SPIDER Retrieval System at TREC-5

Jean-Paul Ballerini, Marco Büchel, Ruxandra Domenig,
Daniel Knaus, Bojidar Mateev, Elke Mittendorf,
Peter Schäuble, Páraic Sheridan, Martin Wechsler

Swiss Federal Institute of Technology (ETH)
CH-8092 Zürich, Switzerland

Abstract

The ETH group participated in this year's TREC in the following tracks: automatic adhoc (long and short), the manual adhoc, routing, and confusion. We also did some experiments on the chinese data which were not submitted.

While for adhoc we relied mainly on methods which were well evaluated in previous TRECs, we successfully tried completely new techniques for the routing task and the confusion task: for routing we found an optimal feature selection method and included co-occurrence data into the retrieval function; for confusion we applied a robust probabilistic technique for estimating feature frequencies.

1 Introduction

ETH's contribution to TREC-5 consists of a new routing approach and robust probabilistic technique for the confusion track. Both approaches have been tuned by means of an essentially known reference method (Lnu.ltn, phrase indexing, query expansion). For comparison reasons, this reference method has been applied to the adhoc task and the results were submitted as official runs.

For the routing task, we focused on feature selection and using co-occurrence data. The latter approach evolved from an approach which has been successfully used in image synthesis. The basic idea is that a synthetic surface and a real surface look similar if their second order statistics are similar. In computer graphics this relationship is used to generate artificial surfaces, e.g. for virtual surgery [Hug, 1996].

Our contribution to the confusion track mainly relied on earlier projects where we learned how to deal with corrupted data. In speech retrieval phoneme recognition rates of 50% are observed when speech recognition is independent of the speakers [Schäuble et al., 1997]. Optical Character Recognition (OCR) may achieve word recognition rates of about 70% in the case of old and partially handwritten documents [Mittendorf et al., 1995]. Fortunately, low recognition rates do not necessarily imply poor retrieval effectiveness. It was shown that the effects of recognition error may cancel out to certain extent if the documents are long enough [Mittendorf & Schäuble, 1996].

The paper is structured as follows. We first describe the reference method applied to adhoc test (Section 2). Our approaches to the routing and confusion problem are described in Section 3 and 4. Our first attempt to Chinese text retrieval is described in Section 5.

In our terminology we denote the TREC disks, disk2, disk3, disk4, by D2, D3, and D4, last years routing data is denoted by R4 (mainly consisting of net data), and this years routing data is denoted by R5 (consisting of FBIS). For the definitions of our symbols refer to the symbols table in appendix A.

2 Adhoc Retrieval

For comparison reasons we applied our *reference method* to the adhoc task and submitted the results as official runs. For the adhoc task our group mainly adapted the methods reported by the SMART group in last year's TREC [Buckley et al., 1996], namely the Lnu.ltn weighting scheme, "automatic relevance feedback" by top ranked documents

for automatic adhoc and manual feedback for manual adhoc. Both feedback methods use Rocchio reweighting. We also implemented phrase indexing. We modified these well-known methods slightly by changing the feature selection method for the Rocchio reweighting and including an indexing capability to index both, British and American English simultaneously.

Indexing

Our indexing vocabulary $\Phi := \Phi_1 \cup \Phi_2$ consists of single words reduced by the Porter algorithm [Porter, 1980] $\varphi_h \in \Phi_1$ called the one-word features, and 2-word-phrases $\varphi_i \in \Phi_2$, which will be called 2-word features. A valid single word has a minimal length of three characters. The removal of important words shorter than this is prevented by using an anti-stopword list [Knaus et al., 1995]. Also, stopwords are removed [Salton, 1971]. A potential 2-word-phrase consists of any pair of adjacent non-stopwords as proposed in [Buckley et al., 1992]. The final list of phrases was composed using only phrases with a document frequency greater than 25 in Disk2 and Disk3 (D2D3) of the TREC collection. Table 1 shows the number of phrases for different minimal document frequencies. As explained later in this section, phrases do not contribute to the document length in the weighting schemes we used.

minimal df	# 2-word features
1	7'000'000
2	2'700'000
5	1'000'000
25	200'000
50	99'000

Table 1: Number of 2-word features for various document frequencies

For documents written in British English we introduced an additional procedure that maps single words into American English. For British-American synonyms (e.g. film–movie, queue–line) we constructed a lookup table (41 words) that was applied before reducing words. Another lookup table (272 words) was used after word reduction to account for different spellings (e.g. analyse–analyze, colour–color). Both lists were collected from various sources in the World Wide Web.

Basic Weighting Scheme

The Retrieval Status Values (RSV) are obtained by

$$\mathrm{RSV}(q, d_j) = \sum_{\varphi_i \in \Phi(q,d_j)} a_{ij} \cdot b_i, \tag{1}$$

where $\Phi(q, d_j)$ denotes the set of 1-word or 2-word indexing features occurring in both the query q and the document d_j, a_{ij} denotes the weight of the document feature and b_i denotes the weight of the query feature.

