FINDING THE RIGHT EXPERT

Discriminative Models for Expert Retrieval

Philipp Sorg[1] and Philipp Cimiano[2]
[1]*AIFB, Karlsruhe Institute of Technology, Germany*
[2]*CITEC, University of Bielefeld, Germany*
philipp.sorg@kit.edu, cimiano@cit-ec.uni-bielefeld.de

Keywords: Expert Retrieval, Learning to Rank, Language Models, Feature Design, Machine Learning

Abstract: We tackle the problem of expert retrieval in Social Question Answering (SQA) sites. In particular, we consider the task of, given an information need in the form of a question posted in a SQA site, ranking potential experts according to the likelihood that they can answer the question. We propose a discriminative model (DM) that allows to combine different sources of evidence in a single retrieval model using machine learning techniques. The features used as input for the discriminative model comprise features derived from language models, standard probabilistic retrieval functions and features quantifying the popularity of an expert in the category of the question. As input for the DM, we propose a novel feature design that allows to exploit language models as features. We perform experiments and evaluate our approach on a dataset extracted from Yahoo! Answers, recently used as benchmark in the CriES Workshop, and show that our proposed approach outperforms i) standard probabilistic retrieval models, ii) a state-of-the-art expert retrieval approach based on language models as well as iii) an established learning to rank model.

1 INTRODUCTION

Imagine you have a non-trivial information need or question for which you seek an expert that you can directly interact with. In fact, for very specific information needs, the general knowledge available on the Web might not be detailed enough and the expertise of an expert in the domain might be needed. An example for such an information need – taken from our dataset – is: *Which cat has the largest variety of prey?*

This setting gives raise to the so called *expert retrieval* problem (Yimam-Seid and Kobsa, 2003), i.e. the task of, given a certain information need, retrieving those experts that can most likely contribute to answering the question. Such an expert retrieval use case arises naturally in Social Question Answering (SQA) portals such as *Yahoo! Answers*[1] or *Answers.com*[2]. In these portals, users post questions which are then answered by other users. In this scenario, new questions could be directly routed to the most promising experts. The input to the expert retrieval task as defined in this paper is a question and the output is a ranked list of experts that could help to answer this question.

In general, we can assume that the expertise of people is essentially defined by their behavior on the Web, in a specific social network (LinkedIn, Facebook, Xing etc.) or a social website like Yahoo! Answers or Wikipedia. By constructing and indexing text profiles consisting of postings by the given expert, standard IR models can be applied to rank experts. This is the approach followed in many approaches to expert retrieval (Balog et al., 2009; Craswell et al., 2005). However, there are other evidence sources for expertise than the overlap between the question and the (textual) profile of the expert. Our focus in this paper is on approaches that allow to combine different models of expertise in a single retrieval model.

In particular, we propose to use a discriminative model to score experts that is optimized using machine learning (ML) techniques. In a supervised setting, a classifier is trained using the question/answer history and relevance assessments of former topics. This classifier is then applied in the retrieval scenario to score each expert for a given query, thus producing a ranking according to the scores. This approach allows to combine different sources of evidence in a principled manner as the combination parameters are

[1] http://answers.yahoo.com/

[2] http://www.answers.com/

optimized through the training of the classifier.

The contributions of the paper are the following:

- We compare the performance of different expert retrieval models on a subset of the Yahoo! Answers dataset used as benchmarking dataset in the Cross-lingual Expert Search (CriES) workshop recently hosted at CLEF.[3]
- We propose a novel feature design that allows to build feature vectors from language models in a retrieval scenario. In our detailed feature analysis, we evaluate different subsets of features and show that our feature design indeed captures the relevance of experts. We also identify those features which contribute most to the retrieval performance.
- We show that the discriminative model based on the novel feature design allows to combine different sources of evidences in a principled way and outperforms all the other models we consider, including an alternative learning-to-rank approach.

  As sources of evidence, we propose to exploit different language models that are estimated using the text profile of experts, the text profiles of categories and on the basis of the popularity of an expert in a given category. Additionally, we use established probabilistic retrieval models as further source of evidence.

The structure of the paper is as follows: We first present different expert retrieval models in Section 2 that we use as baselines and that are also used as sources of evidence for the discriminative model. Then, we introduce the discriminative model and our feature design in Section 3. Afterwards, we describe our datasets, experiments and results obtained in Section 4. After discussing related work in Section 5, we finally wrap up in Section 6.

2 EXPERT RETRIEVAL MODELS

In this section we present the retrieval models used in our experiments. Firstly, we present a popularity model. Secondly, we introduce different standard approaches to information retrieval (IR). Then we introduce a mixture language modeling approach that allows to factor in information about categories. Finally, we present a learning to rank approach that is used as a reference model in our experiments.

