Clause Restructuring For SMT Not Absolutely Helpful

Susan Howlett and **Mark Dras**
Centre for Language Technology
Macquarie University
Sydney, Australia
susan.howlett@students.mq.edu.au, mark.dras@mq.edu.au

Abstract

There are a number of systems that use a syntax-based reordering step prior to phrase-based statistical MT. An early work proposing this idea showed improved translation performance, but subsequent work has had mixed results. Speculations as to cause have suggested the parser, the data, or other factors. We systematically investigate possible factors to give an initial answer to the question: Under what conditions does this use of syntax help PSMT?

1 Introduction

Phrase-based statistical machine translation (PSMT) translates documents from one human language to another by dividing text into contiguous sequences of words (*phrases*), translating each, and finally reordering them according to a *distortion model*.

The PSMT distortion model typically does not consider linguistic information, and as such encounters difficulty in language pairs that require specific long-distance reorderings, such as German–English.

Collins et al. (2005) address this problem by reordering German sentences to more closely parallel English word order, prior to translation by a PSMT system. They find that this *reordering-as-preprocessing* approach results in a significant improvement in translation performance over the baseline. However, there have been several other systems using the reordering-as-preprocessing approach, and they have met with mixed success.

We systematically explore possible explanations for these contradictory results, and conclude that, while reordering is helpful for some sentences, potential improvement can be eroded by many aspects of the PSMT system, independent of the reordering.

2 Prior Work

Reordering-as-preprocessing systems typically involve three steps. First, the input sentence is parsed. Second, the parse is used to permute the words according to some reordering rules, which may be automatically or manually determined. Finally, a phrase-based SMT system is trained and tested using the reordered sentences as input, in place of the original sentences. Many such systems exist, with results being mixed; we review several here.

Xia and McCord (2004) (English-to-French translation, using automatically-extracted reordering rules) train on the Canadian Hansard. On a Hansard test set, an improvement over the baseline was only seen if the translation system's phrase table was restricted to phrases of length at most four. On a news test set, the reordered system performed significantly better than the baseline regardless of the maximum length of phrases. However, this improvement was only apparent with monotonic decoding; when using a distortion model, the difference disappeared. Xia and McCord attribute the drop-off in performance on the Hansard set to similarity of training and test data.

Collins et al. (2005) (German-to-English) use six hand-crafted reordering rules targeting the placement of verbs, subjects, particles and negation. They train and evaluate their system on Europarl text and obtain a BLEU score (Papineni et al., 2002) of 26.8, with the baseline PSMT system achieving 25.2. A human evaluation confirms that reordered translations are generally (but not universally) better.

On Web text, Xu et al. (2009) report significant improvements applying one set of hand-crafted rules to translation from English to each of five SOV lan-

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics:shortpapers, pages 384–388,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

guages: Korean, Japanese, Hindi, Urdu and Turkish.

Training on news text, Wang et al. (2007) (Chinese-to-English, hand-crafted rules) report a significant improvement over the baseline system on the NIST 2006 test set, using a distance-based distortion model. Similar results are mentioned in passing for a lexicalised distortion model.

Also on news text, Habash (2007) (automatically-extracted rules, Arabic-to-English) reports a very large improvement when phrases are limited to length 1 and translation is monotonic. However, allowing phrases up to 7 words in length or using a distance-based distortion model causes the difference in performance to disappear. Habash attributes this to parser and alignment performance. He also includes oracle experiments, in which each system outperforms the other on 40–50% of sentences, suggesting that reordering is useful for many sentences.

Zwarts and Dras (2007) implement six rules for Dutch-to-English translation, analogous to those of Collins et al. (2005), as part of an exploration of dependency distance in syntax-augmented PSMT. Considering only their baseline and reordered systems, the improvement is from 20.7 to only 20.8; they attribute their poor result to the parser used.

Howlett and Dras (2010) reimplement the Collins et al. (2005) system for use in lattice-based translation. In addition to their main system, they give results for the baseline and reordered systems, training and testing on Europarl and news text. In strong contrast to the results of Collins et al. (2005), Howlett and Dras (2010) report 20.04 for the reordered system, below the baseline at 20.77. They explain their lower absolute scores as a consequence of the different test set, but do not explore the reversal in conclusion. Like Habash (2007), Howlett and Dras (2010) include oracle experiments which demonstrate that the reordering is useful for some sentences.

