Word Lattices for Multi-Source Translation

Josh Schroeder, Trevor Cohn, and Philipp Koehn
School of Informatics
University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
Scotland, United Kingdom
{jschroe1, tcohn, pkoehn}@inf.ed.ac.uk

Abstract

Multi-source statistical machine translation is the process of generating a single translation from multiple inputs. Previous work has focused primarily on selecting from potential outputs of separate translation systems, and solely on multi-parallel corpora and test sets. We demonstrate how multi-source translation can be adapted for multiple monolingual inputs. We also examine different approaches to dealing with multiple sources, including consensus decoding, and we present a novel method of input combination to generate lattices for multi-source translation within a single translation model.

1 Introduction

Multi-source statistical machine translation was first formally defined by Och and Ney (2001) as the process of translating multiple meaning-equivalent source language texts into a single target language. Multi-source translation is of particular use when translating a document that has already been translated into several languages, either by humans or machines, and needs to be further translated into other target languages. This situation occurs often in large multi-lingual organisations such as the United Nations and the European Parliament, which must translate their proceedings into the languages of the member institutions. It is also common in multi-national companies, which need to translate product and marketing documentation for their different markets. Clearly, any existing translations for a document can help automatic translation into other languages. These different versions of the input can resolve deficiencies and ambiguities (e.g., syntactic and semantic ambiguity) present in a single input, resulting in higher quality translation output.

In this paper, we present three models of multi-source translation, with increasing degrees of sophistication, which we compare empirically on a number of different corpora. We generalize the definition of multi-source translation to include any translation case with multiple inputs and a single output, allowing for, e.g., multiple paraphrased inputs in a single language. Our methods include simple *output selection*, which treats the multi-source translation task as many independent translation steps followed by selection of one of their outputs (Och and Ney, 2001), and *output combination*, which uses consensus decoding to construct a string from n-gram fragments of the translation outputs (Bangalore et al., 2001). We also present a novel method, *input combination*, in which we compile the input texts into a compact lattice, over which we perform a single decoding pass. We show that as we add additional inputs, the simplest output selection method performs quite poorly relative to a single input translation system, while the latter two methods are able to make better use of the additional inputs.

The paper is structured as follows. §2 presents the three methods for multi-source translation in detail: output selection, output combination, and our novel lattice-based method for input combination. We report experiments applying these techniques to three different corpora, with both monolingual inputs (§3) and multilingual inputs (§4). We finish in §5 by analyzing the benefits and drawbacks of these approaches.

2 Approaches to Multi-Source Translation

We now present three ways to combine multiple inputs into a single output translation, in the context of related work for each technique.

Proceedings of the 12th Conference of the European Chapter of the ACL, pages 719–727,
Athens, Greece, 30 March – 3 April 2009. ©2009 Association for Computational Linguistics

2.1 Output Selection

The most straightforward approach to multi-source translation, proposed by Och and Ney (2001), is to independently translate each of the N source languages and then select a single translation from the outputs. Given N sources $\mathbf{s}_1^N = \mathbf{s}_1, \ldots, \mathbf{s}_N$, first translate each with a separate translation system, $p_1, \ldots, p_N$, to obtain N target translations, $\mathbf{t}_1^N = \mathbf{t}_1, \ldots, \mathbf{t}_N$. Och and Ney present two approaches for selecting a single target from these outputs.

The first, PROD, finds the maximiser of the product, $\arg\max_{\mathbf{t}\in\mathbf{t}_1^N} p(\mathbf{t}) \prod_{n=1}^{N} p_n(\mathbf{s}_n|\mathbf{t})$, where $p(\mathbf{t})$ is the language model probability. For reasons of tractability, the maximisation is performed only over targets generated by the translation systems, $\mathbf{t}_1^N$, not the full space of all translations. The PROD method requires each model to provide a model score for each $\mathbf{t}_n$ generated by the other models. However, this is often impossible due to the models' highly divergent output spaces (Schwartz, 2008), and therefore the technique cannot be easily applied.

The second approach, MAX, solves $\arg\max_{\mathbf{t}\in\mathbf{t}_1^N} \max_{n=1}^{N} p(\mathbf{t}) p_n(\mathbf{s}_n|\mathbf{t})$, which is much easier to calculate. As with PROD, the translation models' outputs are used for the candidate translations. While different models may have different score ranges, Och and Ney (2001) state that there is little benefit in weighting these scores to normalise the output range. In their experiments, they show that MAX used on pairs or triples of language inputs can outperform a model with single language input, but that performance degrades as more languages are added.

