Conceptual Language Models for Context-Aware Text Retrieval

Henning Rode Djoerd Hiemstra

University of Twente, The Netherlands
{h.rode, d.hiemstra}@cs.utwente.nl

Abstract

While participating in the HARD track our first question was, what an IR-application should look like that takes into account preference meta-data from the user, without the need of explicit (manual) meta-data tagging of the collection. Especially, we touch the question how contextual information can be described in an abstract model appropriate for the IR-task, which further allows improving and fine-tuning of the context representations by learning from the user. As a first result, we roughly sketch a system architecture and context representation based on statistical language models that fits well to the task of the HARD track. Furthermore, we discuss issues of ranking and score normalizations on this background.

Keywords Contextual Information Retrieval, Context Modeling, Language Models

1 Introduction

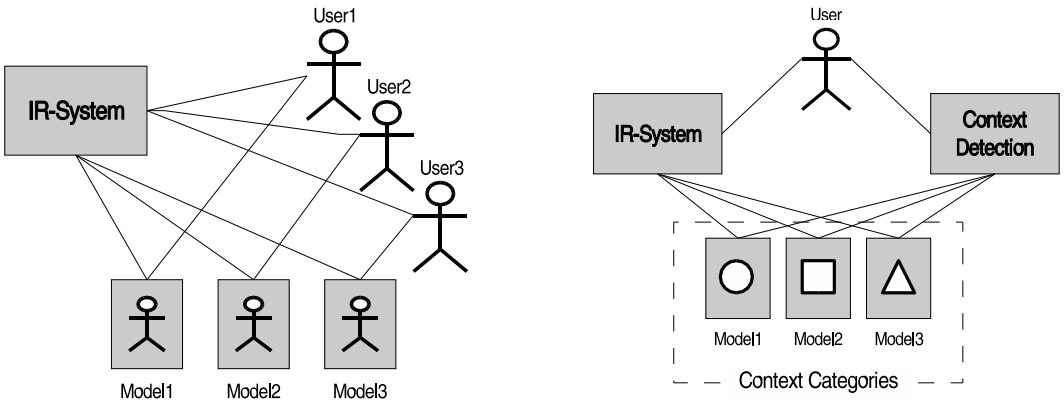
Observing that humans are thinking about, searching for and working with information highly depending on their current (working) context, leads directly to the hypothesis that information systems could increase their performance by learning how to deal with such contextual information. Among other ongoing research projects the HARD track is trying to tackle these issues. It especially addresses the question, how already available information about the user's context can be employed effectively to gain highly precise search results.

A user's information need is only vaguely described by the typical short query, the user states him-/herself to the system. There are at least two reasons for this lack of input precision. First of all, users who search for a certain piece of information have only a limited knowledge about it themselves. The difficulty to describe it is thus an immanent problem of any information need and hardly to overcome. A second reason for insufficient query input, however, touches the area of context information and might in principle be easier to address. Although a humans' search context provides a lot of information about his/her specific information need, a searcher is often not able and not used to explicitly mention it to a system. Comparing the situation to question another human, the counterpart would be able to derive contextual information him-/herself.

In order to outline this paper, we start with an overview on context modeling in the area of information retrieval. The section thereafter will introduce our own approach of context modeling in more detail and sketch a context-aware retrieval system. We proceed further by taking a closer look at incorporating context information in ranking algorithms. Finally, the last section describes our HARD track experiments and presents some empirical results.

2 Context Modeling for Information Retrieval

Aiming at a context-aware text retrieval system, we first have to investigate how context can be modeled appropriately that an IR-system can take advantage of this information. One of the first upcoming matters will probably be described by the following question: Should we try to build a model for each individual user or should it classify the user with respect to user-independent predefined



(a) User-Dependent Models

(b) User-Independent Models

Figure 1: Context Modeling: User vs. Category Models

context-categories? Both kind of systems are outlined in Figure 1. We will choose here the second option, but first discuss the advantages and disadvantages of both by pointing to some related research in this area.

2.1 User-Dependent Models

A first and typical example for this approach is shown by Diaz and Allan in [6]. The authors suggested to build a user preference language model from documents taken out of the browsing-history. Since the model reflects the browsing behavior of each individual user, it describes his/her preferences in a very specific way.

