Combining Orthogonal Information in Large-Scale Cross-Language Information Retrieval

Shigehiko Schamoni
Department of Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
schamoni@cl.uni-heidelberg.de

Stefan Riezler
Department of Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
riezler@cl.uni-heidelberg.de

ABSTRACT

System combination is an effective strategy to boost retrieval performance, especially in complex applications such as cross-language information retrieval (CLIR) where the aspects of translation and retrieval have to be optimized jointly. We focus on machine learning-based approaches to CLIR that need large sets of relevance-ranked data to train high-dimensional models. We compare these models under various measures of orthogonality, and present an experimental evaluation on two different domains (patents, Wikipedia) and two different language pairs (Japanese-English, German-English). We show that gains of over 10 points in MAP/NDCG can be achieved over the best single model by a linear combination of the models that contribute the most orthogonal information, rather than by combining the models with the best standalone retrieval performance.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval; I.2.7 [**Artificial Intelligence**]: Natural Language Processing

General Terms

Algorithms, Experimentation

Keywords

Machine translation, cross-lingual retrieval, patent search

1. INTRODUCTION

Cross-Language Information Retrieval (CLIR) needs to jointly optimize the tasks of translation and retrieval, however, it is standardly approached with a focus on one aspect. For example, the industry standard leverages state-of-the-art statistical machine translation (SMT) to translate the query into the target language, in which standard retrieval is performed [4]. Most research approaches start from a retrieval perspective [13], or, more recently, from a machine learning direction [11]. Besides two different tasks, CLIR also needs to incorporate different languages and specialized domains. Thus, techniques that combine specialized systems into an improved joint system are a promising research direction. In this paper we show that a linear system combination can yield improvements of more than 10 MAP/NDCG points over the best single system, if the combined systems represent orthogonal information. We focus on machine learning-based approaches to CLIR that need large sets of relevance-ranked data to train high dimensional models. The systems investigated in this paper are systems based on direct use of SMT technology, systems that apply learning-to-rank techniques, systems based on probabilistic neural networks, and methods that incorporate domain-specific meta-information into linear learners. We present various measures of correlation/orthogonality on the level of scores (Pearson's correlation coefficient and principal component analysis), ranks (Kendall's rank correlation coefficient), and retrieved documents (Jaccard coefficient), and show on two different domains (patents, Wikipedia) and two different language pairs (Japanese-English, German-English) that the contribution of a single system to the combination is best determined by the orthogonality of the information it represents, rather than by its standalone retrieval performance.

SIGIR '15, August 09-13 2015, Santiago, Chile

DOI: http://dx.doi.org/10.1145/2600428.2609539

2. RELATED WORK

Various publications have investigated different methods of system combination for CLIR, including logical operations on retrieved sets [3], voting procedures based on retrieval scores [1], or machine learning techniques that learn combination weights directly from relevance rankings [14]. The focus of this paper is on machine learning-based CLIR approaches and on metrics to measure orthogonality between these systems. Since all of our models require large sets of relevance-ranked training data, e.g. for learning high-dimensional cross-lingual word matrices, we cannot use standard CLIR datasets from CLEF or TREC campaigns that consist of a few hundred queries with precomputed features. Instead, we use specialized domains such as patents or Wikipedia where relevance information can be induced from the citation or link structure.

3. CLIR MODELS

Translation Models. SMT-based models translate a query and then perform monolingual retrieval. Our first model is called *Direct Translation (DT)* and uses the SMT framework `cdec` [5] to generate a single best query translation.

A second model is called *Probabilistic Structured Queries*

(PSQ). The central idea of this approach is to project query terms into the target language by probabilistically weighted translations from the n-best list of a full SMT system [17].

In both models, we use the Okapi BM25 scoring scheme for document retrieval.