For the (automatic and manual) adhoc tracks as well as for baseline methods for routing and confusion tracks we used *Lnu.ltn* weighting [Singhal et al., 1996] as our basic weighting scheme. Subsequently $\mathrm{ff}(\varphi_i, d_j)$ denotes the feature frequency (number of occurrences) of φ_i in d_j and $\mathrm{nidf}(\varphi_i)$ denotes the normalized inverse document frequency $\mathrm{nidf}(\varphi_i) := 1 - \frac{\log(\mathrm{df}(\varphi_i)+1))}{\log(n+1)}$, where n denotes the number of documents in the collection and $\mathrm{df}(\varphi_i)$ denotes the document frequency (the number of documents φ_i occurs in). The document length $l^{\Phi_1}(d_j)$ is the number of one-word features only, whereas the average feature frequency ($\overline{\mathrm{ff}}^{\Phi}(d_j)$) is always the average over all features (regardless of the feature type):

$$a_{ij} = \frac{1}{(1-s)\cdot p + s \cdot l^{\Phi_1}(d_j)} \cdot \frac{1+\log(\mathrm{ff}(\varphi_i, d_j)}{1+\log(\overline{\mathrm{ff}}^{\Phi}(d_j))} \tag{2}$$

$$b_i = \mathrm{nidf}(\varphi_i) \frac{1+\log(\mathrm{ff}(\varphi_i, q))}{1+\log(\overline{\mathrm{ff}}^{\Phi}(q))}, \tag{3}$$

where s denotes the slope and p denotes the pivot. The pivot is computed as the average document length ($p = \overline{l}^{\Phi_1}(d_j)$).

Query Expansion with Relevance Feedback and Feature Selection

Depending on the task there are different *sources of relevance information*. We denote the set of documents assumed to be relevant as D^{rel}. For both automatic adhoc and manual adhoc the same relevance feedback method is applied given D^{rel}.

- **Automatic Adhoc**
  In a first pass all original queries are evaluated without any feedback information and ranked lists are retrieved. We then assume that the r top ranked documents are relevant. In the second pass the original queries are expanded by using this relevance information and the 1000 top ranked documents are determined.

- **Manual Adhoc**
  An interactive manual process done by the users determines a set of relevant documents for each topic. The queries are formulated by the user and may be modified during the searching process.
  The system used allowed relevance feedback searching as well as highlighting 1-word and 2-word features. The users were requested to justify their relevance judgements for each document. Each topic was investigated by only one user, therefore the relevance judgements are strongly user dependent.
  The user population was highly homogeneous since all of them are non-native English speakers and work in the field of information retrieval.
  No time constraints were set.

Given the set of relevant documents we *select h features* from the original query q and the relevant documents. Let $\Phi^{sel} = \Phi(q) \cup \bigcup_{d_j \in D^{rel}} \Phi(d_j)$ be the set of features from the original query and from all relevant documents. All features of Φ^{sel} are ranked by the following method (called ltn-selection) and the h top ranked features are selected to build together with all original query features the new query q'.

$$sel_{ltn}(\varphi_i) = \alpha \cdot b_i + \beta \cdot \frac{1}{|D^{rel}|} \cdot \sum_{d_k^{rel} \in D^{rel}} b_{ik}^{rel} \tag{4}$$

where for all $d_k^{rel} \in D^{rel}$

$$b_{ik}^{rel} = \begin{cases} \mathrm{nidf}(\varphi_i) \frac{1+\log(\mathrm{ff}(\varphi_i, d_k^{rel}))}{1+\log(\overline{\mathrm{ff}}^{\Phi}(d_k^{rel}))} & : \quad \varphi_i \in d_k^{rel} \\ 0 & : \quad else \end{cases}$$

The normalisation by the average feature frequency $\log(\overline{\mathrm{ff}}^{\Phi}(q))$ of the original query is needed to have the weights of original query features and the weights of feedback document features on the same scale.

We also tried other feature selection methods, but they all performed worse (in terms of effectiveness). The only one that performed nearly equally well was the df_{rel}-selection where the number of documents in D^{rel} containing the feature was used to rank the features ($sel_{df}(\varphi_i) = |\{d_k^{rel} \in D^{rel} | \varphi_i \in d_k^{rel}\}|$). This is the feature selection method reported in [Buckley et al., 1996].

After query expansion the following Rocchio reweighting was applied to the expanded query q':

$$\mathrm{RSV}^{\mathrm{feedback}}(q', d_j) = \sum_{\varphi_i \in \Phi(q', d_j)} b_i' \cdot a_{ij} \tag{5}$$

where

$$b_i' = sel_{ltn}(\varphi_i)$$

Final runs

The TREC-4 data has been used to optimize some of the parameters. For TREC-5 we submitted the following experiments:

	$\geq$ Median	$<$ Median	Average Precision (over all topics)
automatic short	25	25	0.1726
automatic long	41	9	0.2425
manual feedback	43	7	0.3165

Table 2: ETH Results for Adhoc Tasks.

