Multi-Target Machine Translation with Multi-Synchronous Context-free Grammars

Graham Neubig, Philip Arthur, Kevin Duh
Graduate School of Information Science
Nara Institute of Science and Technology
8916-5 Takayama-cho, Ikoma-shi, Nara, Japan
{neubig,philip.arthur.om0,kevinduh}@is.naist.jp

Abstract

We propose a method for simultaneously translating from a single source language to multiple target languages T1, T2, etc. The motivation behind this method is that if we only have a weak language model for T1 and translations in T1 and T2 are associated, we can use the information from a strong language model over T2 to disambiguate the translations in T1, providing better translation results. As a specific framework to realize multi-target translation, we expand the formalism of synchronous context-free grammars to handle multiple targets, and describe methods for rule extraction, scoring, pruning, and search with these models. Experiments find that multi-target translation with a strong language model in a similar second target language can provide gains of up to 0.8-1.5 BLEU points.[1]

1 Introduction

In statistical machine translation (SMT), the great majority of work focuses on translation of a single language pair, from the source F to the target E. However, in many actual translation situations, identical documents are translated not from one language to another, but between a large number of different languages. Examples of this abound in commercial translation, and prominent open data sets used widely by the MT community include UN documents in 6 languages (Eisele and Chen, 2010), European Parliament Proceedings in 21 languages (Koehn, 2005), and video subtitles on TED in as many as 50 languages (Cettolo et al., 2012).

Figure 1: An example of multi-target translation, where a second target language is used to assess the quality of the first target language.

However, despite this abundance of multilingual data, there have been few attempts to take advantage of it. One exception is the *multi-source* SMT method of Och and Ney (2001), which assumes a situation where we have multiple source sentences, and would like to combine the translations from these sentences to create a better, single target translation.

In this paper, we propose a framework of *multi-target* SMT. In multi-target translation, we translate F to not a single target E, but to a set of sentences $\mathcal{E} = \langle E_1, E_2, \ldots, E_{|\mathcal{E}|} \rangle$ in multiple target languages (which we will abbreviate T1, T2, etc.). This, in a way, can be viewed as the automated version of the multi-lingual dissemination of content performed by human translators when creating data for the UN, EuroParl, or TED corpora mentioned above.

But what, one might ask, do we expect to gain by generating multiple target sentences at the same time? An illustrative example in Figure 1 shows three potential Chinese T1 translations for an Arabic input sentence. If an English speaker was asked to simply choose one of the Chinese translations, they

[1] Code and data to replicate the experiments can be found at http://phontron.com/project/naacl2015

likely could not decide which is correct. However, if they were additionally given English T2 translations corresponding to each of the Chinese translations, they could easily choose the third as the most natural, even without knowing a word of Chinese.

Translating this into MT terminology, this is equivalent to generating two corresponding target sentences E_1 and E_2, and using the naturalness of E_2 to help decide which E_1 to generate. Language models (LMs) are the traditional tool for assessing the naturalness of sentences, and it is widely known that larger and stronger LMs greatly help translation (Brants et al., 2007). It is easy to think of a situation where we can only create a weak LM for T1, but much more easily create a strong LM for T2. For example, T1 could be an under-resourced language, or a new entrant to the EU or UN.

As a concrete method to realize multi-target translation, we build upon Chiang (2007)'s framework of synchronous context free grammars (SCFGs), which we first overview in Section 2.[2] SCFGs are an extension of context-free grammars that define rules that synchronously generate source and target strings F and E. We expand this to a new formalism of multi-synchronous CFGs (MSCFGs, Section 3) that simultaneously generate not just two, but an arbitrary number of strings $\langle F, E_1, E_2, \ldots, E_N \rangle$. We describe how to acquire these from data (Section 4), and how to perform search, including calculation of LM probabilities over multiple target language strings (Section 5).

To evaluate the effectiveness of multi-target translation in the context of having a strong T2 LM to help with T1 translation, we perform experiments on translation of United Nations documents (Section 6). These experiments, and our subsequent analysis, show that the framework of multi-target translation can, indeed, provide significant gains in accuracy (of up to 1.5 BLEU points), particularly when the two target languages in question are similar.

