Interest based selection of user generated content for rich multimedia services

Olivier Van Laere, Matthias Strobbe, Samuel Dauwe, Bart Dhoedt, Filip De Turck, Piet Demeester
Ghent University - IBBT - IMEC, Department of Information Technology
Gaston Crommenlaan 8 bus 201, 9050 Gent, Belgium
Matthias.Strobbe@intec.ugent.be

Orlando Verde
Alcatel-Lucent
Research & Innovation
Copernicuslaan 50, 2018 Antwerpen, Belgium
Orlando.Verde@alcatel-lucent.be

Frank Hülsken
Fraunhofer-Institut für Intelligente Analyse und Informationssysteme IAIS
Schloss Birlinghoven, D-53754 Sankt Augustin, Germany
Frank.Huelsken@iais.fraunhofer.de

Abstract

In view of the overwhelming popularity of user generated content, both in terms of production and consumption, new intelligent services are needed to help users finding the content they need and enhance existing services with suitably selected content. In this paper we present a set of algorithms for retrieving content, based on dynamic user profiles and learning capabilities (e.g. based on user feedback). The profile information is used in content searches as well as for assisting the user input analysis process (i.e. speech recognition). To illustrate the approach taken, a rich communication service is presented. Here, the basic service (i.e. voice/video conferencing) is enhanced by showing pictures in real time to the users based on the topic of their conversation and their specific interests.

1 Introduction

During the last few years, we have witnessed the emergence of a whole range of extremely popular web sites hosting user generated content. The amount of content is so overwhelming that users are experiencing more and more problems identifying content matching their interests, so much that these sites could become a victim of their own success. A possible solution for this problem are new intelligent services that take the interests of the user into account for ranking the results of a search in such a way that the user easily finds the content he wants. Approaches following this track are descibed in [5] and [3]. To achieve this goal the metadata attached to the content (usually tags) has to be matched with the user interests and user feedback has to be taken into account carefully to keep the user interests up to date.

In this paper a promising approach for modeling user interests and matching these interests with user generated content is presented (section 2). An enhanced communication use case illustrates applicability in section 3. Simulation results are presented in section 4 and finally future work and conclusions are stated in sections 5 and 6.

2 User Interests Matching

[6] provides an overview of content-based recommendation systems. User profiles consisting of a user model and historic information are presented, together with techniques to learn preferences and to classify content. An approach to modeling user preferences by means of a tree was presented in [4]. They use the tree to rank documents based on the user preferences. However, no user feedback is used in the ranking so the system has no learning capabilities. In this paper, we present the use of a tree for modeling the user preferences combined with the learning of his preferences to recommend content.

2.1 Keyword Tree

In our approach user interests are modeled using a keyword tree with added weight values. Top level keywords represent categories. Lower level keywords represent subcategories and specific interests. A weight value represents the importance of a keyword for a certain user. The sum of the weight values on a specific node level is 1. These weight values are adapted when input or feedback is received from the user. An example of this tree structure is shown in figure 1.

