Report on Thomson Legal and Regulatory Experiments at CLEF-2004

Isabelle Moulinier and Ken Williams
Thomson Legal and Regulatory
610 Opperman Drive
Eagan, MN 55123, USA
{Isabelle.Moulinier,Ken.Williams}@thomson.com

Abstract

Thomson Legal and Regulatory participated in the CLEF-2004 monolingual and bilingual tracks. Monolingual experiments included Portuguese, Russian and Finnish. We investigated a new query structure to handle Finnish compounds.

Our main focus was bilingual search from German to French. Our approach used query translation and post-translation pseudo-relevance feedback. We compared two translation models for query translation, and captured compound translations through fertility probabilities. While the fertility-based approach picks good terms, it does not help improve bilingual retrieval. Pseudo-relevance feedback, on the other hand, resulted in improved average precision.

1 Introduction

During the 2004 CLEF campaign, Thomson Legal and Regulatory participated in monolingual and bilingual information retrieval. With our monolingual experiments, we revisited our approach to handling compounds for Finnish retrieval. Previously, we would attempt to match on compounds and phrases. Our new approach restricts matches to compounds only.

Removing stopwords is generally beneficial. With no language expertise in Finnish, Russian, and Portuguese, we investigated building stopword lists using collection and query log statistics, with no manual editing. Our experiments measured the effect of these lists on retrieval.

Our main focus, however, was bilingual search. Our approach relied on word-based query translation, and we investigated building bilingual lexicons from corpora using statistical machine translation. We were particularly interested in assessing whether a more sophisticated model (IBM Model 3) would outperform a simpler model (IBM Model 1). We focused on the notion of fertility introduced by Model 3, which allows a source term to translate to zero or more target terms. In the case of German to French translations, we used fertilities to capture translating German compounds into French phrases.

In addition to investigating translation approaches, we introduced post-translation pseudo-relevance feedback in our runs. That lead to improved average precision. As reported in prior research, we observed a great variability on a per-query basis.

We present our experimental platform and some background in Section 2. Section 3 presents our bilingual effort, while monolingual experiments are described in 4.

2 Background

We briefly describe the retrieval system we used during our CLEF participation, and the pseudo-relevance feedback approach we adopted.

2.1 The WIN system

The WIN system is a full-text natural language search engine, and corresponds to TLR/West Group's implementation of the inference network retrieval model. While based on the same retrieval model as the INQUERY system [CCB92], WIN has evolved separately and focused on the retrieval of legal material in large collections in a commercial environment that supports both Boolean and natural language searches [Tur94].

Indexing Indexing of European languages considers tokens (words) as indexing units. Tokens are identified by localized tokenization rules (e.g. detecting apostrophes in French). Tokens are also stemmed using a morphological stemmer[1] which also identifies compounds and their parts for compound-rich languages such as Finnish or German.

WIN does not apply a stopword list during indexing, but it does when searches are performed. As a result, all terms are indexed, although it is possible to omit some terms in document length statistics.

Document retrieval Document retrieval in WIN can be decomposed into two components: query formulation and document scoring. Query formulation identifies query concepts, while scoring find matches for such concepts in documents.

Query formulation identifies "concepts" in natural language text, and imposes a Bayesian belief structure on these concepts. In many cases, each term in the natural language text represents a concept, and a flat structure gives the same weight to all concepts. However, phrases, compounds or misspellings can introduce more complex concepts, using operators such as "natural phrase," "compound," or "synonym."

We used a standard *tf-idf* scheme for computing term beliefs in all our runs. The belief of a single concept is given by:

$$bel_{term}(Q) = 0.4 + 0.6 * tf_{norm} * idf_{norm}$$

where

$$tf_{norm} = \frac{\log(tf + 0.5)}{\log(tf_{max} + 1.0)} \quad \text{and} \quad idf_{norm} = \frac{log(C + 0.5) - log(df)}{log(C + 1.0)}$$

and tf is the number of occurrences of the term within the document, tf_{max} is the maximum number of occurrences of any term within the document, df is the number of documents containing the term and C the total number of documents in the collection. tf_{max} is a weak approximation for document length.