2 Synchronous Context-Free Grammars

We first briefly cover SCFGs, which are widely used in MT, most notably in the framework of hierarchical phrase-based translation (Hiero; Chiang (2007)). SCFGs are based on synchronous rules defined as tuples of X, γ, and α

$$X \rightarrow \langle \gamma, \alpha \rangle, \tag{1}$$

where X is the head of the rule, and γ and α are strings of terminals and indexed non-terminals on the source and target side of the grammar. Each non-terminal on the right side is indexed, with non-terminals with identical indices corresponding to each-other. For example, a synchronous rule could take the form of[3]

$$X \rightarrow \langle X_0 \text{ of the } X_1,\ X_0 \text{ des } X_1 \rangle. \tag{2}$$

By simply generating from this grammar, it is possible to generate a string in two languages synchronously, as shown in Figure 2 (a). When we are already given a source side sentence and would like to using an SCFG to generate the translation, we find all rules that match the source side and perform search using the CKY+ algorithm (Chappelier et al., 1998). When we would additionally like to consider

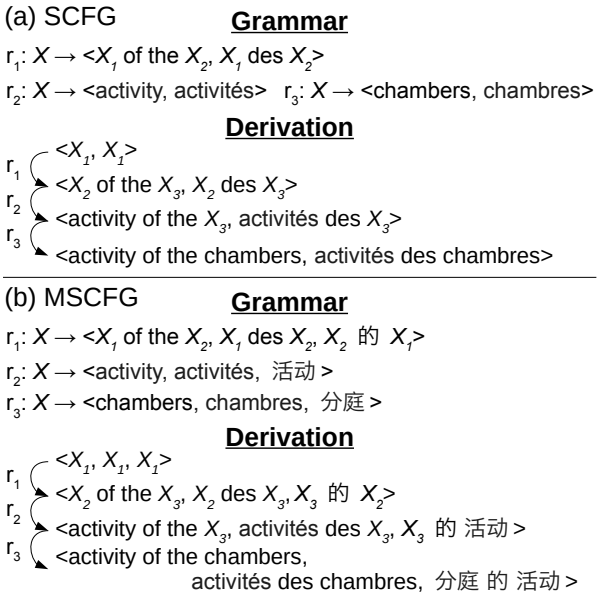

Figure 2: Synchronous grammars and derivations using (a) standard SCFGs and (b) the proposed MSCFGs.

[2] One could also consider a multi-target formulation of phrase-based translation (Koehn et al., 2003), but generating multiple targets while considering reordering in phrase-based search is not trivial. We leave this to future work.

[3] It is possible to use symbols other than X (e.g.: NP, VP) to restrict rule application to follow grammatical structure, but we focus on the case with a single non-terminal.

an LM, as is standard in SMT, we perform a modified version of CKY+ that approximately explores the search space using a method such as cube pruning (Chiang, 2007).

3 Multi-Synchronous CFGs

In this section, we present the basic formalism that will drive our attempts at multi-target translation. Specifically, we propose a generalization of SCFGs, which we will call multi-synchronous context free grammars (MSCFGs). In an MSCFG, the elementary structures are rewrite rules containing not a source and target, but an arbitrary number M of strings

$$X \rightarrow \langle \eta_1, ..., \eta_M \rangle, \tag{3}$$

where X is the head of the rule and η_m is a string of terminal and non-terminal symbols.[4] In this paper, for notational convenience, we will use a specialized version of Equation 3 in which we define a single γ as the source side string, and $\alpha_1, ...\alpha_N$ as an arbitrary number N of target side strings:

$$X \rightarrow \langle \gamma, \alpha_1, ..., \alpha_N \rangle. \tag{4}$$

Therefore, at each derivation step, one non-terminal in γ is chosen and all the nonterminals with same indices in $\alpha_1, ..., \alpha_N$ will be rewritten using a single rule. Figure 2 (b) gives an example of generating sentences in three languages using MSCFGs. Translation can also be performed by using the CKY+ algorithm to parse the source side, and then generate targets in not one, but multiple languages.

It can be noted that this formalism is a relatively simple expansion of standard SCFGs. However, the additional targets require non-trivial modifications to the standard training and search processes, which we discuss in the following sections.

4 Training Multi-Synchronous Grammars

This section describes how, given a set of parallel sentences in N languages, we can create translation models (TMs) using MSCFGs.

[4]We will also make the restriction that indices are linear and non-deleting, indicating that each non-terminal index present in any of the strings will appear exactly once in all of the strings. Thus, MSCFGs can also be thought of as a subset of the "generalized multi-text grammars" of Melamed et al. (2004).