These methods limit the explored space to a full translation output of one of the inputs, and therefore cannot make good use of the full diversity of the translations. In this paper we present MAX scores as a baseline for output selection, and approximate an oracle using the BLEU metric as an upper bound for the output selection technique.

2.2 Output Combination

Consensus decoding as a form of system combination is typically used to integrate the outputs of multiple translation systems into a single synthetic output that seeks to combine the best fragments from each component system. Multi-source translation can be treated as a special case of consensus decoding. Indeed, several authors have seen improvements in translation quality by performing multi-source translation using generic system combination techniques (Matusov et al., 2006; Paulik et al., 2007).

the	ϵ	dog	barked	very	loudly
a	big	dog	barked	ϵ	loudly
sub	*insert*	–	*shift*	*delete*	–

Table 1: Example minimum TER edit script.

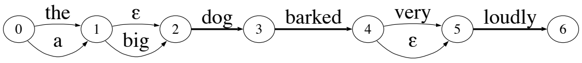

Figure 1: Conversion of TER script from Table 1 to a confusion network.

One class of approaches to consensus decoding focuses on construction of a confusion network or lattice[1] from translation outputs, from which new sentences can be created using different reorderings or combinations of translation fragments (e.g., Bangalore et al. (2001); Rosti et al. (2007b)). These methods differ in the types of lattices used, their means of creation, and scoring method used to extract the best consensus output from the lattice. The system used in this paper is a variant of the one proposed in Rosti et al. (2007a), which we now describe in detail.

The first step in forming a lattice is to align the inputs. Consensus decoding systems often use the script of edit operations that minimises the translation edit rate (TER; Snover et al. (2006)). TER is a word-based measure of edit distance which also allows n-gram shifts when calculating the best match between a hypothesis and reference. Because TER describes the correspondence between the hypothesis and reference as a sequence of insertions, substitutions, deletions, and shifts, the edit script it produces can be used to create a confusion network.

Consider a reference of "The dog barked very loudly" and a hypothesis "A big dog loudly barked." The TER alignment is shown in Table 1, along with the edit operations. Note that the matching "barked" tokens are labelled *shift*, as one needs to be shifted for this match to occur. Using the shifted hypothesis, we can form a confusion

[1] Different authors refer to "lattices," "confusion networks," "word sausages," etc. to describe these data structures, and specific terminology varies from author to author. We define a lattice as a weighted directed acyclic graph, and a confusion network as a special case where each node n in the ordered graph has word arcs only to node $n + 1$.

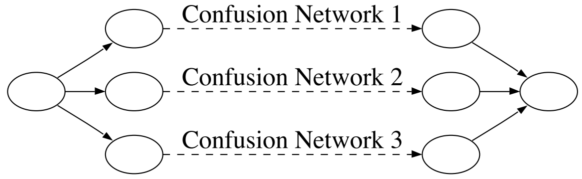

Figure 2: Structure of a lattice of confusion networks for consensus decoding.

network as in Figure 1. Additional sentences can be added by aligning them to the reference as well. Each link is weighted by the number of component sentences sharing that particular word at the given location.

Similar to Rosti et al. (2007a), we let each hypothesis take a turn as the "reference" for TER, using it as a skeleton for a confusion network. We then form a lattice of confusion networks (Figure 2), assigning a prior weight to each confusion network based on the average TER of the selected skeleton with the other hypotheses. This allows each system to set the word order for a component confusion network, but at the cost of a more complex lattice structure.

We can score paths $\mathcal{P}$ through these lattices with the assistance of a language model. Formally, the path score is given by:

$$\begin{aligned} w(\mathcal{P}) &= \nu \log p_{LM}(t(\mathcal{P})) \\ &+ \sum_{d \in \mathcal{P}} \Bigg[\sum_{n=1}^{N} \lambda_n \log p_n(d|\mathbf{s}_n) \\ &+ \mu\delta(d,\epsilon) + \xi(1-\delta(d,\epsilon)) \Bigg] \end{aligned}$$

where p_{LM} is the language model probability of the target string specified by the lattice path, $t(\mathcal{P})$, $p_n(d|s_n)$ is the proportion of system n's k-best outputs that use arc d in path $\mathcal{P}$, and the last two terms count the number of epsilon and non-epsilon transitions in the path. The model parameters are $\lambda_1, \ldots, \lambda_n, \nu, \mu, \xi$, which are trained using Powell's search to maximise the BLEU score for the highest scoring path, $\arg\max_{\mathcal{P}} w(\mathcal{P})$.

