2014 IEEE/WIC/ACM International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)

Dynamic learning of keyword-based preferences for news recommendation

Antonio Moreno, Lucas Marin, David Isern, David Perelló
Department of Computer Science and Mathematics, Universitat Rovira i Virgili, ITAKA Research Group
Av. Països Catalans, 26, 43007 Tarragona, Catalonia (Spain)
antonio.moreno@urv.cat, lucas.marin@urv.cat, david.isern@urv.cat, david.perello@estudiants.urv.cat

***Abstract* — The accurate recommendation of daily news requires a detailed knowledge of the topics of interest to the user. The dynamic and continuous analysis of the content of the news that are read (or ignored) by the user every day may lead to the automatic, unsupervised and non-intrusive learning of the positive (and negative) preferences of the user with respect to a set of keywords. These preferences may then be used to rank the daily news, so that the user is recommended those items that match better with his/her interests. The cyclic preference learning methodology described in this paper is illustrated with a case example based on real news from the British newspaper *The Guardian*, in which promising results have been obtained.**

Keywords—recommender systems; user profile; preference learning; profile adaptation

I. Introduction

In the current Information Society users are constantly confronted with situations in which they must choose, from a set of possible alternatives, the subset of those options that are more relevant for them. *Recommender systems* (RSs, [18]) aim to support users in their decision making processes, by making an automated analysis of the available items, ranking them and filtering those that are more relevant for the user. In order to provide accurate recommendations, RSs need to know with the maximum precision the interests or preferences of the user, that are usually stored in the *user profile*. Preferences may be *explicitly* declared by the user when he/she starts to use the recommender system (*e.g.* by filling in a questionnaire [15]), or they may be *learned* automatically by the system in an implicit way, by analysing the interaction of the user with the recommended items (*e.g.* taking into account those items that the user selects or discards) [3].

Implicit learning has several advantages, the most relevant being that it does not require any explicit effort from the user (most users are actually reluctant to give away their preferences explicitly [13]) and that it may adapt the information on the user profile to dynamic changes on the user's preferences [8]. However, most of the previous works on implicit preference learning are based on a single domain and they are hardly reusable in other areas, since they include in the preference learning algorithm specific knowledge on the influence that each user action on the system should have on the information contained in his/her profile [2, 5, 19].

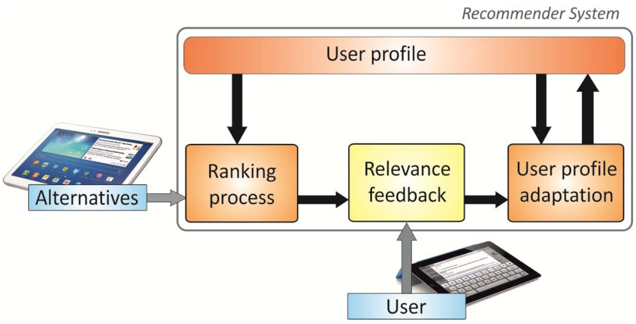

Fig. 1. Recommender system architecture

The preference learning method presented in this work is based on the architecture shown in Figure 1, which follows a cyclic procedure. Given a set of alternatives (in this case, news) the information on the user profile is used to evaluate and order them. The user is then shown a list of ranked alternatives. When the user decides to read one of the news, the system can use that information to update automatically its knowledge about the user's preferences. The basic intuitive idea, as will be described below in more detail, is that the system should increase the preference associated to the basic keywords of the news selected by the user, whereas it should decrease the preference associated to the keywords of those news that were located above the selected option in the ranked list of news.

This idea was already employed successfully in the automatic learning of preferences when the domain items were represented with numerical and/or categorical attributes [10]. However, in this paper we focus on the case in which the items to be recommended are purely textual. The proposed methodology is applied to news, but it does not use any kind of domain specific knowledge and it could be applied to any set of textual items, such as blog entries, tweets, Facebook messages, electronic mails, scientific papers, etc.

Thus, the main contributions of the paper can be summarized as follows:

- The definition of a novel procedure for the automatic implicit learning of the positive and negative preferences of the user with respect to a dynamic set of keywords.
- The successful application of this technique to the recommendation of news.

978-1-4799-4143-8/14 $31.00 © 2014 IEEE
DOI 10.1109/WI-IAT.2014.55

The rest of the paper is structured as follows. The next section comments briefly some recent works on news recommendation. Section 3 explains how the user profile is represented, how a textual item (a news) is represented as a set of keywords and how each alternative is evaluated taking into account the user's preferences. The following section explains how the information in the user profile evolves dynamically as the system analyses the items that are read (or ignored) by the user. Section 5 describes the simulator that has been developed to evaluate the system and the results that have been obtained in a specific case study with news from *The Guardian*. The final section summarizes the contributions of the paper, describes some of its limitations and outlines points of future work.

