Learning to Translate Queries for CLIR

Artem Sokolov*
Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
sokolov@cl.uni-heidelberg.de

Felix Hieber*
Computational Linguistics
Heidelberg University
69120 Heidelberg, Germany
hieber@cl.uni-heidelberg.de

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

ABSTRACT

The statistical machine translation (SMT) component of cross-lingual information retrieval (CLIR) systems is often regarded as black box that is optimized for translation quality independent from the retrieval task. In recent work [10], SMT has been tuned for retrieval by training a reranker on k-best translations ordered according to their retrieval performance. In this paper we propose a decomposable proxy for retrieval quality that obviates the need for costly intermediate retrieval. Furthermore, we explore the full search space of the SMT decoder by directly optimizing decoder parameters under a retrieval-based objective. Experimental results for patent retrieval show our approach to be a promising alternative to the standard pipeline approach.

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-Lingual Information Retrieval (CLIR) addresses the problem of ranking documents whose language differs from the query language. One of the simplest yet well performing approaches to CLIR is based on query translation using an existing Statistical Machine Translation (SMT) system which is treated as a black box. Thus, a monolingual retrieval engine does not need to be altered after translating queries into the target language. This approach is justified in the absence of cross-lingual relevance annotations, but in the presence of large parallel text corpora for SMT training[1].

In this work we argue that one should not only "look inside" the black box of the SMT system [16], but directly optimize SMT for the CLIR task at hand. We address this problem by discriminative training techniques which are widely used in the SMT community, and use automatically constructed relevance judgments from linked data. We show that a decomposable proxy for retrieval quality in training alleviates the problem of a costly intermediate retrieval step in reranking frameworks [10], and allows us to make use of the full, and lexically more diverse, decoder search space to optimize query translations for the CLIR task.

Our approach combines information specific to translation and to retrieval in one model targeted to CLIR: Basic translation units (phrases [7] or hierarchical phrase rules [1]) are estimated on parallel training data, while parameter optimization for lexicalized features that can boost or demote word/phrase translations is done on relevance judgments of existing queries. We present experiments in the domain of patent prior-art search where parallel training data for machine translation and relevance judgments for retrieval are available in large amounts. The results from our experimental evaluation shows our approach to be a promising alternative to the standard pipeline approach.

2. RELATED WORK

Common techniques for modulating query expansion with lexical variations use either comparable corpus statistics [14] or the k-best lists of an SMT system [16]. Experimental results show the latter approach to be superior to state-of-the-art approaches based on direct translation. In [9], consistent preprocessing of MT and IR training data yielded some improvements for retrieval and translation speed.

[10] is the work closest to our approach. They present an approach to learn a reranking model on k-best translations that are ordered according to retrieval performance. The approach requires expensive retrieval for each derivation in the k-best list. They show improvements over a regular SMT baseline on a small set of parallel queries. However, besides the need for costly retrieval in training, the features of the reranking mode cannot be integrated into an SMT decoder, thus limiting the usefulness of their approach.

A tighter integration with a decoder requires the target quality to be decomposable over transductions of its search space. Such approximations were proposed and evaluated

*First two authors contributed equally to the work.

[1]For example, see Google's CLIR approach [3].

SIGIR '14, July 06-11 2014, Gold Coast, QLD, Australia

in [13], however, only for translation-specific measures. Similar to this work, we design a decomposable approximation for CLIR measures (MAP, NDCG) and present a learning algorithm for tuning SMT towards retrieval quality.

3. QUERY TRANSLATION FOR CLIR

3.1 Cross-Lingual Information Retrieval

In this paper, we will use the following notational conventions: For a translation q of a foreign query f, a (monolingual) real-valued scoring function $S_{ir}(q,d)$ assigns a *retrieval score* to each document d in a collection $\mathcal{C}$. Relevance judgments for $\mathcal{C}$ are expressed by a function $rel(f,d) \geq 0$ that assigns to each query f and document d a *relevance level* (which is zero for irrelevant documents, and increasing values indicate higher relevance). The ranking induced by $S_{ir}(q,d)$ can be evaluated using common rank-based metrics, such as Mean Average Precision (MAP) or Normalized Discounted Cumulative Gain (NDCG). For a term-based scoring function $S_{ir}(q,d)$, queries and documents are represented as bag-of-word vectors, and $S_{ir}(q,d)$ is decomposable over query terms t in q. In this work we use Okapi BM25[2]: $S_{ir}(q,d) \equiv bm25(q,d) = \sum_{t \in q} bm25(t,d)$.