The final document score is an average of the document score as a whole and the score of the best portion, where the best portion is dynamically computed based on query concept occurrences.

2.2 Pseudo-relevance feedback

Past research has reported on the benefits of pseudo-relevance feedback. For example, the relevance feedback incorporated in OKAPI BM-25 model has been successful at CLEF (cf. [Sav01]). Recently, alternative approaches to selecting relevant documents have been introduced; for example, Sakai and Sparck-Jones [SSJ01] investigated using document summaries to support pseudo-relevance feedback.

Our approach to pseudo-relevance feedback follows the work outlined by Haines and Croft [HC93] where feedback was added to Inquery.

Term selection We use a Rocchio-like formula to select terms for expansion:

$$sw = \frac{\beta}{|R|} \sum_{d \in R} (tf_{norm} * idf_{norm}) - \frac{\gamma}{|\overline{R}|} \sum_{d \in \overline{R}} (tf_{norm} * idf_{norm}) \tag{1}$$

[1] We are using the stemmer commercialized by Inxight within the LinguistX platform.

where R is the set of documents considered relevant, $\overline{R}$ the set of documents considered not relevant, and $|X|$ denotes the cardinality of set X. tf_{norm} and idf_{norm} are defined in the previous section. The β and γ weights are set experimentally.

We select terms for expansion solely on the basis of documents. We do not favor terms that appear in the original query during term selection. The sets of documents R and $\overline{R}$ are extracted from the document list returned by the original search: R correspond to the top n documents, and $\overline{R}$ to the bottom m, where n and m are determined through experiments on training data.

Reformulated query We append N selected terms to the query, eliminating any terms already present in the original query. In addition, each added term is weighted by the tf_{norm} part of the selection weight. Weights of original query terms remain unchanged.

3 Bilingual experiments

Our approach to bilingual search relies on word-by-word query translation using bilingual lexicons. We build our lexicons from parallel corpora using a statistical machine translation toolkit. In particular, we investigate how parameters from the translation models can be leveraged for selecting translations for German compounds.

3.1 Background

In a cross-lingual search system, user queries and documents may not share the same language. Before matching between documents and queries can happen, some level of translation is required. Conventional approaches separate the translation and retrieval processes, with translation occurring prior to retrieval. However, recent efforts use language modeling [KNS03] to integrate translation and retrieval in a unified model.

We focused on query translation [HG96], rather than document translation [OH97] or the translation of both queries and documents [BRS01]. Query translation can be performed using machine translation tools [Sav01] such as Systran, machine readable dictionaries [HG96], and bilingual lexicons learned from parallel or comparable corpora. Such bilingual lexicons include similarity thesauri [SBS97] which capture the notion of translation and related terms at once; and probability tables from statistical machine translation (e.g. the table constructed by IBM Model 1) which attempt to encode exact translations only.

With queries being translated term-by-term using bilingual lexicons, a term may have multiple possible translations. By taking advantage of query structures available in INQUERY, Pirkola [Pir98] has shown that grouping translations for a given term is a better technique than allowing all translations to contribute equally.

3.2 German to French translation: translating compounds

In previous CLEF campaigns, we constructed similarity thesauri from comparable corpora, and used the thesauri to translate queries concept-by-concept. Such an approach worked fairly well, and we obtained promising results on French-English and Spanish-English retrieval. This year, we used the statistical machine translation toolkit GIZA++ [ON00] to build bilingual lexicons. Future work will include comparing both translation approaches for term-by-term query translation.

Translation models 1 and 3 [BDPDPM93] introduced five models of increasing complexity. We chose to compare Model 1 and Model 3. Model 1 is intended to capture individual word translations, while Model 3 introduces modeling of local alignments and fertilities. We were particularly interested in the notion of fertility, which allows a source term to translate to zero or more target terms. In the case of German to French translations, we hoped that fertilities would capture translating German compounds into French phrases.

