Learning Translational and Knowledge-based Similarities from Relevance Rankings for Cross-Language Retrieval

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

Abstract

We present an approach to cross-language retrieval that combines dense knowledge-based features and sparse word translations. Both feature types are learned directly from relevance rankings of bilingual documents in a pairwise ranking framework. In large-scale experiments for patent prior art search and cross-lingual retrieval in Wikipedia, our approach yields considerable improvements over learning-to-rank with either only dense or only sparse features, and over very competitive baselines that combine state-of-the-art machine translation and retrieval.

1 Introduction

Cross-Language Information Retrieval (CLIR) for the domain of web search successfully leverages state-of-the-art Statistical Machine Translation (SMT) to either produce a single most probable translation, or a weighted list of alternatives, that is used as search query to a standard search engine (Chin et al., 2008; Ture et al., 2012). This approach is advantageous if large amounts of in-domain sentence-parallel data are available to train SMT systems, but relevance rankings to train retrieval models are not.

The situation is different for CLIR in special domains such as patents or Wikipedia. Parallel data for translation have to be extracted with some effort from comparable or noisy parallel data (Utiyama and Isahara, 2007; Smith et al., 2010), however, relevance judgments are often straightforwardly encoded in special domains. For example, in patent prior art search, patents granted at any patent office worldwide are considered relevant if they constitute prior art with respect to the invention claimed in the query patent. Since patent applicants and lawyers are required to list relevant prior work explicitly in the patent application, patent citations can be used to automatically extract large amounts of relevance judgments across languages (Graf and Azzopardi, 2008). In Wikipedia search, one can imagine a Wikipedia author trying to investigate whether a Wikipedia article covering the subject the author intends to write about already exists in another language. Since authors are encouraged to avoid orphan articles and to cite their sources, Wikipedia has a rich linking structure between related articles, which can be exploited to create relevance links between articles across languages (Bai et al., 2010).

Besides a rich citation structure, patent documents and Wikipedia articles contain a number of further cues on relatedness that can be exploited as features in learning-to-rank approaches. For monolingual patent retrieval, Guo and Gomes (2009) and Oh et al. (2013) advocate the use of dense features encoding domain knowledge on inventors, assignees, location and date, together with dense similarity scores based on bag-of-word representations of patents. Bai et al. (2010) show that for the domain of Wikipedia, learning a sparse matrix of word associations between the query and document vocabularies from relevance rankings is useful in monolingual and cross-lingual retrieval. Sokolov et al. (2013) apply the idea of learning a sparse matrix of bilingual phrase associations from relevance rankings to cross-lingual retrieval in the patent domain. Both show improvements of learning-to-rank on relevance data over SMT-based approaches on their respective domains.

The main contribution of this paper is a thorough evaluation of dense and sparse features for learning-to-rank that have so far been used only monolingually or only on either patents or Wikipedia. We show that for both domains, patents and Wikipedia, jointly learning bilingual sparse word associations and dense knowledge-based similarities directly on relevance ranked

Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Short Papers), pages 488–494,
Baltimore, Maryland, USA, June 23-25 2014.

data improves significantly over approaches that use either only sparse or only dense features, and over approaches that combine query translation by SMT with standard retrieval in the target language. Furthermore, we show that our approach can be seen as supervised model combination that allows to combine SMT-based and ranking-based approaches for further substantial improvements. We conjecture that the gains are due to orthogonal information contributed by domain-knowledge, ranking-based word associations, and translation-based information.

2 Related Work

CLIR addresses the problem of translating or projecting a query into the language of the document repository across which retrieval is performed. In a *direct translation* approach (DT), a state-of-the-art SMT system is used to produce a single best translation that is used as search query in the target language. For example, Google's CLIR approach combines their state-of-the-art SMT system with their proprietary search engine (Chin et al., 2008).

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 (PSQ). 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). Ture et al. (2012) brought SMT back into this paradigm by projecting terms from n-best translations from synchronous context-free grammars.

Ranking approaches have been presented by Guo and Gomes (2009) and Oh et al. (2013). Their method is a classical learning-to-rank setup where pairwise ranking is applied to a few hundred dense features. Methods to learn sparse word-based translation correspondences from supervised ranking signals have been presented by Bai et al. (2010) and Sokolov et al. (2013). Both approaches work in a cross-lingual setting, the former on Wikipedia data, the latter on patents.