2.3 Input Combination

Loosely defined, *input combination* refers to finding a compact single representation of N translation inputs. The hope is that the new input preserves as many of the salient differences between the inputs as possible, while eliminating redundant information. Lattices are well suited to this task.

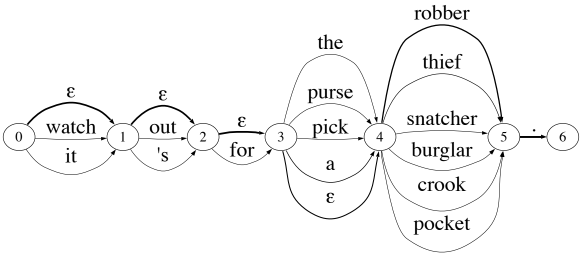

Figure 3: A monolingual confusion network. Thicker lines indicate higher probability word arcs.

When translating speech recognition output, previous work has shown that representing the ambiguity in the recognized text via confusion networks leads to better translations than simply translating the single best hypothesis of the speech recognition system (Bertoldi et al., 2007). The application of input lattices to other forms of input ambiguity has been limited to encoding input reorderings, word segmentation, or morphological segmentation, all showing improvements in translation quality (Costa-jussà et al., 2007; Xu et al., 2005; Dyer et al., 2008). However, these applications encode the ambiguity arising from a single input, while in this work we combine distinct inputs into a more compact and expressive single input format.

When given many monolingual inputs, we can apply TER and construct a confusion network as in Section 2.2.[2] In this application of confusion networks, arc weights are calculated by summing votes from each input for a given word, and normalizing all arcs leaving a node to sum to 1.

Figure 3 shows an example of a TER-derived input from IWSLT data. Because the decoder will handle reordering, we select the input with the lowest average TER against the other inputs to serve as the skeleton system, and do not create a lattice with multiple skeletons.

The problem becomes more complex when we consider cases of multi-lingual multi-source translation. We cannot easily apply TER across languages because there is no clear notion of an exact match between words. Matusov et al. (2006) propose using a statistical word alignment algorithm as a more robust way of aligning (monolingual) outputs into a confusion network for system com-

[2] Barzilay and Lee (2003) construct lattices over paraphrases using an iterative pairwise multiple sequence alignment (MSA) algorithm. Unlike our approach, MSA does not allow reordering of inputs.

bination. We take a similar approach for multilingual lattice generation.

Our process consists of four steps: (i) Align words for each of the $N(N-1)$ pairs of inputs; (ii) choose an input (or many inputs) to be the lattice skeleton; (iii) extract all *minimal consistent alignments* between the skeleton and the other inputs; and (iv) add links to the lattice for each aligned phrase pair.

A multi-parallel corpus such as Europarl (Koehn, 2005) is ideally suited for training this setup, as training data is available for each pair of input languages needed by the word aligner. We used the GIZA++ word alignment tool (Och and Ney, 2003) for aligning inputs, trained on a portion of the Europarl training data for each pair.

We select a skeleton input based on which single-language translation system performs the best when translating a development set. For our Europarl test condition, this was French.

We define a *minimal consistent alignment* (MCA) as a member of the set of multi-word alignment pairs that can be extracted from a many-to-many word alignment between skeleton sentence $\mathbf{x}$ and non-skeleton sentence $\mathbf{y}$ with the following restrictions: (i) no word in $\mathbf{x}$ or $\mathbf{y}$ is used more than once in the set of MCAs; (ii) words and phrases selected from $\mathbf{y}$ cannot be aligned to *null*; and (iii) no smaller MCA can be decomposed from a given pair. This definition is similar to that of *minimal translation units* as described in Quirk and Menezes (2006), although they allow *null* words on either side.

