of the one that we presented (in the time since the first presentation, the same improvements have been applied to the search engine as well). What is specific to this new scenario is the way in which the context is applied. In *Cœus*, a few keywords that the user typed as search terms, were mapped into the context and used to deform it. The difference between the deformed context and the original one was used to create the modified query. The query terms were in general assumed to be relevant, that is, they were assumed to be terms that were part of teh context. If this was not the case, the query was not modified and the query terms themselves were used for a keyword search.

In this case, we have a stream of documents arriving from various news sources, and we are interested in using the context to *filter* them. There is no context deformation in this application, but each news item is mapped to a point in the context space, and the relevance of the news is a function of their distance from the *latent semantic manifold* that forms the context. An incoming news item will in general contain a certain number of words that appear in the context and a certain number of words that don't, and we need to deal carefully with these words to avoid false positives, as we shall see in the following sections.

* * *

Our context representation is based on a self-organizing map that we lay out in a suitable space of words to constitute a *latent semantic manifold*, that is, a non-linear (as opposed to the linear latent semantic subspace of [1]) low-dimensional subspace of the word space that capture important semantic regularities among words. The technique is based on the self-organizing map WEBSOM [4, 3].

We shall divide the construction of the representation in two parts. First, we build a low level representation that captures the syntactic regularities that exist in the documents of the context. This process results in a context representation in the form of a *point cloud* in a vector space whose axes represent words. In this space we then train the self-organizing map to constitute the *latent semantic manifold* that will be our final context representation and our query tool. We shall consider the two constructions separately.

2.1 The point cloud

In each work area of the user computer we build a complete context representation, all the way to the latent semantics manifold, that depend on the documents contained in a specific working context. In the original version of the context representation, we took a rather rigid view of the organization of the context. We assumed that contexts were determined by the working directory in which one was working at a given time, and the documents that we considered in order to build a context were those of the working directory and those of the descendants (with different weights). This fairly inflexible way of building the context created problems when it came to include documents that didn't quite fit into the directory organization. In some cases, for instance, we might want to include the e-mail messages sent while carrying out a certain activity (received messages and the corresponding answers follow a very haphazard pattern, and there is no obvious way to use the moment when they arrive to connect them to a specific activity), or the web pages accessed and read during a certain activity. We have several algorithms under study to collect this kind of information, but all them require a more general, simpler, and more flexible definition of context than that bound to the directory structure. We use the following simple definition:

Definition 1. *A* raw context *is a set of pairs* $\mathfrak{R} = \{(w_1, D_1), \ldots, (w_n, D_n)\}$*, where* $D_i = [t_i^1, t_i^2, \ldots, t_i^{n_i}]$ *is a (ordered) list of words (terms), and* $w_i \in \mathbb{R}^+$ *is a positive weight.*

With this definition we decouple the problem of representing the context from that of building it: we can use any number of algorithms to determine the weights of documents in a working context, without affecting the algorithms that will then build a representation for it.

The point cloud representation of the context $\mathfrak{R}$, is called the *index* of $\mathfrak{R}$, and is built using fairly standard techniques derived from information retrieval. We begin by applying to all the documents D_i stopword removal and stemming. The result is a series of *stems* of significant words[1] (see figure 2.1). From each sequence of stems we consider groups of n consecutive stems, which form what we call *word groups*. In figure 2.1 we have illustrated the case $n = 2$, which is the one that we shall consider in this paper. Correspondingly, we will talk about *word pairs* rather than groups[2]. Each document is now represented as a bag of pairs. Let $\{t_1, \ldots, t_W\}$ be the set of all terms that appear in all documents, and let $(uv)_i$ be a pair formed by the words t_u followed by the word t_v as they appear in the document D_i. Let $|uv|_i$ be the number of times such pair appears in the document D_i. The *raw pair frequency* of the pair (uv) in the context $\mathfrak{R}$ is the sum of all the occurrences of the pair in the various documents weighted by the weight

[1] In the following we will ignore the distinction between a stem and a word, and refer to the stems as *words* whenever no confusion may arise.

[2] In the information retrieval literature, these groups are called *word contexts* [9]. Since the expression *context* is one of the essential concepts in this paper, and since we are using it in quite a different connotation, we prefer to depart from the standard terminology rather than risking a significant confusion.

