Boosting Cross-Language Retrieval by Learning Bilingual Phrase Associations from Relevance Rankings

Artem Sokolov and **Laura Jehl** and **Felix Hieber** and **Stefan Riezler**
Department of Computational Linguistics
Heidelberg University, 69120 Heidelberg, Germany
{sokolov,jehl,hieber,riezler}@cl.uni-heidelberg.de

Abstract

We present an approach to learning bilingual n-gram correspondences from relevance rankings of English documents for Japanese queries. We show that directly optimizing cross-lingual rankings rivals and complements machine translation-based cross-language information retrieval (CLIR). We propose an efficient boosting algorithm that deals with very large cross-product spaces of word correspondences. We show in an experimental evaluation on patent prior art search that our approach, and in particular a consensus-based combination of boosting and translation-based approaches, yields substantial improvements in CLIR performance. Our training and test data are made publicly available.

1 Introduction

The central problem addressed in Cross-Language Information Retrieval (CLIR) is that of translating or projecting a query into the language of the document repository across which retrieval is performed. There are two main approaches to tackle this problem: The first approach leverages the standard Statistical Machine Translation (SMT) machinery to produce a single best translation that is used as search query in the target language. We will henceforth call this the *direct translation* approach. This technique is particularly useful if large amounts of data are available in domain-specific form.

Alternative approaches avoid to solve the hard problem of word reordering, and instead rely on token-to-token translations that are used to project the query terms into the target language with a probabilistic weighting of the standard term tf-idf scheme. Darwish and Oard (2003) termed this method the *probabilistic structured query* approach. The advantage of this technique is an implicit query expansion effect due to the use of probability distributions over term translations (Xu et al., 2001). Recent research has shown that leveraging query context by extracting term translation probabilities from n-best direct translations of queries instead of using context-free translation tables outperforms both direct translation and context-free projection (Ture et al., 2012b; Ture et al., 2012a).

While direct translation as well as probabilistic structured query approaches use machine learning to optimize the SMT module, retrieval is done by standard search algorithms in both approaches. For example, Google's CLIR approach uses their standard proprietary search engine (Chin et al., 2008). Ture et al. (2012b; 2012a) use standard retrieval algorithms such as BM25 (Robertson et al., 1998). That means, machine learning in SMT-based approaches concentrates on the cross-language aspect of CLIR and is agnostic of the ultimate ranking task.

In this paper, we present a method to project search queries into the target language that is complementary to SMT-based CLIR approaches. Our method learns a table of n-gram correspondences by direct optimization of a ranking objective on relevance rankings of English documents for Japanese queries. Our model is similar to the approach of Bai et al. (2010) who characterize their technique as "Learning to rank with (a Lot of) Word Features". Given a set of search queries $\mathbf{q} \in \mathbb{R}^Q$ and docu-

Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, pages 1688–1699,
Seattle, Washington, USA, 18-21 October 2013.

ments $\mathbf{d} \in \mathbb{R}^D$, where the j^{th} dimension of a vector indicates the occurrence of the j^{th} word for dictionaries of size Q and D, we want to learn a score $f(\mathbf{q}, \mathbf{d})$ between a query and a given document using the model[1]

$$f(\mathbf{q}, \mathbf{d}) = \mathbf{q}^\top W \mathbf{d} = \sum_{i=1}^{Q} \sum_{j=1}^{D} q_i W_{ij} d_j.$$

We take a pairwise ranking approach to optimization. That is, given labeled data in the form of a set $\mathcal{R}$ of tuples $(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-)$, where $\mathbf{d}^+$ is a relevant (or higher ranked) document and $\mathbf{d}^-$ an irrelevant (or lower ranked) document for query $\mathbf{q}$, the goal is to find a weight matrix $W \in \mathbb{R}^{Q \times D}$ such that $f(\mathbf{q}, \mathbf{d}^+) > f(\mathbf{q}, \mathbf{d}^-)$ for all data tuples from $\mathcal{R}$. The scoring model learns weights for all possible correspondences of query terms and document terms by directly optimizing the ranking objective at hand. Such a phrase table contains domain-specific word associations that are useful to discern relevant from irrelevant documents, something that is orthogonal and complementary to standard SMT models.

The challenge of our approach can be explained by constructing a joint feature map ϕ from the outer product of the vectors $\mathbf{q}$ and $\mathbf{d}$ where

$$\phi_{((i-1)D+j)}(\mathbf{q}, \mathbf{d}) = (\mathbf{q} \otimes \mathbf{d})_{ij} = (\mathbf{q}\mathbf{d}^\top)_{ij}. \quad (1)$$

Using this feature map, we see that the score function f can be written in the standard form of a linear model that computes the inner product between a weight vector w and a feature vector ϕ where $w, \phi \in \mathbb{R}^{Q \times D}$ and

$$f(\mathbf{q}, \mathbf{d}) = \langle w, \phi(\mathbf{q}, \mathbf{d}) \rangle. \quad (2)$$

While various standard algorithms exist to optimize linear models, the difficulty lies in the memory footprint and capacity of the word-based model. A full-sized model includes $Q \times D$ parameters which is easily in the billions even for moderately sized dictionaries. Clearly, an efficient implementation and remedies against overfitting are essential.

The main contribution of our paper is the presentation of algorithms that make learning a phrase table by direct rank optimization feasible, and an experimental verification of the benefits of this approach, especially with regard to a combination of the orthogonal information sources of ranking-based and SMT-based CLIR approaches. Our approach builds upon a boosting framework for pairwise ranking (Freund et al., 2003) that allows the model to grow incrementally, thus avoiding having to deal with the full matrix W. Furthermore, we present an implementation of boosting that utilizes parallel estimation on bootstrap samples from the training set for increased efficiency and reduced error (Breiman, 1996). Our "bagged boosting" approach allows to combine incremental feature selection, parallel training, and efficient management of large data structures.

We show in an experimental evaluation on large-scale retrieval on patent abstracts that our boosting approach is comparable in MAP and improves significantly by 13-15 PRES points over very competitive translation-based CLIR systems that are trained on 1.8 million parallel sentence pairs from Japanese-English patent documents. Moreover, a combination of the orthogonal information learned in ranking-based and translation-based approaches improves over 7 MAP points and over 15 PRES points over the respective translation-based system in a consensus-based voting approach following the Borda Count technique (Aslam and Montague, 2001).