II. RELATED WORK

In this section we will briefly comment some recent proposals on news recommendation. In some cases, as in the work presented in this paper, the recommender system constructs a keyword-based user profile, and it makes a content-based analysis by comparing each news with the profile to decide whether to show it to the user. For instance, Li et al. [4] define a two-leveled user profile, with long-term and short-term reading preferences. The former are supposed to refer to general categories (*e.g.* sport) and be quite stable, whereas the latter may change more quickly in order to have a more refined view of the current interests of the user (*e.g.* a particular sport event with a certain temporal duration). Long-term preferences are used to select sets of news, and short-term preferences choose the items to be finally recommended within these sets. TF-IDF, as suggested in this paper, is a popular method to identify the most relevant keywords of each news [5, 21, 22]. For example, in the PNFS system [22] both user profiles and news are represented as sets of words, and content-based recommendation techniques based on k-nearest neighbor and Naïve Bayes are employed.

Unlike the single-user approach considered in this paper, other works propose the use of collaborative filtering methods, taking into account the reading behavior of a group of users. For instance, Baraglia et al. [1] suggest a P2P-based system, in which there isn't a single user profile per node but the information is composed by the intersection of URLs visited by at least two members of the group. Medo et al. [12] propose a framework based on how news spread in online systems. The system recommends to users the most suitable leader candidates (information sources) by analyzing their historical reading records and their implicit ratings [20]. The goal of this collaborative system is to maintain a set of leaders for each user. They use an agent-based framework to simulate the emergent behavior of entities. However, as noticed in [6], these adaptive models require detailed information of the user's reading behaviors, named approval or disapproval of the news, and without this information the accuracy of the system decreases. The PENETRATE recommendation system [23] constructs groups of users with similar interests (reading histories) and builds a hierarchy of news for each group, from which, with the aid of personal information, a set of news is selected for recommendation to each particular user.

Finally, it is also possible to find hybrid news recommenders that combine in some way the content-based and collaborative approaches. For example, the Premise system [1] combines a probabilistic analysis of the content of each news with the collaborative information of the opinions of trusted social experts. Wen et al. [21] propose another hybrid recommender system that combines explicit and implicit information collected from users. The system has been designed as a dispatcher where a pool of news is delivered to the most interested users. The feedback of the users and the information collected from trusted users is used to maintain the user profiles. Another example is Wesomender [14], a recommender system for journalists that integrates a content-based (that takes into account factors such as time, location, content and reliability) and a collaborative filtering component. Recently, Mourão, et al. [16] presented a recommender system that uses features such as popularity, recency and similarity of recommended items crawled from the user and other users.

In general, most of the news recommenders proposed in the literature in the last years tend to employ collaborative filtering, whereas in this paper we want to focus on the single-user case, as we intend to make a completely personalised recommender. We also want to minimize the interaction of the user with the system, so we don't base our preference learning algorithms on explicit ratings, but just on the observation of the news that are read by the user. Moreover, our approach is completely applicable to any situation in which there is only textual information about the recommendable items (not just news) and it combines information provided by the content of the documents (main keywords) with the preferential information provided by the user's feedback.

III. USER PROFILE MANAGEMENT

A. Structure of the user profile

In this work it has been decided to store in the user profile a numeric value of positive (or negative) preference of the user with respect to a dynamic set of keywords. The user profile is initially empty, and, as it will be described later, keywords are added as the algorithm analyses over time the interaction of the user with the recommended news.

The preference associated to each word is a value in the range [-100,100], where -100 represents the situation in which the user is not interested at all in reading news about that topic and 100 the case in which the user is very interested in news related to that keyword. When a new word is added to the profile its initial preference value is 0.

B. Keyword-based representation of news

As shown in figure 1, the recommender system takes into consideration a set of alternatives (news), and it must be able to compare the content of each of them with the information about the user's preferences. In this work we have employed one of the most common keyword-based representations of news, based on the use of the well-known TF-IDF (*Term Frequency – Inverse Document Frequency*) measure to extract the most relevant terms of each news [17].

The TF-IDF measure reflects how important is a particular word of a certain document within a given collection or corpus. It is often used as a weighting factor in information retrieval and text mining tasks. The TF-IDF value increases proportionally to the number of times a word appears in the document, but is offset by the overall frequency of the word in the corpus, which helps to control the fact that some words are more common than others.