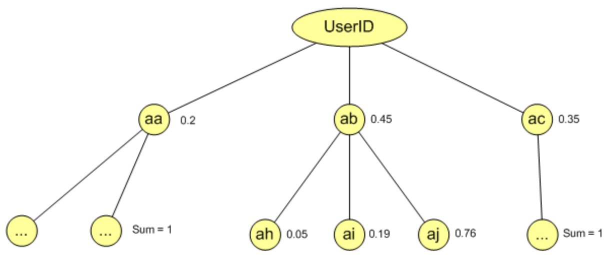


Figure 1. Example of a keyword tree

2.2 Algorithms

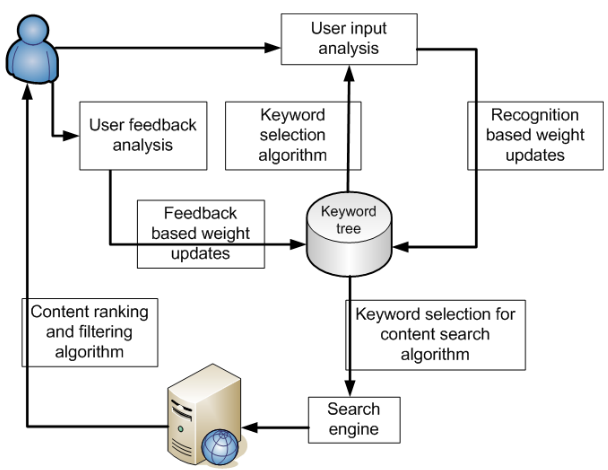


Figure 2. Overview of the interactions between the algorithmic components

Figure 2 shows a general overview of the interactions between the implemented algorithms.

2.2.1 User Input Algorithm

A user can provide input in several ways, by entering search terms, by means of speech or implicitly when talking to another user using VoIP. We assume that the recognized keywords are an indication of the interests of the user. So the weight value of a recognized keyword is increased and the weight values of the siblings are lowered.

To increase the weight value of a keyword we use a logarithmic function (1), with a and b parameters determining the steepness of the function and the maximum value. W_i is the weight value of recognized keyword i.

$$W_{i'} = \log_e(a * W_i + 1)/b \quad (1)$$

The weight values of the siblings (W_s) are renormalized so that the sum of the weights of siblings is 1 at all times.

$$W_{s'} = \frac{(1 - W_i)}{\sum_{siblings} W_j} W_s \quad (2)$$

2.2.2 Keyword Selection for Content Search Algorithm

Based on the recognized keyword, and the ones related to it (e.g. residing in the same category (branch), siblings, children, ...), a selection of search terms is made. The recognized keyword will be included for sure. Related keywords are accounted for if their weight value exceeds a certain threshold value, indicating that the word has been recognized and that it is useful. This set of search terms is then used to search content.

2.2.3 Content Ranking Algorithm

After a search the results are ranked. Ranking is based on the matching between the tags attached to the content and the keywords used for the search. The more tags, attached to a particular piece of content, matching provided keywords and the higher the weight values of these keywords, the higher a result is ranked. Also, the keywords representing the user interest are taken into account.

2.2.4 User Feedback Algorithm

When a number of results are returned to the user, he will typically choose the content he prefers to see in more detail. The tags attached to the content will often match with the keywords of the keyword tree. So, the weight values of these matching keywords will be increased and the siblings lowered. The same formulas as for the User Input Algorithm 2.2.1 are used, but with different parameter values to have bigger increases as user feedback is very valuable information that tells more about someone's interests than a generic search term or the topic of a conversation.

The specific values of the parameters can depend on the kind of user feedback. Suppose the returned results are pictures, then a user can click on a picture to see it in more detail but if he really likes the picture he can also recommend it to somebody else.

2.2.5 Keyword Selection Algorithm

The Keyword Selection Algorithm identifies a relevant subset (the ’current keyword list’) of the keyword tree. This list keeps track of the keywords that can be recognized for a specified user (e.g. by means of speech recognition). The algorithm starts with providing a number of initial keywords. Depending on recognized keywords, related keywords (i.e. the children of the recognized keyword) are added to the subset and keywords with low weight values not belonging to the current topic of the conversation are removed.

When a lot of keywords of a certain branch (e.g. topic, category) are recognized, this might result in a current keyword list containing few keywords from other branches, resulting in a system that is not very adaptive to topic changes. In a basic version of the algorithm, only the top-level keywords from the different branches are always present in the current keyword list. A more advanced version of the algorithm assumes the topic of the conversation has changed when no keywords are recognized for a while and switches in that case to a general keyword list, consisting of high-level keywords from all branches in the tree.