2 Related Work

Recent research in CLIR follows the two main paradigms of direct translation and probabilistic structured query approaches. An example for the first approach is the work of Magdy and Jones (2011) who presented an efficient technique to adapt off-the-shelf SMT systems for CLIR by training them on data pre-processed for retrieval (case folding, stopword removal, stemming). Nikoulina et al. (2012) presented an approach to direct translation-based CLIR where the n-best list of an SMT system is re-ranked according to the MAP performance of the translated queries. The probabilistic structured query approach has seen a lot of work on context-aware query expansion across languages, based on various similarity statistics (Ballesteros and Croft, 1998; Gao et al., 2001; Lavrenko et al., 2002; Gao

[1] With bold letters we denote vectors for query $\mathbf{q}$ and document $\mathbf{d}$. Vector components are denoted with normal font letters and indices (e.g., q_i).

et al., 2007). At the time of writing this paper, the most recent extension to this paradigm is Ture et al. (2012a). In addition to projecting terms from n-best translations, they propose a projection extracted from the hierarchical phrase- based grammar models, and a scoring method based on multi-token terms. Since the latter techniques achieved only marginal improvements over the context-sensitive query translation from n-best lists, we did not pursue them in our work.

CLIR in the context of patent prior art search was done as extrinsic evaluation at the NTCIR PatentMT[2] workshops until 2010, and has been ongoing in the CLEF-IP[3] benchmarking workshops since 2009. However, most workshop participants did either not make use of automatic translation at all, or they used an off-the-shelf translation tool. This is due to the CLEF-IP data collection where parts of patent documents are provided as manual translations into three languages. In order to evaluate CLIR in a truly cross-lingual scenario, we created a large patent CLIR dataset where queries and documents are Japanese and English patent abstracts, respectively.

Ranking approaches to CLIR have been presented by Guo and Gomes (2009) who use pairwise ranking for patent retrieval. Their method is a classical learning-to-rank setup where retrieval scores such as tf-idf or BM25 are combined with domain knowledge on patent class, inventor, date, location, etc. into a dense feature vector of a few hundred features. Methods to learn word-based translation correspondences from supervised ranking signals have been presented by Bai et al. (2010) and Chen et al. (2010). These approaches tackle the problem of complexity and capacity of the cross product matrix of word correspondences from different directions. The first proposes to learn a low rank representation of the matrix; the second deploys sparse online learning under ℓ_1 regularization to keep the matrix small. Both approaches are mainly evaluated in a monolingual setting. The cross-lingual evaluation presented in Bai et al. (2010) uses weak translation-based baselines and non-public data such that a direct comparison is not possible.

A combination of bagging and boosting in the context of retrieval has been presented by Pavlov et al. (2010) and Ganjisaffar et al. (2011). This work is done in a standard learning-to-rank setup using a few hundred dense features trained on hundreds of thousands of pairs. Our setup deals with billions of sparse features (from the cross-product of the unrestricted dictionaries) trained on millions of pairs (sampled from a much larger space). Parallel boosting where all feature weights are updated simultaneously has been presented by Collins et al. (2002) and Canini et al. (2010). The first method distributes the gradient calculation for different features among different compute nodes. This is not possible in our approach because we construct the cross-product matrix on-the-fly. The second approach requires substantial efforts in changing the data representation to use the MapReduce framework. Overall, one of the goals of our work is sequential updating for implicit feature selection, something that runs contrary to parallel boosting.

3 CLIR Approaches

3.1 Direct translation approach

For direct translation, we use the SCFG decoder cdec (Dyer et al., 2010)[4] and build grammars using its implementation of the suffix array extraction method described in Lopez (2007). Word alignments are built from all parallel data using mgiza[5] and the Moses scripts[6]. SCFG models use the same settings as described in Chiang (2007). Training and querying of a modified Kneser-Ney smoothed 5-gram language model are done on the English side of the training data using KenLM (Heafield, 2011)[7]. Model parameters were optimized using cdec's implementation of MERT (Och (2003)).

At retrieval time, all queries are translated sentence-wise and subsequently re-joined to form one query per patent. Our baseline retrieval system uses the Okapi BM25 scores for document ranking.

[2] http://research.nii.ac.jp/ntcir/ntcir/

[3] http://www.ifs.tuwien.ac.at/~clef-ip/

[4] https://github.com/redpony/cdec

[5] http://www.kyloo.net/software/doku.php/mgiza:overview

[6] http://www.statmt.org/moses/?n=Moses.SupportTools

[7] http://kheafield.com/code/kenlm/estimation/

3.2 Probabilistic structured query approach

Early Probabilistic Structured Query approaches (Xu et al., 2001; Darwish and Oard, 2003) represent translation options by lexical, i.e., token-to-token translation tables that are estimated using standard word alignment techniques (Och and Ney, 2000). Later approaches (Ture et al., 2012b; Ture et al., 2012a) extract translation options from the decoder's n-best list for translating a particular query. The central idea is to let the language model choose fluent, context-aware translations for each query term during decoding. This retains the desired query-expansion effect of probabilistic structured models, but it reduces query drift by filtering translations with respect to the context of the full query.

A projection of source language query terms $f \in F$ into the target language is achieved by representing each source token f by its probabilistically weighted translations. The score of target document E, given source language query F, is computed by calculating the BM25 rank over projected term frequency and document frequency weights as follows:

$$score(E|F) = \sum_{f \in F} BM25(tf(f,E), df(f)) \quad (3)$$

$$tf(f,E) = \sum_{e \in E_f} tf(e,E)p(e|f)$$

$$df(f) = \sum_{e \in E_f} df(e)p(e|f)$$

where $E_f = \{e \in E | p(e|f) > p_L\}$ is the set of translation options for query term f with probability greater than p_L. We also use a cumulative threshold p_C so that only the most probable options are added until p_C is reached.