The term frequency (TF, equation 1) is the frequency of occurrence of the term t in the document d. It measures the ratio between the occurrences of t in d (*occur(t,d)*) with respect to the total number of words of document d (*words(d)*).

$$TF(t,d) = \frac{occur(t,d)}{words(d)} \tag{1}$$

The *inverse document frequency* (IDF, equation 2) is a measure of whether the term is common or rare across all the documents of the corpus. It is obtained by dividing the total number of documents by the number of documents containing the term, and then taking the logarithm of that quotient.

$$IDF(t,\mathrm{D}) = \log\left(\frac{|D|}{|\{d \in D : t \in d\}|}\right) \tag{2}$$

Finally, the *TF_IDF* measure of the relevance of the term t of document d within the set of documents D is obtained as follows:

$$TF\_IDF(t,d,D) = TF(t,d) * IDF(t,D) \tag{3}$$

Using this measure over all the words of a document it is possible to determine which of them have a higher relevance. The number of representative words used per document is one of the parameters of the algorithm.

C. Evaluation of an alternative

The first task of the system, as shown in Figure 1, is to rank a set of alternatives (the news that can be potentially recommended to the user) taking into account the information about the user's preferences stored in the user profile. As has been described in the previous sections, a news is represented as a set of keywords, and the preferences are represented as a set of pairs (keyword k, preference v_k in [-100.. 100]). The numerical evaluation of each document d (represented with n keywords) with respect to the user profile P is performed as follows:

$$V(d,P) = \sum_{i=1}^{n} f(v_i,P) \text{, where } f(w,P) = \begin{cases} v_w \text{ if } w \in P \\ 0 \text{ if } w \notin P \end{cases} \tag{4}$$

The valuation (V) is the aggregation of all the preference values of the keywords of the news that appear in the user profile (P). The keywords that do not appear in the user profile are given a neutral evaluation of 0. The function f checks if a term w is contained in the profile and, if that is the case, it returns its associated preference (v_w).

IV. Automatic Learning of User Interests

After having evaluated and ranked all the news with equation (4), the process of updating the keyword-based preference in the user profile is conducted on two steps, described in the following subsections:

- The first one involves gathering information from the user about his/her interests. In this step the system receives an implicit relevance feedback, which is the alternative selected by the user from the sorted list of options (the news actually read by the user).
- In the second step the adaptation algorithm analyzes this selection and decides which changes have to be made to the user profile.

A. Relevance feedback

Our approach uses implicit feedback to obtain information about the user interests. When the user asks for a recommendation and finally selects one as his/her favorite, two pieces of implicit feedback are extracted from this action: the alternative the user selected and the alternatives that were incorrectly ranked by the system above the selection.

The example shown in Table 1 represents a set of alternatives sorted by the RS and shown to the user. As an example, the fifth alternative represents the user's final selection. The information obtained from this interaction with the user that will be used by the adaptation algorithm is the set of the first four alternatives (which we call *over-ranked* alternatives) and the fifth alternative, which was the one preferred by the user.

TABLE I. Ranked list of options and user selection

Pos.	**Keywords**						**User selection**
1	West Indies	England	Wood	Gale	T20	Wicket	
2	Australia	South Africa	Warner	Wicket	Harris	T20	
3	Phone	Market	Android	Company	UK	Technology	
4	Music	Rock	Album	Band	Year	Concert	
5	**Captain**	**South Africa**	**Career**	**Smith**	**World Cup**	**Wicket**	●
			…				
N	Minister	Politic	Jones	UK	News	Government	

The intuitive idea, as shown in figure 2, is that the preference on the keywords of the selected alternative should be increased, whereas the degree of preference associated to the keywords of the over-ranked alternatives should be decreased. The alternatives that were ranked below the selected one are not considered by the preference learning algorithm.

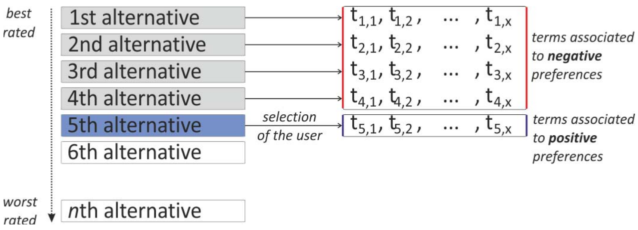

Fig. 2. Intuitive idea of the preference learning algorithm

B. *Adaptation of the user profile*

In this final step the user profile is changed taking into account the keywords that appear in the selected and the over-ranked alternatives. Keep in mind that user preferences can evolve over time and therefore the system must be able to adapt dynamically the user profile. For example, a user can have a strong interest in subjects related to sports, but due to personal circumstances it may change this interest by other interests such as politics or society.

The adapting process can increase or decrease the degree of interest in a term, depending on whether the term appears in the selected news and/or in one (or several) of the over-ranked alternatives (news that have been valued too highly by the RS, but have not been read by the user, showing his/her lack of interest on them).

Each keyword v that appears in the selected news and/or in the over-ranked options is evaluated with the following function, in which *freq(v)* is the number of times that v appears in the selection and the over-ranked alternatives, and K is the number of over-ranked alternatives:

$$rank(v) = \left(\frac{freq(v)}{K}\right) * 100 \qquad (5)$$

The value K is always larger than 0, because preferences are updated only if there is at least one over-ranked alternative. If the selection of the user is the first option, the recommender system is assumed to have rated the news correctly, so the preferences should not be updated.