In this paper, we focus on the Collins et al. (2005) and Howlett and Dras (2010) systems (hereafter CKK and HD), as they are the most similar but have perhaps the most divergent results. Possible explanations for the difference are differences in the reordering process, from either parser performance or implementation of the rules, and differences in the translation process, including PSMT system setup and data used. We examine parser performance in §3 and the remaining possibilities in §4–5.

	Precision	Recall
Dubey and Keller (2003)	65.49	70.45
Petrov and Klein (2008)	69.23	70.41
Howlett and Dras (2010)	72.78	73.15
This paper, lowercased	71.09	73.16
This paper, 50% data	68.65	70.86
This paper, 50% data, lowerc.	67.59	70.23
This paper, 25% data	65.24	67.13
This paper, 10% data	61.56	63.01

Table 1: Precision and recall for the parsers mentioned in §3. The numbers are collated for reference only and are not directly comparable; see the text for details.

3 Parser Performance

We first compare the performance of the two parsers used. CKK uses the Dubey and Keller (2003) parser, which is trained on the Negra corpus (Skut et al., 1997). HD instead uses the Berkeley parser (Petrov et al., 2006), trained on Negra's successor, the larger Tiger corpus (Brants et al., 2002).

Refer to Table 1 for precision and recall for each model. Note that the CKK reordering requires not just category labels (e.g. NP) but also function labels (e.g. SB for subject); parser performance typically goes down when these are learnt, due to sparsity. All models in Table 1 include function labels.

Dubey and Keller (2003) train and test on the Negra corpus, with 18,602 sentences for training, 1,000 development and 1,000 test, removing sentences longer than 40 words.

Petrov and Klein (2008) train and test the Berkeley parser on part of the Tiger corpus, with 20,894 sentences for training and 2,611 sentences for each of development and test, all at most 40 words long.

The parsing model used by HD is trained on the full Tiger corpus, unrestricted for length, with 38,020 sentences for training and 2,000 sentences for development. The figures reported in Table 1 are the model's performance on this development set. With twice as much data, the increase in performance is unsurprising.

From these figures, we conclude that sheer parser grunt is unlikely to be responsible for the discrepancy between CKK and HD. It is possible that parser output differs qualitatively in some important way; parser figures alone do not reveal this.

Here, we reuse the HD parsing model, plus five

Data		Set name	Size
CKK	Train		751,088
	Test		2,000
WMT	Train	europarl-v4	1,418,115
	Tuning	test2007	2,000
		news-test2008	2,051
	Test	test2008	2,000
		newstest2009	2,525

Table 2: Corpora used, and # of sentence pairs in each.

additional models trained by the same method. The first is trained on the same data, lowercased; the next two use only 19,000 training sentences (for one model, lowercased); the fourth uses 9,500 sentences; the fifth only 3,800 sentences. The 50% data models are closer to the amount of data available to CKK, and the 25% and 10% models are to investigate the effects of further reduced parser quality.

4 Experiments

We conduct a number of experiments with the HD system to attempt to replicate the CKK and HD findings. All parts of the system are available online.[1]

Each experiment is paired: the reordered system reuses the recasing and language models of its corresponding baseline system, to eliminate one source of possible variation. Training the parser with less data affects only the reordered systems; for experiments using these models, the corresponding baselines (and thus the shared models) are not retrained.

For each system pair, we also run the HD oracle.

4.1 System Variations

CKK uses the PSMT system Pharaoh (Koehn et al., 2003), whereas HD uses its successor Moses (Koehn et al., 2007). In itself, this should not cause a dramatic difference in performance, as the two systems perform similarly (Hoang and Koehn, 2008).

However, there are a number of other differences between the two systems. Koehn et al. (2003) (and thus presumably CKK) use an unlexicalised distortion model, whereas HD uses a lexicalised model. CKK does not include a tuning (minimum error rate training) phase, unlike HD. Finally, HD uses a 5-gram language model. The CKK language model is unspecified; we assume a 3-gram model would be more likely for the time. We explore combinations of all these choices.