Different word alignment approaches will result in different sets of MCAs. For input lattices, we want sets of MCAs with as many aligned words as possible, while minimising the average number of words in each pair in the set. Experiments with GIZA++ on the Europarl data showed that the "grow-diag-final-and" word alignment symmetrization heuristic had the best balance between coverage and pair length: over 85% of skeleton words were part of a non-*null* minimal pair, and the average length of each pair was roughly 1.5 words. This indicates that our lattices will preserve most of the input space while collapsing easily alignable sub-segments.

Once a set of phrase alignments has been found, we construct a lattice over the skeleton sentence $\mathbf{x}$. For each additional input $\mathbf{y}_n$ we add a set of links and nodes for each word in $\mathbf{x}$ to any relevant words in $\mathbf{y}_n$, rejoining at the last word in $\mathbf{x}$ that is covered by the pair. Figures 4 and 5 show an example of the alignments and lattice generated by using a French skeleton with Spanish and Swedish sentences.

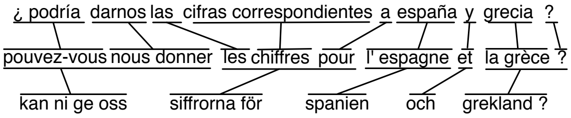
¿ podría darnos las cifras correspondientes a españa y grecia ?
pouvez-vous nous donner les chiffres pour l' espagne et la grèce ?
kan ni ge oss siffrorna för spanien och grekland ?

Figure 4: A multi-lingual alignment between French, Spanish and Swedish, showing the minimal consistent alignments. The lattice generated by this alignment is shown in Figure 5.

Once a lattice is created, we can submit it to a phrase-based decoder in place of text input. The decoder traverses lattice nodes in a manner similar to how words are traversed in text translation. Instead of one input word represented by each location in the coverage vector as in text input, with lattices there are a set of possible input word arcs, each with its own translation possibilities. The concept of compatible coverage vectors for the locations of translated words becomes the notion of reachability between frontier nodes in the lattice (Dyer et al., 2008).

It is possible to construct multi-skeleton lattices by connecting up a set of N lattices, each built around a different skeleton $\mathbf{x}_n$, in much the same manner as multiple confusion networks can be connected to form a lattice in output combination. With sufficient diversity in the input ordering of each skeleton, the decoder need not perform reordering. Because of the size and complexity of these multi-skeleton lattices, we attempt only monotonic decoding. In this scenario, as in consensus decoding, we hope to exploit the additional word order information provided by the alternative skeletons.

3 Experiments: Monolingual Input

We start our experimental evaluation by translating multiple monolingual inputs into a foreign language. This is a best-case scenario for testing and analytic purposes because we have a single translation model from one source language to one target language. While translating from multiple monolingual inputs is not a common use for machine translation, it could be useful in situations where we have a number of paraphrases of the input text, e.g., cross-language information retrieval and summarization.

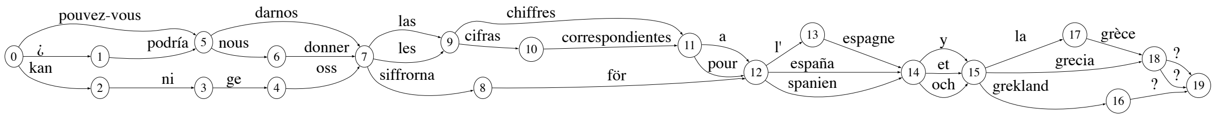

Figure 5: A multi-lingual lattice input for French, Spanish, and Swedish from Europarl dev2006.

Data sets for this condition are readily available in the form of test sets created for machine translation evaluation, which contains multiple target references for each source sentence. By flipping these test sets around, we create multiple monolingual inputs (the original references) and a single reference output (the original source text). We examine two datasets: the BTEC Italian-English corpus (Takezawa et al., 2002), and the Multiple Translation Chinese to English (MTC) corpora,[3] as used in past years' NIST MT evaluations.

All of our translation experiments use the Moses decoder (Koehn et al., 2007), and are evaluated using BLEU-4. Moses is a phrase-based decoder with features for lexicalized reordering, distance-based reordering, phrase and word translation probabilities, phrase and word counts, and an n-gram language model.