The result of Eq. (5) is a value between 0 and 100 per each term appearing in the selection and/or the over-ranked news. On this process, a term can be in one of these three situations:

- A term may appear only in the selected alternative (e.g. “Smith”). In this case, its preference should be increased. The more over-ranked alternatives there are, the bigger should be the increase.
- A term may appear (one or several times) only in the over-ranked alternatives (e.g. “Australia”, “T20”). In this case, its preference should be decreased; the more times the term appears, the bigger should be its decrease.
- A term may appear both in the selected news and in one (or several) of the over-ranked alternatives (e.g. “Wicket”, “South Africa”). In this case it has been decided to prioritize the selection of the user and to increase the preference value. However, the more times the term appears in the over-ranked items, the smaller will be the increase.

These updating decisions have a deep impact on the results. The first one is straightforward, because we aim to prioritize the selections made by the user (we assume that the keywords they contain are interesting). The rationale of the second one is that, if we receive again the keywords of the over-ranked alternatives in the future, they should get a lower evaluation and not be recommended to the user. The last decision merges both assumptions and we designed a mechanism to use both positive (selection) and negative (over-ranked) information.

The amount of increase/decrease of each term has been decided empirically, and it is shown in Figure 3. These figures could be different in other domains. At the top, the blue lines show how the preference is increased in the first and third cases commented above. At the bottom, the red lines show how the preference is decreased for those terms that appear only in the over-ranked alternatives.

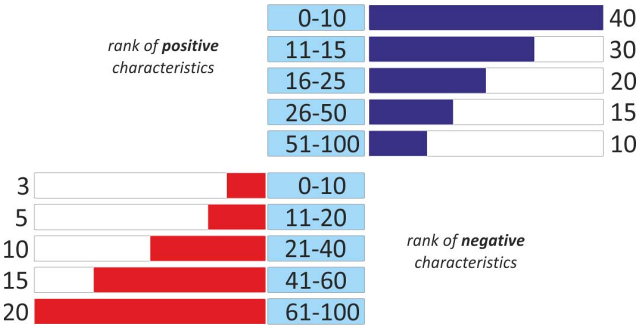

Fig. 3. Rules to calculate the evolution of the user’s preferences

Let us consider some examples of each of the three cases (see Figure 1):

- The keyword “Smith” only appears in the selected option, and it has a rank (1/4)*100 = 25. Thus, according to Figure 3 (blue side), its preference level would be increased in 20 points.
- The keyword “Australia” appears in the over-ranked alternatives (only once). Therefore, its rank will also be (1/4)*100 = 25 and, looking at Figure 3 (red side) we can see that its preference value would be decreased in 10 points. However, the term “T20” appears twice in the over-ranked alternatives, so it would have a rank of (2/4)*100=50 and its preference value would suffer a decrease of 15 points.
- Finally, we can analyze terms that appear both in the selection in the over-ranked alternatives. “Wicket” is in the selection and in 2 over-ranked alternatives, so it has a rank (3/4)*100=75, and its preference would be increased in only 10 points. “South Africa” is in the selection and in one over-ranked option, so its rank would be (2/4)*100=50 and its preference increase would be higher (15 points).

If one of these terms did not appear yet in the user profile, it would be added with an initial preference value of 0 and then the first increase/decrease would be applied.

V. EVALUATION AND RESULTS

In order to evaluate the accuracy of the preference learning mechanism introduced in the previous section and to analyse the influence of its parameters, a test using news from the British newspaper *The Guardian* has been conducted. A simulator has been implemented in Java to perform this experiment. This section begins with a description of the domain in which the test has taken place; after that, the simulator is described and the results of the evaluation of the learning algorithm and the analysis of its parameters are presented.

A. Work domain

The tests have been performed through the analysis of news from *The Guardian* newspaper through its open platform free-to-use API (http://www.theguardian.com/open-platform). The API permits to retrieve news associated to different sections of the newspaper.

A set of 6000 news belonging to 10 different sections (sport, football, technology, music, film, politics, science, society, education and media) was obtained. It may be observed that some sections are quite interrelated (*e.g.* sport and football), whereas others seem independent (*e.g.* film and science). In this way, it is possible to study a situation in which different sections may be interrelated and analyze whether this fact hinders the preference learning algorithm.

B. Simulator of long-term user interaction

The simulator permits to analyse how the user preferences would change over time as he/she reads some of the news (those on which he/she is more interested) and ignores others (see Figure 4).