2.1 Popularity Models

The popularity model assumes that the category of a new question is known. Experts are ranked according to their "popularity" for the given category. In particular, the popularity models rank experts according to their likelihood of being able to answer **any** question from a given category, estimated on the available question/answer history and other features of this category. We define two different measures of popularity:

Definition 1 (Frequency Pop. Model ($\mathrm{PM}_{\mathrm{freq}}$)). *$PM_{freq}$ is defined as the share of answers that an expert e has contributed to category c:*

$$PM_{freq}(e,c) = \frac{|ANSWERS(c) \cap ANSWERS(e)|}{|ANSWERS(c)|}$$

The second approach applies the Pagerank algorithm (Page et al., 1999) to the expert network G_c of category c. Nodes in G_c are defined by the users of the SQA portal. There is a directed link from user u_1 to user u_2 if u_2 provided the best answer to a question posted by u_1.

Definition 2 (Pagerank Pop. Model ($\mathrm{PM}_{\mathrm{pagerank}}$)). *Given the expert network G_c of category c, the Pagerank based popularity model of user e is defined as:*

$$PM_{pagerank}(e,c) = \frac{PAGERANK(G_c,e)}{\sum_{e' \in E} PAGERANK(G_c,e')}$$

2.2 Probabilistic Models and Language Models

We have selected two different standard MLIR models as baselines for our experiments:

BM25: As experts are defined through textual profiles, the task of retrieving experts can be reduced to a standard document retrieval scenario. As state-of-the-art probabilistic retrieval model to compare to we chose the BM25 model (Robertson and Walker, 1994). As we consider a multilingual expert retrieval scenario, results from language-specific indices are combined using Z-Score normalization (Savoy, 2005), which has proven to be effective for multilingual IR (see e.g. (Kürsten, 2009)).

BM25 + Popularity: In the experiments we assume that the category of a new question is known. In this case, the popularity of experts in this target category can be combined with any vector space or probabilistic retrieval model by simply adding the popularity to the score of each expert. In our experiments, we will combine BM25 with the frequency based popularity model $\mathrm{PM}_{\mathrm{freq}}$. To ensure compatible values, we will

[3] `http://www.multipla-project.org/cries`

again first apply Z-Score normalization, resulting in s^*_{BM25} and $\text{PM}^*_{\text{freq}}$. The final score is then defined by the weighted sum of the normalized values:

$$s(e,q) = \alpha \cdot s^*_{\text{BM25}}(e,q) + (1-\alpha) \cdot \text{PM}^*_{\text{freq}}(e,c_q)$$

Language Models: An alternative approach to IR is based on the theory of language models (LMs). In line with Model 1 presented by Balog et al. (Balog et al., 2009) we use the following basic LM for expert search based on answers of users:

$$P_{\text{LM}}(q|e) = \prod_{t\in q}\left[\alpha\left(\sum_{a\in A} P(t|a)P(a|e)\right) + (1-\alpha)P_{\text{bg}}(t)\right]$$

where $P(t|a)$ is the probability of an answer generating a term t and $P(a|e)$ is the probability that expert e provides answer a. We estimate $P(t|a)$ via Maximum Likelihood Estimation: $P(t|a) = \frac{\text{TF}_{a(t)}}{|a|}$. In order to estimate the probability $P(a|e)$ that expert e generates answer a, we apply Bayes' Theorem: $P(a|e) = \frac{P(e|a)P(a)}{P(e)}$ with $P(e|a) = 1$ in case e is actually the author of a, 0 otherwise. The priors $P(a)$ and $P(e)$ are assumed to be uniformly distributed, i.e. $P(a) = \frac{1}{|A|}$ and $P(e) = \frac{1}{|E|}$. The background language model $P_{\text{bg}}(t)$ is estimated by considering the distribution of terms in the entire answer corpus.

As the data we use in our experiments contains both questions in different languages as well as experts posting answers in these languages, we extend the model from Balog by constructing a translated query q^*, consisting of all terms of the translations of the original query to all corpus languages. For translated queries we use a multilingual background language model $P^*_{\text{bg}}(t)$ that estimates priors of terms in different languages based on the language-specific term distributions in the entire corpus.

2.3 Mixture Language Model

We also compare our discriminative approach to a mixture language model (MLM), which extends the model of Balog at al. and allows to combine different evidence sources to estimate the relevance of experts into a single generative model. The evidence sources are modeled as conditional probabilities $P_i(t|e)$ of an expert e generating a query term t and combined in a mixture model of probabilities using weights α_i:

$$P_{\text{MLM}}(q^*|e) = \prod_{t\in q^*}\left[\sum_i \alpha_i P_i(t|e) + (1-\sum_i \alpha_i)P^*_{\text{bg}}(t)\right]$$

The α_i weights thus determine the influence of each component or source of evidence of the mixture model. Overall, in the MLM approach we combine the following models that are used to estimate $P_i(t|e)$:

Text Profile Model: This is a generative model that quantifies the probability of an expert e generating a query term t on the basis of its text profile, consisting of all answers that are associated to this expert:

$$P_{\text{profile}}(t|e) = \sum_{a\in A} P(t|a)P(a|e)$$

Category-restricted Text Profile Model: Given a target category c, the text profiles of experts can be restricted to answers $a \in c$. This results in the following estimation:

$$P_{\text{profile+c}}(t|e) = \sum_{a\in c} P(t|a)P(a|e)$$

In this model, only answers in the target category are considered in order to estimate the relevance of experts. Other answers of experts which might introduce noise to the language models of expert in respect to the given query are ignored.