3.1 English to Italian

We use the portion of the BTEC data made available for the Italian-English translation task at IWSLT 2007, consisting of approximately 24,000 sentences. We also use the Europarl English-Italian parallel corpus to supplement our training data with approximately 1.2 million out-of-domain sentences. We train a 5-gram language model over both training corpora using SRILM (Stolcke, 2002) with Kneser-Ney smoothing and linear interpolation, the interpolation weight chosen to minimise perplexity on the Italian side of the development tuning set.

For multiple translation data, we use IWSLT test sets devset1-3 which have sixteen English translations for each Italian sentence. The Italian version of the BTEC corpus was created after the original Japanese-English version, and only the first English translation was used to generate the Italian data. The other fifteen versions of each English sentence were generated as paraphrases of the primary English translation. We explore translation conditions using only the fifteen paraphrased inputs ("Para." in Table 2), as well as using all sixteen English inputs ("All").

	All	Para.
BEST	40.06	24.02
ORACLE	51.64	47.27
MAX	29.32	23.94
SYSCOMB	**32.89**	**30.39**
CN INPUT	31.86	27.62

Table 2: BLEU scores on the BTEC test set for translating English inputs into Italian.

We tune our translation models on devset1, system combination on devset2 and report results on devset3 for each condition.

When tuning the single input "Para." and "All" baselines, we include all relevant copies of the 506 lines of devset1 English data, and repeat the Italian reference fifteen or sixteen times on the target side, resulting in a total of 7,590 and 8,096 sentence pairs respectively.

The results for devset3 are shown in Table 2. For comparison, we show the BEST score any input produced, as well as an approximated ORACLE output selection generated by choosing the best BLEU-scoring output for each sentence using a greedy search. Our output combination method, SYSCOMB, uses no system-specific weights to distinguish the inputs. For SYSCOMB and MAX, we translated all versions of the English input separately, and we use the top ten distinct hypotheses from each input sentence for n-best input to SYSCOMB.

For input combination, CN INPUT, we used the TER-based monolingual input lattice approach described in Section 2.3, choosing as a skeleton the input with the lowest average TER score when compared with the other inputs (assessed separately for each sentence). Each input was given equal probability in the confusion network links.

Note that the quality of output from translating the primary English input is much higher than from translating any of the paraphrases. The primary input sentence scores a BLEU of 40.06, while the highest scoring paraphrased input manages only a 24.02. When we look at "Para." the difference in the scores when using a single input

[3] LDC2002T01, LDC2003T17, LDC2004T07 and LDC2006T04.

(Best) versus all the inputs (SysComb and CN Input) is striking – clearly there is considerable information in the other inputs which can radically improve the translation output. Removing the primary input from Oracle reinforces this observation: the score drops by only 4.37 Bleu despite the nearly 16 Bleu drop for the single best input.

Interestingly, the output selection technique, Max, performs at a similar level to the combination techniques when we include the primary input, but degrades when given only the lower quality translations of paraphrased input under condition “Para.” In previous work on multi-lingual output selection, the Max score degraded after two or three outputs were combined, but even without the primary reference it maintains a score near the best single paraphrased input when combining fifteen outputs. One possible explanation for this is that the inputs are all being translated with the same translation model, so comparing their scores can give a more accurate ranking of their relative translation quality according to the model. The input combination method, CN Input, performs better than Max and only slightly worse than the output combination approach.

3.2 English to Chinese

We can add an extra dimension to monolingual multi-source translation by considering inputs of differing quality. A multi-source translation system can exploit features indicating the origin of the input to improve output quality. For these experiments, we use the MTC English-Chinese corpus, parts 1–4. This data was translated from Chinese into English by four teams of annotators, denoted E01–E04. This allows us to examine the results for translating the same team’s work over multiple years.

We train on the news domain portion of the official NIST data[4] (excluding the UN and Hong Kong data) for both the translation model and the 5-gram Chinese language model.

While we still have a single translation model, all of our inputs are now of a traceable origin and are known to have quality differences when judged by human evaluators. With this information we can tune one of two ways: We can create a set of all input systems and replicate the reference as we did for English to Italian translation (“All tuned”), or we can tune each input using only the version of the tuning data generated by the same translation team (“Self tuned”).[5] For example, we can tune a system with the MTC Part 2 data provided by translation team E01, and then decode E01’s translations of parts 3 and 4 with the weights obtained in tuning. The results for each system are shown in Table 3. Despite the different tuning conditions, there is no clear advantage to tuning to all inputs versus tuning to each input separately – on average we see a 0.06 Bleu score advantage by using “All” weights.