The input parameters of the algorithm are the corpus of q alternatives (in this case, the set of 6000 news), the ideal user profile I that the system should learn (*i.e.*, the real preferences of the user), the maximum number n of iterations (or "reading simulations") to be executed and the number k of alternatives (news) to be considered in each iteration. Starting with an empty user profile, the steps to be taken in each iteration (until the news have exhausted or the maximum number of iterations has been reached) are the following:

- Select the next k news from the corpus.
- Order those k alternatives using the information in the user profile (this is the order in which these news would be shown to the user).
- Simulate the reading selection of the real user, by computing, from this list of k alternatives, which is the one that is more similar to the ideal profile I (the one with a higher evaluation over I using the function V described in equation 4). Let us say this selected alternative is in position p of the ordered list.
- Take the selected alternative and the over-ranked ones (the news in positions 1 to p-1) and apply the adaptation algorithm described in the previous section to update the user profile.

In order to evaluate the performance of the learning algorithm, the simulator calculates after each iteration which is the distance between the current user profile P and the ideal profile I is as follows:

$$dist(P,I) = \frac{1}{n}\sum_{j=1}^{n}\frac{\left|f(w_j,P) - f(w_j,I)\right|}{r} \qquad (6)$$

In this equation n is the number of words contained in the current user profile P, and $f(w_j,P)$ and $f(w_j,I)$ are the preference values for word w_j according to the current and the ideal profiles respectively. If a word is not contained in I its preference is 0. The constant r is the range of the preferences, which in this case is 200 (since the preference values go from -100 to 100).

The distance between these two profiles tends to 0 when most of the keywords of the ideal profile I appear in the current profile P and they have a similar level of (positive or negative) preference. The maximum distance (1) would be achieved if all the keywords in I have a maximum positive (or negative preference) and they appear in P with the opposite maximum preference. This function is commutative, as $dist(P,I)$ = $dist(I,P)$.

```
ADAPTATION-ALGORITHM-EVALUATION(
E(a^0,...,a^q), //corpus of alternatives
n, //iterations to simulate
I, //ideal profile
k //alternatives per iteration
)
begin
P=initial-profile();  // current profile
d=empty-list();  // evolution of distance
beg, end, iter =0;
while ( iter < n) do
  beg=iter*k;
  end=(beg+k)-1;
  R(a^0,...,a^k)=rating-and-ranking(P, E(a^beg,...,a^end));
  p=calculate-ideal-selection(R, I);
  posVal=calculate-positive-values(R, p);
  negVal=calculate-negative-values(R, p);
  P=apply-changes(P, pVal, nVal);
  d=add(dist(P, I)); // d stores the evolution
  iter=iter+1;
end while;
```

Fig. 4. Pseudocode of the simulator

C. Results

All the results reported in this section correspond to the average behaviour of 10 runs of the simulation with the same ideal profile but different random initial profiles. Figure 5 shows the general performance of the adaptation algorithm in a setting in which the 6000 news were treated in 400 iterations (15 news per iteration), considering 40 keywords/news. It depicts the distance between the current profile and the ideal profile, separating the analysis of the words in the profile obtained through the analysis of news of different sections. Six of the ten sections are shown in the figure. The parameters used in this simulation are the ones which gave the best results in the parameter's analysis, described below.

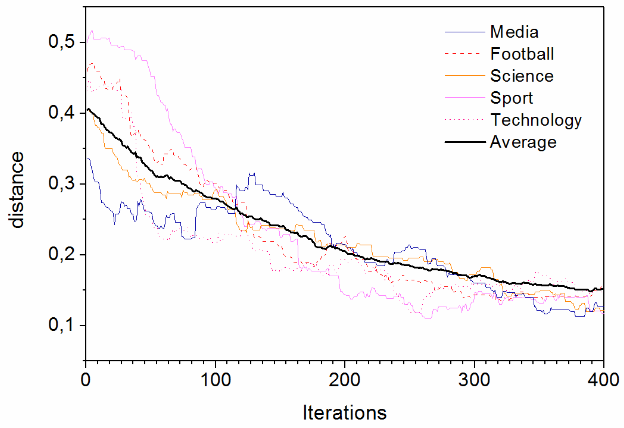

Fig. 5. General performance of the algorithm

The average distance among all the sections shows the general behaviour of the evolution, and it can be noticed that the distance between the profiles gradually decreases through the iterations. It can be observed that the learning rate is greater in the first iterations and it decreases until a quite stable state is reached. Concretely, the initial average distance, that is greater than 0.4, is reduced to approximately 0.3 in around 50 iterations, to 0.2 in 200 iterations and to 0.15 in 400 iterations. The figure also shows the learning behaviour for each section. The preferences on the keywords associated to Sport and Football seem to be learnt very quickly, whereas the preferences of other sections (*e.g.* Media) take a longer time to learn. However, after 400 iterations the result for each section is very similar.

The distance between the current and the ideal profiles gives a general evaluation of the performance of the adaptation algorithm. Another interesting result is provided by the analysis of the position of the selected alternative in each iteration (1^{st}-15^{th}).