3 Use Case: Content Selection Based on Communication

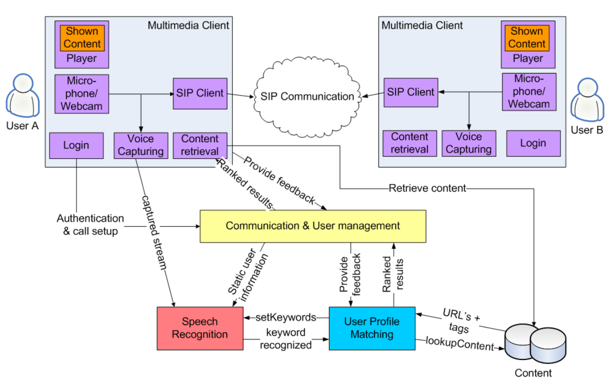


Figure 3. Architecture Content Selection Based on Communication use case

The goal of this use case is to provide users of a multimedia communication client with content that is an added value to their conversation, i.e. pictures about the topic they are discussing that at the same time match with their interests. Figure 3 gives an overview of the architecture of the presented use case. Two users establish a communication session using SIP. Their conversation is captured and redirected to a speech recognizer searching for keywords provided by the User Profile Matching component (via the Keyword Selection Algorithm). When a keyword is recognized the User Profile Matching component is notified and pictures are looked up in Flickr [2] based on the recognized keyword and similar keywords with high weight values. The results are sent to the multimedia clients and presented to the users. A user can click on a suggested picture to see a larger version and he can also recommend a picture to the other user. At that moment the User Profile Matching component is notified of this user feedback and the keyword tree for that user is updated. This use case was developed for the Citizen Media project [1] in collaboration with Alcatel-Lucent Research & Innovation and Fraunhofer-Institut für Intelligente Analyse und Informationssysteme.

4 Evaluation

To evaluate our algorithms a number of simulations were performed. For that purpose arbitrary keyword trees were generated and populated with code words. The construction of the trees was bound to a specified minimum and maximum number of children on each level, but the exact number was randomly chosen between them. The size of the tree was specified and fixed, depending on the experiment. The code words are unique string identifiers generated as all possible combinations of the letters of the alphabet (e.g. ”aa”, ”ab”, ...).

4.1 Keyword Selection Algorithm

In a first series of simulations the Keyword Selection Algorithm was tested for its adaptivity to switch to the current topic of the conversation. Conversations were simulated by generating random keywords from the keyword tree. Consecutive keywords come from the same branch and every 250 keywords a branch switch occurs to simulate a change in the conversation topic. The number of recognized keywords was compared with the case of a randomly filled current keyword list. In that case one can expect for example 10% recognized keywords for a keyword tree of 500 nodes and a current keyword list of size 50.

First, the basic version of the algorithm was tested for different sized keyword trees (from 100 to 1000 nodes) and varying current keyword list sizes (from 10 to 100 keywords). The current keyword list (defined in section 2.2.5) holds the list of keywords that can be recognized for a user. When a conversation swap occurs, this list will dynamically remove the keywords of the old conversation topic, and refill (gradually) with keywords about the new topic. Figure 4 shows the results for three keyword tree sizes.

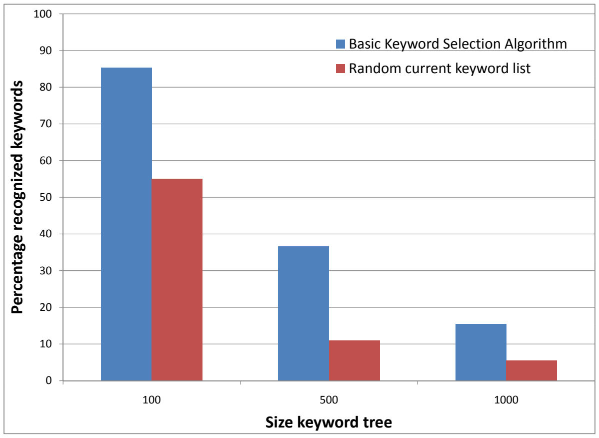


Figure 4. Comparison of the basic keyword selection algorithm and a random current keyword list

The results of the algorithm are quite good in comparison with a randomly filled current keyword list. The improvement is due to the fact that our current keyword list dynamically follows the topic(s) of the conversation, whereas a randomly chosen current keyword list never changes. On average our algorithm recognizes 3,3 times more keywords compared to using a randomly filled current keyword list.