However, humans work and search for information often in multitasking environments (see [11]). Thus, their information need changes frequently, or even overlaps between different tasks. A static profile of each user is not appropriate to take into account rapid contextual changes. For this reason, Diaz and Allan [6] also tested the more dynamic version of session models derived from the most recent visited documents only. With the same intention but following a more "exotic" approach, Bauer and Leake [3] introduced a genetic sieve algorithm, that filters out temporally frequent words occurring in a stream of documents, whereas it stays unaffected by longterm front-runners like stop words. The system is thus aware of sudden contextual changes, but cannot come up directly with sound models describing the new situation.

Summarizing the observations, individual user models enable the most user specific systems, but either lack a balanced and complete description or remain unaware of alternating contexts.

2.2 User-Independent Models

Although context itself is by definition user-dependent, it is possible to approximately describe a specific situation by selecting best-matching predefined concepts, that are themselves independent of any specific user. A concept in this respect might range from a subject description (e.g. Music) to geographical and temporal information (e.g. Netherlands, 16th century).

Examples for this approach can be found among others in last years' HARD track. Along with the query, a set of meta-data items characterized the context of each specific information need. Since the meta-data was structured in categories restricted to certain set of pre-defined items, it was in theory possible to build stable and sound models to classify documents according to each of these concepts.

Following this approach of context modeling, it needs to be explained where the additional context meta-data comes from. Whereas Belkin et al. [4] preferred to think of it as derived by automatic context-detection from the users' behavior, He and Demner-Fushman [7] described the collecting of contextual information in a framework of explicit negotiation between the search-system and user. Further experiments in this area are presented in [10]. The authors tried to employ a conceptual hierarchy of subjects, as established by the "open directory project" [1] or "Yahoo" [2], as contextual models. In a first experiment, queries were compared to these concepts and the best-

matching subjects were displayed to the user for explicit selection. In order to avoid this negotiation process, long-term user profiles were introduced for automatic derivation of matching subjects, which cluster the former interests of the user in suitable groups. However, these user-dependent models suffer from the same limitations as mentioned above.

Although the question is not answered satisfyingly, how automatic context detection can be performed, user-independent context modeling comes up with a good deal of advantages:

- Whereas user modeling suffers often from sparse data, conceptual models are trained by all users of the systems and therefore will become more balanced and complete.
- Conceptual models do not counteract search on topics entirely new to the user.
- Assuming a perfect context detection unit, the search system can react more flexible with respect to a changing context of a user.
- New users can search efficiently without the need to train their user preference models in advance.
- It is theoretically possible to switch back anytime from automatic context detection to a negotiation mode, which enables the user to control the system effectively.

Taking a closer look on conceptual context modeling, the first task will be to identify appropriate categories of the users situation with respect to the information retrieval task. Whereas we can call almost everything surrounding the user as context, we only need those data that allows to further specify the information need of the user. For instance, the HARD track comes with the categories familiarity, genre, subject, geography and related documents. We can easily extend this set by further categories like language or time/date of the desired information.

One might notice, that the chosen categories originate more from a document than from a user centered view. Since we want to fine-tune the retrieval process, it is handy to have categories that directly support the document search, however, starting from the users context, this already requires a first translation. For instance the situation of a biology scientist sitting at his work, might be translated to the following context categorization: familiarity with search-topic: *high*, search genre: *scientific articles*, general subject: *biology*.

The translation of the users situation into the desired context categorization is, of course, itself an error-prone process. Thus, the before-mentioned possibility allowing the user to explicitly edit the automatically performed categorization of his/her context might be an important issue.

3 Conceptual Language Models

The retrieval process itself is enhanced by multiple text-categorizations based on the selected context models best-matching the users' situation. Thus, the maintained models for each context concept will be used by the system as classifiers, e.g. a model for scientific articles should be applicable to filter out scientific articles from an arbitrary set of documents.