Ranking Models. Let $\mathbf{q} \in \{0,1\}^Q$ be a query and $\mathbf{d} \in \{0,1\}^D$ be a document where the n^{th} vector dimension indicates the occurrence of the n^{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 [2, 14] . 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 $\mathbf{q}$. We compare two methods 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 $\mathcal{R}$.

The first method uses the *Vowpal Wabbit (VW)* toolkit [6] to optimize the following ℓ_1-regularized hinge loss objective:

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

where $(x)_+ = \max(0, m - x)$ with margin m and λ is the regularization parameter. VW 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.

The second method is a *boosting model (BM)* that optimizes an exponential loss [16]:

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

where $\mathcal{D}(\mathbf{q}, \mathbf{d}^+, \mathbf{d}^-)$ is a non-negative importance function on tuples. The algorithm combines batch boosting with bagging over independently drawn bootstrap data samples of 100k instances each from $\mathcal{R}$. In every step, the single word pair feature is selected that provides the largest decrease of $\mathcal{L}_{exp}$. The final scoring function comprises the averaged resulting models. For regularization we rely on early stopping.

Neural Network Models. These models utilize the bilingual compositional vector model (biCVM) of [9] to train a retrieval system based on a bilingual autoencoder. The training task is to learn two functions $f : Q \to \mathbb{R}^d$ and $g : D \to \mathbb{R}^d$, which map a query $\mathbf{q}$ and a relevant document $\mathbf{d}$ from a corpus C onto a distributed semantic representation in $\mathbb{R}^d$. The energy of a query-document pair $(\mathbf{q}, \mathbf{d})$ is defined by $E_{bi}(\mathbf{q}, \mathbf{d}) = ||f(\mathbf{q}) - g(\mathbf{d})||^2$. Introducing a large margin m into the noise-contrastive update prevents the model from degenerating. This results in the following regularized hinge-loss objective:

$$\mathcal{H} = \sum_{(\mathbf{q},\mathbf{d}^+) \in C} \left(\sum_{i=1}^{k} \big(m + E_{bi}(\mathbf{q}, \mathbf{d}^+) - E_{bi}(\mathbf{q}, \mathbf{d}^-)\big)_+ \right) + \frac{\lambda}{2} ||\theta||^2,$$

where we treat less relevant documents $\mathbf{d}^-$ as noise samples during training. θ represents the model parameters.

While [9] train their system exclusively on parallel data on sentence and document level, we examine different training setups where we let the architecture learn distributed representations from: (a) data based on expert translations (family patents) and comparable data (Wikipedia articles on the same topic in different languages), which we call CVM_{FM}, and, (b) generally relevant documents (cited patents, linked Wikipedia articles), which we refer to as CVM_R.

Domain Knowledge Model. The final model (DK) for comparison uses highly informative dense features which capture similar aspects of e.g. patents or Wikipedia articles. Domain knowledge features for patents were inspired by [8]: a feature fires if two patents share similar aspects, e.g. a common inventor, similar number of claims, or common patent classes in the IPC hierarchy.

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 (hypernym and hyponym relations).

4. MEASURES OF ORTHOGONALITY

Jaccard similarity coefficient. This coefficient measures the percentage of overlap between two sets. In the retrieval setup, we limit our attention to the *relevant documents* within the top-k results for each query. The overlap metric expressing the similarity of two candidate systems is then:

$$J@k_{s_i \cap s_j} = \frac{|\mathit{retrieved@k}_{s_i} \cap \mathit{retrieved@k}_{s_j}|}{|\mathit{retrieved@k}_{s_i} \cup \mathit{retrieved@k}_{s_j}|},$$

where *retrieved@k* are the relevant documents retrieved within the top-k results. We report the pairwise overlap of two systems s_i and s_j for $k = 100$.

Pearson's ρ and Kendall's τ. The Pearson product-moment correlation coefficient, Pearson's ρ, is used to measure the linear correlation between the scores assigned to each retrieved relevant document. Kendall's τ works directly on the ranks and is insensitive to the absolute score values.