Ture et al. (2012b; 2012a) achieved best retrieval performance by interpolating between (context-free) lexical translation probabilities p_{lex} estimated on symmetrized word alignments, and (context-aware) translation probabilities p_{nbest} estimated on the n-best list of an SMT decoder:

$$p(e|f) = \lambda p_{nbest}(e|f) + (1-\lambda)p_{lex}(e|f) \quad (4)$$

$p_{nbest}(e|f)$ is estimated by calculating expectations of term translations from k-best translations:

$$p_{nbest}(e|f) = \frac{\sum_{k=1}^{n} a_k(e,f)\mathcal{D}(k,F)}{\sum_{k=1}^{n}\sum_{e'} a_k(e',f)\mathcal{D}(k,F)}$$

where $a_k(e,f)$ is a function indicating an alignment of target term e to source term f in the k^{th} derivation of query F, and $\mathcal{D}(k,F)$ is the model score of the k^{th} derivation in the n-best list for query F.

We use the same hierarchical phrase-based system that was used for direct translation to calculate n-best translations for the probabilistic structured query approach. This allows us to extract word alignments between source and target text for F from the SCFG rules used in the derivation. The concept of self-translation is covered by the decoder's ability to use pass-through rules if words or phrases cannot be translated.

Probabilistic structured queries that include context-aware estimates of translation probabilities require a preservation of sentence-wise context-sensitivity also in retrieval. Thus, unlike the direct translation approach, we compute weighted term and document frequencies for each sentence s in query F separately. The scoring (3) of a target document for a multiple sentence query then becomes:

$$score(E|F) = \sum_{s \text{ in } F} \sum_{f \in s} BM25(tf(f,E), df(f))$$

3.3 Direct Phrase Table Learning from Relevance Rankings

Pairwise Ranking using Boosting The general form of the RankBoost algorithm (Freund et al., 2003; Collins and Koo, 2005) defines a scoring function $f(\mathbf{q},\mathbf{d})$ on query $\mathbf{q}$ and document $\mathbf{d}$ as a weighted linear combination of T *weak learners* h_t such that $f(\mathbf{q},\mathbf{d}) = \sum_{t=1}^{T} w_t h_t(\mathbf{q},\mathbf{d})$. Weak learners can belong to an arbitrary family of functions, but in our case they are restricted to the simplest case of unparameterized indicator functions selecting components of the feature vector $\phi(\mathbf{q},\mathbf{d})$ in (1) such that f is of the standard linear form (2). In our experiments, these features indicate the presence of pairs of uni- and bi-grams from the source-side vocabulary of query terms and the target-side vocabulary of document-terms, respectively. Furthermore, in order to simulate the pass-through behavior of SMT, we introduce additional features to the model that indicate the identity of terms in source and target. All identity features have the same fixed weight β, which is found on the development set.

For training, we are given labeled data in the form

of a set $\mathcal{R}$ of tuples $(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-)$, where $\mathbf{d}^+$ is a relevant (or higher ranked) document and $\mathbf{d}^-$ an irrelevant (or lower ranked) document for query $\mathbf{q}$. RankBoost's objective is to correctly rank query-document pairs such that $f(\mathbf{q}, \mathbf{d}^+) > f(\mathbf{q}, \mathbf{d}^-)$ for all data tuples from $\mathcal{R}$. RankBoost achieves this by optimizing the following convex exponential loss:

$$\mathcal{L}_{exp} = \sum_{(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-)\in\mathcal{R}} D(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-) e^{f(\mathbf{q},\mathbf{d}^-)-f(\mathbf{q},\mathbf{d}^+)},$$

where $D(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-)$ is a non-negative importance function on pairs of documents for a given $\mathbf{q}$.

We optimize $\mathcal{L}_{exp}$ in a greedy iterative fashion, which closely follows an efficient algorithm of Collins and Koo (2005) for the case of binary-valued h. In each step, the single feature h is selected that provides the largest decrease of $\mathcal{L}_{exp}$, i.e., that has the largest projection on the direction of the gradient $\nabla_h \mathcal{L}_{exp}$. Because of the sequential nature of the algorithm, RankBoost implicitly performs automatic feature selection and regularization (Rosset et al., 2004), which is crucial to reduce complexity and capacity for our application.

Parallelization and Bagging To achieve parallelization we use a variant of bagging (Breiman, 1996) on top of boosting, which has been observed to improve performance, reduce variance and is trivial to parallelize. The procedure is described as part of Algorithm 1: From the set of preference pairs $\mathcal{R}$, draw S equal-sized samples with replacement and distribute to nodes. Then, using each of the samples as a training set, separate boosting models $\{w_t^s, h_t^s\}, s = 1 \dots S$ are trained that contain the same number of features $t = 1 \dots T$. Finally the models are averaged: $f(\mathbf{q}, \mathbf{d}) = \frac{1}{S} \sum_t \sum_s w_t^s h_t^s(\mathbf{q}, \mathbf{d})$.

Algorithm The entire training procedure is outlined in Algorithm 1. For each possible feature h we maintain auxiliary variables W_h^+ and W_h^-:

$$W_h^{\pm} = \sum_{(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-):h(\mathbf{q},\mathbf{d}^+)-h(\mathbf{q},\mathbf{d}^-)=\pm 1} D(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-),$$

which are the cumulative weights of correctly and incorrectly ranked instances by a candidate feature h. The absolute value of $\partial \mathcal{L}_{exp} / \partial h$ can be expressed as $\left|\sqrt{W_h^+} - \sqrt{W_h^-}\right|$ which is used as feature selection criterion (Collins and Koo, 2005). The optimum of minimizing $\mathcal{L}_{exp}$ over w (with fixed h) can be shown to be $w = \frac{1}{2} \ln \frac{W_h^+ + \epsilon Z}{W_h^- + \epsilon Z}$, where ϵ is a smoothing parameter to avoid problems with small $W_h^{\pm}$ (Schapire and Singer, 1999), and $Z = \sum_{(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-)\in\mathcal{R}} D(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-)$. Furthermore, for each step t of the learning process, values of D are updated to concentrate on pairs that have not been correctly ranked so far:

$$D_{t+1} = D_t \cdot e^{w_t \left(h_t(\mathbf{q},\mathbf{d}^-) - h_t(\mathbf{q},\mathbf{d}^+)\right)}. \qquad (5)$$

Finally, to speed up learning, on iteration t we recalculate $W_h^{\pm}$ only for those h that cooccur with previously selected h_t and keep the rest unchanged (Collins and Koo, 2005).