LM	DM	T	Base.	Reord.	Diff.	Oracle
3	dist	–	25.58	26.73	+1.15	28.11
				26.63	+1.05	28.03

Table 3: Replicating CKK. Top row: full parsing model; second row: 50% parsing model. Columns as for Table 4.

4.2 Data

A likely cause of the results difference between HD and CKK is the data used. CKK used Europarl for training and test, while HD used Europarl and news for training, with news for tuning and test.

Our first experiment attempts to replicate CKK as closely as possible, using the CKK training and test data. This data came already tokenized and lowercased; we thus skip tokenisation in preprocessing, use the lowercased parsing models, and skip tokenisation and casing steps in the PSMT system. We try both the full data and 50% data parsing models.

Our next experiments use untokenised and cased text from the Workshop on Statistical Machine Translation. To remain close to CKK, we use data from the 2009 Workshop,[2] which provided Europarl sets for both training and development. We use `europarl-v4` for training, `test2007` for tuning, and `test2008` for testing.

We also run the 3-gram systems of this set with each of the reduced parser models.

Our final experiments start to bridge the gap to HD. We still train on `europarl-v4` (diverging from HD), but substitute one or both of the tuning and test sets with those of HD: `news-test2008` and `newstest2009` from the 2010 Workshop.[3]

For the language model, HD uses both Europarl and news text. To remain close to CKK, we train our language models only on the Europarl training data, and thus use considerably less data than HD here.

4.3 Evaluation

All systems are evaluated using case-insensitive BLEU (Papineni et al., 2002). HD used the NIST BLEU scorer, which requires SGML format. The CKK data is plain text, so instead we report scores

[1] http://www.showlett.id.au/

[2] http://www.statmt.org/wmt09/translation-task.html

[3] http://www.statmt.org/wmt10/translation-task.html

LM	DM	T	Base.	Reord.	Diff.	Oracle
3	dist	–	26.53	27.34	+0.81	28.93
		E	27.58	28.65	+1.07	30.31
		N	26.99	27.16	+0.17	29.37
	lex	–	27.35	27.88	+0.53	29.55
		E	28.34	28.76	+0.42	30.79
		N	27.77	28.27	+0.50	30.10
5	dist	–	27.23	28.12	+0.89	29.69
		E	28.28	28.94	+0.66	30.81
		N	27.42	28.38	+0.96	30.08
	lex	–	28.24	28.70	+0.46	30.47
		E	28.81	29.14	+0.33	31.24
		N	28.32	28.59	+0.27	30.69

Table 4: BLEU scores for each experiment on Europarl test set. Columns give: language model order, distortion model (distance, lexicalised), tuning data (none (–), Europarl, News), baseline BLEU score, reordered system BLEU score, performance increase, oracle BLEU score.

LM	DM	T	Base.	Reord.	Diff.	Oracle
3	dist	–	16.28	15.96	-0.32	17.12
		E	16.43	16.39	-0.04	17.92
		N	17.25	16.51	-0.74	18.40
	lex	–	16.81	16.34	-0.47	17.82
		E	16.75	16.35	-0.40	18.19
		N	17.75	17.02	-0.73	18.73
5	dist	–	16.44	15.97	-0.47	17.28
		E	16.21	15.89	-0.32	17.55
		N	17.27	16.96	-0.31	18.21
	lex	–	17.10	16.58	-0.52	18.16
		E	17.03	17.04	+0.01	18.76
		N	17.73	17.11	-0.62	19.01

Table 5: Results on news test set. Columns as for Table 4.