4.1 SCFG Rule Extraction

First, we briefly outline rule extraction for SCFGs in the standard two-language case, as proposed by Chiang (2007). We first start by preparing two corpora in the source and target language, $\mathcal{F}$ and $\mathcal{E}$, and obtaining word alignments for each sentence automatically, using a technique such as the IBM models implemented by GIZA++ (Och and Ney, 2003).

We then extract initial phrases for each sentence. Given a source f_1^J, target e_1^I, and alignment $A = \{\langle i_1, i'_1 \rangle, \ldots, \langle i_{|A|}, i'_{|A|} \rangle\}$ where i and i' represent indices of aligned words in F and E respectively. First, based on this alignment, we extract all pairs of phrases $BP = \{\langle f_{i_1}^{j_1}, e_{i'_1}^{j'_1} \rangle, \ldots, \langle f_{i_{|BP|}}^{j_{|BP|}}, e_{i'_{|BP|}}^{j'_{|BP|}} \rangle\}$, where $f_{i_1}^{j_1}$ is a substring of f_1^J spanning from i_1 to j_1, and $e_{i'_1}^{j'_1}$ is analogous for the target side. The criterion for whether a phrase $\langle f_i^j, e_{i'}^{j'} \rangle$ can be extracted or not is whether there exists at least one alignment in A that falls within the bounds of both f_i^j and $e_{i'}^{j'}$, and no alignments that fall within the bounds of one, but not the other. It is also common to limit the maximum length of phrases to be less than a constant S (in our experiments, 10). The `phrase-extract` algorithm of Och (2002) can be used to extract phrases that meet these criteria.

Next, to create synchronous grammar rules, we cycle through the phrases in BP, and extract all potential rules encompassed by this phrase. This is done by finding all sets of 0 or more non-overlapping sub-phrases of initial phrase $\langle f_i^j, e_{i'}^{j'} \rangle$, and replacing them by non-terminals to form rules. In addition, it is common to limit the number of non-terminals to two and not allow consecutive non-terminals on the source side to ensure search efficiency, and limit the number of terminals to limit model size (in our experiments, we set this limit to five).

4.2 MSCFG Rule Extraction

In this section, we generalize the rule extraction process in the previous section to accommodate multiple targets. We do so by first independently creating alignments between the source corpus $\mathcal{F}$, and each of N target corpora $\{\mathcal{E}_1, \ldots, \mathcal{E}_N\}$.

Given a particular sentence we now have source F, N target strings $\{E_1, \ldots, E_N\}$, and N alignments $\{A_1, \ldots, A_N\}$. We next in-

dependently extract initial phrases for each of the N languages using the standard bilingual `phrase-extract` algorithm, yielding initial phrase sets $\{BP_1, \ldots, BP_N\}$. Finally, we convert these bilingual sets of phrases into a single set of multilingual phrases. This can be done by noting that all source phrases f_i^j will be associated with a set of 0 or more phrases in each target language. We define the set of multilingual phrases associated with f_i^j as the cross product of these sets. In other words, if f_i^j is associated with 2 phrases in T1, and 3 phrases in T2, then there will be a total of $2*3 = 6$ phrase triples extracted as associated with f_i^j.[5]

Once we have extracted multilingual phrases, the remaining creation of rules is essentially the same as the bilingual case, with sub-phrases being turned into non-terminals for the source and all targets.

4.3 Rule Scoring

After we have extracted rules, we assign them feature functions. In traditional SCFGs, given a source and target γ and α_1, it is standard to calculate the log forward and backward translation probabilities $P(\gamma|\alpha_1)$ and $P(\alpha_1|\gamma)$, log forward and backward lexical translation probabilities $P_{lex}(\gamma|\alpha_1)$ and $P_{lex}(\alpha_1|\gamma)$, a word penalty counting the non-terminals in α_1, and a constant phrase penalty of 1.

In our MSCFG formalism, we also add new features regarding the additional targets. Specifically in the case where we have one additional target α_2, we add the log translation probabilities $P(\gamma|\alpha_2)$ and $P(\alpha_2|\gamma)$, log lexical probabilities $P_{lex}(\gamma|\alpha_2)$ and $P_{lex}(\alpha_2|\gamma)$, and word penalty for α_2. In addition, we add log translation probabilities that consider both targets at the same time $P(\gamma|\alpha_1, \alpha_2)$ and $P(\alpha_1, \alpha_2|\gamma)$. As a result, compared to the 6-feature set in standard SCFGs, an MSCFG rule with two targets will have 13 features.