Using translation and fertility probabilities Using the GIZA++ toolkit, we trained models 1 and 3 on the Europarl corpus [Koe02]. We did not use the decoding phase typically associated with statistical machine translation. We simply used the translation and fertility probabilities generated by GIZA++ for each source term d and target term f:

- $t^1(f|d)$, model 1 translation probabilities for model 1,
- $t^3(f|d)$, model 3 translation probabilities for model 3,
- $n(\phi|d)$ where $\phi = 0 \ldots 9$, fertility probabilities for model 3, and
- p_0, the fertility probability for the empty notion.

We subsequently defined two translation methods: a word-based method and a fertility-based method.

The word-based method `lex` selects the n most probable translations of each source term d using the translation probabilities. To limit adding spurious translations, we threshold translation probilities to a fixed value p_{min}. Consequently, the `lex` method may select 0 to n translations for a given term.

The fertility-based approach `fert` represents our attempt at capturing the translation of German compounds. With this approach, we select one translation per source term, but each translation may include multiple terms. The `fert` model generates for each source term d a translation set of the m most probable target terms $f_1, \ldots, f_m$, ranked according to their translation probabilities $t^3(f_i|d)$. The number of selected terms m is given by

$$\underset{\phi}{\text{ArgMax}} \begin{cases} n(\phi|d) * p_0 & \text{if } \phi = 0 \\ n(\phi|d) * \sum_{i=1}^{\phi} t^3(f_i|d) & \text{if } \phi > 0 \end{cases}$$

Examples of selected translations are reported in Table 1. The first three examples capture the adequate translation for the German term. The last example, "Lawinenunglücken," is only partially translated to "avalanches" (the disaster aspect is missing). In addition, the `fert` method selects far too many terms because the mass of translation probabilities outweighs the fertility factor.

Additional processing of non-translated terms We performed some additional processing for non-translated terms, i.e. terms with no entry in the bilingual lexicons. In particular, we focused on compounds that did not appear in the parallel corpus.

When no translation was found for a German term, we first stemmed the German term. If translations were found for the stemmed term, we associated these translations to the original term. If still no translation was found and the stemmed term was identified as a compound, we applied the translation process to each stemmed part. The original term was associated with the translations of the compound parts. Finally, when no translation was found, the original German term was kept as the translation. Examples of compounds translated via this additional processing are given in Table 2.

Query formulation We followed [Pir98] and others in structuring translated queries to give the same importance to each original term, regardless of the number of translations. We grouped multiple translations under a weighted #SUM node. The weight associated with each translation is its translation probability.

We also investigated using a proximity operator when translating compound terms. When the original German term was a compound, we grouped all translations under the #NPHR operator[2].

[2] The WIN #NPHR operator corresponds to Inquery phrase operator, and includes partial credit. Partial credit enables both the operator and its children to contribute to document belief scores.

Term: **globale**

f	$t^3(f\|d)$	ϕ	$n(\phi\|d)$	`lex` translation	`fert` translation
globale	0.306778	1	0.746871	globale	globale
mondiale	0.152177	0	0.165741	mondiale	
global	0.115814	2	0.0617001	global	
mondial	0.0928475	3	0.0207158		
chelle	0.0456918		...		

Term: **Klimaveränderungen**

f	$t^3(f\|d)$	ϕ	$n(\phi\|d)$	`lex` translation	`fert` translation
climatiques	0.269569	2	0.589625	climatiques	climatiques
changements	0.258488	1	0.105312	changements	changements
changement	0.105622	3	0.0936477	changement	
climatique	0.103034	4	0.07117		
climat	0.0250892		...		