Algorithm 1: Bagged Boosting

Input: training tuples $\mathcal{R}$, max number of features T, initial D_0, smoothing param. $\epsilon \simeq 10^{-5}$
Initialize:
from $\mathcal{R}$ draw S samples with replacement and distribute to nodes
Learn:
for *all samples* $s = 1 \dots S$ *in parallel* **do**
  calculate W_h^+,W_h^-, Z on sample's data
  for *all* $t = 1 \dots T$ **do**
    choose $h_t = \arg\max_h \left|\sqrt{W_h^+} - \sqrt{W_h^-}\right|$
    and $w_t = \frac{1}{2} \ln \frac{W_h^+ + \epsilon Z}{W_h^- + \epsilon Z}$
    update D_t according to (5)
    update $W_h^{\pm}$ for all h that cooccur with h_t
  end
  return to master $\{h_t^s, w_t^s\}, t = 1 \dots T$
end
Bagging:
return scoring function
$f(\mathbf{q}, \mathbf{d}) = \frac{1}{S} \sum_t \sum_s w_t^s h_t^s(\mathbf{q}, \mathbf{d})$

Implementation Because of the total number of features (billions) there are several obstacles for the straight-forward implementation of Algorithm 1.

First, we cannot directly access all pairs $(\mathbf{q}, \mathbf{d})$ containing a particular feature h needed for calculating $W_h^{\pm}$. Building an inverted index is complicated as it needs to fit into memory for fast fre-

quent access[8]. We resort to the on-the-fly creation of the cross-product space of features, following prior work by Grangier and Bengio (2008) and Goel et al. (2008). That is, while processing a pair $(\mathbf{q}, \mathbf{d})$, we update $W_h^{\pm}$ for all h found for the pair.

Second, even if the explicit representation of all features is avoided by on-the-fly feature construction, we still need to keep all $W_h^{\pm}$ in addressable RAM. To achieve that we use hash kernels (Shi et al., 2009) and map original features into b-bit integer hashes. The values $W_{h'}^{\pm}$ for new, "hashed", features h' become $W_{h'}^{\pm} = \sum_{h:HASH(h)=h'} W_h^{\pm}$. We used the MurmurHash3 function on the UTF-8 representations of features and $b = 30$ (resulting in more than 1 billion distinct hashes).

4 Model Combination by Borda Counts

SMT-based approaches to CLIR and our boosting approach have different strengths. The SMT-based approaches produce fluent translations that are useful for matching general passages written in natural language. Both baseline SMT-based approaches presented above are agnostic of the ultimate retrieval task and are not specifically adapted for it. The boosting method, on the other hand, learns domain-specific word associations that are useful to discern relevant from irrelevant documents. In order to combine these orthogonal sources of information in a way that democratically respects each approach we use Borda Counts, i.e., a consensus-based voting procedure that has been successfully employed to aggregate ranked lists of documents for metasearch (Aslam and Montague, 2001).

We implemented a weighted version of the Borda Count method where each voter has a fixed amount of voting points which she is free to distribute among the candidates to indicate the amount of preference she is giving to each of them. In the case of retrieval, for each $\mathbf{q}$, the candidates are the scored documents in the retrieved subset of the whole document set. The aggregate score f_{agg} for two rankings $f_1(\mathbf{q}, \mathbf{d})$ and $f_2(\mathbf{q}, \mathbf{d})$ for all $(\mathbf{q}, \mathbf{d})$ in the test set is then:

$$f_{agg}(\mathbf{q}, \mathbf{d}) = \kappa \frac{f_1(\mathbf{q}, \mathbf{d})}{\sum_{\mathbf{d}} f_1(\mathbf{q}, \mathbf{d})} + (1-\kappa) \frac{f_2(\mathbf{q}, \mathbf{d})}{\sum_{\mathbf{d}} f_2(\mathbf{q}, \mathbf{d})}.$$

In practice, the normalizations sum over the top K retrieved documents. If a document is present only in the top-K list of one system, its score is considered zero for the other system. The aggregated scores $f_{agg}(\mathbf{q}, \mathbf{d})$ are sorted in descending order and top K scores are kept for evaluation.

Using the terminology proposed by Belkin et al. (1995), combining several systems' scores with Borda Counts can be viewed as the "data fusion" approach to IR, that merges outputs of the systems, while the PSQ baseline is an example of the "query combination" approach that extends the query at the input. Both techniques were earlier found to have similar performance in CLIR tasks based on direct translation, with a preference for the data fusion approach (Jones and Lam-Adesina, 2002).

5 Translation and Ranking Data

5.1 Parallel Translation Data

For Japanese-to-English patent translation we used data provided by the organizers of the NTCIR[9] workshop for the JP-EN PatentMT subtask. In particular, we used the data provided for NTCIR-7 (Fujii et al., 2008), consisting of 1.8 million parallel sentence pairs from the years 1993-2002 for training. For parameter tuning we used the development set of the NTCIR-8 test collection, consisting of 2,000 sentence pairs. The data were extracted from the description section of patents published by the Japanese Patent Office (JPO) and the United States Patent and Trademark Office (USPTO) by the method described in Utiyama and Isahara (2007).

Japanese text was segmented using the MeCab[10] toolkit. Following Feng et al. (2011), we applied a modified version of the compound splitter described in Koehn and Knight (2003) to katakana terms, which are often transliterations of English compound words. As these are usually not split by MeCab, they can cause a large number of out-of-vocabulary terms.

[8] It is possible to construct separate query and document inverted indices and intersect them on the fly to determine the set of documents that contains some pair of words. In practice, however, we found the overhead of set intersection during each feature access prohibitive.

[9] http://research.nii.ac.jp/ntcir/ntcir/

[10] https://code.google.com/p/mecab/

	#queries	#relevant	#unique docs
train	107,061	1,422,253	888,127
dev	2,000	26,478	25,669
test	2,000	25,173	24,668

Table 1: Statistics of ranking data.

For the English side of the training data, we applied a modified version of the tokenizer included in the Moses scripts. This tokenizer relies on a list of non-breaking prefixes which mark expressions that are usually followed by a "." (period). We customized the list of prefixes by adding some abbreviations like "Chem", "FIG" or "Pat", which are specific to patent documents.