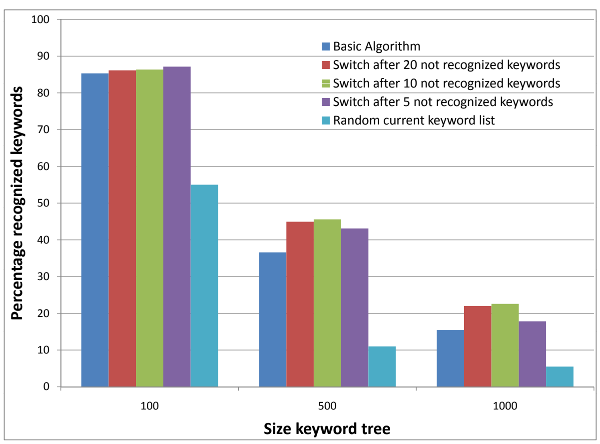


Figure 5. Comparison of the basic and advanced Keyword Selection Algorithm

The same experiment was repeated for the advanced algorithm with recovering to a general current keyword list when no keywords are recognized during a certain period. Figure 5 compares the basic algorithm with three variants of this advanced algorithm (recover after 5, 10 and 20 consecutive unrecognized keywords) and with a randomly filled current keyword list for three keyword tree sizes.

Table 1. Average percentage of recognized keywords

Basic Algorithm	39,57%
Switch after 20 keywords	46,34%
Switch after 10 keywords	46,66%
Switch after 5 keywords	44,43%

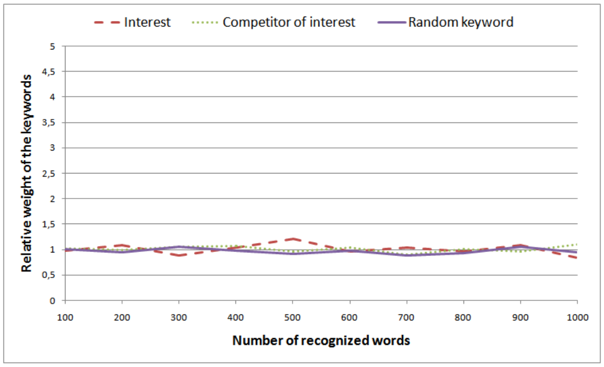
As can be seen on the figure with the advanced version of the algorithm 5 to 10% more keywords are recognized. Only for a keyword tree of 100 nodes the differences are small, but in that case the size of the current keyword list is relatively large so that there are always enough keywords of other categories present. Note that the algorithm can not be too adaptive. The version where the current keyword list is already redirected after only 5 consecutive unrecognized keywords clearly performs worse than the versions with a switch after 10 or 20 keywords. This is also illustrated in table 1. It shows the average percentage of recognized keywords for all performed simulations. In real-time speech recognition, an estimate of the number of unrecognized keywords can be extracted from information about conversations (number of words generally spoken in a certain time interval) and the time the conversation is going on.

The version with a switch after 10 unrecognized keywords now performs 4 times better than an algorithm with a randomly filled current keyword list.