Term: **Treibhauseffektes**

f	$t^3(f\|d)$	ϕ	$n(\phi\|d)$	`lex` translation	`fert` translation
effet	0.265273	2	0.283692	effet	effet
serre	0.26525	1	0.246126	serre	serre
venir	0.0380016	3	0.174969		
mes	0.0191118	9	0.0651408		

Term: **Lawinenunglücken**

f	$t^3(f\|d)$	ϕ	$n(\phi\|d)$	`lex` translation	`fert` translation
avalanches	0.10976	1	0.404492	avalanches	avalanches
programmer	0.10976	2	0.231625	programmer	programmer
servir	0.10976	3	0.1003	servir	servir
court	0.10976	0	0.0752761		court
interventions	0.10976	9	0.0611943		interventions
diverses	0.109759	4	0.0435146		diverses
série	0.109759				série
pourquoi	0.109759		⋮		pourquoi
zones	0.109624				zones

Table 1: Examples of German to French translations. We used the probabilities $t^3(.|.)$ to select translations in the `lex` method. We used both $t^3(.|.)$, the translation probabilities, and $n(\phi|.)$, the fertilities from Model 3 to generate translation in the `fert` approach.

Compound term	Identified Translations ($t(f\|d)$)	
Wohnungsbrände	logement	(0.453006)
	incendie	(0.319306)
	au	(0.256685)
	feu	(0.153006)
Weltmeisterin	du	(0.172959)
	champions	(0.135024)
	monde	(0.135023)

Table 2: Examples of compounds translated through additional processing for terms outside the lexicon. Translation is performed using `lex`, $n = 3$, $p_{min} = 0.1$ and Model 3 translation probabilities.

Run	Avg. Prec,	R-Prec.	Prec. at 20 doc.
t^1, lex, #SUM	0.2934	0.2951	0.2224
t^3, lex, #SUM	0.3225	0.3250	0.2541
t^3, fert, #SUM	0.2717	0.2868	0.2133

Table 3: Comparisons between bilingual base runs. The lex approach using Models 1 (t^1) and 3 (t^3) used $n = 3$ and $p_{min} = 0.1$.

Run	Avg Prec.	R-Prec.	Prec. at 20 doc.
t^3, fert, #SUM	0.2717	0.2868	0.2133
t^3, fert, #NPHR	0.2708	0.2779	0.2153

Table 4: Capturing the translation of German compounds. Comparison between the #SUM and the #NPHR operators.

3.3 Results and discussion

Base runs We ran our first set of experiments set to determine whether the more sophisticated translation model (Model 3) improves retrieval performance over the simpler Model 1. We compared Model 1 (t^1) and Model 3 (t^3) with the lex translation selection. Results are reported in Table 3. Model 3 with lex provided a strong baseline with the #SUM operator. The lex method using Model 3 outperforms the lex method using Model 1, although the difference is not statistically significant. We found that many queries improved by a noticeable margin when Model 3 was introduced, and that some of that queries that degraded were affected by poor post-translation stopword removal.

Fertility runs Our attempt to capture the translation of compounds using fertilities had limited success. We find the fert method promising inasmuch as it is able to identify adequate compound translations but suffers from selecting a single, possibly multi-term translation. The difference between runs lex and fert using Model 3 (cf. Table 3) is statistically significant[3]. We have already seen ("Lawinenunglücken") that the fert approach may select too many terms when the probability mass of the candidate set outweighs the fertility probability factor. In addition, selecting a single translation as does the fert approach, limits the effectiveness of retrieval. We evaluated the lex approach selecting a single translation ($n = 1$), and the average precision dropped to 0.2641[4]. This result confirms our suspicion that the fert approach is hindered by selecting a single, possibly multi-term translation.

Translated compounds as phrases Next we studied the impact of query formulation with the fert approach. The fert approach captures the translation of German compounds into multiple French terms. We expected that introducing the #NPHR operator would positively impact retrieval, since French phrases were a better representation of German compounds. The results reported in Table 4 did not support our intuition: the #NPHR operator did not improve average precision. We think that partial credit diluted results, because with partial credit the children of a phrase contribute independently as concepts to document scores. In future work, we will explore alternative scoring approaches for the phrase proximity to retain the translations of a source term as a single concept.

