Flat vs. Hierarchical Phrase-Based Translation Models for Cross-Language Information Retrieval

Ferhan Ture[1,2], Jimmy Lin[3,2,1]

[1]Dept. of Computer Science, [2]Institute for Advanced Computer Studies, [3]The iSchool
University of Maryland, College Park

fture@cs.umd.edu, jimmylin@umd.edu

ABSTRACT

Although context-independent word-based approaches remain popular for cross-language information retrieval, many recent studies have shown that integrating insights from modern statistical machine translation systems can lead to substantial improvements in effectiveness. In this paper, we compare *flat* and *hierarchical* phrase-based translation models for query translation. Both approaches yield significantly better results than either a token-based or a one-best translation baseline on standard test collections. The choice of model manifests interesting tradeoffs in terms of effectiveness, efficiency, and model compactness.

Categories and Subject Descriptors: H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval

General Terms: Algorithms, Experimentation

Keywords: SCFG, query translation

1. INTRODUCTION

Despite the prevalence of context-independent word-based approaches for cross-language information retrieval (CLIR) derived from the IBM translation models [4], recent studies have shown that exploiting ideas from machine translation (MT) for context-sensitive query translation produces higher-quality results [17, 19, 24]. State-of-the-art MT systems take advantage of sophisticated models with "deeper" representations of translation units, e.g., phrase-based [13], syntax-based [25, 27], and even semantics-based [11] models. In particular, *hierarchical* phrase-based machine translation (PBMT) systems [5] provide a middle ground between efficient "flat" phrase-based models and expressive but slow syntax-based models. In terms of translation quality, efficiency, and practicality, flat and hierarchical PBMT systems have become very popular, partly due to successful open-source implementations.

This paper explores flat and hierarchical PBMT systems for query translation in CLIR. Previously, we have shown that integrating techniques from hierarchical models lead to significant gains in effectiveness—however, it is unclear if such gains could have been achieved from "flat" representations. This question is interesting because it opens up a different region in the design space: flat representations are faster, more scalable, and exhibit less complexity—encoding a different tradeoff between efficiency and effectiveness.

There are two main contributions to this work: First, we test the robustness of query translation techniques introduced in earlier work [24] by comparing flat and hierarchical phrase-based translation models. In addition, we examine the effects of three different heuristics for handling one-to-many word alignments. We show that a combination-of-evidence approach consistently outperforms a strong token-based baseline as well as a one-best translation baseline for three different languages, Arabic (Ar), Chinese (Zh) and French (Fr), using either flat or hierarchical translation grammars. Second, we discuss differences between the two MT models and provide insights on the tradeoffs each represent. Experiments show that a hierarchical translation model yields higher effectiveness, which suggests that there is value in more sophisticated modeling of linguistic phenomena.

2. BACKGROUND AND RELATED WORK

Although word-by-word translation provides the starting point for query translation approaches to CLIR, there has been much work on using term co-occurrence statistics to select the most appropriate translations [10, 15, 1, 21]. Explicitly expressing term dependency relations has produced good results in monolingual retrieval [9, 18], but extending that idea to CLIR has not proven to be straightforward. Another thread of research has focused on translating multi-word expressions in order to deal with ambiguity [2, 28].

Borrowing ideas from MT for IR dates back to at least Ponte and Croft's work on retrieval using language modeling [20]. That work was later extended to translation models for retrieval [3], followed by a series of successful adaptations to the cross-language case [26, 14, 8].

As MT systems have evolved away from the token-based translation approach, researchers have started exploring ways to integrate various components of modern MT systems for better CLIR effectiveness. Magdy et al. [17] showed that preprocessing text consistently for MT and IR systems is beneficial. Nikoulina et al. [19] built MT models tailored to query translation by tuning model weights with queries and reranking the top n translations to maximize effectiveness on a held-out query set. While improvements were more substantial using the latter method, another interesting finding was the low correlation between translation and

SIGIR'13, July 28–August 1, 2013, Dublin, Ireland.

retrieval quality. This indicates that better translation may not necessarily help retrieval.