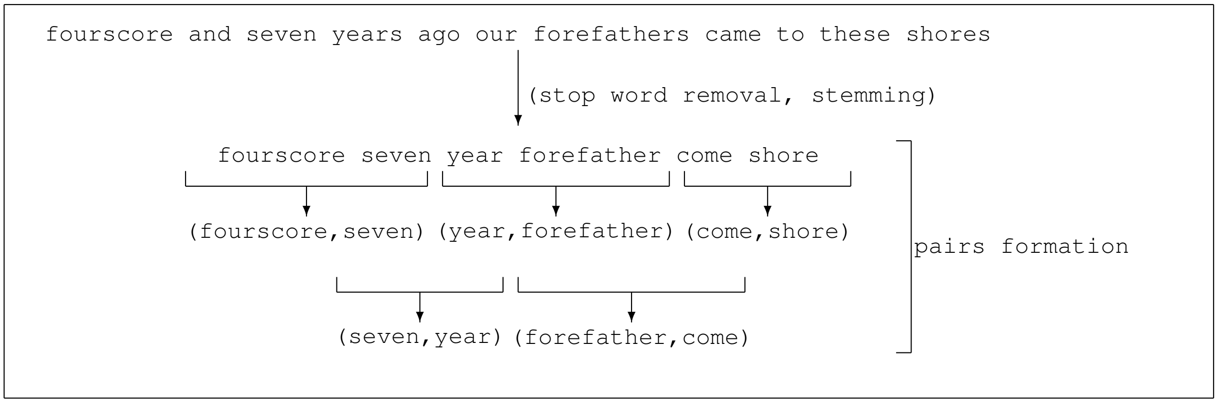
fourscore and seven years ago our forefathers came to these shores
(stop word removal, stemming)
fourscore seven year forefather come shore
(fourscore,seven) (year,forefather) (come,shore)
pairs formation
(seven,year) (forefather,come)

Fig. 1. Initial steps for the construction of the generator. The text of the documents in the directory is first processed by removing stop-words and doing stemming on the remaining words. From the list of stems we then extract all pairs of consecutive words.

associated to the documents:

$$\mathrm{pf}_{(uv)} = \sum_i w_i |uv|_i \tag{1}$$

In information retrieval, this weight typically undergoes some kind of normalization, usually multiplying it by a monotonically increasing function of the inverse of the fraction of the corpus documents in which the term appears. This normalization has the purpose of giving more weight to highly discriminative terms: a term that appears many times in a document D_i but appears also in many documents of the collection is less useful, as a discriminating factor, than a term that appears just as frequently, but which appears in just a few documents. Terms that appear sparingly in the corpus are assumed to be more characteristic of the few documents where they appear. The extreme case of this analysis is that of stop-words: words that appear many times in a given document but that are so common so as to provide virtually no indication as to its contents. In the case of stop words, this phenomenon is so accentuated that the only sensible thing to do is to eliminate the word altogether; in other cases we may want to retain the word, but giving it a smaller weight.

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: the only corpus we have is the context itself. What we do have is the frequency with which each word appears in the general corpus of English writing. Statistical data in this sense is available on a number of web sites, and it gives us the relative frequency e_u of the term t_u in the whole corpus of English texts. We use this *inverse frequency* in order to normalize our weights. Unfortunately, we were not able to find data on the relative frequency of *pairs* of words in English, so we were only able to approximate the pair frequency as the product of the frequency of the individual words, thereby making an independence assumption. We define the inverse frequency of the pair (uv) as

$$\text{if}_{(uv)} = \log \frac{1}{e_u e_v} = -(\log e_u + \log e_v) \tag{2}$$

With this frequency, we define the *raw weight* of the pair (uv) as its *pf-if* weight:

$$\tilde{\omega}_{(uv)} = \frac{1}{C} \text{pf}_{(uv)} \, \text{if}_{(uv)} = -\frac{1}{C} (\log e_u + \log e_v) \sum_i w_i |uv|_i \tag{3}$$

where $C = \max\{\tilde{\omega}_{(uv)}\}$ 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 pairs 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:

$$\omega_{(uv)} = (\tilde{\omega}_{(uv)})^\alpha = -\frac{1}{C^\alpha} \left((\log e_u + \log e_v) \sum_i w_i |uv|_i \right)^\alpha . \tag{4}$$