Seeding translation models We also investigated seeding the translation models with a machine-readable dictionary. We tested only with Model 3 and found no differences between the two translation probability tables.

[3] We used the Wilcoxon signed-rank test, with $\alpha = 0.05$.

[4] This difference is also found statistically significant.

Run	Avg Prec.	R-Prec.	Prec. at 20 doc.	Above/equal/below Median
t^1, lex, nd, NoPRF	0.2934	0.2951	0.2224	–
t^1, lex, nd, $\gamma = 4$ (tlrde2fr4)	0.3289	0.3005	0.2531	23 / 1 / 24
t^3, lex, nd, NoPRF (tlrde2fr2)	0.3225	0.3250	0.2541	32 / 2 / 14
t^3, lex, nd, $\gamma = 1$ (tlrde2fr3)	0.3750$\star$	0.3409	0.3000	31 / 2 / 15
t^3, fert, d (tlrde2fr1)	0.2723	0.2877	0.2153	25 / 1 / 22
t^3, fert, d, $\gamma = 1$	0.3250	0.2915	0.2571	–

Table 5: Experimental results using post-translation pseudo-relevance feedback. All runs with PRF used $N = 20$, $n = 5$, $m = 20$, $\beta = 1$. $\star$ indicates that PRF improves over the base run, and the difference is statistically significant with $\alpha = 0.01$ using the Wilcoxon signed-rank test.

Runs using pseudo-relevance feedback Finally we report on experiments using post-translation pseudo-relevance feedback (PRF). After the initial retrieval, we selected the five highest-ranked documents as relevant documents. We also selected the twenty lowest-ranked documents as non-relevant. We use the non-relevant documents as a filter to prevent common words from being selected by PRF.

As can be observed in Table 5, the introduction of PRF was beneficial. We observed the typical behavior when comparing base runs and PRF. In the best case (run "t^3, lex"), PRF helps improve the performance of 59% of queries and degrade 38% of the queries. In the two other runs, PRF is helpful for 50% of queries, and not so helpful for 44% of queries.

A point of interest is the comparison to the median. There is a significant difference in average precision between the base run and the PRF run using lex and Model 3; however each run compares similarly to the median of all runs. After analysis, we observed the well-documented seesaw effect of pseudo-relevance feedback: 10 queries fell below the median when PRF was added, while 8 queries rose above the median.

4 Monolingual experiments

We participated in the monolingual track with three new languages: Finnish, Portuguese and Russian. We revisited our approach to compound handling and experimented with the creation of stopword lists.

4.1 Compound handling in Finnish retrieval

Prior research During past CLEF campaigns, the handling of compounds has received a fair amount of attention. Prior research has found that, for German, Dutch, or Finnish, breaking compounds into parts and searching on the parts was beneficial to both monolingual and crosslingual retrieval [HKP+02, Md02]. Alternatively, some researchers have focused on character n-grams as indexing units for European languages (cf. [MMP01]), limiting the reliance on compound identification. Indeed character n-grams may capture compound parts without explicitly identifying compounds.

Compounds are not like phrases At CLEF 2000, we investigated the impact of decompounding on monolingual retrieval for German. In those experiments, we found that decompounding was useful and that representing compounds using the #NPHR operator with partial credit was the most effective. The #NPHR operator corresponds to an unordered proximity of 3, and partial credit allows the children of the proximity operator to contribute to the final belief score, independently of the operator.