Looking back at last years' HARD track experiments (e.g. at [4, 8]), every context category is handled with different techniques ranging from a set of simple heuristic rules as used for classifying the genre to applying external ill-founded though efficient algorithms like the "Fog-Index Measure" to rate the readability. The techniques might enable an IR-system to utilize the specific given meta-data, but the approaches lack a uniform framework that enables extending the system to work with other meta-data categories as well.

Instead of introducing another set of new techniques, our basic hypothesis is that statistical language models are a sufficient mean to be applied as a universal representation for all context categories as long as they are used to support text retrieval. Obviously, language models can be utilized effectively as subject classifiers, but we think, it is also possible to use them to judge about the genre or readability of a document. In the latter case, we can for instance assume that easily readable articles will probably consist of common rather than of special terms. For geography models, on the other hand, we would expect a higher probability to see certain city names and persons, whereas genre models might contain often occurring specific verbs.

In order to make our text retrieval system context-aware, it is thus sufficient to enhance it by a set of language model classifiers for each context category. For the purpose of the HARD track, we assume a context detection unit is able to perform the translation process from a concrete user

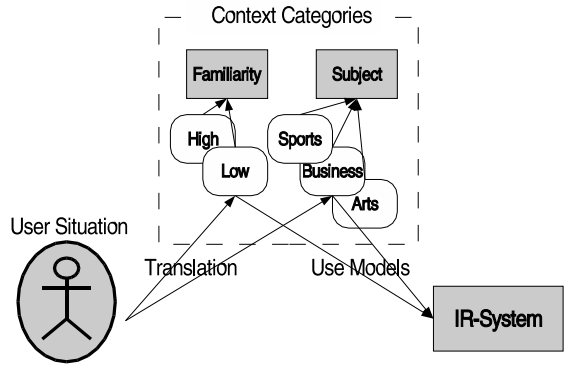


Figure 2: Context Modeling with Conceptual Language Models

situation to a characteristic selection of conceptual models. The remaining task to perform all document classifications and to combine them for a final ranking according to the entire search topic will be addressed in the next section. Figure 2 sketches roughly the described system.

3.1 Learning Application

Apparently, an IR-system working with conceptual models will profit from being a self-learning application. It might be possible to start the system with basic models for each category, but in order to have an easily extendable application, which enables to build new categories and models, it is beneficial to have a system that is able to train its models itself by the feedback of the user.

Anytime a user indicates (explictly or observed by his browsing behavior) that a certain document matches her/his information need, we can assume that it also matches the selected conceptual models. Therefore, the content of such a document can be used to train the context models. In the setting of the HARD track we will use the LDC annotation of the training topics to improve our models in the same way.

4 Ranking Algorithms

Having language models at hand that describe the users context, we are able to classify the documents according to each single context category, but we need to come up with one final ranking including every single source of relevance evidence. There are basically three options to perform this task:

- *Query Expansion* or any kind of uniting of terms, taken from all context models, to build one large final query.
- *Reranking* or *filtering* of the results according to each classifier.
- Using *combined ranking* algorithms to aggregate the scores of single classifications.

Using query expansion techniques would lead to the difficult task to select a certain number of representative terms from each model. Since the query and "meta-query" models differ highly in length, we cannot simply unite all terms to one combined query. Filtering or reranking, if it is used in a more "aggressive" way, can be regarded as black-and-white decisions for or against a document. However, thinking of several meta-query categories it is likely that a document is judged relevant by a user even if it does not match one of the associated classifiers. Therefore, we opt here for a combined ranking solution, which is comparable to "softer" reranking strategies and allows to consider each context-classification step adequately.

4.1 Combined Ranking of Query and Meta-Query

For discussing the ranking of documents according to the query and meta-query we first introduce some common notation. Let the random variables Q, D denote the choice of a query, respectively document, and $r/\bar{r}$ mark the event, that D is regarded as relevant/not relevant. Further, M represents in our case the *meta-query*, consisting of several single models for each involved category M_i:

$$M = \{M_1, M_2, \ldots, M_n\}.$$

Using the odds of relevance as a basis, we can deduce it to probabilities we are able to estimate. Q and M are assumed to be independent given D and r.