2.1 Context-Independent Baseline

As a baseline, we consider the technique presented by Darwish and Oard [6]. Given a source-language query s, we represent each token s_j by its translations in the target language, weighted by the bilingual translation probability. These token-to-token translation probabilities, called Pr_{token}, are learned independently from a parallel bilingual corpus using automatic word alignment techniques [4]. In this approach, the score of document d, given source-language query s, is computed by the following equations:

$$\text{Score}(d|s) = \sum_j \text{Weight}(\text{tf}(s_j, d), \text{df}(s_j)) \quad (1)$$

$$\text{tf}(s_j, d) = \sum_{t_i} \text{tf}(t_i, d) Pr_{\text{token}}(t_i|s_j) \quad (2)$$

$$\text{df}(s_j) = \sum_{t_i} \text{df}(t_i) Pr_{\text{token}}(t_i|s_j) \quad (3)$$

In order to reduce noise from incorrect alignments, we impose a lower bound on the token translation probability, and also a cumulative probability threshold, so that translation alternatives of s_j are added (in decreasing order of probability) until the cumulative probability has reached the threshold. Any weighting function can be used in conjunction with the tf and df values, and we chose the Okapi BM25 term weighting function (with parameters $k_1 = 1.2$, $b = 0.75$).

2.2 Flat vs. Hierarchical Phrase-based MT

Machine translation can be divided into three steps: training the translation model, tuning parameters, and decoding. We will mostly focus on the first step, since that is where flat and hierarchical MT approaches differ the most.

The output of the first step is the translation model (called TM hereafter). For both flat and hierarchical variants, the TM consists of a set of rules (i.e., the translation grammar) in the following format:

$$\alpha = \alpha_0\alpha_1\ldots \;||\; \beta = \beta_0\beta_1\ldots \;||\; \mathcal{A} \;||\; \ell(\alpha \to \beta)$$

We call the sequence of α_i's the source side of the rule, and sequence of β_j's the target side of the rule. The above indicates that the source side translates into the target side with a likelihood of $\ell(\alpha \to \beta)$.[1] $\mathcal{A}$ contains token alignments in the format i-j, indicating that source token α_i is aligned to target token β_j.

A *hierarchical* model [5] differs from a *flat* model [13] in terms of rule expressivity: rules are allowed to contain one or more nonterminals, each acting as a variable that can be expanded into other expressions using the grammar, carried out in a recursive fashion. These grammars are called synchronous context-free grammars (SCFG), as each rule describes a context-free expansion on both sides.

Consider the following two rules from an SCFG:

R_1. `[X] leave in europe || congé de [X] en europe || 1-0 2-3 3-4 || 1`
R_2. `maternal || maternité || 0-0 || 0.69`

[1]The likelihood function ℓ is not a probability density function because it is not normalized.

In R_1, the non-terminal variable `[X]` allows an arbitrarily long part of the sentence to be moved from the left of the sentence in English to the middle of the sentence in French, even though it generates a single token (i.e., *maternal*) using R_2 in this particular example. As a result, an SCFG can capture distant dependencies in language that may not be realized in flat models.

Each sequence of rules that covers the entire input is called a *derivation*, D, and produces a translation candidate, t, which is scored by a linear combination of features. One can use many features to score a candidate, but two features are the most important: the product of rule likelihood values indicates how well the candidate preserves the original meaning, $\text{TM}(t, D|s)$, whereas the language model score, $\text{LM}(t)$, indicates how well-formed the translation is. Combining the two, the *decoder* searches for the best translation:

$$t^{(1)} = \arg\max_t [\max_{D \in \mathcal{D}(s,t)} \text{TM}(t, D|s)\text{LM}(t)] \quad (4)$$

There is a tradeoff between using either flat or hierarchical grammars. The latter provides more expressivity in representing linguistic phenomena, but at the cost of slower decoding [16]. On the other hand, flat models are faster but less expressive. Also, due to the lack of variables, flat grammars contain more rules, resulting in a more verbose translation grammar.

3. QUERY TRANSLATION WITH MT

In our previous work [24], we described two ways to construct a context-sensitive term translation probability distribution using internal representations from an MT system. These distributions can then be used to retrieve ranked documents using equations (1)–(3).