Team	Tuning	Part 3	Part 4
E01	All	**16.18**	15.52
E01	Self	16.02	**15.63**
E02	All	**14.29**	14.00
E02	Self	13.88	**14.05**
E03	All	14.99	**15.06**
E03	Self	**15.10**	14.94
E04	All	14.03	**12.65**
E04	Self	14.03	12.59

Table 3: Bleu scores using single inputs from each different team on the MTC. Bold indicates the better score between All and Self tuning.

Approach	Tuning	Part 3	Part 4
Max	All	**15.06**	**15.08**
Max	Self	14.97	13.75
SysComb	All	16.82	16.24
SysComb	Self	**16.87**	**16.45**

Table 4: Bleu scores for multi-source translations of MTC test sets. Better score for each output-based multi-source method is shown in bold.

With four different inputs to our multi-source translation system, and two ways of weighting the features for each input, how can we best utilize these systems in output selection and combination? We perform system combination and Max selection and obtain the scores shown in Table 4. The consensus decoding approach uses system-specific features as described in Section 2.2 to distinguish between E01-E04.

As with English to Italian, output combination performs the best of the multi-source techniques. Max performs better with translations generated by “All” weights than with “Self”, and the con-

[4] http://www.nist.gov/speech/tests/mt/2008

[5] Note that in the “Self tuned” setting we have only a quarter as much tuning data as for “All tuned”.

Input Language	test2006	test2007
French (FR)	29.72	30.21
Spanish (ES)	29.55	29.62
Swedish (SV)	29.33	29.44
Portuguese (PT)	28.75	28.79
Danish (DA)	27.20	27.48
Greek (EL)	26.93	26.78
Italian (IT)	26.82	26.51
German (DE)	24.04	24.41
Dutch (NL)	23.79	24.28
Finnish (FI)	18.96	18.85

Table 5: BLEU scores for individual translation systems into English trained on Europarl, from best to worst.

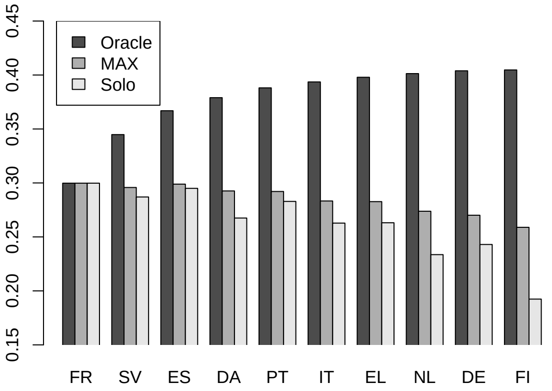

Figure 6: Performance for multilingual multi-source translation (test2005) as each language input is added, showing Oracle target selection, MAX score, or just a single language input (Solo).

verse is true for SYSCOMB. Given the robust performance of MAX when translation scores originated from the same translation model in English to Italian, it is not surprising that it favors the case where all the outputs are scored by the same model ("All tuned"). On the other hand, diversity amongst the system outputs has been shown to be important to the performance of system combination techniques (Macherey and Och, 2007). This may give an indication as to why the "Self tuned" data produced higher scores in consensus decoding – the outputs will be more highly divergent due to their different tuning conditions.

4 Experiments: Multilingual Input

Multilingual cases are the traditional realm of multi-source translation. We no longer have directly comparable translation models; instead each input language has a separate set of rules for translating to the output language. However, the availability of (and demand for) multi-parallel corpora makes this form of multi-source translation of great practical use.

4.1 Lattice Inputs

As described in Section 2.3, lattices can be used to provide a compact format for translating multilingual inputs to a multi-source translation system. We trim all non-skeleton node paths to a maximum length of four to reduce complexity when decoding. Such long paths are mostly a result of errors in the original word alignments, and therefore pruning these links is largely innocuous.

We train on the Europarl corpus and use the in-domain test sets provided for previous years' Workshops on Statistical Machine Translation. Because of the computational complexity of dealing with so many models, we train on only the first 100,000 sentences of each parallel corpus. Single system baseline scores for each language are shown in Table 5.