- **ETHal1**
  automatic adhoc: queries 251-300 versus D2D4, long queries (all fields),
  automatic query expansion with $r = 20$,
  feature selection with ltn-selection, $h = 50, \alpha = 1.0, \beta = 1.0$
  $Lnu.ltn$-weighting with slope $s = 0.24$, top 1000 ranks retrieved.
- **ETHas1**
  automatic adhoc: queries 251-300 versus D2D4, short queries (description field only),
  automatic query expansion with $r = 20$,
  feature selection with ltn-selection, $h = 50, \alpha = 1.0, \beta = 1.0$
  $Lnu.ltn$-weighting with slope $s = 0.24$, top 1000 ranks retrieved.
- **ETHme1**
  manual adhoc: queries 251-300 versus D2D4, long queries (all fields),
  manual search for relevant documents,
  query expansion with all found relevant documents,
  feature selection with ltn-selection, $h = 50, \alpha = 1.0, \beta = 1.0$
  $Lnu.ltn$-weighting with slope $s = 0.24$, top 1000 ranks retrieved.

Results We report the results of the final runs in table 2. As expected, the automatic long method had better results than the automatic short method, even though some topic descriptions could have been misleading.

The manual feedback improved the average precision significantly. This indicates, that—in contrast to last year's interactive TREC—the user's and the assessor's relevance judgements were rather similar although our users were non-native English speakers. In addition, the lack of any time constraints enables the searcher to search in his/her own pace without any hurry. However, this is comparable to a real user's search-task, where the search is finished, when the user does not expect the system to retrieve any more relevant information. For these experiments users needed 30 to 40 minutes per topic.

3 Routing

For the first time the ETHZ group attacked the routing problem seriously. We investigated two problems: optimal feature selection and the use of co-occurrence data in the routing environment.

3.1 Feature Selection

We evaluated feature selection methods that are based on contingency tables:

	$\#\{d_j \mid d_j \in R\}$	$\#\{d_j \mid d_j \notin R\}$	
$\#\{d_j \mid \varphi_i \in d_j\}$	x_{11}	x_{12}	$x_{1.}$
$\#\{d_j \mid \varphi_i \notin d_j\}$	x_{21}	x_{22}	$x_{2.}$
	$x_{.1}$	$x_{.2}$	$x_{..}$

We used the U-measure that is defined for the U-test [Andersen, 1992] for feature selection:

$$\mu_U(\varphi_i) := \sqrt{x_{..}} \frac{x_{11}x_{22} - x_{12}x_{21}}{\sqrt{x_{1.}x_{.1}x_{2.}x_{.2}}}. \tag{6}$$

Note the similarity to the χ^2-measure,

$$\mu_{\chi^2}(\varphi_i) := x_{..} \frac{(x_{11}x_{22} - x_{12}x_{21})^2}{x_{1.}x_{.1}x_{2.}x_{.2}}. \tag{7}$$

That means, that

$$\mu_U(\varphi_i) = \begin{cases} \sqrt{\mu_{\chi^2}(\varphi_i)} & x_{11}x_{22} \geq x_{12}x_{21} \\ -\sqrt{\mu_{\chi^2}(\varphi_i)} & x_{11}x_{22} < x_{12}x_{21}. \end{cases} \tag{8}$$

Experiments which used the χ^2-measure for feature selection showed only moderate results [Schütze et al., 1995]. We briefly explain why the U-measure is well suited for feature selection, and certainly better suited than the χ^2-measure. The χ^2-measure rewards positive *and* negative correlation, i.e. μ_{χ^2} yields large values if $x_{11}x_{22} \ll x_{12}x_{21}$ *and* if $x_{11}x_{22} \gg x_{12}x_{21}$. The measure μ_U only rewards positive correlation. Thus a feature whose absence implies relevance or for which presence implies irrelevance is a good χ^2-feature but a bad U-feature (negative numbers with large absolute values). Since almost all retrieval functions do not make use of negative evidence the U-measure is the better feature selection method for most retrieval methods.

Experiments with U-features

We evaluated $\mu = \mu_U$, $\mu = \mu_{\chi^2}(\varphi_i)$ and several other measures μ (Robertson Selection Value [Robertson et al., 1995], F4-formula [Eftimiadis, 1993], measure used in [Allan et al., 1996], etc.) on the contingency table with the following experimental setting:

1. Build contingency tables $(x_{11}, x_{12}, x_{21}, x_{22})$ on the training data for all features which occur in at least one relevant document ($x_{11} \geq 1$), do not occur in at least one irrelevant document ($x_{22} \geq 1$), occur in at least 15 documents ($x_{1.} \geq 15$), and do not occur in at least 15 documents ($x_{2.} \geq 15$),
2. Rank all features according to $\mu(\varphi_i)$.
3. Select the N_1 top ranked 1-word features $\varphi_l \in \Phi_1$ and the N_2 top ranked 2-word features $\varphi_k \in \Phi_2$ and let $q := \{\varphi_{l_0}, \ldots, \varphi_{l_{N_1-1}}, \varphi_{k_0}, \ldots, \varphi_{k_{N_2-1}}\}$.
4. Rank the documents of the test collection according to $\mathrm{RSV}_{Lnu.ltn}(q, d_j)$.