3.2 Tuning SMT for CLIR

State-of-the-art SMT systems compute the target-language query q of a foreign query f by recombining, through concatenation and reordering, small bilingual translation units called phrases (contiguous substrings in phrase-based SMT) or synchronous grammar rules (in hierarchical phrase-based SMT). These units are the result of a complex process that starts with word-to-word alignments and culminates with assigning various numerical confidence scores (*feature functions* or *models*) to the extracted units [7].

The union of complete hypotheses over the large number of possible input sentence splits, applicable translation options, and reordering possibilities, is called the *search space*, and is commonly structured as directed acyclic graphs (*lattices*) or *hypergraphs*. Inference (*decoding*) in SMT relies on maximizing the hypothesis score over the search space, i.e., maximizing the likeliness of obtaining a word alignment a of the target q given source f. This is usually parameterized as a linear model, $S_{smt}(q,a,f) = \mathbf{w} \cdot \mathbf{h}_{q,a,f}$, where $\mathbf{h}_{q,a,f}$ is a numerical vector of features and $\mathbf{w}$ is a parameter vector:

$$q_f = \arg\max_{a,q \in \mathcal{E}_f} \mathbf{w} \cdot \mathbf{h}_{q,a,f}, \tag{1}$$

$\mathcal{E}_f$ is the set of *reachable* translations/alignments that the SMT system can produce for the input f. For notational convenience we will omit dependence of the max operator and features $\mathbf{h}$ on a.

An important computational property of the quantity under arg max is that its components can be decomposed (through summation) over the scores of the individual units that are used in the alignment of q and f. This property is required to obtain a compact representation of the decoder search space, which can then be explored efficiently with dynamic programming (e.g., quantities like (1) are computed on lattices using shortest path algorithms). The optimal value for $\mathbf{w}$ is found in a tuning process that tries to replicate human reference translations by maximizing n-gram-based precision measures such as BLEU [11] on a development set consisting of pairs of source and target sentences.

We use the structured SVM margin-rescaling framework [15] to learn a new $\mathbf{w}$ adapted to the CLIR task. The framework assumes a unit-decomposable penalty $\Delta(q,q') \geq 0$, defined on structured outputs (translations), suffered for producing q instead of q'; it is zero if $q = q'$ and gracefully increases as q deviates more and more from q'. When optimizing for translation quality, the following loss function is minimized:

$$\mathcal{L} = \sum_f \max_{q \in \mathcal{E}_f} (\Delta(q, q_f^*) + \mathbf{w} \cdot \mathbf{h}_q) - \mathbf{w} \cdot \mathbf{h}_{q_f^*},$$

where q_f^* is either a desired reference translation r_f, or its reachable substitute $q_f^* = \max_q(-\Delta(q, r_f))$ with Δ approximating an inverted SMT quality measure.

In CLIR, a single desired output does not exist, but a set $\mathcal{C}_f^+$ of relevant documents for each foreign query f. Therefore we define a new function $\Delta(q, \mathcal{C}_f^+) = \max_q \left(S_{rel}(q, \mathcal{C}_f^+) \right) - S_{rel}(q, \mathcal{C}_f^+)$, that is the difference in best achievable approximate retrieval quality and retrieval quality for translation q. We will define $S_{rel}(q, \mathcal{C}_f^+)$ in section 3.3. Let us define *fear*, *hope* and *oracle* derivations [2, 5] for a foreign query f:

$$\begin{aligned} q^{fear} &= \arg\max_{q \in \mathcal{E}_f} (\mathbf{w} \cdot \mathbf{h}_q + \Delta(q, \mathcal{C}_f^+)), \\ q^{hope} &= \arg\max_{q \in \mathcal{E}_f} (\mathbf{w} \cdot \mathbf{h}_q - \Delta(q, \mathcal{C}_f^+)), \\ q^{oracle} &= \arg\max_{q \in \mathcal{E}_f} (-\Delta(q, \mathcal{C}_f^+)), \end{aligned}$$

and the corresponding feature vectors, $\mathbf{h}_q^{fear} \equiv \mathbf{h}_{q^{fear}}$ etc. The oracle derivation is the best derivation possible, i.e. with the smallest penalty in $\mathcal{E}_f$. The fear is the derivation maximizing the model score minus a confidence margin equal to the penalty (remember that $\Delta = 0$ if $q = q^{hope}$). As the static oracle derivation can be too idiosyncratic for the linear model to produce, the hope includes the model score to find a reasonable compromise. Additionally, a hope depending on the (changing) model score increases exploration of the search space during training.