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

The pairs of terms are represented as points in a vector space whose axes are the words $t_1, \ldots, t_W$. In this space, the pair (uv), with weight $\omega_{(uv)}$ is represented by the point

$$p_{uv} = (\underbrace{\overbrace{0, \ldots, \omega_{pruv}}^{u}, 0, \ldots, \omega_{(uv)}}_{v}, 0, \ldots, 0) \tag{5}$$

Each point p_{uv} lies in the two-dimensional sub-space determined by the axes t_u and t_v (see figure 2.1) At the end of this step, the context $\mathfrak{R}$ is represented by a set of points $I_{\mathfrak{R}}$ in this space.

2.2 The latent semantics manifold

The point cloud thus built is used as the training data for a self-organizing map deployed in the term space. 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:

$$[\mu\nu] = ([\mu\nu]_1, \ldots, [\mu\nu]_T) \quad 1 \le \mu \le N, 1 \le \nu \le M \tag{6}$$

The map is discrete, two dimensional with the 4-neighborhood topology. That is, given the neuron $[\mu\nu]$, its neighbors are the neurons $[(\mu-1)\nu]$, $[(\mu+1)\nu]$, $[\mu(\nu-1)]$, and $[\mu(\nu+1)]$. We can visualize the map as a grid laid out in the word space with rods joining neighboring neurons (figure 2.2). Given two neurons, $[\mu\nu]$ and $[\zeta\xi]$, we can measure their distance in two ways:

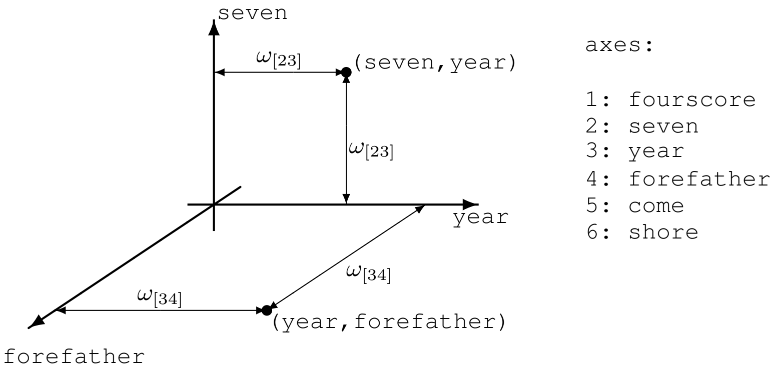

Fig. 2. Position of the points of a *point cloud* in the word space.

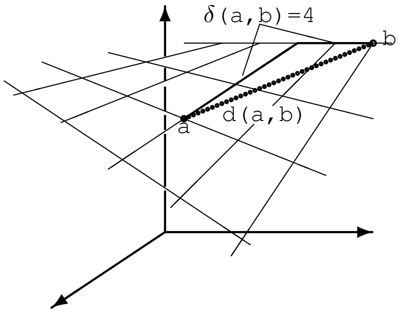

Fig. 3. The self-organizing map in the input space with the two distances defined in it: the distance d between neurons considered as elements of the input space, and the distance δ between the same neurons measured on the map.

i) as points in the word space, that is, assuming that the metric is Euclidean, as

$$d([\zeta\xi],[\mu\nu]) = \left[\sum_{i=1}^{T}([\zeta\xi]_i - [\mu\nu]_i)^2\right]^{\frac{1}{2}} \tag{7}$$

Note that this distance can be computed between any point in the word space and a neuron:

$$d(p,[\mu\nu]) = \left[\sum_{i=1}^{T}(p_i - [\mu\nu]_i)^2\right]^{\frac{1}{2}} \tag{8}$$

ii) as points in the grid using the *graph distance* between them (also called the *chemical distance*):

$$\delta([\zeta\xi],[\mu\nu]) = |\zeta - \mu| + |\xi - \nu| \tag{9}$$

If the neurons are dense and form a continuum, this distance reduces to a geodesic distance in the two-dimensional manifold of the map [7].