$$\begin{aligned} \frac{P(r|Q,M,D)}{P(\bar{r}|Q,M,D)} &= \frac{P(Q,M,D|r) * P(r)}{P(Q,M,D|\bar{r}) * P(\bar{r})} \\ &= \frac{P(Q|D,r) * P(M|D,r) * P(r|D)}{P(Q|D,\bar{r}) * P(M|D,\bar{r}) * P(\bar{r}|D)} \\ &\propto \frac{P(Q|D,r)}{P(Q|D,\bar{r})} * \frac{P(M|D,r)}{P(M|D,\bar{r})} \end{aligned}$$

The prior document relevance $P(r|D)/P(\bar{r}|D)$ is dropped from the formula in the last step. We assume that there is not a-priori reason that a user would like one document over another, effectively making the prior document relevance constant in this case.

The simple derivation now allows to handle query and meta-query separately but in a similar manner. In terms of the user's information need we can regard Q and M as alternative incomplete and noisy query representations. Combining the resulting document rankings from both queries gathers different pieces of evidence about relevance and thus helps to improve retrieval effectiveness [5].

The remaining probabilities can be estimated following the language modeling approach. D, Q and M are interpreted here as a sequence of terms, and the probability to "generate" a sequence X out of Y is estimated by the sum of the log likelihood probabilities of the terms occurring in X interpolated by a smoothing factor. We denote a maximum likelihood probability of a term t within a sequence of words X by $P(t|X)$. According to Kraaij [9] the probabilities of the form $P(X|t,\bar{r})$ where a term t is non-relevant for the generation of X can be estimated by the collection likelihood of the term. Combing both enables to determine the required relevance odds, also called the *logarithmic likelihood ratio* of a query Q given a document D:

$$\begin{aligned} & \log\left(\frac{P(Q|D,r)}{P(Q|D,\bar{r})}\right) \\ = & \sum_{t\in Q} |t \text{ in } Q| * \log\left(\frac{(1-\lambda)P(t|D)+\lambda P(t|C)}{P(t|C)}\right) \\ = & \; LLR(Q|D). \end{aligned}$$

Here, C represents the entire collection and λ the smoothing factor to interpolate document and collection likelihood.

Since we want to relate the scores of the query and meta-query to each other, we have to ensure that their probability estimates deliver "compatible" values (see [5]). Especially query length normalization plays a crucial role in this case. Notice, that Q and M differ widely with respect to their length. Thus, a simple *LLR*-ranking would produce by far higher values when it is applied to the meta-query. Using *NLLR* instead, the query *length normalized* variant of the above measurement, helps to avoid score incompatibilities.

4.2 Ranking according to the Meta-Query

As mentioned above, we would like to rank documents according to query and meta-query in the same way. However, since M consists of several single language models $M_1,\ldots,M_n$ we need to take a closer look to this matter as well.

If M is substituted by $M_1,\ldots,M_n$, the resulting formula can be factorized, given the independence of $M_1,\ldots,M_n$:

$$\begin{aligned} & \log\left(\frac{P(M_1,\ldots,M_n|D,r)}{P(M_1,\ldots,M_n|D,\bar{r})}\right) \\ = & \log\left(\frac{P(M_1|D,r)}{P(M_1|D,\bar{r})} * \ldots * \frac{P(M_n|D,r)}{P(M_n|D,\bar{r})}\right) \\ \propto & \; \frac{1}{n}\left(NLLR(M_1|D)+\ldots+NLLR(M_n|D)\right). \end{aligned}$$

We introduced the factor $\frac{1}{n}$ in the last row as a second normalization step due to the number of involved meta-data models. Especially if the number n of models is growing, not only the overall score of the documents would increase, but also the entire meta-query would outweigh the original query in the final rank.

A last remark concerns the choice of the smoothing factor λ. In contrast to typical short queries, the role of smoothing is less important here, since we can assume that the model is a good representation of relevant documents and therefore contains most of their words itself. We thus argue to use a smaller value for λ here than in case of the query ranking to stress the selectivity of the models.

5 Experiments

As apparent from the description of our system, our HARD track research focuses on the usage of the meta-data coming along with the search topics. We have neither tried to take advantage of clarification forms for relevance feedback nor have we applied our algorithms to perform retrieval on passage level. Furthermore, we abstained completely from techniques like stemming and stopword removal, since they might degrade the effectivity of classifiers, like genre language models.