3.1 Using the Translation Model

With appropriate data structures, it is possible to efficiently extract all rules in a TM (either flat or hierarchical) that apply to a given source query, s, called TM_s. For each such applicable rule r, we identify each source token s_j in r, ignoring any non-terminal symbols. From the token alignment information included in the rule structure, we can find all target tokens aligned to s_j. For each such target token t_i, the likelihood value of s_j being translated as t_i is increased by the likelihood score of r. At the end of the process, we have a list of possible translations and associated likelihood values for each source token that has appeared in any of the rules. We can then convert each list into a probability distribution, called Pr_{PBMT} for flat and Pr_{SCFG} for hierarchical grammars by normalizing the sum of likelihood scores:

$$Pr_{\text{SCFG/PBMT}}(t_i|s_j) = \frac{1}{\psi} \sum_{\substack{r \in \text{TM}_s \\ s_j \leftrightarrow t_i \text{ in } r}} \ell(r) \quad (5)$$

where $s_j \leftrightarrow t_i$ represents an alignment between tokens s_j and t_i and ψ is the normalization factor.

When a source token s_j is aligned to multiple target tokens in a rule, it is not obvious how to distribute the probability mass. In our previous implementation [24], each alignment was treated as an independent event with the same probability. We call this the *one-to-one* heuristic, and introduce two alternatives due to the following drawback: the target tokens aligned to s_j are usually not independent. For example, the token *brand* is aligned to three tokens *marque*, *de*,

fabrique (En. *brand, of, factory*), which is an appropriate translation when put together. Even if *de* is discarded as a stopword, the one-to-one heuristic will learn the token pair (*brand*, *fabrique*) incorrectly. An alternative heuristic is to ignore these rules altogether, assuming that good translation pairs will appear in other rules, thus discarding these cases would not cause any harm (we call this the *one-to-none* technique). A third approach is to combine the target tokens into a multi-token expression. Thus, in the above example, we would learn the translation of *brand* as *marque de fabrique*, which is a useful mapping that we might not learn otherwise. We call the third technique *one-to-many*, and compare these three heuristics in our evaluation.

3.2 Using N-best Translations

Given $t^{(1)}$, the most probable translation of query s computed by equation (4), we can score a document d as follows:

$$\text{Score}(d|s) = \sum_i \text{Weight}(\text{tf}(t_i^{(1)}, d), \text{df}(t_i^{(1)})) \quad (6)$$

Since MT systems generate a set of candidate translations in the process of computing equation (4), we can consider the n most likely candidates. For each candidate translation $t^{(k)}$, and for each source token s_j, we use token alignments to determine which tokens in $t^{(k)}$ are associated with s_j. If there are multiple target tokens, we apply one of the three methods introduced previously: *one-to-none*, *one-to-one*, or *one-to-many*. By the end of the process, we obtain a probability distribution of translations for each s_j based on the n best query translations. If source token s_j is aligned to (i.e., translated as) t_i in the k^{th} best translation, the value $\ell(t^{(k)}|s)$ is added to its probability mass, producing the following for Pr_{nbest} (where φ is the normalization factor):

$$Pr_{\text{nbest}}(t_i|s_j) = \frac{1}{\varphi} \sum_{\substack{k=1 \\ s_j \leftrightarrow t_i \text{ in } t^{(k)}}}^{n} \ell(t^{(k)}|s) \quad (7)$$

3.3 Evidence Combination

For Pr_{token}, translation probabilities are learned from all sentence pairs in a parallel corpus, whereas $Pr_{\text{SCFG/PBMT}}$ only uses portions that apply to the source query, which reduces ambiguity in the probability distribution based on this context. Pr_{nbest} uses the same set of rules in addition to a language model to search for most probable translations. This process filters out some irrelevant translations at the cost of less diversity, even among the top 10 or 100 translations. Since the three approaches have complementary strengths, we can perform a linear interpolation of the three probability distributions:

$$\begin{aligned} Pr_{\text{c}}(t_i|s_j; \lambda_1, \lambda_2) =& \lambda_1 Pr_{\text{nbest}}(t_i|s_j) + \lambda_2 Pr_{\text{SCFG/PBMT}}(t_i|s_j) \\ &+ (1 - \lambda_1 - \lambda_2) Pr_{\text{token}}(t_i|s_j) \end{aligned} \quad (8)$$

Replacing any of these probability distributions introduced above for Pr_{token} in equations (1)–(3) yields the respective scoring formula.

4. EVALUATION

We performed experiments on three CLIR test collections: TREC 2002 En-Ar CLIR, NTCIR-8 En-Zh Advanced Cross-Lingual Information Access (ACLIA), and CLEF 2006 En-Fr CLIR, with sizes 383,872, 388,589 and 177,452 documents, respectively. We used the title text of the 50 topics for the Arabic and French collections, and we treated the 73 well-formed questions in NTCIR-8 as queries.