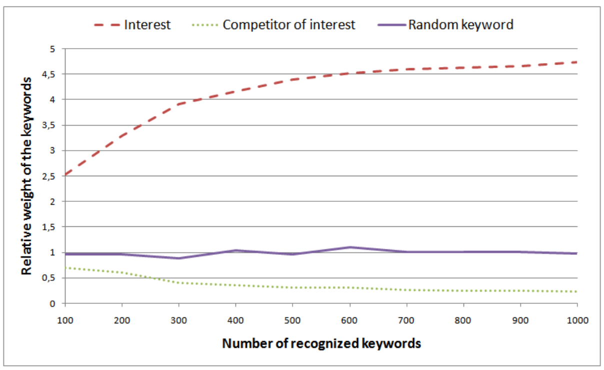
4.2 User Input & Feedback Algorithm

In a second series of simulations, the impact of the (user) feedback on the weight values of the keywords is verified. For this simulation, a keyword tree was used consisting of 200 nodes. We varied the number of words spoken in the conversation starting at 100 words up to 1000 words. Each result is the average value, for the specified number of words spoken, of 25 runs of the experiment. The weight values presented are relative weights (being the actual weight, times the number of children on the level of the keyword). This to be able to compare weight values of nodes in the tree in a correct way. A node having value 0.25 while having 8 siblings, has a better weight value than a node with weight 0.3 while only having one sibling (which means his sibling has 0.7), although this is not visible at first sight. The experiment is performed using just recognition of the keywords, and then repeated using feedback.

For this experiment, we tracked three keywords. First, we monitored the value of a keyword that is of interest to the user (it is in his profile). The second keyword is a keyword that is located on the same tree level as the interest of the user. The third keyword is a randomly chosen one, with the constraint that it is not competing with any interests.



(a) No feedback



(b) Feedback

Figure 6. Comparison of the evolution of keyword weights, without and with feedback.

As can be seen in figure 6(a), the keyword weight values are of the same order of magnitude. This is what we expect, as weight values only change when a word is recognized, and no particular measures are performed on a recognition.

If we take a look at figure 6(b), there is a clear distinction between the keyword representing a user interest (dashed line), and the competing keyword on it's level (dotted line). The random keyword just undergoes it's recognitions, and stays around his normal relative weight value (full line).

5 Future Work

At the moment the structure of the keyword tree is predefined and fixed for all users. In the future the structure of the tree could be adapted and extended based on tags that return often with content consumed by the user.

The algorithms will also be extended to take context information (e.g. location, time of the day, presence info, . . .) into account. For example, when a user is located at his office, he only gets work related results.

The enhanced communication use case could be improved by taking the interests of both involved users (combined profile) into account for suggesting content.

The use case was implemented in a prototype and will be evaluated by users.

6 Conclusion

In this paper we presented a new way to match user generated content with user interests. We used a keyword tree to model the interests of the user in combination with five algorithms to update the keyword tree, incorporate user feedback and select relevant keywords for search and real-time communication services.

A content enhanced communication service was presented to illustrate the potential of the concepts.

To conclude the paper, the algorithms were evaluated through simulations.

Acknowledgment

The research presented in this paper is partially funded by the IST-FP6 Citizen Media project.

Matthias Strobbe is a research assistant of the Fund for Scientific Research - Flanders (F.W.O.-Vlaanderen). Filip De Turck is a postdoctoral Fellow of the Fund for Scientific Research - Flanders (F.W.O.-Vlaanderen).

References

[1] The citizen media project. http://www.ist-citizenmedia.org/.

[2] Flickr API. http://www.flickr.com/services/api/.

[3] B. Rousseau, P. Browne, P. Malone, M. Foghl. User Profiling for Content Personalisation in Information Retrieval. In *Proceedings of the ACM Symposium on Applied Computing*, March 2004.

[4] H. H. Syed, P. Andritsos. A Lightweight Tree Structure to Model User Preferences. In *Proceedings of PersDL*, June 2007.

[5] K. Bradley, R. Rafter, B. Smyth. Case-Based User Profiling for Content Personalisation. In *Proceedings of the International Conference on Adaptive Hy- permedia and Adaptive Web-based Systems*, 2000.

[6] M. J. Pazzani, D. Billsus. Content-based Recommendation Systems. *Lecture Notes in Computer Science*, 4321/2007:325–341, 2007.