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 a given neuron (the neuron whose neighborhood we are determining) and another neuron, 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; for it we postulate the following properties:

i) $\forall t.(t \geq 0 \;\Rightarrow\; h(t,0) = 1)$;
ii) $\forall t,n.(t \geq 0 \wedge n \geq 0 \;\Rightarrow\; 0 \leq h(t,n) \leq 1)$;
iii) $\forall t,n.(t \geq 0 \wedge n \geq 0 \;\Rightarrow\; h(t,n) \geq h(t+1,n))$;
iv) $\forall t,n.(t \geq 0 \wedge n \geq 0 \;\Rightarrow\; h(t,n) \geq h(t,n+1))$;

The degree to which neuron $[\zeta\xi]$ belongs to the neighborhood of neuron $[\mu\nu]$ at time t is given by $h(t,\delta([\zeta\xi],[\mu\nu]))$. Condition iv) localizes the neighborhood around $[\mu\nu]$, while condition iii) causes it to "shrink" in time. In addition to the neighborhood we define a *learning parameter* $\alpha(t), t \in \mathbb{N}$ such that

i) $\forall t.(t \geq 0 \;\Rightarrow\; 0 \leq \alpha(t) \leq 1)$;
ii) $\forall t.(t \geq 0 \;\Rightarrow\; \alpha(t) \geq \alpha(t+1))$;

In order to create the latent semantic manifold for a context $\mathfrak{R}$, all the points in the index $I_{\mathfrak{R}}$ are presented to the map, and the training algorithm is applied. We call the presentation of a point $p \in I_{\mathfrak{R}}$ an *event* of learning, and the presentation of all the points of $I_{\mathfrak{R}}$ 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 following learning steps take place:

i) the neuron that is closest to p according to the word space distance is found:

$$[*] = \arg\min_{[\mu\nu]} d(p,[\mu\nu]); \tag{10}$$

ii) 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:

$$\forall[\mu\nu] \quad [\mu\nu] \leftarrow [\mu\nu] + \alpha(t)h(t,\delta([*],[\mu\nu])) \cdot (p - [\mu\nu]) \tag{11}$$

3 News filtering

Consider now a news item received from some suitable news service. We are interested in using the context calculated in the previous section to determine whether this item is of interest or not. The first decision one has to make is on what text should this comparison be made. Each news item in an RSS format, for example, comes with a short synopsis of the story and with a link to the whole story. We may decide to use only the synopsis or to traverse the link and read the whole text. The first solution is preferable in many respect, mainly because it avoid overloading the server with the full text with request for news that then, maybe, will not be accepted. On the other hand, access to the whole html text will give us a better idea of the story and therefore allow us to make a better judgment (although it can sometimes be misleading, as we shall comment later on). Here, we are testing the method with both solution, to obtain a better evaluation of how it works.

The processing to which we submit the new element is quite similar to that used for the generation of the point cloud representation, albeit with a few relevant differences. First of all, we process individual words rather than pairs. The reason for this choice is that a news story, especially those found on the web, are rather short, and do not contain enough words to come up with a reliable statistics on the occurrence of pairs. Therefore we consider the terms t_u in the selected part of the story (description or full text, depending on the test that we are running) and apply a *tf-if* weighting scheme similar to that of the point cloud, obtaining the weight:

$$V_u^k = -\frac{1}{C}\log e_u|u| \tag{12}$$

where $|u|$ is the number of times t_u appears, and C is a constant that keeps the weights in the range $[0,1]$, and k an identifier for the story. To this weight we associate the point

$$p_u^k = (\overbrace{0,\ldots,V_u}^{u},0,\ldots,0), \tag{13}$$

which lies on the axis t_u is the word space. The story is then represented by the point $p^k = \sum_u p_u^k$.