We calculate the metrics on a per-query basis and report the arithmetic mean. Again, we discard all irrelevant documents from the retrieved results by assigning them a score of 0. Then for each pair of systems, we select the queries which have at least 3 data points (i.e. relevant documents) in common, as 2 data points are always correlated. On average, this method selects about 75% of the queries for evaluation.

Principal Component Analysis (PCA). PCA is a method to find the set of n principal components (PC) that span the subspace of the data where most of the data variance resides. The straightforward approach would be to identify all the PCs (or eigenvectors) describing the retrieved data and to compare them. Our experiments showed that at least 850 PCs are required to capture more than 90% of the data variance, making a thorough comparison infeasible. Thus, we opt for a simplified approach where we consider only a small subset of the most important PCs.

We start by creating $|q| \times |d|$ matrices of retrieval scores for each system, where $|q|$ and $|d|$ are the numbers of queries and documents. PCA returns the first k principal components for each system. By calculating their dot products we obtain a sequence of values, or a k-dimensional vector, which describes the difference between the retrieval results of two candidate systems. To further reduce this vector to a single value, we report the normalized ℓ_2-norm of this vector of the top-k PC's similarity. This reflects our requirement

		#q	#d	#d$^+$/q
Patents (JP-EN)	train	107,061	888,127	13.28
	dev	2,000	100,000	13.24
	test	2,000	100,000	12.59
Wikipedia (DE-EN)	train	225,294	1,226,741	13.04
	dev	10,000	113,553	12.97
	test	10,000	115,131	13.22

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

that only the *dimension* of variance is of interest:

$$||PC||@k_{s_i,s_j} = \sqrt{\frac{1}{k}\sum_{n=1}^{k}\langle \mathbf{b}_i^n, \mathbf{b}_j^n\rangle^2}$$

The vector $\mathbf{b}_i^n$ represents the n^{th} normalized PC describing the space of relevant documents retrieved by system s_i, thus the range of values for $||PC||@k$ lies between 0 (all orthogonal) and 1 (all similar). In this sense, our PCA-based analysis is directly connected to the notion of orthogonality. We used $k = 10$ principal components in our experiments.

5. EXPERIMENTS

Patent Prior-art Search. Our first dataset consists of a Japanese-English (JP-EN) corpus of patent abstracts from the MAREC and NTCIR data.[1] It contains automatically induced relevance judgments for patent abstracts [7]: EN patents are regarded as relevant to a JP query patent with level (3) if they are in a family relationship (e.g., same invention), (2) if cited by the patent examiner, or (1) if cited by the applicant. On average, queries and documents contain about 5 sentences. Table 1 shows the size of the dataset, consisting of over 100k queries and nearly 1M documents, with approximately 13 relevant documents per query.

Wikipedia Article Retrieval. Our second dataset consists of relevance-linked Wikipedia pages.[2] Relevance judgments were extracted by aligning German (DE) queries with their English (EN) counterparts ("mates") via the graph of interlanguage links available in articles and Wikidata. The highest relevance level is assigned to the EN mate, the next relevance level to all other EN articles that link to the mate, *and* are linked to by the mate. Instead of using all outgoing links from the mate, only articles with bidirectional links are used. EN documents are restricted to the first 200 words 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, title words are removed from German queries. Data statistics are given in Table 1.

Parallel Data for SMT Models. *DT* and *PSQ* require an SMT system trained on parallel corpora. A JP-EN system was trained on 1.8M parallel sentences from the NTCIR-7 JP-EN PatentMT subtask. For Wikipedia, we trained a DE-EN system on 4.1M parallel sentences provided by WMT[3].

System Combination. We reapply the *VW* ranking approach described in Section 3 on dev set data for system combination. This method shows stable gains over three different IR-metrics: the precision-based MAP [11] and NDCG [10], where the latter considers relevance levels, and the recall-oriented PRES [12]. All scores were computed on the top 1,000 retrieved documents.