Our approach extends the work of Sokolov et al. (2013) by presenting an alternative learning-to-rank approach that can be used for supervised model combination to integrate dense and sparse features, and by evaluating both approaches on cross-lingual retrieval for patents and Wikipedia. This relates our work to supervised model merging approaches (Sheldon et al., 2011).

3 Translation and Ranking for CLIR

SMT-based Models. We will refer to DT and PSQ as SMT-based models that translate a query, and then perform monolingual retrieval using BM25. Translation is agnostic of the retrieval task.

Linear Ranking for Word-Based Models. Let $\mathbf{q} \in \{0,1\}^Q$ be a query and $\mathbf{d} \in \{0,1\}^D$ be a document where the j^{th} vector dimension indicates the occurrence of the j^{th} word for dictionaries of size Q and D. A linear ranking model is defined as

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

where $W \in \mathbb{R}^{Q\times D}$ encodes a matrix of ranking-specific word associations (Bai et al., 2010) . We optimize this model by pairwise ranking, which assumes 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 such that an inequality $f(\mathbf{q},\mathbf{d}^+) > f(\mathbf{q},\mathbf{d}^-)$ is violated for the fewest number of tuples from $\mathcal{R}$. We present two methods for optimizing W in the following.

Pairwise Ranking using Boosting (BM). The Boosting-based Ranking baseline (Freund et al., 2003) optimizes an exponential loss:

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

where $\mathcal{D}(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-)$ is a non-negative importance function on tuples. The algorithm of Sokolov et al. (2013) combines batch boosting with bagging over a number of independently drawn bootstrap data samples from $\mathcal{R}$. In each step, the single word pair feature is selected that provides the largest decrease of $\mathcal{L}_{exp}$. The found corresponding models are averaged. To reduce memory requirements we used random feature hashing with the size of the hash of 30 bits (Shi et al., 2009). For regularization we rely on early stopping.

Pairwise Ranking with SGD (VW). The second objective is an ℓ_1-regularized hinge loss:

$$\mathcal{L}_{hng} = \sum_{(\mathbf{q},\mathbf{d}^+,\mathbf{d}^-)\in\mathcal{R}} \big(f(\mathbf{q},\mathbf{d}^+) - f(\mathbf{q},\mathbf{d}^-)\big)_+ + \lambda||W||_1,$$

where $(x)_+ = \max(0, 1-x)$ and λ is the regularization parameter. This newly added model utilizes the standard implementation of online SGD from the Vowpal Wabbit (VW) toolkit (Goel et al., 2008) and was run on a data sample of 5M to 10M tuples from $\mathcal{R}$. On each step, W is updated with a scaled gradient vector $\nabla_W \mathcal{L}_{hng}$ and clipped to account for ℓ_1-regularization. Memory usage was reduced using the same hashing technique as for boosting.

Domain Knowledge Models. Domain knowledge features for patents were inspired by Guo and Gomes (2009): a feature fires if two patents share similar aspects, e.g. a common inventor. As we do not have access to address data, we omit geolocation features and instead add features that evaluate similarity w.r.t. patent classes extracted from IPC codes. Documents within a patent section, i.e. the topmost hierarchy, are too diverse to provide useful information but more detailed classes and the count of matching classes do.

For Wikipedia, we implemented features that compare the relative length of documents, number of links and images, the number of common links and common images, and Wikipedia categories: Given the categories associated with a foreign query, we use the language links on the Wikipedia category pages to generate a set of "translated" English categories S. The English-side category graph is used to construct sets of super- and subcategories related to the candidate document's categories. This expansion is done in both directions for two levels resulting in 5 category sets. The intersection between target set T_n and the source category set S reflects the category level similarity between query and document, which we calculate as a mutual containment score $\mathrm{s}_n = \frac{1}{2}(|S \cap T_n|/|S| + |S \cap T_n|/|T_n|)$ for $n \in \{-2, -1, 0, +1, +2\}$ (Broder, 1997).

Optimization for these additional models including domain knowledge features was done by overloading the vector representation of queries $\mathbf{q}$ and documents $\mathbf{d}$ in the VW linear learner: Instead of sparse word-based features, $\mathbf{q}$ and $\mathbf{d}$ are represented by real-valued vectors of dense domain-knowledge features. Optimization for the overloaded vectors is done as described above for VW.