4.4 Rule Table Pruning

In MT, it is standard practice to limit the number of rules used for any particular source γ to ensure realistic search times and memory usage. This limit is generally imposed by ordering rules by the phrase probability $P(\alpha_1|\gamma)$ and only using the top few (in our case, 10) for each source γ. However, in the MSCFG case, this is not so simple. As the previous section mentioned, in the two-target MSCFG, we have a total of three probabilities conditioned on γ: $P(\alpha_1, \alpha_2|\gamma)$, $P(\alpha_1|\gamma)$, $P(\alpha_2|\gamma)$. As our main motivation for multi-target translation is to use T2 to help translation of T1, we can assume that the final of these three probabilities, which only concerns T2, is of less use. Thus, we propose two ways for pruning the rule table based on the former two.

The first method, which we will call *T1+T2*, is based on $P(\alpha_1, \alpha_2|\gamma)$. The use of this probability is straightforward, as it is possible to simply list the top rules based on this probability. However, this method also has a significant drawback. If we are mainly interested in accurate generation of the T1 sentence, there is a possibility that the addition of the T2 phrase α_2 will fragment the probabilities for α_1. This is particularly true when the source and T1 are similar, while T2 is a very different language. For example, in the case of a source of English, T1 of French, and T2 of Chinese, translations of English to French will have much less variation than translastions of English to Chinese, due to less freedom of translation and higher alignment accuracy between English and French. In this situation, the pruned model will have a variety of translations in T2, but almost no variety in T1, which is not conducive to translating T1 accurately.

As a potential solution to this problem, we also test a *T1* method, which is designed to maintain variety of T1 translations for each rule. In order to do so, we first list the top α_1 candidates based only on the $P(\alpha_1|\gamma)$ probability. Each α_1 will be associated with one or more α_2 rule, and thus we choose the α_2 resulting in the highest joint probability of the two targets $P(\alpha_1, \alpha_2|\gamma)$ as the representative rule for α_1. This pruning method has the potential advantage of increasing the variety in the T1 translations, but also has the potential disadvantage of artificially reducing genuine variety in T2. We examine which method is more effective in the experiments section.

5 Search with Multiple LMs

LMs computes the probability $P(E)$ of observing a particular target sentence, and are a fundamental

[5] Taking the cross-product here has the potential for combinatorial explosion as more languages are added, but in our current experiments with two target languages this did not cause significant problems, and we took no preventative measures.

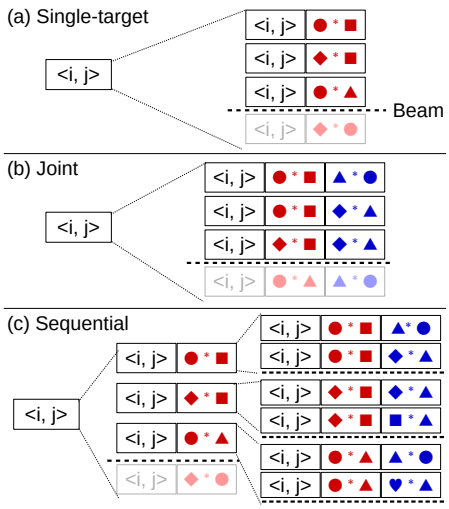

Figure 3: State splitting with (a) one LM, (b) two LMs with joint search, and (c) two LMs with sequential search, where T1 and T2 are the first (red) and second (blue) columns respectively.

part of both standard SMT systems and the proposed method. Unlike the other features assigned to rules in Section 4.3, LM probabilities are non-local features, and cannot be decomposed over rules. In case of n-gram LMs, this probability is defined as:

$$P_{LM}(E) = \prod_{i=1}^{|E|+1} p(e_i|e_{i-n+1}, \ldots, e_{i-2}, e_{i-1}) \quad (5)$$

where the probabilities of the next word e_i depend on the previous $n-1$ words.

When not considering an LM, it is possible to efficiently find the best translation for an input sentence f_1^J using the CKY+ algorithm, which performs dynamic programming remembering the most probable translation rule for each state corresponding to source span f_i^j. When using an LM, it is further necessary split each state corresponding to f_i^j to distinguish between not only the span, but also the strings of $n-1$ boundary words on the left and right side of the translation hypothesis, as illustrated in Figure 3 (a). As this expands the search space to an intractable size, this space is further reduced based on a limit on expanded edges (the pop limit), or total states per span (the stack limit), through a procedure such as cube pruning (Chiang, 2007).