Unfortunately, the submission deadline for the runs has met a quite early phase of our research. The submitted runs, though they performed quite well, do not reflect all considerations presented in this paper. In order to provide a consistent presentation, we decided to show only "post-track" runs in this section, which directly follow the given description.

5.1 Collecting Data for the Models

For this years' experiments, we have used only a part of the meta-data that came along with the queries, namely the *subject*, *geography* and *related text* sections. Having appropriate models at hand is a crucial requirement for any kind of experiments

	utwenteB21	utwenteM21	utwenteB111	utwenteM111
R-Precision (Hard)	0.267	0.289	0.294	0.349
R-Precision (Hard+Soft)	0.268	0.308	0.308	0.366

Table 1: R-Precision for Baseline and Meta-data Runs

and the need to construct them ourselves has led to this limitation.

The *subject* data was chosen, because it was considered to work best with respect to the purpose to classify texts. It is probably easier to identify sport articles by their typical vocabulary then to distinguish between genres. *Geography* data, on the contrary, can be regarded as a less typical domain for applying language model classifiers. And finally *related text* documents were used to demonstrate their straightforward integration in the proposed context modeling framework.

In order to construct language models for subject classification, we used three different sources of data: manual annotation, APE keywords, and the training data.

Firstly, we manually annotated 500 documents for each chosen subject among the queries, e.g. sports, health and technology. The 500 documents have been preselected by a simple query containing the subject term and additional terms found in a thesaurus. The aim of this step was to detect 150-200 relevant documents as a basic model representing its subject. For construction of a language model all terms occurring in those documents were simply united to build one large "vocabulary" and probability distribution.

Although the number of documents might look appropriate for building a basic text classifier, the way we gathered the documents cannot ensure the models to be unbiased. In order to further improve the models, we used the keyword annotation coming along with the documents. During the manual classification process we observed that the keyword section of documents from the Associated Press Newswire (APE) provide very useful hints and in many cases HARD subjects can easily be assigned to APE keywords. It seemed admissible from research perspective to employ this information as long as we restrict it to a small part of the corpus, in this case APE news only. However, since HARD subjects cannot be mapped one-to-one to APE keywords, our subject models differed considerably afterwards in length and quality. For the geography models, the link between query meta-data and document keywords was easier to establish. Therefore, the geography models highly profit by this technique.

In a last step, we automatically enhanced the models by data obtained from the annotated training topics as mentioned above. If any document was judged as relevant to a specific training query, this also means that the document matches all the meta-data constraints of that query. Thus, all relevant documents belonging to a query asking e.g. for sport articles, apparently are sport articles themselves, and can therefore be used to enrich the sport articles model.

Baseline Runs Every HARD track topic was specified by a title, description and topic-narrative section, which could be used for the baseline runs. The easiest solution to compose a query of this data was thus to unite all occurring terms in the three sections to one query model. The queries of our first baseline run `utwenteB111` were composed in this way. The second baseline run `utwenteB21` reflected the more realistic situation of shorter user queries, combing only the title and description section. Since the number and expressiveness of terms in the title section differs from the topic-description, we weighted title terms higher here by adding them two times to the query.

For both runs we computed a ranking according to $NLLR(Q|D)$ as a basis for later comparisons with the meta-data runs. The two baseline runs gives us the possibility to examine the influence of the initial query length to improvements by context meta-data.

Meta-data Runs Corresponding to the equally named baseline runs, `utwenteM111` and `utwenteM21` were calculated as shown in the last section and take into account subject, geography, and related texts as $M_1 \dots M_3$. Table 1 gives an overview on the achieved average R-Precision of all runs. It shows first of all that our approach for handling contextual data is able to improve retrieval results, for soft as well as for hard relevance. We expected higher relative improvements when using context information together with short user queries, however, our