Category Model: Language models of categories can be defined using the text of all answers in each category. These language models can then be used for an implicit classification of query terms, resulting in the following category-based estimation of $P(t|e)$:

$$P_{\text{cat}}(t|e) = \sum_{c\in C} P(t|c)P(c|e)$$

The probability $P(c|e)$ of category c given expert e is estimated by the popularity models as presented above: $P_{\text{cat-freq}}(c|e) \approx \text{PM}_{\text{freq}}(e,c)$ and $P_{\text{cat-pagerank}}(c|e) \approx \text{PM}_{\text{pagerank}}(e,c)$.

2.4 Learning to Rank

Joachims (Joachims, 2002) has presented a modified version of support vector machines (SVMs) – called ranking SVMs – that learn a retrieval function on the basis of rankings provided as training input. The ranking for training is essentially given by a preference function as formalized in the following definition:

Definition 3 (Learning to Rank)**.** *Given a question q, learning to rank approaches compare pairs of experts to compute a preference of which expert to rank before the other. The ranking of experts can then be reduced to the preference function* pref*:*

$$pref : E \times E \times Q \to \{0,1\}$$

with $pref(e_1,e_2,q) = 1$ meaning that e_1 should be ranked before e_2 for query q.

In our experiments, we applied the *SVMRank* software published by Joachims to the problem of expert retrieval. Thereby, we used the same features and the same training data as defined for the discriminative model that will be presented in Section 3. This allows to compare the effectiveness of the ML model applied to the ranking problem, as the same feature vectors for expert-query tuples are used as input in both cases.

3 DISCRIMINATIVE MODEL

In Section 2 we presented several expert retrieval models. Combining different retrieval models is a promising approach to further boost the retrieval performance. However, this combination often depends on parameters, for example when using the mixture language model as presented above. Exploring the parameter space using for instance gradient descent methods is time consuming and prone to lead to local optima. Therefore, we reduce retrieval to a classification/regression problem and thus make ML techniques applicable to this task:

Definition 4 (ER as Regression Problem)**.** *The problem of ranking experts E for a given question q can be reduced to a regression problem of learning the optimal parameters for a function* expertise*:*

$$expertise : E \times Q \rightarrow [0,1]$$

The experts can then be ranked according to $expertise(e,q)$ *for a given question q.*

The clear advantage of this approach is that efficient optimization techniques are known from machine learning research which can then be applied to the retrieval problem.

In case we use a standard discrete or binary classifier instead of a regression function, $expertise(e,q)$ could be set to the probability that a pair (e,q) belongs to the positive (relevant) class, i.e. $rank(e) \approx P(class = 1|q,e)$.

3.1 Feature Design

As input for the regression function, feature representations of candidate experts have to be defined. As the feature vectors need to have the same length independent of the length of the query (in the number of terms), the features need to be designed in such a way that their number is constant for any query length. Therefore, we propose to use aggregated values over all query terms to define single features.

Given expert e and query q, we use the following features to describe (e,q):

Feature	*Description*
MIN	$\min_{t \in q} P_i(t\|e)$
MAX	$\max_{t \in q} P_i(t\|e)$
AVG	$\frac{\sum_{t \in q} P_i(t\|q)}{\|q\|}$
MEDIAN	Median in $\{P_i(t\|e) \mid t \in q\}$
STD	Standard deviation in $\{P_i(t\|e) \mid t \in q\}$

Table 1: Aggregated features that are defined by the conditional probability distribution $P_i(t|e)$. These features describe properties of expert e given query $q = (t_1, t_2, \dots)$ according to generative model i.

Language Model Features: As described above, different sources of evidence are used to model the probability $P_i(t|e)$ of query term t being generated by expert e: P_{profile}, $P_{\text{profile+c}}$, $P_{\text{cat-freq}}$, $P_{\text{cat-pagerank}}$ and the popularity models PM_{freq} and $\text{PM}_{\text{pagerank}}$. We use this probability distribution to define aggregated features over all query terms. These aggregated features are described in detail in Table 1. We thus assume the generation of any query term to be conditionally independent from the generation of other query terms.

Products of Language Model Features: As the above aggregate features do not capture the dependencies between different sources of evidence on term level, we additionally consider products of these conditional probability distributions. For example given two evidence sources i and j, we define the combined distribution as $P_{ij}(t|e) = P_i(t|e)P_j(t|e)$. The aggregated features as described in Table 1 are then computed based on P_{ij}. In our experiments, we add features for all permutations of up to three models.

Probabilistic Model Features: In addition to the aggregated features over query terms for the different generative models, we also use standard IR retrieval scores based on the text profile of each expert. Features are then defined as the score of expert e given query q. In particular, we used retrieval models based on BM25, TF.IDF and DLH13 term weighting in combination with Z-Score normalization for multilingual retrieval, resulting in exactly one feature for each retrieval model.