In a multi-target translation situation with one LM for each target, managing the LM state becomes more involved, as we need to keep track of the $n-1$ boundary words for both targets. We propose two methods for handling this problem.

The first method, which we will dub the *joint* search method, is based on consecutively expanding the LM states of both T1 and T2. As shown in the illustration in Figure 3 (b), this means that each post-split search state will be annotated with boundary words from both targets. This is a natural and simple expansion of the standard search algorithm, simply using a more complicated representation of the LM state. On the other hand, because the new state space is the cross-product of all sets of boundary words in the two languages, the search space becomes significantly larger, with the side-effect of reducing the diversity of T1 translations for the same beam size. For example, in the figure, it can be seen that despite the fact that 3 hypotheses have been expanded, we only have 2 unique T1 LM states.

Our second method, which we will dub the *sequential* search method, first expands the state space of T1, then later expands the search space of T2. This procedure can be found in Figure 3 (c). It can be seen that by first expanding the T1 space we ensure diversity in the T1 search space, then additionally expand the states necessary for scoring with the T2 LM. On the other hand, if the T2 LM is important for creating high-quality translations, it is possible that the first pass of search will be less accurate and prune important hypotheses.

6 Experiments

6.1 Experimental Setup

We evaluate the proposed multi-target translation method through translation experiments on the MultiUN corpus (Eisele and Chen, 2010). We choose this corpus as it contains a large number of parallel documents in Arabic (ar), English (en), Spanish (es), French (fr), Russian (ru), and Chinese (zh), languages with varying degrees of similarity. We use English as our source sentence in all cases, as it is the most common actual source language for UN documents. To prepare the data, we first deduplicate the sentences in the corpus, then hold out 1,500 sentences each for tuning and test. In our basic training setup, we use 100k sentences for training both the TM and the T1 LM. This somewhat small

number is to simulate a T1 language that has relatively few resources. For the T2 language, we assume we have a large language model trained on all of the UN data, amounting to 3.5M sentences total.

As a decoder, we use the Travatar (Neubig, 2013) toolkit, and implement all necessary extensions to the decoder and rule extraction code to allow for multiple targets. Unless otherwise specified, we use joint search with a pop limit of 2,000, and T1 rule pruning with a limit of 10 rules per source rule. BLEU is used for both tuning and evaluating all models. In particular, we tune and evaluate all models based on T1 BLEU, simulating a situation similar to that in the introduction, where we want to use a large LM in T2 to help translation in T1. In order to control for optimizer instability, we follow Clark et al. (2011)'s recommendation of performing tuning 3 times, and reporting the average of the runs along with statistical significance obtained by pairwise bootstrap resampling (Koehn, 2004).

6.2 Main Experimental Results

In this section we first perform experiments to investigate the effectiveness of the overall framework of multi-target translation.

We assess four models, starting with standard single-target SCFGs and moving gradually towards our full MSCFG model:

SCFG: A standard SCFG grammar with only the source and T1.

SCFG+T2Al: SCFG constrained during rule extraction to only extract rules that also match the T2 alignments. This will help measure the effect, if any, of being limited by T2 alignments in rule extraction.

MSCFG-T2LM: The MSCFG, without using the T2 LM. Compared to SCFG+T2Al, this will examine the effect caused by adding T2 rules in scoring (Section 4.3) and pruning (Section 4.4) the rule table.

MSCFG: The full MSCFG model with the T2 LM.

The result of experiments using all five languages as T1, and the remaining four languages as T2 for all of these methods is shown in Table 1.

T1	T2	SCFG	SCFG +T2Al	MSCFG -T2LM	MSCFG
ar	es	24.97	25.11	24.79	†**25.19**
	fr		24.70	24.73	24.89
	ru		24.54	24.62	24.48
	zh		24.21	24.16	23.95
es	ar	42.15	41.73	41.21	41.22
	fr		42.20	41.84	‡**42.91**
	ru		41.62	41.90	41.98
	zh		41.80	41.61	41.65
fr	ar	37.21	37.26	37.03	37.41
	es		37.25	37.22	‡**38.67**
	ru		37.11	37.31	‡37.79
	zh		37.14	37.29	36.99
ru	ar	26.20	25.91	25.67	25.86
	es		26.17	26.01	†**26.45**
	fr		26.07	25.77	26.29
	zh		25.53	25.57	25.52
zh	ar	21.16	21.06	20.85	20.84
	es		21.39	21.31	21.33
	fr		†**21.60**	21.28	21.16
	ru		20.50	21.15	21.14