Figure 6 shows the position of the chosen alternative (the one that is closer to the ideal profile) in the first 50 iterations and in the last 50 iterations. In the first case, in more than 60% of the cases the best alternative was not located in the 5 first positions; thus, the recommender was not rating the set of 15 news appropriately (or, at least, it was not detecting adequately the best option). This result was expected, as the simulator starts with an empty profile without any information about the user's preferences. However, it may be seen that when the last 50 iterations (351-400) are reached and the ideal user profile has been learnt, in 23 iterations (46%) the best option was the first one, and in 40 iterations (80%) the best option was among the first three alternatives. Only in 7 of those last 50 iterations (14%) the news that is more interesting for the user had not been rated among the first five items.

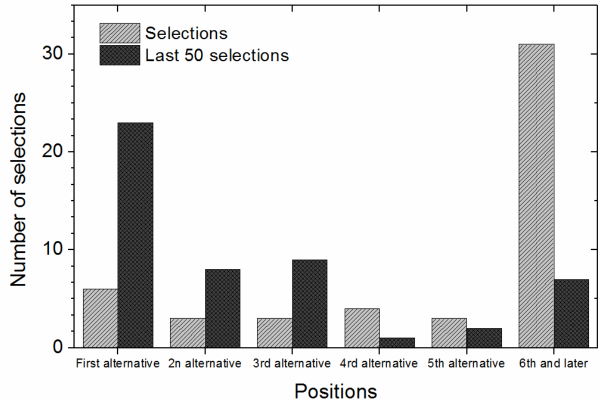

Fig. 6. Analysis of the positions of the alternatives selected by the user

The proposed implicit adaptation algorithm introduces several parameters that should be properly customised. The rest of this section explains the influence of these parameters in the final result.

1) Number of terms per alternative

In this test we analyse how the distance between the current and the ideal profile evolves if different number of keywords/news are considered. Figure 7 shows the results obtained with 20, 30, 40 or 50 keywords per news.

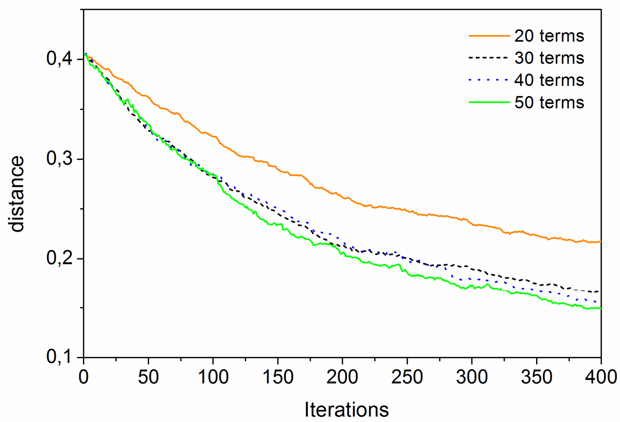

Fig. 7. System performance depending on the number of terms per alternative considered

In general, it may be said that, the more terms are considered, the better is the learning process (the system can probably make a more fine grained analysis of each news and it also has more detailed information on the user's preferences). It may be noticed that the performance when 20 keywords/news is clearly worse than the others, but there is not much difference between 30, 40 and 50. It should also be taken into account that the running time of the preference learning algorithm and the usage of memory will increase if more keywords are considered. After this analysis we chose to consider 40 keywords/news, reaching a compromise between learning quality and computational cost.

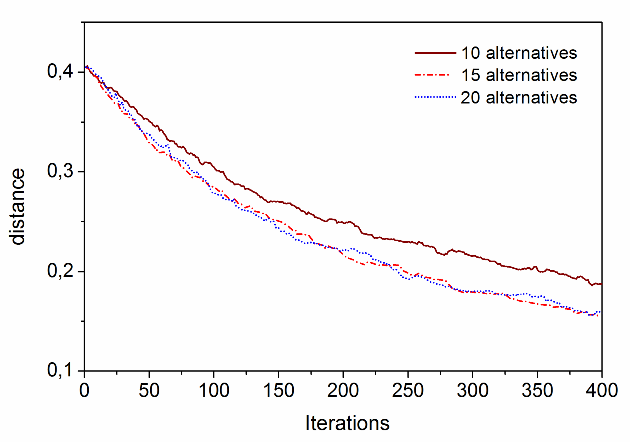

Fig. 8. System performance depending on the number of alternatives considered per iteration

2) Number of alternatives considered in each iteration

Another important element is the number k of proposals shown to the user in each iteration. This set is used to infer the changes to be made on the current user's profile.

Figure 8 shows the performance of the learning algorithm when 10, 15 or 20 alternatives are considered in each iteration. If only 10 alternatives are taken, the result is worse than if 15 or 20 alternatives are analysed. In these two latter cases, the results are almost identical. In our tests, as commented above, we decided to consider 15 alternatives per iteration.