[1] `www.cl.uni-heidelberg.de/boostclir`
[2] `www.cl.uni-heidelberg.de/wikiclir`
[3] `www.statmt.org/wmt11/translation-task.html`

	models	MAP	NDCG	PRES
Patents (JP-EN)	DT	0.2554	0.5397	0.5680
	PSQ	0.2659	0.5508	0.5851
	VW	0.2205	0.4989	0.4911
	BM	0.1730	0.4335	0.5431
	CVM_{FM}	0.2504	0.5399	0.6104
	CVM_{R}	0.1767	0.4229	0.6121
	DK	0.2203	0.4874	0.5171
Wikipedia (DE-EN)	DT	0.3678	0.5691	0.7219
	PSQ	0.3642	0.5671	0.7165
	VW	0.1249	0.3389	0.6466
	BM	0.1386	0.3418	0.6145
	CVM_{FM}	0.1467	0.3326	0.5584
	CVM_{R}	0.1686	0.3515	0.6178
	DK	0.1824	0.3393	0.4937

Table 2: Test results for *standalone* CLIR models using direct translation (*DT*), probabilistic structured queries (*PSQ*), sparse ranking model (*VW*), sparse boosting model (*BM*), compositional vector model trained on parallel/comparable documents (CVM_{FM}) and on all relevant documents (CVM_R), and dense domain knowledge features (*DK*).

Results. Table 2 shows the performance of single retrieval systems according to MAP, NDCG, and PRES. SMT-based CLIR-methods clearly outperform all others. Only on specialized domains like patent-prior-art-search and by training on very clean data (expert translations), the neural network-based CVM_{FM} model is competitive. On the task of Wikipedia article retrieval, SMT-based methods outperform other approaches by a large margin.

Our hypothesis is that rather than combining the systems with the best standalone retrieval performance, the best overall system is gained by combining systems that are least similar and contribute orthogonal information to the combination. Table 3 lists all possible pairwise system combinations, together with their retrieval performance and their orthogonality/correlation.

An inspection of the patents in Table 3 shows that all measures of orthogonality/correlation capture the high similarity of the two SMT-based methods, *DT* and *PSQ*. Combining these two models results only in a small improvement in retrieval performance. Similar relations are found for all pairs of systems from same groups: ranking-based approaches such as *VW* and *BM* or neural network approaches such as CVM_{FM} and CVM_R are similar according to all measures of orthogonality/correlation, and lead to small improvements in retrieval performance in combination. Picking the least similar systems among the four groups, irrespective of their standalone retrieval performance, yields much higher improvements in combination. This is very pronounced for the *DK* system that is orthogonal to all other models. The biCVM-models also seem to contribute new information, where the gains are mostly higher for combinations with CVM_R despite its lower performance as a standalone model compared to CVM_{FM}. The last row in the Patents section presents the best performing combination of the four groups' systems, showing that the improvements by orthogonal combinations add up.

On Wikipedia data, shown in the lower part of Table 3, we find similar relations. The lower similarity between CVM_R and CVM_{FM} can be explained by training data dif-