5.2 Ranking Data from Patent Citations

Graf and Azzopardi (2008) describe a method to extract relevance judgements for patent retrieval from patent citations. The key idea is to regard patent documents that are cited in a query patent, either by the patent applicant, or by the patent examiner or in a patent office's search report, as relevant for the query patent. Furthermore, patent documents that are related to the query patent via a patent family relationship, i.e., patents granted by different patent authorities but related to the same invention, are regarded as relevant. We assign three integer relevance levels to these three categories of relationships, with highest relevance (3) for family patents, lower relevance for patents cited in search reports by patent examiners (2), and lowest relevance level (1) for applicants' citations. We also include all patents which are in the same patent family as an applicant or examiner citation to avoid false negatives. This methodology has been used to create patent retrieval data at CLEF-IP[11] and proved very useful to automatically create a patent retrieval dataset for our experiments.

For the creation of our dataset, we used the MAREC[12] citation graph to extract patents in citation or family relation. Since the Japanese portion of the MAREC corpus only contains English abstracts, but not the Japanese full texts, we merged the patent documents in the NTCIR-10 test collection described above with the Japanese (JP) section of MAREC. Title, abstract, description and claims were added to the MAREC-JP data if the document was available in NTCIR. In order to keep parallel data for SMT training separate from ranking data, we used only data from the years 2003-2005 to extract training data for ranking, and two small datasets of 2,000 queries each from the years 2006-2007 for development and testing. Table 1 gives an overview over the data used for ranking. For development and test data, we randomly added irrelevant documents from the NTCIR-10 collection until we obtained two pools of 100,000 documents. The necessary information to reproduce the exact train, development and test data samples is downloadable from authors' webpage[13].

The experiments reported here use only the abstract of the Japanese and English patents in our training, development and test collection.

6 Experiments

6.1 System Development

System development and evaluation in our experiments was done on the ranking data described in the previous section (see Table 1). We report Mean Average Precision (MAP) scores, using the `trec_eval` (ver. 8.1) script from the TREC evaluation campaign[14], with a limit of top $K = 1,000$ retrieved documents for each query. Furthermore, we use the Patent Retrieval Evaluation Score (PRES)[15] introduced by Magdy and Jones (2010). This metric accounts for both precision and recall. In the study by Magdy and Jones (2010), PRES agreed with MAP in almost 80% of cases, and both agreed on the ranks of the best and the worst IR system. Both MAP and PRES scores are reported in the same range $[0, 1]$, and 0.01 stands for 1 MAP (PRES) point. Statistical significance of pairwise system comparisons was assessed using the paired randomization test (Noreen, 1989; Smucker et al., 2007).

For each system, optimal meta-parameter settings were found by choosing the configuration with highest MAP score on the development set. These results

[11] http://www.ifs.tuwien.ac.at/~clef-ip/

[12] http://www.ifs.tuwien.ac.at/imp/marec.shtml

[13] http://www.cl.uni-heidelberg.de/statnlpgroup/boostclir

[14] http://trec.nist.gov/trec_eval

[15] http://www.computing.dcu.ie/~wmagdy/Scripts/PRESeval.htm

method	MAP		PRES	
	dev	test	dev	test
1 DT	0.2636	0.2555	0.5669	0.5681
2 PSQ lexical table	0.2520	0.2444	0.5445	0.5498
3 PSQ n-best table	0.2698	0.2659	0.5789	0.5851
Boost-1g	0.2064	^{123}0.1982	0.5850	^{12}0.6122
Boost-2g	0.2526	^{3}0.2474	0.6900	^{123}0.7196

Table 2: MAP and PRES scores for CLIR methods (best configurations) on the development and test sets. Prefixed numbers denote statistical significance of a pairwise comparison with the baseline indicated by the superscript. For example, the bottom right result shows that Boost-2g is significantly better than DT (method 1), PSQ lexical table (method 2) and PSQ n-best table (method 3).

(together with PRES results) are shown in the second and fourth column of Table 2.

The direct translation approach (DT) was developed in three configurations: no stopword filtering, small stopword list (52 words) and a large stopword list (543 words). The last configuration achieved the highest score (MAP 0.2636).

The probabilistic structured query (PSQ) approach was developed using the lexical translation table and the translation table estimated on the decoder's n-best list, both optionally pruned with a variable lower p_L and cumulative p_C threshold on the word pair probability in the table (Section 3.2). A further meta-parameter of PSQ was whether to use standard or unique n-best lists. Finally, all variants were coupled with the same stopword filters as in the DT approach. The configurations that achieved the highest scores were: MAP 0.2520 for PSQ with a lexical table ($p_L = 0.01, p_C = 0.95$, no stopword filtering), and MAP 0.2698 for PSQ with a translation table estimated on the n-best list ($p_L = 0.005, p_C = 0.95$, large stopword list). Interpolating between lexical and n-best tables did not improve results in our experiments, thus we set $\lambda = 1$ in equation (4).

Each SMT-based system was run with 4 different MERT optimizations, leading to variations of less than 1 MAP point for each system. The best configurations for DT and PSQ on the development set were fixed and used for evaluation on the test set.

Training of the boosting approach (Boost) was done in parallel on bootstrap samples from the training data. First, a query $\mathbf{q}$ (i.e., a Japanese abstract) was sampled uniformly from all training queries.

method	MAP		PRES	
	dev	test	dev	test
DT + PSQ n-best	0.2778	*0.2726	0.5884	*0.5942
DT + Boost-1g	0.2778	*0.2728	0.6157	*0.6225
DT + Boost-2g	0.3309	***0.3300**	0.7132	***0.7279**
PSQ lexical + Boost-1g	0.2695	*0.2653	0.6068	*0.6131
PSQ lexical + Boost-2g	0.3215	***0.3187**	0.7071	***0.7240**
PSQ n-best + Boost-1g	0.2863	*0.2850	0.6309	*0.6402
PSQ n-best + Boost-2g	0.3439	***0.3416**	0.7212	***0.7376**

Table 3: MAP and PRES scores of the aggregated models on the development and test sets. Development scores correspond to peaks in Figures 1 and 3, respectively, for MAP and PRES; test scores are given for the κ's delivering these peaks on the development set. Prefixed $*$ indicates statistical significance of the result difference between aggregated system and the respective translation-based system used in the aggregation.