* * *

In order to determine the significance of the story, we compare it with all the neurons in the map for similarity. We use a simple variant of the cosine similarity [6] in which we normalize only with respect to the news vector, that is, we determine the similarity between the kth story and the neuron $[\mu\nu]$ by the semi-normalized dot product:

$$S(k,[\mu\nu]) = \frac{\sum_i^T p_i^k[\mu\nu]_i}{\sqrt{\sum_i (p_i^k)^2}} = \frac{\sum_i^T V_i^k[\mu\nu]_i}{\sqrt{\sum_i (V_i^k)^2}} \tag{14}$$

Note that in this comparison we normalize relative to the news vector but not to the vector of the neuron. This solution was taken based on two considerations. On the one hand, we wanted to compensate for the effects of the length of the news story. Longer stories will contain more instances of relevant words, even though these repetitions do not necessarily result in more relevance. To make a trivial example, if we build a story S' by concatenating two copies of a story S, then S' will contain twice the number of relevant words as the story S, although S' has exactly the same relevance as S. Normalization will take care of this bias towards long stories. On the other hand, in the case of the context, the absolute value of the weights does contain some indication on the relevance of a certain neuron and on its importance for characterizing the context. In the SMART [5] notation (or a little extension thereof) this would be a schema of type *nlc.nln*, where we use the letter *l* for *language frequency normalization* as opposed to the letter *t* for *tdf normalization*.

The maximal dot-product between the story and all the neurons in the context is taken as the relevance of the story for the context:

$$R(k) = \max_{[\mu\nu]} S(k, [\mu\nu]) \tag{15}$$

and all the stories for which the relevance $R(k)$ is above a specified threshold are considered relevant and shown to the user.

4 Test

We have tested the news reading system with two contexts. The first one (henceforth *computing*) was composed of documents of one of the authors (Santini) containing professional documents, drafts of papers, professional emails, etc. The second (henceforth *tennis*) was composed of a collection of articles on tennis. The general idea was to try two contexts as different as possible, based on the following considerations:

i) The language of computing is composed in great part of terms borrowed from other disciplines. Apart from the profusion of acronyms, computing doesn't seem to create many new terms, mutuating them from other fields. Tennis, on the other hand, has a more varied mix of terms, some of which are fairly generic (*net*, *out*), while others are more specific or, in any case, not in widespread use outside the tennis milieu (*back-hand*, *fifteen-love*, *setpoint*).

ii) Computing has diluted its identity due to the prevalence of business practices in its milieu, so many news stories that contain a predominantly computing terminology are in reality business articles and, therefore, irrelevant for the purposes of our test. So, the context of computing is marred by a considerable ambiguity, and it is interesting to see how the context system will be able to deal with it. Articles on tennis, on the other hand, seem to be more technical: an article about tennis players is almost always about tennis (this is not always true, and it is not true for all sports; at the time of this writing, for instance, a similar assertion could not be made about golf, due to perduring gossip about a known golf player) and, in this sense, the *tennis* context is less ambiguous than the computing.

We have used the method outlined in the previous section to create a *tennis* and a *computing* latent semantic manifold. We have then collected a set of about 2600 news stories from the streaming servers of non-specialized newspapers and news services. We have not concentrated on the topics of the contexts, but have made sure that we collected a wide range of different news. We have then manually classified the news as being of interest for computing[3] or for tennis. Out of the 2600 stories, about 500 were relevant for the computing context and about 400 for the tennis context.

For each context we have executed two measures: one using only the description that came with the specification of the story, the other using the link that came with the story to get the whole text (in the graphs these two test modalities are indicated as *full html* and *description only*). In the case of full html, we have eliminated the stories whose pages contained less than 200 significant words. We did this in order to eliminate video news. A story that consisted in a video resulted in a page with no story-related information but that often contained links to other "hot stories." In some cases, the text that accompanied these links contained significant words (e.g. together with a video completely unrelated with tennis, there could be a link to some tennis news marked with words related to tennis), giving the page a high rank. With the 200 words limit we assure that only pages containing a story related to the streaming element were retained. We sorted the news for relevance and we determined the recall of the first n results as a function of n.

The results are reported in fig. 4 for *computing* and in fig. 4 for *tennis*. The straight

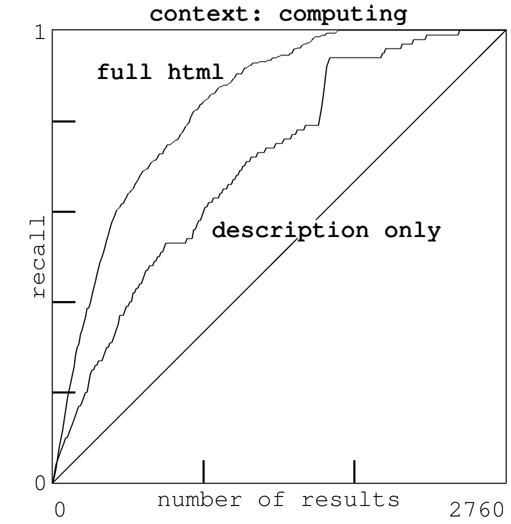

Fig. 4. Recall of the first n results as a function of n; computing context.