combination	MAP	NDCG	PRES	J	ρ	τ	$\|\|PC\|\|$
Patents (JP-EN)							
PSQ + DT	^{L}.2707	.5578	.5941	.7318	.7488	.7591	.4717
PSQ + VW	.2912	.5862	.6286	.5077	.4475	.5387	.2590
PSQ + BM	^{L}.2661	.5611	.6257	.5358	.4964	.5413	.2541
PSQ + CVM$_{FM}$	.3071	.6105	.6808	.5528	.4336	.5154	.2359
PSQ + CVM$_{R}$	.3095	.6140	.7059	.4666	.3139	.3806	.2342
PSQ + DK	.3554	.6560	.7320	.3893	.2001	.3081	.1018
DT + VW	.2799	.5742	.6095	.5345	.4682	.5574	.3129
DT + BM	^{L}.2523	^{L}.5472	.6114	.5485	.5041	.5537	.2707
DT + CVM$_{FM}$	.3068	.6108	.6804	.5357	.4109	.5036	.1470
DT + CVM$_{R}$	.3084	.6139	.7071	.4560	.3082	.3730	.1651
DT + DK	.3515	.6530	.7295	.3870	.2026	.3084	.1147
VW + BM	.2389	.5324	.5985	.4802	.4139	.4899	.2163
VW + CVM$_{FM}$	.2923	.5970	.6623	.4729	.3880	.4832	.2380
VW + CVM$_{R}$	.2883	.5983	.6912	.3853	.2850	.3584	.2237
VW + DK	.3283	.6366	.7104	.3677	.1942	.2998	.0890
BM + CVM$_{FM}$	.2739	.5708	.6490	.4929	.4018	.4607	.3047
BM + CVM$_{R}$	.2402	.5222	.6630	.4290	.3197	.3694	.3567
BM + DK	.3083	.6167	.7092	.3461	.1627	.2500	.1454
DK + CVM$_{FM}$	.3388	.6443	.7493	.3667	.1931	.2919	.1379
DK + CVM$_{R}$	.3169	.6241	.7487	.3217	.1439	.2133	.1505
CVM$_{FM}$+CVM$_{R}$	^{L}.2529	^{L}.5407	.6608	.5787	.5149	.5098	.5358
PSQ+VW +CVM$_{R}$+DK	.3834	.6860	.7804	–	–	–	–
Wikipedia (DE-EN)							
PSQ + DT	.3724	.5758	.7258	.8445	.8535	.8110	.7202
PSQ + VW	^{L}.3623	.5935	.7857	.4092	.2630	.2452	.3054
PSQ + BM	.2908	.5106	^{L}.7207	.4956	.4224	.3949	.2850
PSQ + CVM$_{FM}$	.3718	.5840	.7467	.4017	.2866	.2923	.2061
PSQ + CVM$_{R}$	.3843	.6006	.7888	.3841	.1299	.1376	.1298
PSQ + DK	.3894	.6110	.7772	.3309	.0617	.1207	.0221
DT + VW	^{L}.3714	.5997	.7888	.4042	.2551	.2406	.2985
DT + BM	.2993	.5170	^{L}.7243	.4899	.4119	.3894	.2965
DT + CVM$_{FM}$	.3770	.5873	.7521	.3926	.2711	.2806	.1941
DT + CVM$_{R}$	.3870	.6021	.7911	.3809	.1258	.1345	.1621
DT + DK	.4009	.6186	.7814	.3275	.0557	.1168	.0168
VW + BM	^{R}.1337	.3559	.6792	.3805	.2831	.2446	.2981
VW + CVM$_{FM}$	.1652	.3922	.6952	.3547	.2492	.2079	.2154
VW + CVM$_{R}$	^{R}.1663	.3929	.7189	.3618	.1701	.1300	.2890
VW + DK	.2239	.4616	.7331	.2930	.0918	.1119	.0202
BM + CVM$_{FM}$	.1024	.3006	^{L}.6093	.3648	.2321	.2064	.2097
BM + CVM$_{R}$	.1315	^{L}.3372	.6546	.3546	.1541	.1371	.2390
BM + DK	.1893	.4031	.6669	.2804	.0314	.0772	.0211
DK + CVM$_{FM}$	^{L}.1856	.4023	.6780	.2803	.0267	.0694	.0141
DK + CVM$_{R}$	.2243	.4455	.7226	.2785	.0339	.0550	.0073
CVM$_{FM}$+CVM$_{R}$	.1880	.3905	.6652	.3652	.2224	.1986	.2379
DT+VW +CVM$_{R}$+DK	.4009	.6352	.8312	–	–	–	–