3) Distribution of alternatives

Finally, it was also analysed if the order in which the alternatives are presented to the system affects the result. Two approaches were compared: the sequential one described above (in which the 6000 news are sequentially presented to the system, without any repetition, in groups in which the amount of news of each section is approximately balanced) and a random one in which 15 news are randomly selected from the corpus of 6000 news in each iteration.

Figure 9 shows that there is basically no difference between both approaches, so it seems that it is not necessary to pre-process the corpus and order it in a specific way in order to achieve good final results.

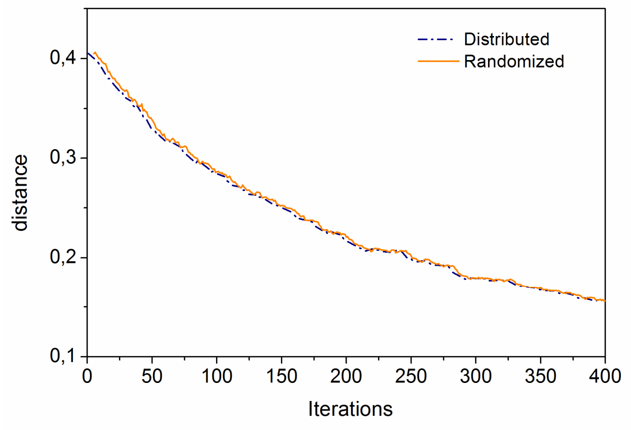

Fig. 9. System performance depending on the distribution of alternatives

VI. CONCLUSIONS AND FUTURE WORK

This works proposes a cyclic process to improve automatically and gradually the accuracy of the recommendations given to users by a news service. The first step rates and ranks the available news using the knowledge about the user's preferences stored in the user profile. Both news and preferences are represented by textual keywords. In the second step, the system analyses the news read by the user and infers in an automatic and unsupervised fashion the changes to be applied in the user profile to represent better the user's interests (without any explicit declaration of preferences).

The system has been tested and evaluated with real news from the British newspaper *The Guardian*, obtaining encouraging results. The preference learning system is totally generic and domain-independent, and it could be applied without any change in any situation in which the objects to be recommended are purely textual (blog entries, tweets, scientific papers, etc.). The stronger requirement of the learning methodology presented in this paper is that it should be applied to situations in which there is a continuous feedback of the user, which permits to learn quite accurately his/her preferences with a relatively small number of iterations. The weakest point of the system, which is the main object of our lines of future work, is its purely syntactic treatment of keywords.

Unlike more complex preference learning systems, that can learn the preferences of the users with respect to numerical and categorical values [7, 9, 10], in this work we have focused on a keyword-based treatment of news, choosing the most relevant terms of each news with the well-known TF-IDF measure, as usual in the literature of news recommenders [5, 21, 22]. In future work, the possibility of making a better (*i.e.*, more semantic) selection and treatment of the relevant terms of a news will be explored. For example, it could be possible to take the more frequent keywords and apply ontology-based semantic similarity measures [11] to obtain WordNet concepts that represent the topics of the news in a more generic way (*e.g.* from the keywords football, tennis and golf it could be inferred that the user is interested in sports). It could even be possible to introduce a hierarchical representation of preferences (for example, football could be a subclass of sports, and we could store different levels of preferences for these two concepts). As shown on section IV.A the final variation of the preferences (increase or decrease) is being calculated by three rules that apply a fixed set of values (see Figure 3). Even though this mechanism was designed to control the evolution of the profile, these values could be replaced by a function which receives the actual preference of the keyword and evaluates the change to apply. This function could be set up according to general policies (*e.g.* conservative, risky, etc.).

Another line of future work is the appropriate treatment of synonyms (football, soccer) and polysemic (bank, operation) words. We also envisage a more complex management of the user profile, for instance introducing a temporal declining factor on preferences so that they tend to zero (and are eventually removed) if they have not influenced the user choices in a certain period of time.

ACKNOWLEDGMENTS

This work has been supported by the Spanish Ministry of Science and Innovation (SHADE project: Semantic and Hierarchical Attributed in Decision Making TIN2012-34369). L. Marin was supported by a URV pre-doctoral research grant.

REFERENCES

[1] R. Baraglia, P. Dazzi, M. Mordacchini, and L. Ricci, "A peer-to-peer recommender system for self-emerging user communities based on gossip overlays," *J. Comp. and Syst, Sci. Int.*, vol. 79, pp. 291-308, 2013.

[2] E. Cheng, F. Jing, M. Li, W. Ma, and H. Jin, "Using Implicit Relevance Feedback to Advance Web Image Search," *In Proc. of IEEE International Conference on Multimedia and Expo, ICME 2006*, Toronto, ON, Canada 2006, pp. 1773-1776.