Besides comparing the different multi-source translation methods discussed above, in this task we also want to examine what happens when we use different numbers of input languages. To determine the best order to add languages, we performed a greedy search over oracle BLEU scores for test set test2005. We started with the best scoring single system, French to English, and in each iteration picked one additional system that would maximise BLEU if we always selected the translation system output closest to the reference. The results are shown in Figure 6.

The oracle selection order differs from the order of the best performing systems, which could be due to the high scoring systems having very similar output while lower scoring systems exhibit greater diversity. Interestingly, the order of the languages chosen iterates between the Roman and Germanic language families and includes Greek early on. This supports our claim that diversity is important. Note though that Finnish, which is also in a separate language family, is selected last, most likely due to difficulties in word alignment and translation stemming from its morphological complexity (Birch et al., 2008). This finding might also carry over to phrase-table triangulation (Cohn and Lapata, 2007), where multi-parallel data is used in training to augment a standard translation

Approach	test2006	test2007
French Only	29.72	30.21
French + Swedish		
MAX	29.86	30.13
LATTICE	29.33	29.97
MULTILATTICE	29.55	29.88
SYSCOMB	31.32	31.77
French + Swedish + Spanish		
MAX	30.18	30.33
LATTICE	29.98	30.45
MULTILATTICE	30.50	30.50
SYSCOMB	33.77	33.87
6 Languages		
MAX	28.37	28.33
LATTICE	30.22	30.91
MULTILATTICE	30.59	30.59
SYSCOMB	35.47	36.03

Table 6: BLEU scores for multi-source translation systems into English trained on Europarl. Single source French decoding is shown as a baseline.

system.

We choose to evaluate translation performance at three combination levels: two languages (French and Swedish), three languages (+Spanish), and six languages (+Danish, Portuguese, Italian). For each combination we apply MAX, SYSCOMB, French skeleton lattice input translation LATTICE, and monotone decoding over multiple skeleton lattices, MULTILATTICE. Results are shown in Table 6.

To enable the decoder used in LATTICE and MULTILATTICE to learn weights for different sources, we add a feature to the phrase table for each of the languages being translated. This feature takes as its value the number of words on the source side of the phrase. By weighting this feature up or down for each language, the decoder can prefer word links from specific languages.

As seen in previous work in multi-source translation, MAX output selection performs well with two or three languages but degrades as more languages are added to the input. Conversely, our lattice input method shows upward trends: LATTICE is comparable with MAX on three inputs and scores increase in the six language case.

Given the higher scores for output combination over input combination, what differences can we observe between the systems? Both systems have features that indicate the contributions of each input language to the final output. With input combination, we are forced by the decoder to take the maximum scoring path through the lattice, but in output combination we have the aggregate vote of word confidences generated by each system. If we could combine word arc scores across inputs, as in output combination, we might get a more robust solution for taking advantage of the available similarities on the target side of the translation. This points to a direction for future research.

Other differences between the systems may explain the score gap between our input and output combination approaches. Consensus decoding allows you to mix and match fragments that aren't necessarily stored as fragments in the phrase table. Another difference is the richer space of reorderings in TER-based lattices, due to the ability of the metric to handle long-distance alignments.

5 Conclusion

We analyzed three approaches for dealing with multi-source translation. While MAX is mostly a poor performer, the upper bound of output selection is stunning. The very positive results for output system combination across all data conditions are quite promising. Output combination achieves these results while the using the limited expressive power of n-best inputs. The potential of using a more expressive format – such as lattices that represent the joint search space of multiple models – is high. Our first attempts at adapting lattices to multi-source translation input show promise for future development. We have only scratched the surface of methods for constructing input lattices, and plan to actively continue research into improving these methods.

Acknowledgments

Thanks to Chris Callison-Burch for many insightful discussions, and to Chris Dyer for his implementation of lattice decoding in Moses.

This work was supported by the EuroMatrix project funded by the European Commission (6th Framework Programme), and has made use of the resources provided by the Edinburgh Compute and Data Facility (`http://www.ecdf.ed.ac.uk/`), which is partially supported by the eDIKT initiative (`http://www.edikt.org`). We also acknowledge the support of the EPSRC (grant GR/T04557/01).

References

Srinivas Bangalore, German Bordel, and Giuseppe Riccardi. 2001. Computing consensus translation from multiple machine translation systems. In *Proceedings of ASRU*, pages 351–354, Trento, Italy, December.