We used the TREC-4-routing queries and D2D3 for training and the TREC-4-routing data (R4) for testing. The features selected by μ_U consistently produced the best average precision values or values that were among the best for different query sizes (N_1, N_2), the experiments are reported in [Mateev, 1996].

We chose $N_1 := 50$ and $N_2 := 20$ which yields an average precision of 0.3103 with $RSV_{Lnu.ltn}$ on $R4$, we will denote this method as $RSV_{Lnu.ltn:50:20}$. We found out later that $N_1 := 60$ and $N_2 := 30$ yield slightly better results with the same experimental setting (0.3172).

The U-measure is a measure that is normalized over all marginal values (i.e. $(x_{1.}, x_{.1}, x_{2.}, x_{.2})$). Therefore it is predestined to rather select all features above a certain threshold rather than a fixed number of top ranked features. However, we did not experiment with such thresholds.

3.2 Using Co-occurrence Data for Routing Retrieval

There are several attempts reported in information retrieval literature that make use of the co-occurrence of features, see e.g. [van Rijsbergen, 1977]. These attempts brought only slight improvements if any at all. However, none of the previous approaches were able to use as much training data as one is able to use for the routing task. We therefore felt encouraged to attempt to use the co-occurrence of features for the routing task.

Description of Co-occurrence Indexing and Retrieval

We tried a straightforward approach of generalizing the Binary Independence Retrieval model (BIR) and the Robertson—Sparck Jones weighting formula.

1. Choose a sequence of feature distances e.g. $\Delta^{(0)} := 0$, $\Delta^{(1)} := 1$, $\Delta^{(2)} := 10$, $\Delta^{(3)} := 30$, $\Delta^{(4)} := 200$.

2. Fix a set of N 1-word features $\varphi_i \in \Phi_1$.

3. Define the description matrix (in generalization of a description vector) $d_j^{(i)} := (a_{j,lk}^{(i)})_{l,k=0}^{N-1}$, such that

$$a_{j,lk}^{(i)} := \begin{cases} 1 & \text{if there exists an occurrence of } \varphi_k \text{ at least } \Delta^{(i-1)}+1 \\ & \text{and at most } \Delta^{(i)} \text{ tokens after an occurrence of } \varphi_l, \\ 0 & \text{else.} \end{cases} \tag{9}$$

   Note that the matrices $d_j^{(i)}, i = 1, \ldots, 4$ are usually not symmetric, and that the matrix $d_j^{(0)}$, is a diagonal matrix with the diagonal describing the binary description vector of a document used for probabilistic retrieval (in this case we define $\Delta^{(-1)} := -1$).

4. Define the query weight matrices $q^{(i)} := (b_{lk}^{(i)})_{l,k=0}^{N-1}$,

$$b_{lk}^{(i)} := \log \frac{p_{lk}^{(i)}(1-q_{lk}^{(i)})}{q_{lk}^{(i)}(1-p_{lk}^{(i)})}, \tag{10}$$

   where $p_{lk}^{(i)} := P(d_{j,lk}^{(i)} = 1|R)$ and $q_{lk}^{(i)} := P(d_{j,lk}^{(i)} = 1|\overline{R})$, The probabilities $p_{lk}^{(i)}$ and $q_{lk}^{(i)}$ are estimated by relative frequencies on the training data. To get robust estimates even for zero-values, we changed to the so-called 0.5-formula,see e.g. [Robertson et al., 1995].

5. Define Retrieval Status Values:

$$\mathrm{RSV}^{(i)}(q, d_j) := \mathrm{Tr}(d_j^{(i)} q^{(i)T}), \tag{11}$$

   where Tr denotes the trace of a matrix and A^T denotes the transposition of matrix A.

To clarify the ideas behind this co-occurrence retrieval methods we make the following remarks.

- The $\mathrm{RSV}^{(0)}$ is exactly the retrieval status value of the BIR, $\mathrm{RSV}^{(1)}$ comprises a retrieval based on a kind of "phrase indexing".
- The choice of $\Delta^{(0)}, \ldots, \Delta^{(4)}$, is very intuitive. The vague idea was to index the original terms $\Delta^{(0)}$, index phrases $\Delta^{(1)}$, index the co-occurrence in a local region of sentence length $\Delta^{(2)}$, of paragraph length $\Delta^{(3)}$, and of average document length $\Delta^{(4)}$.
- None of the retrieval status values makes use of feature frequency information, but implicitly the information of feature frequency is used in $\mathrm{RSV}^{(1)}$, $\mathrm{RSV}^{(2)}$, $\mathrm{RSV}^{(3)}$, and $\mathrm{RSV}^{(4)}$ since a feature with high frequency has a better chance to co-occur with another feature.
- Since we do not explicitly use feature frequency information, document normalization is not necessary (though a good normalization might help).
- Each RSV can be perfectly motivated in the framework of probabilistic retrieval.
- Since only co-occurrences in a window of predefined length are used, the co-occurrence indexing carries valuable local information.