3.2 Classifiers

In order to rank experts, we used Multi-Layer Perceptrons (MLP) and Logistic Regression as regression classifiers in our experiments. The layout of the MLPs was set to one hidden layer with half as many nodes as the number of input features. We used these standard settings as changes in the layout of the MLPs did not show any significant differences when applying the trained classifiers in the retrieval scenario.

We also used decision trees as instance of discrete classifiers. In particular, we used the J48 decision tree, a pruned version of C4.5 (Quinlan, 1993). In this case, the probability that the example belongs to the relevant class is used as expert score. For implementation we relied on the Weka framework[4] and also used the class probabilities for discrete classifiers as given in the output of the Weka API.

4 EXPERIMENTS

In this section, we will describe our experiments on a dataset from Yahoo! Answers, a popular SQA portal. We will present and discuss the retrieval results of the different baselines and of the proposed models.

Dataset: We used a dataset from the Yahoo! Answers portal introduced by (Surdeanu et al., 2008).[5] In our experiments we use the subset of the Yahoo! Answers Webscope dataset defined in the CriES workshop (Sorg et al., 2010). This dataset consists of 780,193 questions posted by 169,819 different users, representing the pool of experts in our experiments. These questions are classified into 305 categories which form a taxonomy. While the biggest share of questions and answers are written in English, the dataset also contains German (1%), French (3%) and Spanish (5%) questions. In our evaluation, we relied on the 60 topics use as part of the benchmarking task associated with the CriES workshop. There were 15 topics for each language in the dataset: English, German, French and Spanish. The Gold Standard was created by manual assessment of a result pool consisting of the top 10 retrieved experts of all submitted runs to the workshop for each topic. In our experiments we rely on the Gold Standard based on strict assessment (Sorg et al., 2010).

Training Data for Discriminative Model: As training data we used the available Gold Standard provided by the CriES workshop. Positive examples are generated from pairs of topics and corresponding experts marked as relevant, while negative examples are generated from pairs of topics and expert marked as non-relevant. Training and evaluation was performed in cross-validation manner with three sets of randomly chosen topics. The discriminative approach is trained on each combination of two folds, using the relevance assessments of the according topic/expert pairs. The evaluation was conducted on the remaining fold. The final results are averaged over three folds.

Evaluation Measures: As our results were not part of the initial result pool used to create the Gold Standard in the context of the CriES workshop we have to deal with incomplete relevance assessments. For some of the retrieved experts there is no relevance information available. Therefore, we chose BPREF as primary evaluation measure as this measure has been shown to be robust against incomplete relevance assessments (Buckley and Voorhees, 2004).

Further, we will use mean average precision (MAP), precision at cut-off level 10 (P@10) and recall at cut-off level 100 (R@100) as standard evaluation measures for retrieval scenarios. These measures were applied to evaluate the results both with respect to strict and lenient assessment.

4.1 Results

The results of our expert search experiments are presented in Table 2. The statistical significance between systems was verified using a paired t-test at a confidence level of .01. Significant results are marked with the ID number of the system compared to.[6] For the experiments based on ML, namely the discriminative models and the learning to rank approach, we performed a 3-fold cross-validation on the set of topics. The standard deviation of the values of each evaluation measure – computed for the three sets of topics – are given after the corresponding values.

In the following, we discuss the results of the baseline models, the mixture language model, the learning-to-rank model (LR) and the discriminative model (DM). We will use the system ID in parenthesis to link to the according results in Table 2. As a first observation, it is important to mention that the usage of the information given by the topic category generally improves retrieval results (runs 1-3 vs. runs 4-11). The results support the following conclusions:

- **Underperformance of probabilistic retrieval models.** Probabilistic retrieval models (1) do not produce comparable results to LMs (2,3) – in most cases worse than half of the performance of LMs.

[4] http://www.cs.waikato.ac.nz/ml/weka/

[5] This dataset is provided by the Yahoo! Research Webscope program (*L6. Yahoo! Answers Comprehensive Questions and Answers (version 1.0)*)

[6] For example, the P@10 value $^{(1)}$.50 of system 2 (P_{LM}) shows a statistical significant difference compared to the P@10 value .19 of system 1 (BM25).