4 Model Combination

Combination by Borda Counts. The baseline consensus-based voting Borda Count procedure endows each voter with a fixed amount of voting points which he is free to distribute among the scored documents (Aslam and Montague, 2001; Sokolov et al., 2013). The aggregate score 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 a simple linear interpolation: $f_{agg}(\mathbf{q}, \mathbf{d}) = \kappa \frac{f_1(\mathbf{q},\mathbf{d})}{\sum_{\mathbf{d}} f_1(\mathbf{q},\mathbf{d})} + (1-\kappa) \frac{f_2(\mathbf{q},\mathbf{d})}{\sum_{\mathbf{d}} f_2(\mathbf{q},\mathbf{d})}$. Parameter κ was adjusted on the dev set.

Combination by Linear Learning. In order to acquire the best combination of more than two models, we created vectors of model scores along with domain knowledge features and reused the VW pairwise ranking approach. This means that the vector representation of queries $\mathbf{q}$ and documents $\mathbf{d}$ in the VW linear learner is overloaded once more: In addition to dense domain-knowledge features, we incorporate arbitrary ranking models as dense features whose value is the score of the ranking model. Training data was sampled from the dev set and processed with VW.

5 Data

Patent Prior Art Search (JP-EN). We use BoostCLIR[1], a Japanese-English (JP-EN) corpus of patent abstracts from the MAREC and NTCIR data (Sokolov et al., 2013). It contains automatically induced relevance judgments for patent abstracts (Graf and Azzopardi, 2008): EN patents are regarded as relevant with level (3) to a JP query patent, if they are in a family relationship (e.g., same invention), cited by the patent examiner (2), or cited by the applicant (1). Statistics on the ranking data are given in Table 1. On average, queries and documents contain about 5 sentences.

Wikipedia Article Retrieval (DE-EN). The intuition behind our Wikipedia retrieval setup is as follows: Consider the situation where the German (DE) Wikipedia article on geological sea *stacks* does not yet exist. A native speaker of German with profound knowledge in geology intends to write it, naming it "*Brandungspfeiler*", while seeking to align its structure with the EN counterpart. The task of a CLIR engine is to return relevant EN Wikipedia articles that may describe the very same concept (*Stack (geology)*), or relevant instances of it (*Bako National Park*, *Lange Anna*). The information need may be paraphrased as a high-level definition of the topic. Since typically the first sentence of any Wikipedia article is such

[1] www.cl.uni-heidelberg.de/boostclir

	$\#\mathbf{q}$	$\#\mathbf{d}$	$\#\mathbf{d}^+/\mathbf{q}$	$\#words/\mathbf{q}$
Patents (JP-EN)				
train	107,061	888,127	13.28	178.74
dev	2,000	100,000	13.24	181.70
test	2,000	100,000	12.59	182.39
Wikipedia (DE-EN)				
train	225,294	1,226,741	13.04	25.80
dev	10,000	113,553	12.97	25.75
test	10,000	115,131	13.22	25.73

Table 1: Ranking data statistics: number of queries and documents, avg. number of relevant documents per query, avg. number of words per query.

a well-formed definition, this allows us to extract a large set of one sentence queries from Wikipedia articles. For example: "*Brandungspfeiler sind vor einer Kliffküste aufragende Felsentürme und vergleichbare Formationen, die durch Brandungserosion gebildet werden.*"[2] Similar to Bai et al. (2010) we induce relevance judgments by aligning DE queries with their EN counterparts ("mates") via the graph of inter-language links available in articles and Wikidata[3]. We assign relevance level (3) to the EN mate and level (2) to all other EN articles that link to the mate, *and* are linked by the mate. Instead of using all outgoing links from the mate, we only use articles with bidirectional links.

To create this data[4] we downloaded XML and SQL dumps of the DE and EN Wikipedia from, resp., 22nd and 4th of November 2013. Wikipedia markup removal and link extraction was carried out using the Cloud9 toolkit[5]. Sentence extraction was done with NLTK[6]. Since Wikipedia articles vary greatly in length, we restricted EN documents to the first 200 words after extracting the link graph to reduce the number of features for BM and VW models. To avoid rendering the task too easy for literal keyword matching of queries about named entities, we removed title words from the German queries. Statistics are given in Table 1.

Preprocessing Ranking Data. In addition to lowercasing and punctuation removal, we applied Correlated Feature Hashing (CFH), that makes collisions more likely for words with close meaning (Bai et al., 2010). For patents, vocabularies contained 60k and 365k words for JP and EN. Filtering special symbols and stopwords reduced the JP vocabulary size to 50k (small enough not to resort to CFH). To reduce the EN vocabulary to a comparable size, we applied similar preprocessing *and* CFH with F=30k and k=5. Since for Wikipedia data, the DE and EN vocabularies were both large (6.7M and 6M), we used the same filtering and preprocessing as for the patent data before applying CFH with F=40k and k=5 on both sides.