Table 1: Results for standard Hiero (SCFG), SCFG with T2 extraction constraints (SCFG+T2Al), a multi-SCFG minus the T2 LM (MSCFG-T2LM), and full multi-target translation (MSCFG). Bold indicates the highest BLEU score, and daggers indicate statistically significant gains over SCFG (†: $p < 0.05$, ‡: $p < 0.01$)

First, looking at the overall results, we can see that MSCFGs with one of the choices of T2 tends to outperform SCFG for all instances of T1. In particular, the gain for the full MSCFG model is large for the cases where the two target languages are French and Spanish, with en-fr/es achieving a gain of 1.46 BLEU points, and en-es/fr achieving a gain of 0.76 BLEU points over the baseline SCFG. This is followed by Arabic, Russian and Chinese, which all saw small gains of less than 0.3 when using Spanish as T2, with no significant difference for Chinese. This result indicates that multi-target MT has the potential to provide gains in T1 accuracy, particularly in cases where the languages involved are similar to each other.

It should be noted however, that positive results are sensitive to the languages chosen for T1 and T2,

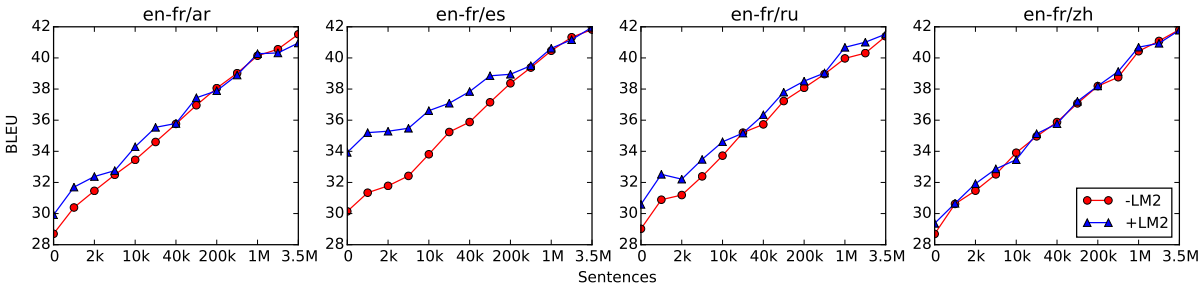

Figure 4: BLEU scores for different T1 LM sizes without (-LM2) or with (+LM2) an LM for the second target.

T2	SCFG	+T2Al
ar	223M	46.5M
es		134M
ru		70.8M
zh		26.0M

Table 2: Differences in rule table sizes for a T1 of French.

and in the cases involving Russian or Chinese, there is often even a drop in accuracy compared to the baseline SCFG. The reason for this can be seen by examining the results for SCFG+T2Al and MSCFG-T2LM. It can be seen that in the cases where there is an overall decrease in accuracy, this decrease can generally be attributed to a decrease when going from SCFG to SCFG+T2Al (indicating that rule extraction suffers from the additional constraints imposed by T2), or a decrease from SCFG+T2Al to MSCFG-LM2 (indicating that rule extraction suffers from fragmentation of the T1 translations by adding the T2 translation). On the other hand, we can see that in the majority of cases, going from MSCFG-LM2 to MSCFG results in at least a small gain in accuracy, indicating that a T2 LM is generally useful, after discounting any negative effects caused by a change in the rule table.

In Table 2 we show additional statistics illustrating the effect of adding a second language on the number of rules that can be extracted. From these results, we can see that all languages reduce the number of rules extracted, with the reduction being greater for languages with a larger difference from English and French, providing a convincing explanation for the drop in accuracy observed between these two settings.

6.3 Effect of T1 Language Model Strength

The motivation for multi-target translation stated in the introduction was that information about T2 may give us hints about the appropriate translation in T1. It is also a reasonable assumption that the less information we have about T1, the more valuable the information about T2 may be. To test this hypothesis, we next show results of experiments in which we vary the size of the training data for the T1 LM in intervals from 0 to 3.5M sentences. For T2, we either use no LM (MSCFG-T2LM) or an LM trained on 3.5M sentences (MSCFG). The results for when French is used as T1 are shown in Figure 4.