ID	*Model*	*BPREF*	*MAP*	*P@10*	*R@100*
1	BM25 + Z-Score	.08	.04	.19	.10
2	P_{LM} (Balog et al., 2009)	$^{(1)}$.33	$^{(1)}$.22	$^{(1)}$.50	$^{(1)}$.45
3	P_{MLM} $(.1P_{\text{profile}} + .9P_{\text{cat}})$	$^{(2)}$.37	$^{(2)}$.26	$^{(2)}$.55	$^{(2)}$.50
4	CriES best run (Iftene et al., 2010)	.23	.21	$^{(2)}$.62	.24
5	PM$_{\text{freq}}$ (Popularity Model)	$^{(2)}$.37	$^{(2)}$.29	$^{(3)}$.67	.43
6	PM$_{\text{pagerank}}$ (Popularity Model)	.35	.25	.56	.42
7	BM25 + PM$_{\text{freq}}$ + Z-Score	$^{(3,5)}$.39	$^{(3,5)}$.31	$^{(3,5,8,9,11)}$**.71**	$^{(5)}$.47
8	P_{MLM} $(.5P_{\text{profile}} + .5\text{PM}_{\text{freq}})$	$^{(3,5)}$.39	$^{(3,5)}$.31	$^{(3)}$.67	$^{(5)}$.47
9	SVMRank	$^{(7,8)}$**.43** ± .02	$^{(3,5)}$.32 ± .03	.58 ± .05	$^{(3,7,8)}$**.60** ± .04
10	DM (Logistic Regression)	.39 ± .03	.27 ± .05	.50 ± .08	.54 ± .05
11	DM (MLP)	$^{(7,8)}$**.43** ± .01	$^{(8)}$**.33** ± .02	$^{(2)}$.60 ± .02	$^{(3,7,8)}$**.60** ± .03

Table 2: Expert retrieval results on the Yahoo! Answers dataset. Runs 4 to 11 are exploiting the a priori knowledge of the target category of new questions. BPREF, mean average precision (MAP), precision at cutoff level 10 (P@10) and recall at cutoff level 100 (R@100) are presented as evaluation measures.

Using probabilistic retrieval models on the problem of multilingual expert search in SQA portals seems therefore not adequate.

- **Popularity models outperform retrieval models that do not exploit the target category.** Using answer frequencies in categories as popularity model (5) beats all results of retrieval models which do not use the target category of new questions (1-3).
- **The best run submitted to the CriES challenge is outperformed by popularity baseline.** The best run out of the five submissions to the CriES workshop (4) is not able to beat the popularity baseline (6) we defined using answer frequencies.
- **BM25 combined with category knowledge and the informed MLM outperform the popularity model.** The MLM combining both text profiles of experts and the popularity model (8) slightly improves retrieval compared to the popularity model, which is significant for R@100, MAP and BPREF. Combining the probabilistic retrieval model BM25 with the popularity model achieves the best results according to P@10 (7). Results for R@100, MAP and BPREF are identical to the MLM. This is an astonishing result, as BM25 performs poorly in the non-informed scenario (1).
- **DMs successfully combine sources of evidence.** Using MLPs as learning approach (11), P@10 drops by .11 compared to the best informed model (7). However R@100 is improved by .13, BPREF by .04 and MAP by .02. This shows that our approach to the combination of different probability distributions using ML is indeed successful as BPREF, MAP and R@100 are significantly improved. When using logistic regression as classifier function, the results are consistently worse. BPREF drops by .04 and MAP by .06. This shows that optimizing the logistic regression function does not result in optimal combination parameters of different sources of evidence.
- **DMs are able to compete with the learning to rank approach.** The learning to rank approach – represented by the SVMRank implementation by (Joachims, 2002) – achieves similar results as the DM based on MLPs. Firstly, this proves that our feature design can also be applied in alternative ML models to solve the ranking problem. Secondly, this shows that the DM is able to compete with state-of-the-art learning to rank models.

4.2 Feature Analysis

In the following, we evaluate the impact of single features and specific feature groups that were defined in Section 3.1. Similar to the experiments presented above we use the expert retrieval scenario and perform a 3-fold cross-validation using the DM based on MLPs or logistic regression. Only the features under consideration are thereby used to train the classifier and as input for the DM.

In summary, our feature analysis consists of the following parts:

- Evaluation of the *aggregation functions* that are used to aggregate the probabilities of each query term to build features.
- Evaluation of the features that are based on *products of language models* that model the dependencies of query terms prior to the aggregation step.
- Evaluation of the *most important features* that contribute most to the retrieval performance.

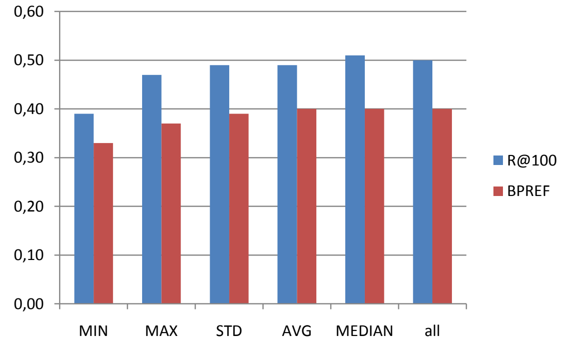

Figure 1: Retrieval results using the DM, MLP classifier and different features subsets. The MIN, MAX, AVG, MEDIAN and STD feature subset consist of all features that are based on the according aggregation functions of the query term probabilities.