Combination

Since each $\mathrm{RSV}^{(i)}$ is based on a different kind of indexing, we can expect the descriptions to be rather independent. Thus $\mathrm{RSV}^{(0)}, \ldots, \mathrm{RSV}^{(4)}$ proffer themselves for combination. Since in our preliminary experiments $\mathrm{RSV}^{(4)}$ did not produce good results we only combined $\mathrm{RSV}^{(0)}, \ldots, \mathrm{RSV}^{(3)}$, *and* $\mathrm{RSV}_{Lnu.ltn:50:20}$, but $\mathrm{RSV}^{(4)}$ was still used for training of the combination parameters. We combined the RSV-values linearly and determined the combination parameters with logistic regression on a per query basis, where the combination values were trained on D1D2D3. The result of this combination was submitted as our final run **ETHru1**.

Data	$RSV_{Lnu.ltn:50:20}$	N	$RSV^{(0)}$	$RSV^{(1)}$	$RSV^{(2)}$	$RSV^{(3)}$	$RSV^{(4)}$
R4	0.3103	20	0.2867	0.2803	0.3099	0.3073	0.2835
R4	—	60	0.2646	0.2976	0.3168	0.3090	0.2610
R5	0.2046	20	0.1839	0.1578	0.1915	0.1863	0.1812

Table 3: Average Precision of 45 Topics vs R5 (FBIS) and of 50 Topics vs R4 (TREC-4 Routing Data)

Data	Average Precision	Percent improvement over Lnu.ltn:50:20
R4	0.3405	10%
R5	0.2401	17%

Table 4: Combination Results

Experiments

We trained the matrices $q^{(i)}$ on D1D2D3. Since at the moment the SPIDER retrieval system does not administer position information we did not have the chance to compute $RSV^{(i)}, i = 1, \ldots, 4$ on the complete set of documents. So we preselected a set of 1000 Documents for each query by computing the $RSV_{Lnu.ltn:50:20}$. We re-ranked the documents according to $RSV^{(0)}, \ldots, RSV^{(4)}$. The average precision values for each of the RSV is reported in table 3.

The results of the combination are reported in table 4.

Table 5 shows how the co-occurrence ranking method does compares with other submitted methods. We report the number of topics that are above or below the median. In brackets we report the numbers for the 39 queries with more than two relevant documents.

Analyzing Co-occurrence Indexing and Retrieval

- The described retrieval method is a good retrieval method for the routing environment though not one of the best.
- The average precision values in table 3 indicate that co-occurrence of terms is a very important source of information for retrieval. It it surprising that probabilistic retrieval on feature pairs that occur within 10 tokens ($RSV^{(2)}$) performs so significantly better than retrieval on conventional 1-word features ($RSV^{(0)}$).
- The occurrence of features ($RSV^{(0)}$), the co-occurrence within a region of more than one and at most 10 tokens ($RSV^{(2)}$), and the co-occurrence within a region of more than 10 and at most 30 tokens ($RSV^{(3)}$) are good indicators of relevance, whereas phrases and the co-occurrence of features that are more than 30 tokens apart from each other seem to be less well suited.
- The methods are well suited for combination (see table 4).
- Table 5 indicates that the preselection of the first 1000 documents is not good enough. A run of the co-occurrence ranking without a preselection would probably be better. However the preselection was only made because of the lack of position information in our index structure. A different implementation would make the preselection obsolete.

	Relevant Retrieved @ 1000	R-Precision	Average Precision
$\geq$ Median	31 (25)	36 (30)	32 (30)
$<$ Median	14 (14)	9 (9)	13 (9)

Table 5: Comparison of Co-occurrence Ranking ETHru1 on 45 Topics (39 Topics)

We gained the conviction that the information about co-occurrence is a very valuable indicator of relevance. However, while developing our co-occurrence retrieval method we made a lot of ad-hoc decisions (selection of $\Delta^{(i)}$, document description with matrices, Robertson—Sparck Jones weighting, combination, etc.) that should be given further thoughts.

4 Confusion track

For the confusion track our goal was mainly to apply the probabilistic matching of terms that was developed for the retrieval of library cards [Mittendorf et al., 1995] and speech retrieval [Wechsler & Schäuble, 1995] to the different problem of retrieval of rather long OCR-corrupted documents in a monolingual collection. In addition to the probabilistic matching we hoped to be able to detect systematic errors for which we could compensate.

Since our probabilistic matching method detects a large number of similar query word occurrences, it is computationally too expensive to run the algorithms on all documents. Therefore, for each query, we preselected 2000 documents of the FR94-degrade5 (d5) collection and 2000 documents of the FR94-degrade20 (d20) collection by indexing documents with overlapping N-gram features (N consecutive alphanumeric characters with at most one separation character) and retrieving the document with the *Lnu.ltn*-weighting scheme. We decided to set $N := 4$ for the d5 collection and after having a short glimpse at d20 we thought that this collection might be too corrupted for 4-grams and set $N := 3$ for this collection. (After submitting our final runs we did some comparisons of the ranking on the perfect documents and the corrupted documents and we now think that the decision for 3-grams was probably too anxious.) We submitted these rankings as ETHD5N and ETHD20N.