[3] In the rather restricted sense that we have mentioned in point i) above: a news story about business was not considered relevant, even if it talked about the computing business. Only news with some technical information about computing were considered relevant.

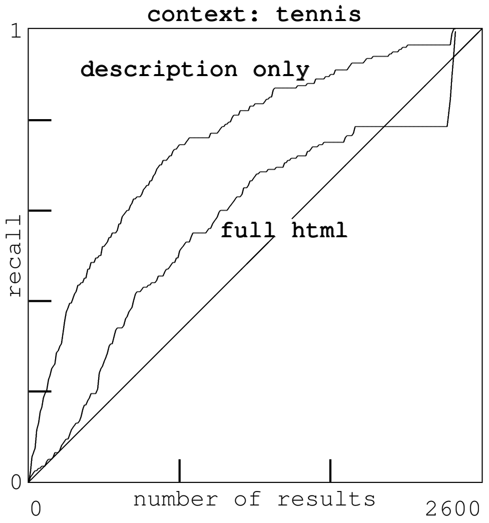

Fig. 5. Recall of the first n results as a function of n; tennis context.

line represents the reference recall, the one we would have by ordering the news randomly.

The comparison of the two figures reveals a curious inversion: in the case of computing, the full html yields better results than the description only, while in the case of tennis the description yields better results than the full html text. We argue that this difference is due to the different style of news reporting. In the case of tennis, like in the case of other sports, the summaries that one finds in the description are often very technical and therefore very discriminative, while in the case of computing the descriptions are more generic due, in part, to the confusion between computing and business that we have mentioned and to the compenetration of the two jargons.

A further curiosity is the bizarre plateau taht we observe in the case of the tennis context with full html, extending from about 1600 to 2400 results. Between 200 and 2500 results this plateau actually makes context perform worse than random! There seems to be a large group of irrelevant news with enough misleading context as to be bumped in front of the following group of relevant news. We haven't observed this phenomenon in any other test, and its causes are, as of yet, unknown.

5 Conclusions

In this paper we have presented a method that uses the context of a person's activity (to the extent that this activity is recorder on that person's computer) to filter potentially interesting news. We have shown how the basic context representation technique developed in [8] can be modified and extended to deal with the present case, and how the *latent semantic manifold* representation can be used as an instrument to filter the stream of news.

We have presented examples of applications to two test domains: computing article and sports articles about tennis, showing the potential of the method.

Acknowledgments

The authors were supported in part by the Min-isterio de Educación y Ciéncia under the grant N. MEC TIN2008-06566-C04-02, Information Retrieval on different media based on multidimensional models: relevance, novelty, personalization and context.

References

1. Scott Deerwester, Susan T. Dumais, George W, Furnas, Thomas K, Landauer, and Richard Harshman. Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, 41(6):391–407, 2000.
2. Hans-Georg Gadamer. *Truth and method*. London, 1975.
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. T. Kohonen. *Self-organizing maps*. Heidelberg, Berlin, New York:Springer-Verlag, 2001.
5. C. Manning, P. Raghavan, and H. Schütze. *An introduction to information retrieval*. Cambridge University Press, 2009.
6. Miranda Lee Pao. *Concepts of Information Retrieval*. Libraries Unlimited, Englewood, Colo, 1989.
7. Simone Santini. The self-organizing field. *IEEE Transactions on Neural Networks*, 7(6):1415–23, 1996.
8. Simone Santini and Alexandra Dumitrescu. Context as a non-ontological determinant of semantics. In *Proceedings of the 3rd International conference on Semantics and digital media technologies*, 2008.
9. S. K. M. Wong Wong, Wojciech Ziarko, and Patrick C. N Wong. Generalized vector spaces model in information retrieval. In AMC Press New York, editor, *Proceedings of the 8th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 18–25, 1985.