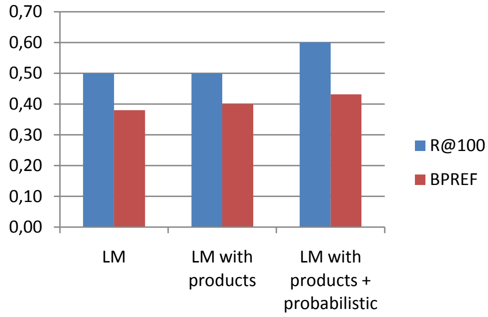

Figure 2: Retrieval results using the DM, MLP classifier and three different features subsets: all language model features, all language model features including products of up to three language models and all features including the three features derived from probabilistic retrieval models.

ID	*Feature*	*BPREF*
#1	MEDIAN[$\mathrm{PM}_{\text{freq}}$]	.37
#2	MEDIAN[$\mathrm{PM}_{\text{pagerank}}$]	.34
#3	TF.IDF	.26
#4	AVG[P_{profileC}]	.25
#5	AVG[P_{profile}]	.23
#6	MEDIAN[$P_{\text{cat-freq}}$]	.23
#7	MEDIAN[$P_{\text{cat-pagerank}}$]	.23
#8	STD[P_{profileC}]	.22
#9	MAX[P_{profileC}]	.21
#10	STD[P_{profile}]	.21
#11	MAX[P_{profile}]	.20
#12	DLH13	.18
#13	AVG[$P_{\text{cat-freq}}$]	.13
#14	BM25	.11
#15	MAX[$P_{\text{cat-freq}}$]	.10

Table 3: Retrieval performance of the DM with logistic regression. The classifier is trained and evaluated using single features.

Aggregation of Term Probabilities: For the feature design we used five aggregation function that aggregate the probabilities of all query term for a specific generative language model into one value: MIN, MAX, AVG, MEDIAN and STD (see Section 3). To evaluate these aggregation functions, we define set of features that only consist of the features constructed using a specific aggregation model. These sets of features were then used to train MLPs which were evaluated using 3-fold cross-validation on the CriES dataset. The results are presented in Figure 1.

The presented R@100 and BPREF values show that all aggregation functions define features that contain information about the relevance of experts. For all feature sets the results are comparable or better than the baseline given by the popularity model.

Comparing the different aggregation functions, the features based on the MEDIAN function achieve the best retrieval performance. Actually, using all language model features slightly declines the performance compared to the results using the MEDIAN feature subset. This licenses the conclusion that the median of all query term probabilities contains the most information about the relevance of the according experts.

Features from Products of Language Models: As the query term probabilities are aggregated over all query terms, dependencies between query terms can not be modeled. Therefore, we used products of language models that implicitly contain the dependencies of up to three different language models. In order to evaluate the effect of these features based on products of language models, we defined three different feature subsets: all features based on single language models, all features based on single and products of language models and all features including the features based on probabilistic retrieval models. The retrieval results using these feature subsets are presented in Figure 2.

The results show that the performance gain using products of language models is small, increasing BPREF by .02 only. Our conclusion is that the dependencies on term level across the language models are not important for modeling the relevance of experts.

In contrast, adding the features based on probabilistic language models leads to a substantial improvement of R@100 by .10. This shows that these features add more information about the relevance of experts.

Greedy Feature Selection: To measure the impact of each feature, we performed retrieval experiments based on the input of single features. In contrast to the feature analysis experiments presented above we used logistic regression as underlying ML model of

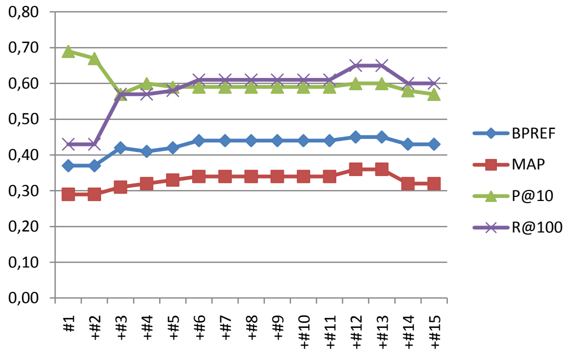

Figure 3: Retrieval results using the discriminative model, logistic regression and different features subsets. The set of features is based on a greedy feature selection of the top performing features.

the discriminative model. The reason for this is that MLPs do not show stable performance with few features only. The retrieval results of the top performing features are presented in Table 3.

Firstly, the ranking of features according to retrieval performance shows that the features based on the two different popularity models, $\mathrm{PM}_{\mathrm{freq}}$ and $\mathrm{PM}_{\mathrm{pagerank}}$, are most important. Secondly, all three features based on the probabilistic retrieval models are present in the top 15 features, which shows that they contain information about the relevance of experts. Thirdly, features based on the language models of expert profiles achieve high retrieval results. This includes features based on the global profile P_{profile} and on the category-specific profile $P_{\mathrm{profile+c}}$. Finally, the category language models $P_{\mathrm{cat\text{-}freq}}$ and $P_{\mathrm{cat\text{-}pagerank}}$ are also valuable features in the retrieval process. Interestingly, these features do not depend on the a priori knowledge of the target category.

In order to evaluate the combination of the most important features, we applied a greedy feature selection approach. Based on the ranking of features as presented in Table 3, we add the best performing set of features in a step-by-step greedy fashion, evaluating the performance of the classifiers at each step. The retrieval results of the greedy feature selection are presented in Figure 3.