Parallel Data for SMT-based CLIR. For both tasks, DT and PSQ require an SMT baseline system trained on parallel corpora that are disjunct from the ranking data. A JP-EN system was trained on data described and preprocessed by Sokolov et al. (2013), consisting of 1.8M parallel sentences from the NTCIR-7 JP-EN PatentMT subtask (Fujii et al., 2008) and 2k parallel sentences for parameter development from the NTCIR-8 test collection. For Wikipedia, we trained a DE-EN system on 4.1M parallel sentences from Europarl, Common Crawl, and News-Commentary. Parameter tuning was done on 3k parallel sentences from the WMT'11 test set.

6 Experiments

Experiment Settings. The SMT-based models use `cdec` (Dyer et al., 2010). Word alignments were created with `mgiza` (JP-EN) and `fast_align` (Dyer et al., 2013) (DE-EN). Language models were trained with the KenLM toolkit (Heafield, 2011). The JP-EN system uses a 5-gram language model from the EN side of the training data. For the DE-EN system, a 4-gram model was built on the EN side of the training data and the EN Wikipedia documents. Weights for the standard feature set were optimized using `cdec`'s MERT (JP-EN) and MIRA (DE-EN) implementations (Och, 2003; Chiang et al., 2008). PSQ on patents reuses settings found by Sokolov et al. (2013); settings for Wikipedia were adjusted on its dev set (n=1000, λ=0.4, L=0, C=1).

Patent retrieval for DT was done by sentence-wise translation and subsequent re-joining to form one query per patent, which was ranked against the documents using BM25. For PSQ, BM25 is computed on expected term and document frequencies.

For ranking-based retrieval, we compare several combinations of learners and features (Table 2). VW denotes a sparse model using word-based features trained with SGD. BM denotes a similar model trained using Boosting. DK denotes VW training of a model that represents queries $\mathbf{q}$ and documents $\mathbf{d}$ by dense domain-knowledge features instead of by sparse word-based vectors. In

[2] `de.wikipedia.org/wiki/Brandungspfeiler`
[3] `www.wikidata.org/`
[4] `www.cl.uni-heidelberg.de/wikiclir`
[5] `lintool.github.io/Cloud9/index.html`
[6] `www.nltk.org/`

order to simulate pass-through behavior of out-of-vocabulary terms in SMT systems, additional features accounting for source and target term identity were added to DK and BM models. The parameter λ for VW was found on dev set. Statistical significance testing was performed using the paired randomization test (Smucker et al., 2007).

Borda denotes model combination by Borda Count voting where the linear interpolation parameter is adjusted for MAP on the respective development sets with grid search. This type of model combination only allows to combine pairs of rankings. We present a combination of SMT-based CLIR, DT+PSQ, a combination of dense and sparse features, DK+VW, and a combination of both combinations, (DT+PSQ)+(DK+VW).

LinLearn denotes model combination by overloading the vector representation of queries $\mathbf{q}$ and documents $\mathbf{d}$ in the VW linear learner by incorporating arbitrary ranking models as dense features. In difference to grid search for *Borda*, optimal weights for the linear combination of incorporated ranking models can be learned automatically. We investigate the same combinations of ranking models as described for *Borda* above. We do not report combination results including the sparse BM model since they were consistently lower than the ones with the sparse VW model.

Test Results. Experimental results on test data are given in Table 2. Results are reported with respect to MAP (Manning et al., 2008), NDCG (Järvelin and Kekäläinen, 2002), and PRES (Magdy and Jones, 2010). Scores were computed on the top 1,000 retrieved documents.

As can be seen from inspecting the two blocks of results, one for patents, one for Wikipedia, we find the same system rankings on both datasets. In both cases, as *standalone* systems, DT and PSQ are very close and far better than any ranking approach, irrespective of the objective function or the choice of sparse or dense features. Model combination of similar models, e.g., DT and PSQ, gives minimal gains, compared to combining orthogonal models, e.g. DK and VW. The best result is achieved by combining DT and PSQ with DK and VW. This is due to the already high scores of the combined models, but also to the combination of yet other types of orthogonal information. *Borda* voting gives the best result under MAP which is probably due to the adjustment of the interpolation parameter for MAP on the development set.