For the flat and hierarchical translation models, we used `Moses` [12] and `cdec` [7], respectively. The training data consisted of Ar-En GALE 2010 evaluation (3.4m sentence pairs), Zh-En FBIS corpus (0.3m pairs), and Fr-En Europarl corpus v7 (2.2m pairs). A 3-gram language model was built for Arabic and Chinese using the target side of the parallel corpora. For French, we trained a 5-gram LM from the monolingual dataset provided for WMT-12. More details of the experimental setup can be found in [23].

Source code for replicating all the results presented in this paper is available in the open-source Ivory toolkit.[2]

4.1 Effectiveness

The baseline token-based model yields a Mean Average Precision (MAP) of 0.271 for Arabic, 0.150 for Chinese, and 0.262 for French. These numbers are competitive when compared to similar techniques applied to these collections. For each collection, we evaluated the three CLIR techniques (Pr_{token}, $Pr_{\text{SCFG/PBMT}}$, and Pr_{nbest}, with $n \in \{1, 10\}$), exploring the effect of the different alignment heuristics as well as flat vs. hierarchical phrase-based translation models. Parameters of the interpolated model were learned by a grid search. Experimental results are summarized in Table 1.[3]

Based on a randomized significance test [22], the interpolated model outperforms (with 95% confidence, marked *) the token-based model for all runs except for Arabic with `Moses`, consistently with the one-to-many heuristic and in some cases with the two other heuristics. Furthermore, in five out of the six conditions, the interpolated model with the one-to-many heuristic is significantly better than the one-best MT approach (marked †). This confirms that combining different query translation approaches is beneficial, and is also robust with respect to the test collection, language, and underlying MT model. The one-to-many term mapping heuristic seems to be the most effective overall.

However, the two MT models display significant differences in the "grammar" column, as the hierarchical model significantly outperforms the flat model. This supports the argument that the former is better at representing translation alternatives since it is more expressive. Also as a result of this difference, the flat grammar is much larger than the hierarchical one, which leads to an order of magnitude increase in processing time for Pr_{PBMT}.[4] These differences become especially important for the Arabic collection, where $Pr_{\text{SCFG/PBMT}}$ performs much better than $Pr_{\text{10-best}}$, using either MT system. An additional benefit of using $Pr_{\text{SCFG/PBMT}}$ is that we do not need to tune model parameters for translation, which is computationally intensive.

It is also interesting that the differences between the two MT models are insignificant for the 10-best approach, where the decoder finds similar translations in both cases. Therefore, it might be better to use flat representations for the 10-best approach for efficiency, since the end-to-end translation process is faster than hierarchical models.

[2] `http://ivory.cc/`

[3] For the 1-best model, one-to-one and one-to-many perform very similarly, so we present only the former for space considerations.

[4] On the other hand, decoding with a flat grammar is substantially faster than decoding with hierarchical MT due to constraints imposed by language modeling.

Language	MT	token	grammar			1-best		10-best			interpolated		
			many	one	none	one	none	many	one	none	many	one	none
Ar	cdec	0.271	0.293	0.282	0.302	0.249	0.249	0.255	0.249	0.248	0.293*†	0.282	0.302*
	Moses		0.274	0.266	0.273	0.249	0.232	0.264	0.254	0.249	0.280†	0.274	0.276
Zh	cdec	0.150	0.182	0.188	0.170	0.155	0.155	0.159	0.159	0.159	0.192*†	0.193*	0.182*
	Moses		0.156	0.167	0.151	0.155	0.146	0.169	0.163	0.163	0.183*†	0.177*	0.188*
Fr	cdec	0.262	0.297	0.288	0.292	0.276	0.235	0.307	0.304	0.295	0.318*†	0.314*	0.315*
	Moses		0.264	0.257	0.262	0.297	0.242	0.289	0.300	0.282	0.307*	0.301	0.300

Table 1: A summary of experimental results under different conditions, for all three CLIR tasks. Superscripts * and † indicate the result is significantly better than the token-based and one-best approaches, respectively.

5. CONCLUSIONS AND FUTURE WORK

In this paper, we extended an MT-based context-sensitive CLIR approach [24], comparing flat and hierarchical phrase-based translation models on three collections in three different languages. We make a number of interesting observations about the tradeoffs in incorporating machine translation techniques for query translation.