Regina Barzilay and Lillian Lee. 2003. Learning to paraphrase: An unsupervised approach using multiple-sequence alignment. In *Proceedings of NAACL: HLT*, pages 16–23, Edmonton, Canada, May.

Nicola Bertoldi, Richard Zens, and Marcello Federico. 2007. Speech translation by confusion network decoding. In *Proceedings of IEEE ICASSP*, pages 1297–1300, Honolulu, Hawaii, USA, April.

Alexandra Birch, Miles Osborne, and Philipp Koehn. 2008. Predicting success in machine translation. In *Proceedings of EMNLP*, pages 745–754, Honolulu, Hawaii, USA, October.

Trevor Cohn and Mirella Lapata. 2007. Machine translation by triangulation: Making effective use of multi-parallel corpora. In *Proceedings of ACL*, pages 728–735, Prague, Czech Republic, June.

Marta Ruiz Costa-jussà, Josep M. Crego, Patrik Lambert, Maxim Khalilov, José A. R. Fonollosa, José B. Mario, and Rafael E. Banchs. 2007. Ngram-based statistical machine translation enhanced with multiple weighted reordering hypotheses. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 167–170, Prague, Czech Republic, June.

Christopher J. Dyer, Smaranda Muresan, and Philip Resnik. 2008. Generalizing word lattice translation. In *Proceedings of ACL: HLT*, pages 1012–1020, Columbus, Ohio, USA, June.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Christopher J. Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of ACL: Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of MT Summit X*, pages 79–86, Phuket, Thailand, September.

Wolfgang Macherey and Franz J. Och. 2007. An empirical study on computing consensus translations from multiple machine translation systems. In *Proceedings of EMNLP-CoNLL*, pages 986–995, Prague, Czech Republic, June.

Evgeny Matusov, Nicola Ueffing, and Hermann Ney. 2006. Computing consensus translation for multiple machine translation systems using enhanced hypothesis alignment. In *Proceedings of EACL*, pages 33–40, Trento, Italy, April.

Franz Josef Och and Hermann Ney. 2001. Statistical multi-source translation. In *Proceedings of MT Summit VIII*, pages 253–258, Santiago de Compostela, Spain, September.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–52.

Matthias Paulik, Kay Rottmann, Jan Niehues, Almut Silja Hildebrand, and Stephan Vogel. 2007. The ISL phrase-based MT system for the 2007 ACL workshop on statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 197–202, Prague, Czech Republic, June.

Chris Quirk and Arul Menezes. 2006. Do we need phrases? Challenging the conventional wisdom in statistical machine translation. In *Proceedings of ACL: HLT, Main Conference*, pages 9–16, New York, New York, USA, June.

Antti-Veikko I. Rosti, Spyros Matsoukas, and Richard Schwartz. 2007a. Improved word-level system combination for machine translation. In *Proceedings of ACL*, pages 312–319, Prague, Czech Republic, June.

Antti-Veikko I. Rosti, Bing Xiang, Spyros Matsoukas, Richard Schwartz, Necip Fazil Ayan, and Bonnie J. Dorr. 2007b. Combining output from multiple machine translation systems. In *Proceedings of NAACL: HLT*, pages 228–235, Rochester, New York, USA, April.

Lane Schwartz. 2008. Multi-source translation methods. In *Proceedings of AMTA*, pages 279–288, Waikiki, Hawaii, USA, October.

Matthew Snover, Bonnie J. Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of AMTA*, pages 223–231, Boston, Massachusetts, USA, August.

Andreas Stolcke. 2002. SRILM - an extensible language modeling toolkit. In *Proceedings of ICSLP*, pages 901–904, Denver, Colorado, USA, October.

Toshiyuki Takezawa, Eiichiro Sumita, Fumiaki Sugaya, Hirofumi Yamamoto, and Seiichi Yamamoto. 2002. Toward a broad-coverage bilingual corpus for speech translation of travel conversations in the real world. In *Proceedings of LREC*, pages 147–152, Las Palmas, Canary Islands, Spain, May.

Jia Xu, Evgeny Matusov, Richard Zens, and Hermann Ney. 2005. Integrated Chinese word segmentation in statistical machine translation. In *Proceedings of IWSLT*, Pittsburgh, Pennsylvania, USA, October.