The best P@10 values are achieved using features #1 and #2, which correspond to the popularity models of experts in the target category. Adding more features, e.g. #3 corresponding to the TF.IDF score, results in a drop of P@10. However, all other measures are improved, with a remarkable improvement w.r.t. R@100. Adding more features slightly improves BPREF, MAP and R@100 until features #13. Thereby P@10 stays on the same level.

Using only features #1-#13 as input for the logistic regression function actually leads to the best retrieval results in our experiments with BPREF of $.60 \pm 0.01$ and MAP of $.65 \pm 0.04$. Compared to the DM based on MLPs, BPREF is further improved by .02, MAP by .03.

This is a well-known phenomena in ML. In some cases, the input of a large set of features might "confuse" the classifier. The restriction of the set of features to the most important ones can then be used to improve the prediction performance. In our greedy feature selection, this effect can be observed when adding features #14 and #15 as the retrieval performance declines.

4.3 Summary of Results

In the following, we summarize all results presented in this paper. We also discuss some negative results obtained in our experiments. We think that these results might be helpful to avoid mistakes and dead ends in future research.

Best Retrieval Results with MLP-based DM: Standard retrieval approaches – probabilistic models as well as language models – do not perform well on the task of expert retrieval using the Yahoo! Answers dataset. Using mixture language models that also take into account the intra-categorial popularity of experts improve retrieval results substantially. However, the DM based on MLPs significantly outperforms the mixture language model in respect to BPREF, MAP and R@100.

Comparing to other learning to rank models, the MLP-based DM achieves similar results as ranking SVMs.

Feature Analysis in DM: The feature analysis showed that our approach to aggregate query term probabilities to define features is reasonable, with MEDIAN being the most predictive aggregation function. Further, our results indicate that the products of different language models prior to aggregation do not provide valuable information for the classification. Finally, the evaluation of single features showed that among the most predictive features are: $\mathrm{MEDIAN}[\mathrm{PM}_{\mathrm{freq}}]$ (the popularity model in the target category), TF.IDF (score of the vector space retrieval model) and $\mathrm{AVG}[P_{\mathrm{profile}}]$ (using the answer profiles of experts). These features are based on different sources of evidence, which shows that all sources contribute to measuring the expertise of users given a query.

Discrete Classifiers in Discriminative Model: Using classifiers with discrete output such as decision trees resulted in poor retrieval results. In this case many experts get the same or very similar scores, which makes them indistinguishable for ranking purposes. Cross-validation on the training set was usually comparable to MLP, but differences were huge when applying the classifier to the actual retrieval task (more than 50% performance drop).

Training Data: We performed experiments training on pairs of questions and experts providing the best answer as specified in the Yahoo! Answers dataset (i.e. not using the relevance judgments provided in the CriES dataset). Let's call these pairs *best-answer-pairs*. This did also yield unsatisfactory results. A possible explanation for the negative performance is the fact that negative examples where randomly sampled from *non-best-answer-pairs*, thus leading to a noisy dataset as many experts might be relevant for a given question in spite of not having provided the best answer.

5 RELATED WORK

There have been previous IR approaches dealing with data from SQA sites. Bian et al. (Bian et al., 2008) have proposed a system for factoid QA over Social Media. They present a general framework for factual IR that also exploits the social structure of the dataset. Their best model is a ranking function learned via supervised ML approaches. A related approach is the one of Agichtein et al. (Agichtein et al., 2008) which predicts the quality of answers in SQA portals and builds on a classifier trained on various features derived from text, meta data and user relationships. We have shown that supervised models can also be successfully applied to the task of ranking experts instead of answers in data from SQA sites.

Cao et al. (Cao et al., 2009) present an approach to use categorization information for Community Question Answering. They build a LM of category profiles and use a mixture model for answer retrieval. In this paper, we extend this idea to expert search by modeling the importance of an expert in a given category which enables the application of category smoothing in expert search. Cao et al. use empirical methods to optimize weights in the mixture model. In this paper we present the DM that relies on supervised ML to build a combined model.

Balog et al. (Balog et al., 2009) have presented an approach to expert retrieval based on LMs. They propose a model that builds on expert profiles comprising of all the documents written by a given expert and is thus similar to our text-profile based approach. In this paper, we compare the DM to a mixture language model approach that extends this model by integrating different evidence sources, in particular a category-based generative model, and by including support for multilingual retrieval. We show that the DM outperforms this mixture language model baseline.

Fang et al. (Fang et al., 2010) propose a discriminative model to integrate document evidence and document-candidate associations for expert search. Similar to the DM in this paper, they use available relevance assessments as training data. The features they use are based on language models as well as on similarity measures defined on specific properties of the dataset. This includes the extraction of names, email addresses and URLs. While they only generate one feature for the language model, we define several features. In addition, our approach can be applied to any type of dataset as it does not require the modeling of specific features.