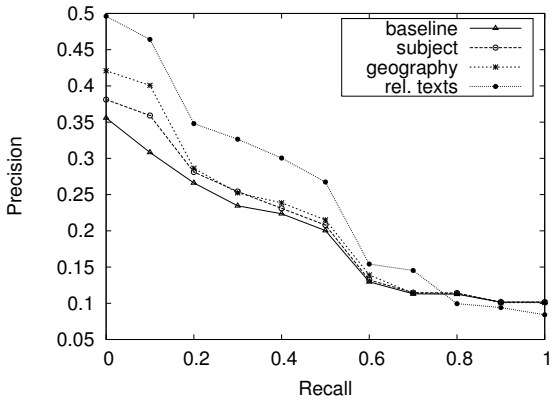


Figure 3: Comparing Precision/Recall for each single Meta-data Category

runs show that scenarios with long queries, like in `utwenteM111`, can profit in the same way from contextual data.

We performed further experiments to find out if the given context categories are equally useful for improving the system performance. Figure 3 presents the resulting precision-recall graph if the queries are associated with only one specific meta-data category. It considers short queries and hard relevance only. In order to get comparable results for all categories, we needed to restrict the evaluation to a small subset of 11 topics that came with geographical and subject requirements we could support with appropriate models. For instance, we dropped topics asking for the subject *society*, since the associated classifier was considered rather weak compared to others. Such a restriction is admissible, since we were interested in the retrieval improvements in the case appropriate models are available, however, the remaining topic set was unfortunately a relative small base for drawing strong conclusions.

The graph suggests that the utilization of geography and subject preferences allow small improvements whereas related texts considerably increase the retrieval performance. In fact, using related text information alone shows even better results than its combination with other meta-data. As a conclusion, it might be interesting to test in further experiments, if a more elaborated approach of combining the rankings according to each single meta-data category is able to correct such effects. The displayed graph shows further that the usage of contextual data especially enhances the precision at small levels of recall, which meets perfectly the "high accuracy" task of the track.

At last, the HARD track provided a very suitable environment for our research and we are looking forward to continue our experiments on other context categories and with different ranking functions.

References

[1] Open Directory Project. Netscape Communication Cooperation. `http://www.dmoz.org`.

[2] Yahoo Directory. Yahoo! Inc. `http://dir.yahoo.com`.

[3] T. Bauer, D. B. Leake. Real Time User Context Modeling for Information Retrieval Agents. In *Proceedings of the 2001 ACM CIKM International Conference on Information and Knowledge Management*, pages 568–570. ACM, New York, NY, USA, 2001.

[4] N. J. Belkin, et al. Rutgers' HARD and Web Interactive Track Experiences at TREC 2003. In *The Twelfth Text Retrieval Conference (TREC 2003)*, pages 418–429. NIST, Gaithersburg, Maryland, USA, 2003.

[5] W. Croft. Combining Approaches to Information Retrieval. In W. Croft (ed.), *Advances in Information Retrieval*, pages 1–36. Kluwer Academic Publishers, Massachusetts, USA, 2000.

[6] F. Diaz, J. Allan. Browsing-based User Language Models for Information Retrieval. Technical report, CIIR University of Massachusetts, 2003.

[7] D. He, D. Demner-Fushman. HARD Experiment at Maryland: from Need Negotiation to Automated HARD Process. In *The Twelfth Text Retrieval Conference (TREC 2003)*, pages 707–714. NIST, Gaithersburg, Maryland, USA, 2003.

[8] N. A. Jaleel, et al. UMass at Trec2003: HARD and QA. In *The Twelfth Text Retrieval Conference (TREC 2003)*, pages 715–725. NIST, Gaithersburg, Maryland, USA, 2003.

[9] W. Kraaij. *Variations on language modeling for information retrieval*. Ph.D. thesis, University of Twente, Netherlands, 2004.

[10] A. Sieg, B. Mobasher, R. Burke. Inferring User's Information Context: Integrating User Profiles and Concept Hierarchies. In *Proceedings of the 2004 Meeting of the International Federation of Classification Societies*. Chicago, USA, 2004.

[11] A. Spink, M. Park. Information Retrieval as Multitasking: An Exploratory Framework. In *ACM SIGIR 2004 Workshop on 'Information Retrieval in Context'*, pages 16–19. Sheffield, UK, 2004.