With the new penalty we consider two losses to minimize:

$$\mathcal{L}_{svm} = \sum_f (\mathbf{w} \cdot \mathbf{h}_q^{fear} + \Delta(q^{fear}, \mathcal{C}_f^+)) - \mathbf{w} \cdot \mathbf{h}_q^{oracle} \tag{2}$$

$$\mathcal{L}_{ramp} = \sum_f (\mathbf{w} \cdot \mathbf{h}_q^{fear} + \Delta(q^{fear}, \mathcal{C}_f^+)) - (\mathbf{w} \cdot \mathbf{h}_q^{hope} - \Delta(q^{hope}, \mathcal{C}_f^+)). \tag{3}$$

For a learning rate α, the respective (sub)gradient descent updates are:

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \alpha \left(\sum_f \mathbf{h}_q^{fear} - \mathbf{h}_q^{oracle} \right) \tag{4}$$

$$\mathbf{w}_{i+1} = \mathbf{w}_i - \alpha \left(\sum_f \mathbf{h}_q^{fear} - \mathbf{h}_q^{hope} \right) \tag{5}$$

The update (4) for the standard structured loss (2) [15] increases weights of the features present in the oracle derivation and decreases for the ones in the fear. The ramp loss objective in equation (3) [5] boosts weights of features found in the current hope derivation.

[2]BM25 parameters were set to $k_1 = 1.2$, $b = 0.75$.

penalty	Moses		cdec	
	MAP	NDCG	MAP	NDCG
junk penalty	0.1797	0.3702	0.1441	0.3236
word penalty	0.1756	0.3663	0.1486	0.3301

Table 1: Oracle performance on the small training set for phrase-based (Moses) and hierarchical phrase-based (cdec) SMT decoders.

3.3 Oracle query translations

In this work, we are interested in tuning an SMT system for retrieval performance. Even though some correlation between BLEU scores and MAP has been shown [4], we note that an n-gram based precision metric like BLEU focuses strongly on the problem of reordering translation units to accommodate for higher n-gram matches and is thus not a suitable optimization metric for retrieval models, that do not take word order into account. A suitable optimization metric should either directly optimize the rank of relevant documents (*learning-to-rank*), or, more related to the task of translation, optimize lexical choices in the translation to improve term matching and adjust weights for reordering and language models correspondingly.

Directly optimizing rank-based metrics is problematic because a full retrieval for each derivation generated by the SMT system is required. This usually restricts the search space for oracle translations to the k-best list of derivations [10]. To alleviate this problem, we abstract away from the ranking problem and approximate retrieval quality of a translation q with its *relevance score* $S_{rel}(q, \mathcal{C}_f^+)$ to the set of relevant documents $\mathcal{C}_f^+ = \{d \in \mathcal{C} | rel(f,d) > 0\}$. Let $\mathcal{C}_{f,k}^+ = \{d \in \mathcal{C} | rel(f,d) = k\}$ be the set of relevant documents in the k-th relevance level. Since BM25 is decomposable over query terms, we directly assign partial relevance scores to terms t in the translated query q:

$$S_{rel}(q, \mathcal{C}_f^+) = \sum_{t \in q} S_{rel}(t, \mathcal{C}_f^+) = \sum_{t \in q} \sum_k \omega_k \frac{\sum_{d \in \mathcal{C}_{f,k}^+} bm25(t,d)}{|\mathcal{C}_{f,k}^+|},$$

where the ω_k are *relevance weights* adjusting the importance of each relevance level k in $\mathcal{C}$. To ensure good quality of the oracle translations, we found optimal values for ω_k by grid search with a step size 0.1 and a constraint $\sum_k |\omega_k| = 1$.

So far we only reward terms that appear in $\mathcal{C}_f^+$. While the SMT system thrives to generate relevant terms it produces them in phrases, together with connecting words as dictated by the translation model. If such 'by-product' terms appear sufficiently often in irrelevant documents, this can inadvertently boost their ranks. To counterbalance this effect we experimented with two penalties, with weight $\omega_0 \leq 0$: (1) a junk-word penalty that fires on insertion of irrelevant terms, or (2) a word penalty that acts on each word in the derivation. A comparison of oracle configurations in terms of the maximal performance (over the tested range of ω_k and ω_0) found on the training set is given in Table 1. For training we used oracles found with junk penalty for Moses, and with word penalty for cdec.

