Deriving Interests from Keywords

Michael Sonntag
Institute for Information Processing and Microprocessor Technology (FIM)
Johannes Kepler University Linz, Altenbergerstr. 69, A-4040 Linz
sonntag@fim.uni-linz.ac.at

1. Introduction

Currently a new version of the distance education platform WeLearn [3] to be combined with an agent system ([1], also developed locally) for providing advanced services [2] is under development. In a previous stage of the project software was created to automatically derive keywords from documents, either from the different versions of metadata already contained (various Microsoft file formats, PDF, CPS-Manifests according to several specifications, HTML) or from plain text. The next stage is deriving a set of keywords with an associated interest measure for all persons (separately for each student by a personal agent). No ontology is used so the system remains universally applicable, but probably at the cost of slightly lower quality. Currently we are considering the approach of a rules-based system with some similarities to neural networks, which is applied to the notifications generated by the user's actions.

2. Deriving interests

Keywords exist for all elements within the system (see above). The actual method for identifying interesting keywords is split in three parts: The input, which is based on the actions observed and their associated keywords, the output, a set of weighted keywords, and the rules/ procedures for calculating the latter form the former. These elements will be discussed in this order.

2.1. Input: Actions observed

The following actions are observed by the system and sent to the personal agents for derivation of interests:

- Authoring a message: This happens either within a forum or a chat. The textual content and the subject are analyzed. Chat messages are difficult because they are very short and so keyword derivation is difficult or impossible, therefore not all of them might actually influence the result. If the message is a reply the text of the message replied to is also used in the calculation.
- Reading a message: Again the content and the subject are used. Also messages the current one is a reply to (and its parent messages up to some limit; as well as child messages in reply to this one) are included.
- Browsing course content: Viewing parts of a course is very important. However, to distinguish between individual students taking the same course additional properties need to be measured. These are the duration of the visit on a certain part (measured by the time elapsed between requests; this is not necessarily always correct but a reasonable approximation) and the time since the very first visit to this page (for recurring visits).
- Configured notifications: The system also supports individually configured notifications. If the user specifies some keywords there explicitly, these are also used.

All the actions received are stored within the agent, enabling calculations which are not based on iteratively adding new elements but rather requiring a complete recalculation. To avoid amassing too much data, in intervals all previous data is removed and their results are stored as initial values.

2.2. Output: Weighted keywords

The result of the process is a list of keywords associated with a value of their measure of applicability (which is normalized to the range [0.0, 1.0]; 0.0=no interest at all, 1.0=very high interest, top priority). In a further step this list is then reduced to the set of keywords to be used for comparison with material or other persons. This step is envisioned to incorporate both the keywords above a certain level (minimum interest e.g. 0.5 to exclude unlikely or weakly supported results) and their absolute number (floating limit to keep the number of interesting keywords within certain bound, e.g. between 5 and 20). Too many keywords will slow the system down and are probably not very descriptive either, as most students will be rather focused on some topics but not all available in the platform.

Practical experiments will later be conducted to determine whether a second property is needed for the resulting keywords: The certainty of the algorithm. The result

could be e.g. that a keyword is of high interest, but that the result has a rather low certainty. Currently these two issues (whether a keyword is of interest and how sure the algorithm is about this assessment) are mixed: A low certainty results in a low contribution to the resulting overall interest level.

2.3. Rules considered

The following rule templates are currently considered for calculating the weight of a certain keyword. Interest in keywords is only calculated for single words, combinations of them (e.g. two keywords appearing both in several materials) are not taken into account. This would add a further layer of complexity (both conceptual and runtime effort) but probably not improve the result very much. This is because keywords are rather few (might be helpful when considering the whole content) and sometimes not absolutely certain themselves (e.g. when automatically derived from plaintext).

- Configured keywords are taken for granted with the maximum value: The user is expected to know exactly about his/her interests. These are reduced by a list of stop words to make them comparable to derived ones.
- The more often a keyword appears in all documents (not only the visited ones!) of a certain area (here: a single course), the more basic weight it receives. However, the very top part (e.g. the top 10 percentile) receives rather low weight: These probably describe the whole course and would therefore apply to all students alike and not provide differentiations between them.
- The more often and the longer some material described by a certain keyword is visited, the more weight it receives: Interesting parts are visited often.
- If a keyword appears in many different visited areas, it is more important than a keyword from a single page visited often: Regularly visited single pages probably are navigation or contain general content (which has little significance for personal interests).
- The longer the first visit to some material is in the past compared to the last visit, the higher the weight: This signifies a long-term interest which is more important than returning to e.g. a post several times during a brief period (short-term interest only).
- Keywords of messages within the same thread diminish in importance with the "distance" from the current visit. E.g. if posting a reply the content of the message replied to is important, but the previous message (two steps in the past), is of less significance.
- Keywords within subjects of messages are given precedence over keywords only appearing in the body. It is assumed that this is an important element of the decision whether to read the message or not and therefore describes the interests better than the content.
- Messages authored are more important than those only read and messages read which are in reply to authored messages are similarly considered more important than reading "unrelated" messages.

Additionally, the farther back in time an action was the less importance it receives until it is finally removed from the calculation or summarized (see above). Through this interests deprecate, resulting in a gradual worsening of the results over time. However, this reduces problems of incorrect derivations and changes of interests.

3. Conclusions and future work

The implementation of this subsystem has just started and first results are expected for the end of this year. Like the rest of the system it will be implemented in Java. As no matching rules engine could be found (which requires additional properties as it will be used for other subsystems too), a custom implementation is required. Fortunately, through implementing this subsystem as autonomous agents, performance is of lower priority as calculations will take place asynchronously. Important parts like the actual weights of the elements of the different rules can only be determined after complete implementation and will be validated through testing.

An area for further research is determining whether a significant difference in quality exists between keywords explicitly provided and those derived automatically, and how much influence this has on the end result. One method to measure this could be a questionnaire for the students at the end of the semester asking them whether the keywords and their importance are correct or not. We suspect that keywords which occur both explicitly and implicitly will be quite accurate, while those entering the set of rules from automatic derivation only produce lower quality results (wrong identification more likely).

4. Acknowledgement

This paper is a result of the project "Integrating Agents into Teleteaching-Webportals" funded by the Austrian FWF (Project number P15947-N04).

5. References

[1] Agentsystem POND: http://www.fim.uni-linz.ac.at/research/Agenten/index.htm
[2] Sonntag, M., Loidl-Reisinger, S.: Cooperative Agent-Supported Learning with WeLearn. In: Chroust, G., Hofer; C. (Eds.): Euromicro 2003. Los Alamitos, IEEE 2003
[3] Web Environment for Learning: http://www.fim.uni-linz.ac.at/research/WeLearn/index.htm