[3] G. Jawaheer, M. Szomszor, and P. Kostkova, "Comparison of Implicit and Explicit Feedback from an Online Music Recommendation Service," *In Proc. of 1st International Workshop on Information Heterogeneity and Fusion in Recommender Systems*, Chicago, US, 2010, pp. 47-51.

[4] L. Li, L. Zheng, F. Yang, and T. Li, "Modeling and broadening temporal user interest in personalized news recommendation," *Expert Syst. Appl.*, vol. 41, pp. 3168-3177, 2014.

[5] C. Lin, R. Xie, X. Guan, L. Li, and T. Li, "Personalized news recommendation via implicit social experts," *Inf. Sci.*, pp. 1-18, 2014.

[6] H. Liu, F. Yu, A. Zeng, and L. Lü, "Recommendation of Leaders in Online Social Systems," *In Proc. of 20th International Symposium on Methodologies for Intelligent Systems, ISMIS 2012*, Macau; China, 2012, pp. 387-396.

[7] L. Marin, D. Isern, and A. Moreno, "Dynamic adaptation of numerical attributes in a user profile," *Appl. Intell.*, vol. 39, pp. 421-437, 2013.

[8] L. Marin, D. Isern, and A. Moreno, "Unsupervised adaptation of the user interests," *In Proc. of 3rd International Conference on Agents and Artificial Intelligence*, Rome, Italy, 2011, pp. 337-342.

[9] L. Marin, D. Isern, A. Moreno, and A. Valls, "On-line dynamic adaptation of fuzzy preferences," *Inf. Sci.*, vol. 220, pp. 5-21, 2013.

[10] L. Marin, A. Moreno, and D. Isern, "Automatic preference learning on numeric and multi-valued categorical attributes," *Know.-Based Syst.*, vol. 56, pp. 201–215, 2014.

[11] S. Martínez, A. Valls, and D. Sánchez, "Semantically-grounded construction of centroids for datasets with textual attributes," *Know.-Based Syst.*, vol. 35, pp. 160–172, 2012.

[12] M. Medo, Y.-C. Zhang, and T. Zhou, "Adaptive model for recommendation of news," *Europhysics Letters*, vol. 88, pp. 38005, 2009.

[13] M. Montaner, B. López, and J. L. de La Rosa, "A taxonomy of recommender agents on the internet," *Artif. Intell. Rev.*, vol. 19, pp. 285-330, 2003.

[14] A. Montes-García, J. M. Álvarez-Rodríguez, J. E. Labra-Gayo, and M. Martínez-Merino, "Towards a journalistic-based news recommendation system: The Wesomender approach," *Expert Syst. Appl.*, vol. 40, pp. 6735-6741, 2013.

[15] A. Moreno, A. Valls, D. Isern, L. Marin, and J. Borràs, "SigTur/E-Destination: Ontology-based personalized recommendation of Tourism and Leisure Activities," *Eng. Appl. Artif. Intell.*, vol. 26, pp. 633–651, 2013.

[16] F. Mourão, L. Rocha, J. A. Konstan, and J. Wagner Meira, "Exploiting non-content preference attributes through hybrid recommendation method," *In Proc. of 7th ACM conference on Recommender systems, RecSys 2013*, Hong Kong, 2013, pp. 177-184.

[17] J. Ramos, "Using TF-IDF to Determine Word Relevance in Document Queries " *In Proc. of First International Conference on Machine Learning*, New Brunswick: NJ, USA, 1994.

[18] P. Resnick and H. R. Varian, "Recommender Systems," *Commun. ACM*, vol. 40, pp. 56-58, 1997.

[19] J. Weakliam, M. Bertolotto, and D. Wilson, "Implicit interaction profiling for recommending spatial content," *In Proc. of 13th ACM International Workshop on Geographic Information Systems, GIS 2005*, Bremen, Germany, 2005, pp. 285-294.

[20] D. Wei, T. Zhou, G. Cimini, P. Wu, W. Liu, and Y.-C. Zhang, "Effective mechanism for social recommendation of news," *Physica A*, vol. 390, pp. 2117-2126, 2011.

[21] H. Wen, L. Fang, and L. Guan, "A hybrid approach for personalized recommendation of news on the Web," *Expert Syst. Appl.*, vol. 39, pp. 5806-5814, 2014.

[22] X. Wu, F. Xie, G. Wu, and W. Ding, "PNFS: personalized web news filtering and summarization," *Int. J. Art. Intell. Tools*, vol. 22, pp. 1360007, 2013.

[23] L. Zheng, L. Li, W. Hong, and T. Li, "PENETRATE: personalized news recommendation using ensemble hierarchical clustering," *Expert Syst. Appl.*, vol. 40, pp. 2127-2136, 2014.