4. EXPERIMENTS

We conducted experiments on the BoostCLIR[3] dataset, a corpus of Japanese (JP) & English (EN) patent abstracts [12], using two open-source MT decoders, phrase-based Moses[4] and hierarchical SCFG decoder cdec[5].

We took NTCIR-7 data (1.8M parallel sentences) from the years 1993-2000 for SMT training and the NTCIR-8 test collection (2k sentences) for parameter tuning. Additionally to a dozen of vanilla dense SMT features, both decoders included lexicalized sparse features based on word alignments, indicating source word deletions, target word insertions, and word-to-word mappings. Both baseline systems were tuned with their respective MIRA [2] implementations. On held-out parallel test data, Moses and cdec achieved 0.2640 and 0.2829 BLEU, respectively.

For the ranking data, EN patents are regarded as relevant to the query JP patent, if they are cited by either the applicant or the patent examiner [6]. We assigned relevance level (2) for examiner citations, level (1) for applicants' own citations, and level (0) otherwise. A patent abstract contains about 5 sentences on average. Before running monolingual BM25-based retrieval, sentence-split query translations were concatenated back into a single query. The data was split into two training subsets of 200 and 1,000 queries (resp., $\simeq$ 1k and $\simeq$ 5k sentences) and dev/test subsets (of 400 queries each), all sampled without replacement. Oracle tuning and the training to determine the best learning configuration (see below) were done on the smaller training set and evaluated on the development set. We ran our training for 20k iterations starting from the MIRA weights found during the SMT tuning step of respective decoders, with the learning rate $\alpha = 0.001$.

Figure 1 shows experimental MAP[6] retrieval results for our approach evaluated for phrase-based (Moses) and hierarchical phrase-based (cdec) translation models. Small but stable improvements are gained only for the phrase-based system.[7] We show results for both updates (4) and (5). We see that ramp loss updates generally perform better than SVM updates for Moses. This is due to the ability to trade off the capabilities of the model against the best possible approximate performance on the retrieval task in the ramp loss setting. The SVM update is forced to perform "bold updates" towards the oracle which can result in updates that overfit to particular oracles [8]. Furthermore, we find it to be beneficial to constrain updates by freezing the dense features after MIRA training on parallel data, and tune only parameters of sparse lexicalized features that promote or demote the insertion, deletion, and translation of particular words. Additionally, we test two decoding evaluation setups of search space *rescoring* and *redecoding*. The former reuses hypergraphs/lattices produced with the MIRA-tuned weights and applies new weights to find an alternative, CLIR-optimized, derivation. The latter runs the decoder directly with the new weights. Both constraints (freezing and rescoring) show that the farther the setup strays away from the original MIRA model, the more difficult becomes generalization to unseen data. This suggests that it is crucial to find the optimal combination of translation- and retrieval-specific information for both inference and learning.

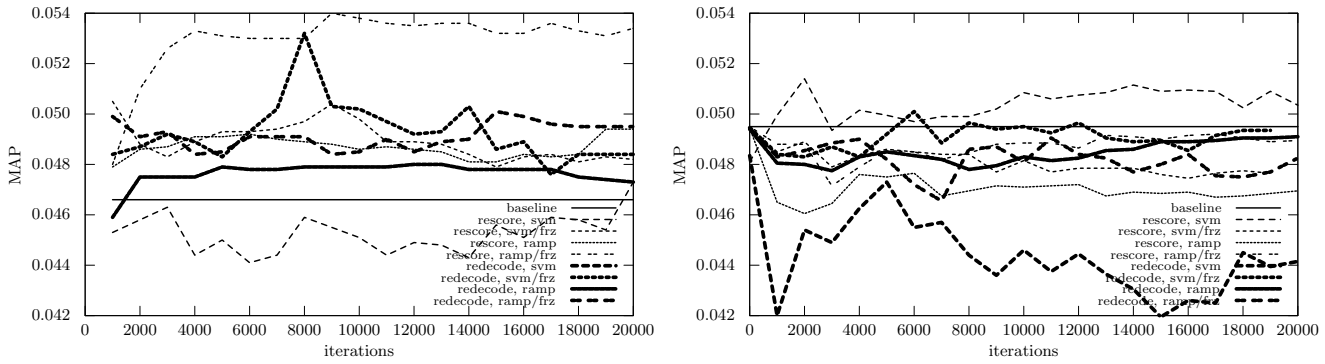
[3] `www.cl.uni-heidelberg.de/statnlpgroup/boostclir`

[4] `www.statmt.org/moses`

[5] `www.cdec-decoder.org`

[6] Evaluating with NDCG results in the same optimal configurations for both decoders.