Starting from those N-gram based ranking lists, we performed a re-ranking using the following strategy. We decided to take a weighting formula with pivoted byte size normalization, suggested by Singhal, Buckley, and Mitra [Singhal et al., 1996]

$$\mathrm{RSV}(q,d_j) := \frac{1}{(1-s)\cdot p + s\cdot l^{\mathrm{byte}}(d_j)} \sum_{\varphi_i \in q, \mathrm{Eff}(\varphi_i,d_j)>0} (1+\log(\mathrm{ff}(\varphi_i,q)))\cdot \mathrm{idf}(\varphi_i)\cdot(1+\log(\mathrm{Eff}(\varphi_i,d_j))) \qquad (12)$$

where $l^{\mathrm{byte}}(d_j)$ denotes the length of the document in bytes. The corrupted $\mathrm{idf}(\varphi_i)$ in d5 and d20 were substituted by the idf in last year's ad-hoc data D2D3. The slope was set to $s := 0.3$ and the pivot was set to the average byte size of documents in d5 or d20, $p := \overline{l}^{\mathrm{byte}}$. The values $\mathrm{Eff}(\varphi_i,d_j)$ are supposed to be robust estimates of the feature frequency in the document, we will describe below how we get these estimates. For each query we performed the following steps:

- Index the query using stopword removal and Porter reduction [Porter, 1980]. For these experiments we did not apply phrase indexing for technical reasons.
- To each position in the document an accumulator is allocated that counts the number of matching characters between the substring starting at this position and the feature. To each character of the feature, the set of accumulators to be incremented is determined using a character index of the document and the character's relative position in the feature. This yields a set of potential feature beginnings. To account for insertion and deletion errors, we investigate the accumulators in a small range around the potential feature beginnings and increment the accumulator with the highest count. The accumulators values are then used to determine the beginnings of valid slots. A valid slot s_i must contain at least P percent of the feature's characters (P is referred to in Table 6). The slot end is either determined by aligning the last two characters or using the feature's length. Note that the features are reduced words with a starting delimiter, e.g. for the word *replacement* the string *_replac* is matched to the documents. This delimiter prohibits subword matching, which we think is useful for English (e.g. the word 'ring' should not match bothering).
- For each slot s_l, compute an edit distance $d(\varphi_i, s_l)$ between the slot and the feature. We used a distance function that has already been described in [Mittendorf et al., 1995].
- For each slot s_l, based on the edit distance, compute the probability p_l that s_l contains an occurrence of the feature using the edit distance. A reliable probability estimation function would require training data (perfect and corrupted data), which we did not have time to produce for the confusion track. So we applied a very crude

	P	d5		d20	
		α	res	α	res
nff	–	1.014	3.44	1.939	5.352
eff	50%	0.765	2.714	0.954	4.428
	60%	0.816	2.465	1.256	3.978
	70%	0.841	2.426	1.315	3.98
	75%	0.847	2.792	1.3602	4.191

Table 6: Residual Standard Error (res) of fitted lines for various P and α.

probability estimation:

$$p_l := \begin{cases} 1 & \text{if } \frac{d(s_l, \varphi_i)}{\text{len}(\varphi_i)} \leq 0.2 \\ 0 & \text{else,} \end{cases} \tag{13}$$

where $\text{len}(\varphi)$ is the number of characters in the query feature. The constant 0.2 is dependent on the edit distance function and was determined empirically. We will conduct more detailed analysis of the edit distance function and probability estimation.

- For each document and each feature, compute the expected feature frequency:

$$\text{eff}(\varphi_i, d_j) = \sum_{l=0}^{p-1} p_l \,. \tag{14}$$

- Search for systematic errors of the expected feature frequency eff and compensate for them:

$$\text{eff}(\varphi_i, d_j) \mapsto \text{Eff}(\varphi_i, d_j) \,.$$

We selected 100 documents arbitrarily out of $d20$ (since $d20 \subset d5 \subset$ FR94) and compared in the 100 triples of documents (FR94, $d5$, $d20$) the feature frequencies ff (in FR94) with the noisy feature frequencies nff (feature frequencies in $d5$ and $d20$) and with the expected feature frequencies eff ($d5$ and $d20$). We fitted scatterplots by a line: $\text{ff}(\varphi_i, d_j) = \alpha\text{nff}(\varphi_i, d_j)$ and $\text{ff}(\varphi_i, d_j) = \alpha\text{eff}(\varphi_i, d_j)$, respectively. The results for different values for P (slot selection threshold) and α are reported in Table 6. To minimize the residual standard error (res), we decided to use the function

$$\text{Eff}(\varphi_i, d_j) := \alpha\text{eff}(\varphi_i, d_j),$$

with the following parameters:

$$d5 : \; P := 70\% \; \alpha := 0.8413 \tag{15}$$

$$d20 : \; P := 60\% \; \alpha := 1.2562. \tag{16}$$

Final runs

The final runs were submitted using the method described above as **ETHD5P** and **ETHD20P**. The pre-selection run on N-grams (4-gram and 3-gram) with Lnu.ltn weighting were submitted as **ETHD5N** and **ETHD20N**. A normal Lnu.ltn run with the basic method described in section 2 was submitted as **ETHFR94**.