Table 3: Test results for *combined* CLIR models (see Table 2). Jaccard index J@100, Pearson's ρ, Kendall's τ, and the PCA-based $||PC||$@10 show correlation/orthogonality of a system pair. Preceding superscript letters indicate non-significant difference of the combined system to the ^{L}eft or ^{R}ight component at $p = .001$ using the paired randomization test described in [15].

ferences: the latter expects pairs of comparable documents, thus we employed the first 200 *unfiltered* article words as queries for training. As a result, both models are less similar and the combination shows notable gains compared to the patent task. The similarity measures between *VW* and *BM* on Wikipedia are blurred for an analogous reason: *BM* is trained on the full vocabulary, while *VW* uses correlated feature hashing to lower the memory footprint [2].

6. CONCLUSION

We presented an empirical validation of the conjecture that best results in CLIR system combination are achieved by combining systems that comprise orthogonal information. We measured correlation/orthogonality on various levels, and identified the groups of translation-based models agnostic of ranking, direct ranking optimizers unapt for translation, distributed semantic representations by neural networks, and linear learners based on meta-information. We showed experimentally that combining models from these orthogonal groups outperforms standalone models or combinations of best-performing models.

Acknowledgments. This research was supported in part by DFG grant RI-2221/1-2 "Weakly Supervised Learning of Cross-Lingual Systems".

7. REFERENCES

[1] J. A. Aslam and M. Montague. Models for metasearch. In *SIGIR*, 2001.

[2] B. Bai, J. Weston, D. Grangier, R. Collobert, K. Sadamasa, Y. Qi, O. Chapelle, and K. Weinberger. Learning to rank with (a lot of) word features. *Information Retrieval Journal*, 13(3), 2010.

[3] N. J. Belkin, P. Kantor, E. A. Fox, and J. A. Shaw. Combining the evidence of multiple query representations for information retrieval. *Inf. Process. Manage.*, 31(3):431–448, 1995.

[4] J. Chin, M. Heymans, A. Kojoukhov, J. Lin, and H. Tan. Cross-language information retrieval. Patent Application, 2008. US 2008/0288474 A1.

[5] C. Dyer, A. Lopez, J. Ganitkevitch, J. Weese, F. Ture, P. Blunsom, H. Setiawan, V. Eidelman, and P. Resnik. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *ACL*, 2010.

[6] S. Goel, J. Langford, and A. L. Strehl. Predictive indexing for fast search. In *NIPS*, 2009.

[7] E. Graf and L. Azzopardi. A methodology for building a patent test collection for prior art search. In *EVIA Workshop*, 2008.

[8] Y. Guo and C. Gomes. Ranking structured documents: A large margin based approach for patent prior art search. In *IJCAI*, 2009.

[9] K. M. Hermann and P. Blunsom. Multilingual models for compositional distributed semantics. In *ACL*, 2014.

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

[11] H. Li. *Learning to Rank for Information Retrieval and Natural Language Processing*. Morgan & Claypool, 2014.

[12] W. Magdy and G. J. Jones. PRES: a score metric for evaluating recall-oriented information retrieval applications. In *SIGIR*, 2010.

[13] J.-Y. Nie. *Cross-Language Information Retrieval*. Morgan & Claypool, 2010.

[14] S. Schamoni, F. Hieber, A. Sokolov, and S. Riezler. Learning translational and knowledge-based similarities from relevance rankings for cross-language retrieval. In *ACL*, 2014.

[15] M. D. Smucker, J. Allan, and B. Carterette. A comparison of statistical significance tests for information retrieval evaluation. In *CIKM*, 2007.

[16] A. Sokolov, L. Jehl, F. Hieber, and S. Riezler. Boosting cross-language retrieval by learning bilingual phrase associations from relevance rankings. In *EMNLP*, 2013.

[17] F. Ture, J. Lin, and D. W. Oard. Looking inside the box: Context-sensitive translation for cross-language information retrieval. In *SIGIR*, 2012.