A combination-of-evidence approach was found to be robust and effective, but we have not examined how the interpolation model parameters can be learned using held-out data—this is the subject of ongoing work. Also, we are exploring ways of leveraging the translation of multi-token source-side expressions. Although we demonstrated the benefits of hierarchical grammars, we still do not explicitly take advantage of non-terminal information in the rules. It might be beneficial to perform a detailed error analysis to see what types of topics are improved with the use of SCFGs over flat grammars. Finally, we briefly discussed interesting tradeoffs between efficiency and effectiveness, but more detailed experiments are required to better understand different operating points and the overall design space.

6. ACKNOWLEDGMENTS

This research was supported in part by the BOLT program of the Defense Advanced Research Projects Agency, Contract No. HR0011-12-C-0015; NSF under awards IIS-0916043 and IIS-1144034. Any opinions, findings, conclusions, or recommendations expressed are those of the authors and do not necessarily reflect views of the sponsors. The second author is grateful to Esther and Kiri for their loving support and dedicates this work to Joshua and Jacob.

7. REFERENCES

[1] M. Adriani and C. Van Rijsbergen. Phrase identification in cross-language information retrieval. *RIAO*, 2000.

[2] L. Ballesteros and W. Croft. Phrasal translation and query expansion techniques for cross-language information retrieval. *SIGIR Forum*, 31:84–91, 1997.

[3] A. Berger and J. Lafferty. Information retrieval as statistical translation. *SIGIR*, 1999.

[4] P. Brown, V. Pietra, S. Pietra, and R. Mercer. The mathematics of statistical machine translation: parameter estimation. *CL*, 19(2):263–311, 1993.

[5] D. Chiang. Hierarchical phrase-based translation. *CL*, 33(2):201–228, 2007.

[6] K. Darwish and D. Oard. Probabilistic structured query methods. *SIGIR*, 2003.

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

[8] M. Federico and N. Bertoldi. Statistical cross-language information retrieval using n-best query translations. *SIGIR*, 2002.

[9] J. Gao, J.-Y. Nie, G. Wu, and G. Cao. Dependence language model for information retrieval. *SIGIR*, 2004.

[10] J. Gao, J.-Y. Nie, and M. Zhou. Statistical query translation models for cross-language information retrieval. *TALIP*, 5(4):323–359, 2006.

[11] B. Jones, J. Andreas, D. Bauer, K. Hermann, and K. Knight. Semantics-based machine translation with hyperedge replacement grammars. *COLING*, 2012.

[12] P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. Moses: open source toolkit for statistical machine translation. *ACL Demos*, 2007.

[13] P. Koehn, F. Och, and D. Marcu. Statistical phrase-based translation. *NAACL-HLT*, 2003.

[14] W. Kraaij, J. Nie, and M. Simard. Embedding web-based statistical translation models in cross-language information retrieval. *CL*, 29(3):381–419, 2003.

[15] Y. Liu, R. Jin, and J. Chai. A maximum coherence model for dictionary-based cross-language information retrieval. *SIGIR*, 2005.

[16] A. Lopez. Statistical machine translation. *ACM Computing Surveys*, 40(3):8:1–8:49, 2008.

[17] W. Magdy and G. Jones. Should MT systems be used as black boxes in CLIR? *ECIR*, 2011.

[18] D. Metzler and W. Croft. A Markov random field model for term dependencies. *SIGIR*, 2005.

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

[20] J. Ponte and W. Croft. A language modeling approach to information retrieval. *SIGIR*, 1998.

[21] H.-C. Seo, S.-B. Kim, H.-C. Rim, and S.-H. Myaeng. Improving query translation in English-Korean cross-language information retrieval. *IP&M*, 41(3):507–522, 2005.

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

[23] F. Ture. *Searching to Translate and Translating to Search: When Information Retrieval Meets Machine Translation.* PhD thesis, University of Maryland, College Park, 2013.

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

[25] D. Wu. A polynomial-time algorithm for statistical machine translation. *ACL*, 1996.

[26] J. Xu and R. Weischedel. Empirical studies on the impact of lexical resources on CLIR performance. *IP&M*, 41(3):475–487, 2005.

[27] K. Yamada and K. Knight. A syntax-based statistical translation model. *ACL*, 2001.

[28] W. Zhang, S. Liu, C. Yu, C. Sun, F. Liu, and W. Meng. Recognition and classification of noun phrases in queries for effective retrieval. *CIKM*, 2007.