From these results, we can see that this hypothesis is correct. When no T1 LM is used, we generally see some gain in translation accuracy by introducing a strong T2 LM, with the exception of Chinese, which never provides a benefit. When using Spanish as T2, this benefit continues even with a relatively strong T1 LM, with the gap closing after we have 400k sentences of data. For Arabic and Russian, on the other hand, the gap closes rather quickly, with consistent gains only being found up to about 20-40k sentences. In general this indicates that the more informative the T2 LM is in general, the more T1 data will be required before the T2 LM is no longer able to provide additional gains.

6.4 Effect of Rule Pruning

Next we examine the effect of the rule pruning methods explained in Section 4.4. We set T1 to either French or Chinese, and use either the naive pruning criterion using T1+T2, or the criterion that picks the top translations in T1 along with their most probable T2 translation. Like previous experiments, we

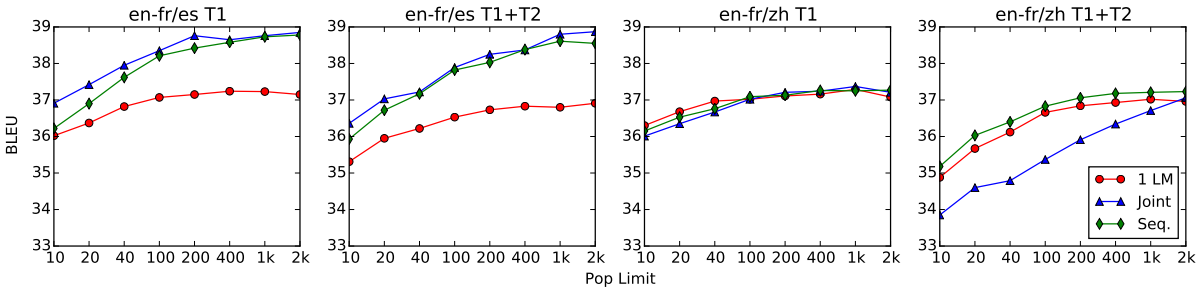

Figure 5: The impact of search on accuracy. Lines indicate a single LM (1 LM), two LMs with joint search (Joint) or two LMs with sequential search (Seq.) for various pop limits and pruning criteria.

	T1=fr		T1=zh	
T2	T1+T2	T1	T1+T2	T1
ar	36.21	**37.41**	20.35	**20.84**
es	**38.68**	38.67	20.73	**21.33**
fr	-		20.49	**21.16**
ru	37.14	**37.79**	19.87	**21.14**
zh	36.41	**36.99**	-	

Table 3: BLEU scores by pruning criterion. Columns indicate T1 (fr or zh) and the pruning criterion (T1+T2 joint probability, or T1 probability plus max T2). Rows indicate T2.

use the top 10 rules for any particular F.

Results are shown in Table 3. From these results we can see that in almost all cases, pruning using T1 achieves better results. This indicates the veracity of the observation in Section 4.4 that considering multiple T2 for a particular T1 causes fragmentation of TM probabilities, and that this has a significant effect on translation results. Interestingly, the one exception to this trend is T1 of French and T2 of Spanish, indicating that with sufficiently similar languages, the fragmentation due to the introduction of T2 translations may not be as much of a problem.

It should be noted that in this section, we are using the joint search algorithm, and the interaction between search and pruning will be examined more completely in the following section.

6.5 Effect of Search

Next we examine the effect of the search algorithms suggested in Section 5. To do so, we perform experiments where we vary the search algorithm (joint or sequential), the TM pruning criterion (T1 or T1+T2), and the pop limit. For sequential search, we set the pop limit of T2 to be 10, as this value did not have a large effect on results. For reference, we also show results when using no T2 LM.

From the BLEU results shown in Figure 5, we can see that the best search algorithm depends on the pruning criterion and language pair.[6] In general, when trimming using T1, we achieve better results using joint search, indicating that maintaining T1 variety in the TM is enough to maintain search accuracy. On the other hand, when using the T1+T2 pruned model when T2 is Chinese, sequential search is better. This shows that in cases where there are large amounts of ambiguity introduced by T2, sequential search effectively maintains necessary T1 variety before expanding the T2 search space. As there is no general conclusion, an interesting direction for future work is search algorithms that can combine the advantages of these two approaches.

7 Related Work

While there is very little previous work on multi-target translation, there is one line of work by González and Casacuberta (2006) and Pérez et al. (2007), which adapts a WFST-based model to output multiple targets. However, this purely monotonic method is unable to perform non-local reordering, and thus is not applicable most language pairs. It is also motivated by efficiency concerns, as opposed to this work's objective of learning from a T2 language.