With this year's experiments, we revisited the operator and proposed a stricter proximity #NPHR0. In order to contribute to the document belief score, parts of the compound must

Run	Avg. Prec.	R-Prec.	Prec. at 20 doc.
#NPHR	0.5418	0.4903	0.2722
#NPHR0	0.5562	0.5027	0.2744

Table 6: Experimental results using different operators in the representation of Finnish compounds. Differences are not statistically significant.

appear in a compound, not in a "phrase" environment. Partial credit is still applied. In other words, we replaced the unordered proximity of 3 with a proximity of 0. This is made possible by our indexing scheme, where compounds and their parts are indexed.

Experimental results and discussion Table 6 summarizes our experimental results with Finnish compounds. We observe a small improvement in both average precision and R-Precision, although the difference is not statistically significant.

Let us note that all documents that satisfy the #NPHR0 operator also satisfy the #NPHR operator, although their belief score may be different under each condition. For some queries, e.g. query 208, the #NPHR run ranks relevant documents higher in the list, suggesting that it finds useful proximities in addition to the exact compounds. On the other hand, for other queries, e.g. query 203, the additional proximities found in documents degrade the ranked list by pushing relevant documents further down the list. We suspect that the difference in ranking is linked to the different *idf* values associated with the #NPHR and #NPHR0 operators.

4.2 Experiments with stopword lists

Two sources to identify stopwords At NTCIR-4, we built upon Savoy's work [Sav01] and we compared using collection and query log statistics to create stopword lists. We found little differences in retrieval effectiveness.

For our CLEF experiments, we merged both approaches. We selected the most frequent terms in the collection as stopwords. We subsequently enriched that list with terms extracted from query logs. No manual review of the list was performed.

For our runs, we selected the most frequent 100 and 200 stemmed terms in collections. To those collection-based lists, we added stemmed terms that occurred in over 20% of the query logs. For each language, a query log consisted of collected CLEF queries from previous campaigns.

Results and discussion In Table 7, we compare our base runs with no stopword removal (`none`) with removing stopwords extracted from collection statistics and query logs. Stopword lists are a useful tool to make search more effective in terms of average precision. We observe statistically significant differences in average precison for all runs.

We conclude our discussion on stopword lists by outlining the need for human review. In the Finnish stopword list, we noticed cities such as Helsinki and Tampere, as well as terms like suomi (Finland, finnish language) and suomalainen (finnish). Similarly the Portuguese list contains Lisboa, Portugal, português, governo or ministro. While such terms are frequent in the collection, they are not truly stopwords and interfered with some queries (e.g. query 231).

5 Conclusion

Our bilingual experiments with IBM Model 3 are promising. Using a word by word translation, we were able to capture the translation of German compounds using translation and fertility probabilities. In the future, we will expand our work to select more than one translation per source term. In addition we will investigate alternative scoring for the #NPHR operator, where partial credit is not allowed to dilute the contribution of other concepts. Finally, we will investigate whether there is added value in using Model 4 or 5 in the context of word-by-word translation.

Run	Avg. Prec.	R-Prec.	Above/equal/below Median
fi, none	0.5466	0.4947	–
fi, 100 (tlrfi1)	0.5551 $\star$	0.4994	23/8/13
fi, 200 (tlrfi2)	0.5562 $\star$	0.5027	23/9/12
pt, none	0.4250	0.3992	–
pt, 100 (tlrpt1)	0.4458 $\star\star$	0.4017	16/15/14
pt, 200 (tlrpt2)	0.4469 $\star\star$	0.4044	16/17/12
ru, none (tlrru2)	0.3176	0.2783	9/7/17
ru, 100 (tlrru1)	0.3702 $\star\star$	0.3183	13/9/11
ru, 200	0.3820 $\star\star$	0.3364	–

Table 7: Summary of our monolingual runs, with an emphasis on using stopword lists. Finnish runs use the #NPHR0 operator. The $\star\star$,$\star$ sign indicates a statistical difference with the base run "none" with $\alpha = 0.01, 0.05$ using the Wilcoxon signed-rank test.