DM	T	%	Base.	Reord.	Diff.	Oracle
dist	–	50	26.53	27.26	+0.73	28.85
		25		27.03	+0.50	28.66
		10		27.01	+0.48	28.75
	E	50	27.58	28.50	+0.92	30.19
		25		28.27	+0.69	30.21
		10		28.17	+0.59	30.18
lex	–	50	27.35	27.90	+0.55	29.52
		25		27.62	+0.27	29.46
		10		27.54	+0.19	29.42
	E	50	28.34	28.56	+0.22	30.55
		25		28.44	+0.10	30.46
		10		28.42	+0.08	30.42

Table 6: Results using the smaller parsing models. Columns are as for Table 4 except LM removed (all are 3-gram), and parser data percentage (%) added.

from the Moses multi-reference BLEU script (multi-bleu), using one reference translation. Comparing the scripts, we found that the NIST scores are always lower than multi-bleu's on `test2008`, but higher on `newstest2009`, with differences at most 0.23. This partially indicates the noise level in the scores.

5 Results

Results for the first experiments, closely replicating CKK, are given in Table 3. The results are very similar to the those CKK reported (baseline 25.2, reordered 26.8). Thus the HD reimplementation is indeed close to the original CKK system. Any qualitative differences in parser output not revealed by §3, in the implementation of the rules, or in the PSMT system, are thus producing only a small effect.

Results for the remaining experiments are given in Tables 4 and 5, which give results on the `test2008` and `newstest2009` test sets respectively, and Table 6, which gives results on the `test2008` test set using the reduced parsing models.

We see that the choice of data can have a profound effect, nullifying or even reversing the overall result, even when the reordering system remains the same. Genre differences are an obvious possibility, but we have demonstrated only a dependence on data set.

The other factors tested—language model order, lexicalisation of the distortion model, and use of a tuning phase—can all affect the overall performance gain of the reordered system, but less distinctly. Reducing the quality of the parsing model (by training on less data) also has a negative effect, but the drop must be substantial before it outweighs other factors.

In all cases, the oracle outperforms both baseline and reordered systems by a large margin. Its selections show that, in changing test sets, the balance shifts from one system to the other, but both still contribute strongly. This shows that improvements are possible across the board if it is possible to correctly choose which sentences will benefit from reordering.

6 Conclusion

Collins et al. (2005) reported that a reordering-as-preprocessing approach improved overall performance in German-to-English translation. The reimplementation of this system by Howlett and Dras (2010) came to the opposite conclusion.

We have systematically varied several aspects of the Howlett and Dras (2010) system and reproduced results close to both papers, plus a full range in between. Our results show that choices in the PSMT system can completely erode potential gains of the reordering preprocessing step, with the largest effect due to simple choice of data. We have shown that a lack of overall improvement using reordering-as-preprocessing need not be due to the usual suspects, language pair and reordering process.

Significantly, our oracle experiments show that in all cases the reordering system does produce better translations for some sentences. We conclude that effort is best directed at determining for which sentences the improvement will appear.

Acknowledgements

Our thanks to Michael Collins for providing the data used in Collins et al. (2005), and to members of the Centre for Language Technology and the anonymous reviewers for their helpful comments.

References

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics*, pages 531–540.

Amit Dubey and Frank Keller. 2003. Probabilistic parsing for German using sister-head dependencies. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*, pages 96–103.

Nizar Habash. 2007. Syntactic preprocessing for statistical machine translation. In *Proceedings of the MT Summit XI*, pages 215–222.

Hieu Hoang and Philipp Koehn. 2008. Design of the Moses decoder for statistical machine translation. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 58–65.

Susan Howlett and Mark Dras. 2010. Dual-path phrase-based statistical machine translation. In *Proceedings of the Australasian Language Technology Association Workshop*, pages 32–40.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference and the North American Association for Computational Linguistics*, pages 48–54.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318.

Slav Petrov and Dan Klein. 2008. Parsing German with latent variable grammars. In *Proceedings of the ACL-08: HLT Workshop on Parsing German*, pages 33–39.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 433–440.

Wojciech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing*, pages 88–95.

Chao Wang, Michael Collins, and Philipp Koehn. 2007. Chinese syntactic reordering for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 737–745.

Fei Xia and Michael McCord. 2004. Improving a statistical MT system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics*, pages 508–514.

Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Och. 2009. Using a dependency parser to improve SMT for subject-object-verb languages. In *Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 245–253.

Simon Zwarts and Mark Dras. 2007. Syntax-based word reordering in phrase-based statistical machine translation: Why does it work? In *Proceedings of the MT Summit XI*, pages 559–566.