Issues for improvement

The key part of this method is the probability estimation function, which was defined very crudely for this track. Alternative functions, maybe with additional information (e.g. the number of equal characters), could help to improve probability estimation. The edit distance function itself might be optimized using a character confusion matrix (which however is difficult to obtain).

On the other hand, the compensation $\text{eff}(\varphi_i, d_j) \mapsto \text{Eff}(\varphi_i, d_j)$ could be made dependent on recognition characteristics of the features, e.g. with the help of a confusion matrix on characters. Further experiments and analysis have to be done in this matter.

Results

We realized that we made a mistake applying the weighting formula (12) as it is since $\mathrm{Eff}(\varphi_i, d_j) \in [0,1]$ can happen. Because of the small compensation factor $\alpha = 0.8413$ for d5 and because of the logarithmic weighting negative weights can occur quite often. This is actually wrong and should not happen. This explains the bad results for ETHD5P (expected run length: 205.06, worse than the median over all submitted rankings). We did not yet do experiments with a better weighting scheme. However, we are sure that our approach is a good approach since for d20 the probabilistic method ETHD20P has the best result of all submitted rankings (expected run length: 115.41), in spite of the "wrong" weighting formula. For d20 we used a compensation factor $\alpha = 1.2562$ and thus there appeared no undesired negative weights.

We were surprised that ETHD5N, which was only thought to be a ranking for preselection, is the best submitted method for d5 (expected run length: 25.10). We will analyze the results further after the conference. In particular, we will adjust the weighting scheme.

5 Chinese Track

Another first for the ETHZ group this year involved running experiments on the Chinese data of the multilingual track, though these experiments were conducted after the submission deadline and so are unofficial. We used simple indexing techniques involving n-grams and then investigated various combinations of retrieval techniques available through the SPIDER system. These include the use of automatic relevance feedback using the top r ranked documents of an initial retrieval run and the use of similarity thesauri for query expansion.

To date we have indexed the Chinese collection using both 1-grams and 2-grams. Even with these two approaches we have noted a very significant difference in the resulting index. Indexing the 164,788 documents with 1-grams (each individual Chinese character becomes an indexing feature) gives an indexing vocabulary of 10,701 features. Using 2-grams results in a vocabulary of 2,089,778 features! This is without stopword removal since we do not yet have a list of Chinese stopwords.

On each of the two indexes, we have run a baseline experiment using cosine weighting ($lnc.ltn$) and experiments with automatic feedback with r set to each of 3, 5, and 10 (ltn-selection, $h = 50, \alpha = 1.0, \beta = 1.0$). Against these baselines we also performed experiments using the Lnu.ltn pivoted normalisation weighting scheme, both with and without automatic relevance feedback loops, plus a series of experiments using further query expansion through the use of similarity thesauri. Our most successful experimental setup on the TREC5 data was the use of Lnu.ltn weighting with an automatic feedback loop of the top ten retrieved documents expanded to 50 terms, and without further expansion through a similarity thesaurus. We hope to validate these results and explore further possibilities through participation in the TREC-6 conference.

Achnowlegements: The authors would like to thank Markus Flückiger and Daniel To for their help with the implementation, and the realization of the experiments.

A Symbols

Symbol	Variable Name	Definition
Φ_1		set of one word features
Φ_2		set of two word features (phrases)
Φ		$\Phi_1 \cup \Phi_2$
$l^{\Phi}(d_j)$	doc_len_all_feat_dj	length of document d_j counted as number of features, $l^{\Phi}(d_j) := \sum_{\varphi_i \in \Phi, \varphi_i \in d_j} 1$
$l^{\Phi_1}(d_j)$	doc_len_1w_feat_dj	length of document d_j counted as number of one word features, $l^{\Phi_1}(d_j) := \sum_{\varphi_i \in \Phi_1, \varphi_i \in d_j} 1$
$l^{\Phi_2}(d_j)$	doc_len_ph_feat_dj	length of document d_j counted as number of two word features, $l^{\Phi_2}(d_j) := \sum_{\varphi_i \in \Phi_2, \varphi_i \in d_j} 1$
$\overline{\text{ff}}^{\Phi}(d_j)$	avg_ff_all_feat_dj	average feature frequency in document d_j over all features, $\frac{1}{l^{\Phi}(d_j)} \sum_{\varphi_i \in \Phi} \text{ff}(\varphi_i, d_j)$
$\overline{\text{ff}}^{\Phi_1}(d_j)$	avg_ff_1w_feat_dj	average feature frequeny in document d_j over all one word features, $\frac{1}{l^{\Phi_1}(d_j)} \sum_{\varphi_i \in \Phi_1} \text{ff}(\varphi_i, d_j)$
$\overline{\text{ff}}^{\Phi_2}(d_j)$	avg_ff_ph_feat_dj	average feature frequency over all two word features in document d_j, $\frac{1}{l^{\Phi_2}(d_j)} \sum_{\varphi_i \in \Phi_2} \text{ff}(\varphi_i, d_j)$
$\overline{l}^{\Phi}$	avg_doc_len_all_feat	average of $l^{\Phi}(d_j)$
$\overline{l}^{\Phi_1}$	avg_doc_len_1w_feat	average of $l^{\Phi_1}(d_j)$
$\overline{l}^{\Phi_2}$	avg_doc_len_ph_feat	average of $l^{\Phi_2}(d_j)$
$l^{T}(d_j)$	doc_len_all_token_dj	length of document d_j counted as number of all tokens, $l^{T}(d_j) = \sum_{\varphi_i \in \Phi} \text{ff}(\varphi_i, d_j)$
$l^{T_1}(d_j)$	doc_len_1w_token_dj	length of document d_j counted as number of one word tokens, $l^{T_1}(d_j) = \sum_{\varphi_i \in \Phi_1} \text{ff}(\varphi_i, d_j)$
$l^{T_2}(d_j)$	doc_len_ph_token_dj	length of document d_j counted as number of two word tokens, $l^{T_2}(d_j) = \sum_{\varphi_i \in \Phi_2} \text{ff}(\varphi_i, d_j)$