[7] An implementation of the reranking approach [10] with our set of features scored about $\simeq$ 0.002 MAP above baseline.



(a) moses: dev

(b) cdec: dev

Figure 1: MAP on dev data for CLIR-tuning of phrase-based (Moses) and hierarchical (cdec) translation. For Moses the 'redecode, svm' curve is below the visible part of the plot.

config	Moses		cdec	
	MAP	NDCG	MAP	NDCG
baseline	0.0438	0.1498	0.0515	0.1600
rescore	$^{0.03}$**0.0498**	$^{0.02}$**0.1575**	$^{0.11}$0.0473	$^{0.08}$0.1548
redecode	$^{0.28}$0.0463	$^{0.26}$0.1532	$^{0.23}$0.0487	$^{0.27}$0.1571

Table 2: Test performance of the chosen learning configurations for Moses (rescore: ramp/frz@9k, redecode: svm/frz@8k) and cdec (svm/frz: rescore@2k, redecode@6k). Superscripts denote p-values obtained by a paired randomization test with respect to the baseline.

Table 2 shows test results for models trained on the bigger training set using the best settings found on the development set (see caption). For the hierarchical system improving over the significantly (at level $p = 0.01$) stronger baseline proves to be difficult. One reason could be a relatively harsh pruning strategy in cdec, governed by the language model, which produces lexically less diverse search spaces. This is supported by much worse oracles (Table 1) and fewer active sparse features in the learned models when compared to Moses (17k vs. 23k on the small training set).

5. CONCLUSION

We presented an approach for tuning an SMT system for cross-lingual retrieval. Our approach is efficient since it uses a decomposable proxy for retrieval quality that can be computed directly on the translation hypergraph or lattice in training. It is effective since optimal weights of retrieval-governing sparse features are accessible to the decoder, which combines this information with translation-specific dense features for optimal query translation in a cross-lingual retrieval setup.

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

6. REFERENCES

[1] D. Chiang. Hierarchical phrase-based translation. *Comp. Ling.*, 33(2), 2007.

[2] D. Chiang, Y. Marton, and P. Resnik. Online large-margin training of syntactic and structural translation features. In *EMNLP*, 2008.

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

[4] A. Fujii, M. Utiyama, M. Yamamoto, and T. Utsuro. Evaluating effects of machine translation accuracy on cross-lingual patent retrieval. In *SIGIR*, 2009.

[5] K. Gimpel and N. Smith. Structured ramp loss minimization for machine translation. In *NAACL*, 2012.

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

[7] P. Koehn. *Statistical Machine Translation*. Cambridge University Press, 1st edition, 2010.

[8] P. Liang, A. Bouchard-Côté, D. Klein, and B. Taskar. An end-to-end discriminative approach to machine translation. In *COLING-ACL*, 2006.

[9] W. Magdy and G. J. Jones. Studying machine translation technologies for large-data CLIR tasks: a patent prior-art search case study. *Inf. Retr.*, 2013.

[10] V. Nikoulina, B. Kovachev, N. Lagos, and C. Monz. Adaptation of statistical machine translation model for cross-lingual information retrieval in a service context. In *EACL*, 2012.

[11] K. Papineni, S. Roukos, T. Ward, and W. Zhu. BLEU: a method for automatic evaluation of machine translation. In *ACL*, 2002.

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

[13] A. Sokolov, G. Wisniewski, and F. Yvon. Lattice BLEU oracles in machine translation. *ACM Trans. Speech Lang. Process.*, 10(4), 2014.

[14] T. Talvensaari, J. Laurikkala, K. Järvelin, M. Juhola, and H. Keskustalo. Creating and exploiting a comparable corpus in cross-language information retrieval. *ACM Trans. Inf. Syst.*, 25(1), 2007.

[15] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *J. Mach. Learn. Res.*, 6, 2005.

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