Then we sampled independently and uniformly a relevant document $\mathbf{d}^+$ (i.e., an English abstract) from the English patents marked relevant for the Japanese patent, and a random document $\mathbf{d}^-$ from the whole pool of English patent abstracts. If $\mathbf{d}^-$ had a relevance score greater or equal to the relevance score of $\mathbf{d}^+$, it was resampled. The initial importance weight D_0 for a triplet $(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-)$ was set to the positive difference in relevance scores for $\mathbf{d}^+$ and $\mathbf{d}^-$. Each bootstrap sample consisted of 10 pairs of documents for each of $10,000$ queries, resulting in $100,000$ training instances per sample.

The Boost approach was developed for uni-gram and combined uni- and bi-gram versions. We observed that the performance of the Boost method continuously improved with the number of iterations T and with the number of samples S, but saturated at about 15-20 samples without visible over-fitting in the tested range of T. Therefore we arbitrarily stopped training after obtaining $5,000$ features per sample, and used 35 samples for uni-gram version and 65 samples for the combined bi-gram version, resulting in models with 104K and 172K unique features, respectively. The optimal values for the pass-through weight β were found to be 0.3 and 0.2 for the uni-gram and bi-gram models on the development set. The best configuration of uni-gram and bi-gram model achieved MAP scores of 0.2064 and 0.2526 the development set. Using stopword filters during training did not improve the results here.

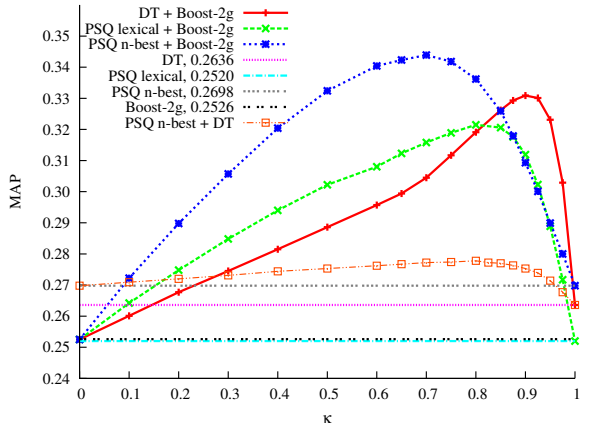

Figure 1: MAP rank aggregation for combinations of the bi-gram boosting and the baselines on the dev set.

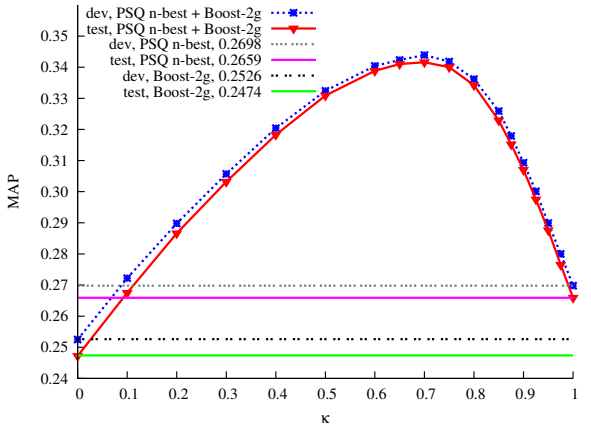

Figure 2: MAP rank aggregation for the bi-gram boosting and the "PSQ n-best table" approach on dev and test sets.

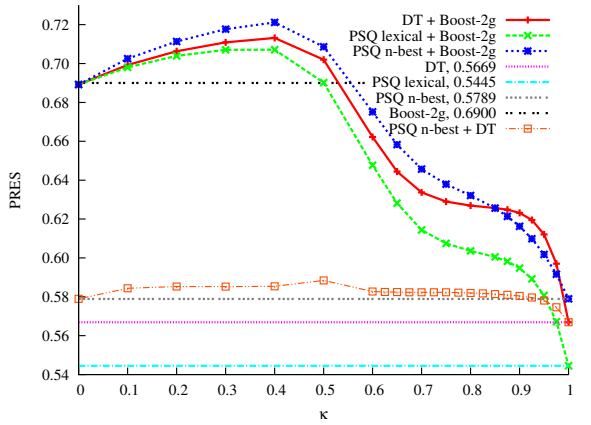

Figure 3: PRES rank aggregation for combinations of the bi-gram boosting and the baselines on the dev set.

6.2 Testing and Model Combination

The third and the fifth columns of Table 2 give a comparison of the MAP scores of the baseline approaches and the Boost model evaluated individually on the test set. Each score corresponds to the best configuration found on the development set. We see that the PSQ approach using n-best lists for projection outperforms all other methods in terms of MAP, but loses to both Boost approaches when evaluated with PRES. Direct translation is about 1 MAP point lower than PSQ n-best; Boost with combined uni- and bi-grams is another 0.8 MAP points worse, but is better in terms of PRES, especially for the bi-gram version. Given the fact that the complex SMT system behind the direct translation and PSQ approach is trained and tuned on very large in-domain datasets, the performance of the bare phrase table induced by the Boost method is respectable.

Our best results are obtained by a combination of the orthogonal information sources of the SMT and the Boost approaches. We evaluated the Borda Count aggregation scheme on the development data in order to find the optimal value for $\kappa \in [0, 1]$. The interpolation was done for the best combined uni- and bi-gram boosting model with the best variants of the DT and PSQ approaches. As can be seen from Figures 1 and 3, rank aggregation by Borda Count outperforms both individual approaches by a large margin. Figure 2 verifies that the results are transferable from the development set to the test set. The best performing system combination on the development data is also optimal on the test data.

Table 3 shows the retrieval performance of the best baseline model (PSQ n-best) combined with the best Boost model (bi-gram), with an impressive gain of over 7 MAP points (15 PRES points) over the best individual baseline result from Table 2. Even when, according to the PRES measure (Figure 3), the Boost-2g system is better on its own, injecting complementary information from the PSQ or DT approach still contributes several points. Similar gains are obtained by model combination of the DT approach with the best Boost model. However, a combination of the SMT-based CLIR approaches DT and PSQ barely improved results over the best input model. In summary, aggregating rankings is helpful for orthogonal systems, but not for systems including similar information.

6.3 Analysis

Table 4 lists some of the top-200 selected features for the boosting approach (the most common translation of the Japanese term is put in subscript).