References

[Allan et al., 1996] Allan, J., Callan, J., Croft, W., & Lu, Z. (1996). Recent Experiments with INQUERY. In *TREC-4*.

[Andersen, 1992] Andersen, E. (1992). *The Statistical Analysis of Categorical Data*. Springer, Berlin, third edition.

[Buckley et al., 1992] Buckley, C., Salton, G., & Allan, J. (1992). Automatic Retrieval With Locality Information Using SMART. In *TREC-1 Proceedings*, pp. 59–72.

[Buckley et al., 1996] Buckley, C., Singhal, A., & Mitra, M. (1996). New Retrieval Approaches Using SMART: TREC-4. In *TREC-4 Proceedings*.

[Eftimiadis, 1993] Eftimiadis, E. (1993). A User-Centered Evaluation of Ranking Algorithms for Interactive Query Expansion. In *ACM SIGIR Conference on R&D in Information Retrieval*, pp. 146–159.

[Hug, 1996] Hug, J. (1996). Analyse und Sybtgese der Textur von Organoberflächen. Master's thesis, Institute for Communication Systems.

[Knaus et al., 1995] Knaus, D., Mittendorf, E., Schäuble, P., & Sheridan, P. (1995). Highlighting Relevant Passages for Users of the Interactive SPIDER Retrieval System. In *TREC-4 Proceedings*.

[Mateev, 1996] Mateev, B. (1996). Stochastic Dependence of Indexing Features and the Routing Problem. Diploma Thesis, Department of Computer Science, ETH Zürich.

[Mittendorf & Schäuble, 1996] Mittendorf, E., & Schäuble, P. (1996). Measuring the Effects of Data Corruption on Information Retrieval. In *Symposium on Document Analysis and Information Retrieval*, pp. 179–189.

[Mittendorf et al., 1995] Mittendorf, E., Schäuble, P., & Sheridan, P. (1995). Applying Probabilistic Term Weighting to OCR Text in the Case of a Large Alphabetic Library Catalogue. In *ACM SIGIR Conference on R&D in Information Retrieval*, pp. 328–335.

[Porter, 1980] Porter, M. F. (1980). An Algorithm for Suffix Stripping. *Program*, 14(3),130–137.

[Robertson et al., 1995] Robertson, S. E., Walker, S., Jones, S., Hancock-Beaulieu, M. M., & Gatford, M. (1995). OKAPI at TREC-3. In *TREC-3 Proceedings*, pp. 109–126.

[Salton, 1971] Salton, G. (1971). *The SMART Retrieval System-Experiments in Automatic Document Processing.* Prentice Hall, Englewood, Cliffs, New Jersey.

[Schäuble et al., 1997] Schäuble, P., Sheridan, P., & Wechsler, M. (1997). Cross-Language Speech Retrieval. submitted for SIGIR'97.

[Schütze et al., 1995] Schütze, H., Hull, D., & Pedersen, J. (1995). A Comparison of Classifiers and Document Representations for the Routing Problem. In *ACM SIGIR Conference on R&D in Information Retrieval*, pp. 229–237.

[Singhal et al., 1996] Singhal, A., Buckley, C., & Mitra, M. (1996). Pivoted Document Length Normalization . In *ACM SIGIR Conference on R&D in Information Retrieval*, pp. 21–29.

[van Rijsbergen, 1977] van Rijsbergen, C. (1977). A Theoretical Basis for the Use of Co-occurrence Data in Information Retrieval. *Journal of Documentation*, 33,106–119.

[Wechsler & Schäuble, 1995] Wechsler, M., & Schäuble, P. (1995). Speech retrieval based on automatic indexing. In Ruthven, I., editor, *Proceedings of the Final Workshop on Multimedia Information Retrieval (MIRO'95)*, Electronic Workshops in Computing, Glasgow. Springer.