Joachims (Joachims, 2002) introduced SVMRank as a learning to rank approach that exploits click-through data from Internet search engines, which is used to define the preference function. In contrast, we rely on the Gold Standard as training data. Our results show that – given the same input features and training data – our learning to rank approach has similar performance as SVMRank.

6 CONCLUSION

We have approached the problem of retrieving relevant experts for given information needs in the context of Social Question Answering sites such as Yahoo! Answers. We have shown that probabilistic models (BM25) and standard language models – considered as state-of-the-art – do not perform well on the task. We have proposed a novel approach to expert retrieval based on discriminative models optimized using machine learning techniques that substantially improve the retrieval performance. As part of this new retrieval model we presented a novel feature design that allows to use language models as input for learning to rank approaches. We use this approach to systematically optimize the combination parameters of different sources of evidence. These sources of evidence essentially comprise of text profiles of experts, category text profiles and approximations of the intra-categorical popularity of experts.

We have shown that our suggested discriminative model outperforms all baselines and has similar performance as SVMRank. We performed a detailed feature analysis and identified the most important features in the retrieval scenario. This shows the appropriateness of our feature design and also allows to further improve the retrieval performance by restricting the set of features that are used in the discriminative model.

ACKNOWLEDGEMENTS

This work was funded by the Multipla project[7] sponsored by the German Research Foundation (DFG) under grant number 38457858 as well as by the Monnet project[8] funded by the European Commission under FP7.

REFERENCES

Agichtein, E., Castillo, C., Donato, D., Gionis, A., and Mishne, G. (2008). Finding high-quality content in social media. In *Proceedings of the International Conference on Web Search and Web Data Mining (WSDM)*, pages 183—194, Palo Alto, California, USA. ACM.

Balog, K., Azzopardi, L., and de Rijke, M. (2009). A language modeling framework for expert finding. *Information Processing & Management*, 45(1):1–19.

Bian, J., Liu, Y., Agichtein, E., and Zha, H. (2008). Finding the right facts in the crowd: Factoid question answering over social media. In *Proceeding of the 17th International Conference on World Wide Web (WWW)*, pages 467–476, Beijing, China. ACM.

Buckley, C. and Voorhees, E. M. (2004). Retrieval evaluation with incomplete information. In *Proceedings of the 27th International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 25—32, Sheffield. ACM.

Cao, X., Cong, G., Cui, B., Jensen, C. S., and Zhang, C. (2009). The use of categorization information in language models for question retrieval. In *Proceeding of the 18th Conference on Information and Knowledge Management (CIKM)*, pages 265–274, Hong Kong, China. ACM.

Craswell, N., de Vries, A., and Soboroff, I. (2005). Overview of the TREC-2005 enterprise track. In *Proceedings of the 14th Text REtrieval Conference (TREC)*, pages 199–205.

Fang, Y., Si, L., and Mathur, A. P. (2010). Discriminative models of integrating document evidence and document-candidate associations for expert search. In *Proceedings of the 33rd International Conference on Research and Development in Infromation Retrieval (SIGIR)*, pages 683—690, Geneva.

Iftene, A., Luca, B., Carausu, G., and Merchez, M. (2010). Identify experts from a domain of interest. In *Notebook Reports of the CLEF Conference*, Padua.

Joachims, T. (2002). Optimizing search engines using clickthrough data. In *Proceedings of the 8th International Conference on Knowledge Discovery and Data Mining (KDD)*, pages 133—142, Edmonton.

Kürsten, J. (2009). Chemnitz at CLEF 2009 Ad-Hoc TEL task: Combining different retrieval models and addressing the multilinguality. In *Working Notes of the Annual CLEF Meeting*, Corfu.

Page, L., Brin, S., Motwani, R., and Winograd, T. (1999). The pagerank citation ranking: Bringing order to the web. Technical Report 1999-66, Stanford InfoLab.

Quinlan, J. R. (1993). *C4. 5: programs for machine learning*. Morgan Kaufmann.

Robertson, S. E. and Walker, S. (1994). Some simple effective approximations to the 2-Poisson model for probabilistic weighted retrieval. In *Proceedings of the 17th International Conference on Research and Development in Information Retrieval (SIGIR)*, pages 232—241, Dublin. Springer.

Savoy, J. (2005). Data fusion for effective european monolingual information retrieval. In *Multilingual Information Access for Text, Speech and Images*, pages 233—244.

Sorg, P., Cimiano, P., Schultz, A., and Sizov, S. (2010). Overview of the cross-lingual expert search (CriES) pilot challenge. In *Notebook Reports of the CLEF Conference*, Padua.

Surdeanu, M., Ciaramita, M., and Zaragoza, H. (2008). Learning to rank answers on large online QA collections. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 719–727, Columbus, Ohio.

Yimam-Seid, D. and Kobsa, A. (2003). Expert-Finding systems for organizations: Problem and domain analysis and the DEMOIR-Approach. *Journal of Organizational Computing and Electronic Commerce*, 13(1):1.

[7] http://www.multipla-project.org/

[8] http://www.monnet-project.eu/