Factored machine translation (Koehn and Hoang, 2007) is also an example where an LM over a second

[6] Results for model score, a more direct measure of search errors, were largely similar.

stream of factors (for example POS tags, classes, or lemmas) has been shown to increase accuracy. These factors are limited, however, by the strong constraint of being associated with a single word and not allowing reordering, and thus are not applicable to our setting of using multiple languages.

There has also been work on using multiple languages to improve the quality of extracted translation lexicons or topic models (Mausam et al., 2009; Baldwin et al., 2010; Mimno et al., 2009). These are not concerned with multi-target translation, but may provide us with useful hints about how to generate more effective multi-target translation models.

8 Conclusion

In this paper, we have proposed a method for multi-target translation using a generalization of SCFGs, and proposed methods to learn and perform search over the models. In experiments, we found that these models are effective in the case when a strong LM exists in a second target that is highly related to the first target of interest.

As the overall framework of multi-target translation is broad-reaching, there are a still many challenges left for future work, a few of which we outline here. First, the current framework relies on data that is entirely parallel in all languages of interest. Can we relax this constraint and use comparable data, or apply MSCFGs to pivot translation? Second, we are currently performing alignment independently for each target. Can we improve results by considering all languages available (Lardilleux and Lepage, 2009)? Finally, in this paper we considered the case where we are only interested in T1 accuracy, but optimizing translation accuracy for two or more targets, possibly through the use of multi-metric optimization techniques (Duh et al., 2012) is also an interesting future direction.

Acknowledgments

The authors thank Taro Watanabe and anonymous reviewers for helpful suggestions. This work was supported in part by JSPS KAKENHI Grant Number 25730136 and the Microsoft CORE project.

References

Timothy Baldwin, Jonathan Pool, and Susan Colowick. 2010. PanLex and LEXTRACT: Translating all words of all languages of the world. In *Proc. COLING*, pages 37–40.

Thorsten Brants, Ashok C. Popat, Peng Xu, Franz J. Och, and Jeffrey Dean. 2007. Large language models in machine translation. In *Proc. EMNLP*, pages 858–867.

Mauro Cettolo, Christian Girardi, and Marcello Federico. 2012. WIT3: web inventory of transcribed and translated talks. In *Proc. EAMT*, pages 261–268.

Jean-Cédric Chappelier, Martin Rajman, et al. 1998. A generalized CYK algorithm for parsing stochastic CFG. In *TAPD*, pages 133–137.

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

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proc. ACL*, pages 176–181.

Kevin Duh, Katsuhito Sudoh, Xianchao Wu, Hajime Tsukada, and Masaaki Nagata. 2012. Learning to translate with multiple objectives. In *Proc. ACL*.

Andreas Eisele and Yu Chen. 2010. MultiUN: A multilingual corpus from United Nation documents. In *Proc. LREC*.

M. Teresa González and Francisco Casacuberta. 2006. Multi-target machine translation using finite-state transducers. In *Proc. of TC-Star Speech to Speech Translation Workshop*, pages 105–110.

Philipp Koehn and Hieu Hoang. 2007. Factored translation models. In *Proc. EMNLP*.

Phillip Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT*, pages 48–54.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. EMNLP*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. MT Summit*, pages 79–86.

Adrien Lardilleux and Yves Lepage. 2009. Sampling-based multilingual alignment. In *Proc. RANLP*, pages 214–218.

Mausam, Stephen Soderland, Oren Etzioni, Daniel Weld, Michael Skinner, and Jeff Bilmes. 2009. Compiling a massive, multilingual dictionary via probabilistic inference. In *Proc. ACL*, pages 262–270.

I. Dan Melamed, Giorgio Satta, and Benjamin Wellington. 2004. Generalized multitext grammars. In *Proc. ACL*, pages 661–668.

David Mimno, Hanna M. Wallach, Jason Naradowsky, David A. Smith, and Andrew McCallum. 2009. Polylingual topic models. In *Proc. EMNLP*, pages 880–889.

Graham Neubig. 2013. Travatar: A forest-to-string machine translation engine based on tree transducers. In *Proc. ACL Demo Track*, pages 91–96.

Franz Josef Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proc. MT Summit*, pages 253–258.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Franz Josef Och. 2002. *Statistical machine translation: from single-word models to alignment templates*. Ph.D. thesis, RWTH Aachen.

Alicia Pérez, M. Teresa González, M. Inés Torres, and Francisco Casacuberta. 2007. Speech-input multitarget machine translation. In *Proc. WMT*, pages 56–63.