We see that the direct ranking approach is able to penalize uni- and bi-gram cooccurrences that are harmful for retrieval by assigning them a negative weight, e.g., the pairing of 解決$_{\text{resolution}}$ with *image*. Pairs of uni- and bi-grams that are useful for retrieval are boosted by positive weights, e.g., the pair 圧縮$_{\text{compression}}$,機$_{\text{machine}}$ and *compressor* captures an important compound. Further examples, not shown in the table, are matches of the same source (target) n-gram with several different target (source) n-grams, e.g., the Japanese term 画像$_{\text{image}}$ is paired not only with its main translation, but also with dozens of related notions: *video*, *picture*, *scanning*, *printing*, *photosensitive*, *pixel*, *background* etc. This has a query expansion effect that is not possible in systems that use one translation or a small list of n-best translations. In addition, associations of source n-grams with overlapping target n-grams help boost the final score: e.g., the same term 画像$_{\text{image}}$ is positively paired with target bi-grams as {*an,original*}, {*original,image*} and {*image,for*}. This has the effect of compensating for the lack of handling phrase overlaps in an SMT decoder.

t	h_t **(uni- & bi-grams)**	w_t
1	層$_{\text{layer}}$ - layer	1.29
2	データ$_{\text{data}}$ - data	1.13
3	回路$_{\text{circuit}}$ - circuit	1.13
76	で$_{\text{in}}$ - voltage	-0.39
77	導$_{\text{guide}}$,電$_{\text{power}}$ - conductive	1.25
81	解決$_{\text{resolution}}$ - image	-0.25
99	変速$_{\text{speed}}$ - transmission	1.68
100	液晶$_{\text{LCD}}$ - liquid,crystal	1.73
123	力$_{\text{power}}$ - force	0.91
124	圧縮$_{\text{compression}}$,機$_{\text{machine}}$ - compressor	2.83
132	ケーブル$_{\text{cable}}$ - cable	1.81
133	超$_{\text{hyper}}$,音波$_{\text{sound wave}}$ - ultrasonic	3.34
169	粒子$_{\text{particle}}$ - particles	1.57
170	算出$_{\text{calculation}}$ - for,each	1.14
184	ロータ$_{\text{rotor}}$ - rotor	2.01
185	検出$_{\text{detection}}$,器$_{\text{vessel}}$ - detector	1.43

Table 4: Examples of the features found by boosting.

7 Conclusion

We presented a boosting approach to induce a table of bilingual n-gram correspondences by direct preference learning on relevance rankings. This table can be seen as a phrase table that encodes word-based information that is orthogonal and complementary to the information in standard translation-based CLIR approaches. We compared our boosting approach to very competitive CLIR baselines that use a complex SMT system trained and tuned on large in-domain datasets. Furthermore, our patent retrieval setup gives SMT-based approaches an advantage in that queries consist of several normal-length sentences, as opposed to the short queries common to web search. Despite this and despite the tiny size (about 170K parameters) of the boosting phrase table, compared to standard SMT phrase tables, this approach reached performance similar to direct translation using a full SMT model in terms of MAP, and was significantly better in terms of PRES. Overall, we obtained the best results by a model combination using consensus- based voting where the best SMT-based approach was combined with the boosting phrase table (gaining more than 7 MAP or 15 PRES points). We attribute this to the fact that the boosting approach augments SMT approaches with valuable information that is hard to get in approaches that are agnostic about the ranking data and the ranking task at hand.

The experimental setup presented in this paper uses relevance links between patent abstracts as ranking data. While this technique is useful to develop patent retrieval systems, it would be interesting to see if our results transfer to patent retrieval scenarios where full patent documents are used instead of only abstracts, or to standard CLIR scenarios that use short search queries in retrieval.

Acknowledgements

The research presented in this paper was supported in part by DFG grant "Cross-language Learning-to-Rank for Patent Retrieval". We would like to thank Eugen Ruppert for his contribution to the ranking data construction.

References

Javed A. Aslam and Mark Montague. 2001. Models for metasearch. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, New Orleans, LA.

Bing Bai, Jason Weston, David Grangier, Ronan Collobert, Kunihiko Sadamasa, Yanjun Qi, Olivier Chapelle, and Kilian Weinberger. 2010. Learning to rank with (a lot of) word features. *Information Retrieval Journal*, 13(3):291–314.

Lisa Ballesteros and W. Bruce Croft. 1998. Resolving ambiguity for cross-language retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'98)*, Melbourne, Australia.

Nicholas J. Belkin, Paul Kantor, Edward A. Fox, and Joseph A. Shaw. 1995. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448.

Leo Breiman. 1996. Bagging predictors. *Journal of Machine Learning Research*, 24:123–140.

Kevin Canini, Tushar Chandra, Eugene Ie, Jim McFadden, Ken Goldman, Mike Gunter, Jeremiah Harmsen, Kristen LeFevre, Dmitry Lepikhin, Tomas Lloret Llinares, Indraneel Mukherjee, Fernando Pereira, Josh Redstone, Tal Shaked, and Yoram Singer. 2010. Sibyl: A system for large scale machine learning. In *LADIS: The 4th ACM SIGOPS/SIGACT Workshop on Large Scale Distributed Systems and Middleware*, Zurich, Switzerland.

Xi Chen, Bing Bai, Yanjun Qi, Qihang Ling, and Jaime Carbonell. 2010. Learning preferences with millions of parameters by enforcing sparsity. In *Proceedings of the IEEE International Conference on Data Mining (ICDM'10)*, Sydney, Australia.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Jeffrey Chin, Maureen Heymans, Alexandre Kojoukhov, Jocelyn Lin, and Hui Tan. 2008. Cross-language information retrieval. Patent Application. US 2008/0288474 A1.

Michael Collins and Terry Koo. 2005. Discriminative reranking for natural language parsing. *Computational Linguistics*, 31(1):25–69.

Michael Collins, Robert E. Schapire, and Yoram Singer. 2002. Logistic regression, AdaBoost and Bregman distances. *Journal of Machine Learning Research*, 48(1-3):253–285.

Kareem Darwish and Douglas W. Oard. 2003. Probabilistic structured query methods. In *Proceedings. of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'03)*, Toronto, Canada.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Jonathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, Uppsala, Sweden.