We find our monolingual runs satisfactory. Our reformulated compound handling in Finnish improved was beneficial when compared to our previous approach. Compound handling may also benefit from improved partial credit. We have observed similar findings with German and Korean. Our stopword experiments confirmed well-established results about stopword removal and retrieval effectiveness.

References

[BDPDPM93] P. Brown, V. Della Pietra, S. Della Pietra, and R. Mercer. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2), June 1993.

[BRS01] M. Braschler, B. Ripplinger, and P. Schäuble. Experiments with the eurospider retrieval system for clef 2001. In Peters et al. [PBGK02].

[CCB92] W. B. Croft, J. Callan, and J. Broglio. The INQUERY retrieval system. In *Proceedings of the 3^{rd} International Conference on Database and Expert Systems Applications*, Spain, 1992.

[HC93] D. Haines and W.B. Croft. Relevance feedback and inference networks. In *Proceedings of the Sixteenth Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 2–11, 1993.

[HG96] D. Hull and G. Grefenstette. Querying across languages: a dictionary-based approach to multilingual information retrieval. In *Proceedings of the 19^{th} Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 49–57, 1996.

[HKP$^+$02] H. Hedlund, H. Keskustalo, A. Pirkola, E. Airio, and K. Järvelin. Utaclir at CLEF 2001 - effects of compound splitting and N-gram techniques. In Peters et al. [PBGK02], pages 118–136.

[KNS03] W. Kraaij, J.-Y. Nie, and M. Simard. Web-based statistical translation models in cross-language information retrieval. *Computational Linguistics*, 29(3):381–419, 2003.

[Koe02] P. Koehn. Europarl: A multilingual corpus for evaluation of machine translation. Draft, 2002.

[Md02] C. Monz and M. de Rijke. Shallow morphological analysis in monolingual information retrieval for dutch, german, and italian. In Peters et al. [PBGK02], pages 262–277.

[MMP01] P. McNamee, J. Mayfield, and C. Piatko. A language-independent approach to european text retrieval. pages 129–139, 2001.

[OH97] D. Oard and P. Hackett. Document translation for cross-language text retrieval at the university of maryland. In *Proceedings of the 6th Text REtrieval Conference (TREC-6)*. National Institute of Standards and Technology (NIST), November 1997.

[ON00] F. J. Och and H. Ney. Improved statistical alignment models. In *The 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hongkong, China, October 2000.

[PBGK01] C. Peters, M. Brashler, J. Gonzalo, and M. Kluck, editors. *Evaluation of Cross-Language Information Retrieval Systems*, number 2406 in LNCS. Springer, September 2001.

[PBGK02] C. Peters, M. Braschler, J. Gonzalo, and M. Kluck, editors. *Evaluation of Cross-Language Information Retrieval Systems: revised papers. Second Workshop of the Cross-Languague Evaluation Forum, CLEF 2001, Darmstadt, Germany, September 3–4, 2000*, volume 2406 of *Lecture Notes in Computer Science*. Springer, 2002.

[Pir98] A. Pirkola. The effects of query structure and dictionary setups in dictionary-based cross-language information retrieval. In *Proceedings of the 21th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 55–63, Melbourne, Australia, 1998.

[Sav01] J. Savoy. Report on CLEF-2001 experiments: Effective combined query-translation approach. In Peters et al. [PBGK01].

[SBS97] P. Sheridan, M. Braschler, and P. Schuble. Cross-lingual information retrieval in a multilingual legal domain. In *Proceedings of the First European Conference on Research and Advanced Technology for Digital Libraries*, pages 253–268, Pisa, Italy, 1997.

[SSJ01] T. Sakai and K. Sparck-Jones. Generic summaries for indexing in information retrieval. In *Proceedings of the 24th annual international ACM SIGIR conference on Research and development in information retrieval*, pages 190–198, 2001.

[Tur94] H. Turtle. Natural language vs. boolean query evaluation: a comparison of retrieval performance. In *Proceedings of the 17th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 212–220, Dublin, Ireland, 1994.