combination		models	MAP	NDCG	PRES
Patents (JP-EN)	*standalone*	DT	0.2554	0.5397	0.5680
		PSQ	0.2659	0.5508	0.5851
		DK	0.2203	0.4874	0.5171
		VW	0.2205	0.4989	0.4911
		BM	0.1669	0.4167	0.4665
	Borda	DT+PSQ	*0.2747	*0.5618	*0.5988
		DK+VW	*0.3023	*0.5980	*0.6137
		(DT+PSQ)+(DK+VW)	*0.3465	*0.6420	*0.6858
	LinLearn	DT+PSQ	†*0.2707	†*0.5578	†*0.5941
		DK+VW	†*0.3283	†*0.6366	†*0.7104
		DT+PSQ+DK+VW	†***0.3739**	†***0.6755**	†***0.7599**
Wikipedia (DE-EN)	*standalone*	DT	0.3678	0.5691	0.7219
		PSQ	0.3642	0.5671	0.7165
		DK	0.2661	0.4584	0.6717
		VW	0.1249	0.3389	0.6466
		BM	0.1386	0.3418	0.6145
	Borda	DT+PSQ	*0.3742	*0.5777	*0.7306
		DK+VW	*0.3238	*0.5484	*0.7736
		(DT+PSQ)+(DK+VW)	***0.4173**	*0.6333	*0.8031
	LinLearn	DT+PSQ	†*0.3718	†*0.5751	†*0.7251
		DK+VW	†*0.3436	†*0.5686	†*0.7914
		DT+PSQ+DK+VW	*0.4137	†***0.6435**	†***0.8233**

Table 2: Test results for *standalone* CLIR models using direct translation (DT), probabilistic structured queries (PSQ), sparse model with CFH (VW), sparse boosting model (BM), dense domain knowledge features (DK), and model combinations using Borda Count voting (*Borda*) or linear supervised model combination (*LinLearn*). Significant differences (at p=0.01) between aggregated systems and all its components are indicated by $*$, between LinLearn and the respective Borda system by $\dagger$.

Under NDCG and PRES, *LinLearn* achieves the best results, showing the advantage of automatically learning combination weights that leads to stable results across various metrics.

7 Conclusion

Special domains such as patents or Wikipedia offer the possibility to extract cross-lingual relevance data from citation and link graphs. These data can be used to directly optimizing cross-lingual ranking models. We showed on two different large-scale ranking scenarios that a supervised combination of orthogonal information sources such as domain-knowledge, translation knowledge, and ranking-specific word associations by far outperforms a pipeline of query translation and retrieval. We conjecture that if these types of information sources are available, a supervised ranking approach will yield superior results in other retrieval scenarios as well.

Acknowledgments

This research was supported in part by DFG grant RI-2221/1-1 “Cross-language Learning-to-Rank for Patent Retrieval”.

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.

Andrei Z. Broder. 1997. On the resemblance and containment of documents. In *Compression and Complexity of Sequences (SEQUENCES'97)*, pages 21–29. IEEE Computer Society.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'08)*, Waikiki, Hawaii.

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

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.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proceedings of the Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Atlanta, GA.

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.

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'08)*, Tokyo, Japan.

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.

Kalervo Järvelin and Jaana Kekäläinen. 2002. Cumulated gain-based evaluation of IR techniques. *ACM Transactions in Information Systems*, 20(4):422–446.

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.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press.

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.

Sooyoung Oh, Zhen Lei, Wang-Chien Lee, Prasenjit Mitra, and John Yen. 2013. CV-PCR: A context-guided value-driven framework for patent citation recommendation. In *Proceedings of the International Conference on Information and Knowledge Management (CIKM'13)*, San Francisco, CA.

Daniel Sheldon, Milad Shokouhi, Martin Szummer, and Nick Craswell. 2011. Lambdamerge: Merging the results of query reformulations. In *Proceedings of WSDM'11*, Hong Kong, China.

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.

Jason R. Smith, Chris Quirk, and Kristina Toutanova. 2010. Extracting parallel sentences from comparable corpora using document level alignment. In *Proceedings of Human Language Technologies: The 11th Annual Conference of the North American Chapter of the Association for Computational Linguistics (NAACL-HLT'10)*, Los Angeles, 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.

Artem Sokolov, Laura Jehl, Felix Hieber, and Stefan Riezler. 2013. Boosting cross-language retrieval by learning bilingual phrase associations from relevance rankings. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*.

Ferhan Ture, Jimmy Lin, and Douglas W. Oard. 2012. Combining statistical translation techniques for cross-language information retrieval. In *Proceedings of the International Conference on Computational Linguistics (COLING'12)*, Bombay, India.

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.