Minwei Feng, Christoph Schmidt, Joern Wuebker, Stephan Peitz, Markus Freitag, and Hermann Ney. 2011. The RWTH Aachen system for NTCIR-9 PatentMT. In *Proceedings of the NTCIR-9 Workshop*, Tokyo, Japan.

Yoav Freund, Ray Iyer, Robert E. Schapire, and Yoram Singer. 2003. An efficient boosting algorithm for combining preferences. *Journal of Machine Learning Research*, 4:933–969.

Atsushi Fujii, Masao Utiyama, Mikio Yamamoto, and Takehito Utsuro. 2008. Overview of the patent translation task at the NTCIR-7 workshop. In *Proceedings of NTCIR-7 Workshop Meeting*, Tokyo, Japan.

Yasser Ganjisaffar, Rich Caruana, and Cristina Videira Lopes. 2011. Bagging gradient-boosted trees for high precision, low variance ranking models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'11)*, Beijing, China.

Jianfeng Gao, Jian-Yun Nie, Endong Xun, Jian Zhang, Ming Zhou, and Changning Huang. 2001. Improving query translation for cross-language information retrieval using statistical models. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, New Orleans, LA.

Wei Gao, Cheng Niu, Jian-Yun Nie, Ming Zhou, Jian Hu, Kam-Fai Wong, and Hsiao-Wuen Hon. 2007. Cross-lingual query suggestion using query logs of different languages. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'07)*, Amsterdam, The Netherlands.

Sharad Goel, John Langford, and Alexander L. Strehl. 2008. Predictive indexing for fast search. In *Advances in Neural Information Processing Systems*, Vancouver, Canada.

Erik Graf and Leif Azzopardi. 2008. A methodology for building a patent test collection for prior art search. In *Proceedings of the 2nd International Workshop on Evaluating Information Access (EVIA)*, Tokyo, Japan.

David Grangier and Samy Bengio. 2008. A discriminative kernel-based approach to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(8):1371–1384.

Yunsong Guo and Carla Gomes. 2009. Ranking structured documents: A large margin based approach for

patent prior art search. In *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI'09)*, Pasadena, CA.

Kenneth Heafield. 2011. KenLM: faster and smaller language model queries. In *Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation (WMT'11)*, Edinburgh, UK.

Gareth J.F. Jones and Adenike M. Lam-Adesina. 2002. Combination methods for improving the reliability of machine translation based cross-language information retrieval. In *Proceedings of the 13th Irish International Conference on Artificial Intelligence and Cognitive Science (AICS'02)*, Limerick, Ireland.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the Conference on European Chapter of the Association for Computational Linguistics (EACL'03)*, Budapest, Hungary.

Victor Lavrenko, Martin Choquette, and W. Bruce Croft. 2002. Cross-lingual relevance models. In *Proceedings of the ACM Conference on Research and Development in Information Retrieval (SIGIR'02)*, Tampere, Finland.

Adam Lopez. 2007. Hierarchical phrase-based translation with suffix arrays. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL 2007)*, Prague, Czech Republic.

Walid Magdy and Gareth J.F. Jones. 2010. PRES: a score metric for evaluating recall-oriented information retrieval applications. In *Proceedings of the ACM SIGIR conference on Research and development in information retrieval (SIGIR'10)*, New York, NY.

Walid Magdy and Gareth J. F. Jones. 2011. An efficient method for using machine translation technologies in cross-language patent search. In *Proceedings of the 20th ACM Conference on Informationand Knowledge Management (CIKM'11)*, Glasgow, Scotland, UK.

Vassilina Nikoulina, Bogomil Kovachev, Nikolaos Lagos, and Christof Monz. 2012. Adaptation of statistical machine translation model for cross-lingual information retrieval in a service context. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL'12)*, Avignon, France.

Eric W. Noreen. 1989. *Computer Intensive Methods for Testing Hypotheses. An Introduction*. Wiley, New York.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Meeting of the Association for Computational Linguistics (ACL'00)*, Hongkong, China.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Meeting on Association for Computational Linguistics (ACL'03)*, Sapporo, Japan.

Dmitry Pavlov, Alexey Gorodilov, and Cliff A. Brunk. 2010. Bagboo: a scalable hybrid bagging-the-boosting model. In *Proceedings of the 19th ACM International Conference on Information and Knowledge Management (CIKM'10)*, Toronto, Canada.

Stephen E. Robertson, Steve Walker, and Micheline Hancock-Beaulieu. 1998. Okapi at TREC-7. In *Proceedings of the Seventh Text REtrieval Conference (TREC-7)*, Gaithersburg, MD.

Saharon Rosset, Ji Zhu, and Trevor Hastie. 2004. Boosting as a regularized path to a maximum margin classifier. *Journal of Machine Learning Research*, 5:941–973.

Robert E. Schapire and Yoram Singer. 1999. Improved boosting algorithms using confidence-rated predictions. *Journal of Machine Learning Research*, 37(3):297–336.

Qinfeng Shi, James Petterson, Gideon Dror, John Langford, Alexander J. Smola, Alexander L. Strehl, and Vishy Vishwanathan. 2009. Hash Kernels. In *Proceedings of the 12th Int. Conference on Artificial Intelligence and Statistics (AISTATS'09)*, Irvine, CA.

Mark D. Smucker, James Allan, and Ben Carterette. 2007. A comparison of statistical significance tests for information retrieval evaluation. In *Proceedings of the 16th ACM conference on Conference on Information and Knowledge Management (CIKM '07)*, New York, NY.

Ferhan Ture, Jimmy Lin, and Douglas W. Oard. 2012a. Combining statistical translation techniques for cross-language information retrieval. In *Proceedings of the International Conference on Computational Linguistics (COLING 2012)*, Bombay, India.

Ferhan Ture, Jimmy Lin, and Douglas W. Oard. 2012b. Looking inside the box: Context-sensitive translation for cross-language information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR 2012)*, Portland, OR.

Masao Utiyama and Hitoshi Isahara. 2007. A Japanese-English patent parallel corpus. In *Proceedings of MT Summit XI*, Copenhagen, Denmark.

Jinxi Xu, Ralph Weischedel, and Chanh Nguyen. 2001. Evaluating a probabilistic model for cross-lingual information retrieval. In *Proceedings of the ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR'01)*, New York, NY.