View metadata, citation and similar papers at core.ac.uk
brought to you by CORE
provided by DCU Online Research Access Service

Using Percolated Dependencies for Phrase Extraction in SMT

Ankit K. Srivastava and Andy Way
Centre for Next Generation Localisation
School of Computing, Dublin City University
Glasnevin, Dublin 9, Ireland
{asrivastava, away}@computing.dcu.ie

Abstract

Statistical Machine Translation (SMT) systems rely heavily on the quality of the phrase pairs induced from large amounts of training data. Apart from the widely used method of heuristic learning of *n*-gram phrase translations from word alignments, there are numerous methods for extracting these phrase pairs. One such class of approaches uses translation information encoded in parallel treebanks to extract phrase pairs. Work to date has demonstrated the usefulness of translation models induced from both constituency structure trees and dependency structure trees. Both syntactic annotations rely on the existence of natural language parsers for both the source and target languages. We depart from the norm by directly obtaining dependency parses from constituency structures using head percolation tables. The paper investigates the use of aligned chunks induced from percolated dependencies in French–English SMT and contrasts it with the aforementioned extracted phrases. We observe that adding phrase pairs from any other method improves translation performance over the baseline *n*-gram-based system, percolated dependencies are a good substitute for parsed dependencies, and that supplementing with our novel head percolation-induced chunks shows a general trend toward improving all system types across two data sets up to a 5.26% relative increase in BLEU.

1 Introduction

The phrase-based statistical machine translation (PB-SMT) (Koehn et al., 2003) model is the most widely researched paradigm in MT today. The standard method of extracting phrase-pairs from parallel data involves using union and intersection heuristics on both source-to-target and target-to-source word alignments, in the Moses system (Koehn et al., 2007). This string-based extraction methodology gives rise to 'non-linguistic' chunk pairs, henceforth known as STR.[1]

In this paper, we seek to investigate performance of the baseline Moses MT system by changing one step only, namely the phrase extraction process. Specifically, this entails using three sets of syntactically motivated phrase pairs such as those extracted from node-aligned parallel treebanks. Tinsley et al. (2007) and Hearne et al. (2008) extracted phrase-pairs from constituency-aligned and dependency-aligned data, giving rise to two types of linguistic chunk pairs: CON and DEP respectively. Both these data sets were obtained by monolingual parsing of training sentences, subtree-aligning the parsed trees, and extracting word and phrase alignments . A prerequisite for this approach is the existence of constituency and dependency parsers for both the source and target languages.

Hearne et al. (2008) demonstrated on a very small set of training data that combining string-based extraction (baseline Moses) with either of the syntax-induced phrase extractions resulted in improved translation accuracy with a general trend toward preferring dependency-based over constituency-based phrases. However, there exist more robust and accu-

[1] In the context of SMT a phrase may be any sequence of consecutive words (*n*-gram), not necessarily syntactic constituents.

rate phrase structure parsers than dependency structure parsers for most languages in NLP applications, which has led to alternate measures of automatically generating dependencies from phrase structure parses, as shown on pages 129–131 of Nivre (2006).

In this paper, we heuristically obtain dependency parses by using lexical head information in constituency parse trees. While the head percolation tables themselves are nothing new, the use of phrase pairs induced from them as a separate knowledge source in PB-SMT phrase tables is novel, to the best of our knowledge. This method of annotating and subsequently aligning percolated dependency parses gives rise to another set of aligned chunks: PERC. We then evaluate the uniqueness and usefulness of these alignments against STR, CON, and DEP alignments, and combinations thereof.

The rest of the paper is organised as follows. After a review of related work in Section 2, we briefly describe the MT system setup and phrase extraction methodologies used to obtain the four types of chunk alignments in Section 3. The experiments and analysis of the results are detailed in Section 4, followed by our concluding remarks together with avenues for further research in Section 5.

2 Background and Related Work

We have taken a technique from statistical parsing and introduced its output as another knowledge source in the framework of syntax-aware PB-SMT. In what follows, we present our novel amalgamation of two pre-existing techniques, namely syntax-aware PB-SMT and generation of dependency structures from phrase-stucture parse trees.

2.1 Syntax-aware PB-SMT

Incorporation of linguistic knowledge into the phrase extraction process has shown mixed results in recent years. For instance, Koehn et al. (2003), demonstrated that using syntax to constrain their phrase-based system actually harmed its quality. In contrast, all of the following approaches have shown that augmenting the baseline string-based translation model with syntax-aware word and phrase alignments causes translation performance to improve.

Groves and Way (2005) extract EBMT phrase pairs by monolingually chunking both the source and target sides using closed-class marker words (Green, 1979) and then aligning the resulting chunks using mutual information techniques. Tinsley et al. (2007) extract phrase pairs by obtaining phrase structure parses for both the source and target sides using monolingual parsers and then aligning the subtrees using a statistical tree aligner. Hearne et al. (2008) go a step further by building on the work of (Tinsley et al., 2007) and adding phrase pairs induced from dependency parse trees. Note that all these approaches work on string-based translation models, i.e. syntactic knowledge is merely used to extract linguistically motivated phrase pairs. The phrase tables still contain translations of strings, just like in Moses. There also exist a number of other approaches (Chiang, 2005; Quirk et al., 2005; Galley et al., 2006; Hassan et al., 2008) which have developed different models where the incorporation of syntax has shown itself to be beneficial. However such models are not restricted to the string-based translation modeling and are thus somewhat out of the scope of this paper.

In this paper, we extend the experiments of Hearne et al. (2008) by adding another syntax-aware phrase extraction methodology, namely *percolated dependencies*. We also scale up the volume of the training data, and compare and contrast the resultant phrase tables (cf. section 4).

2.2 Head Percolation

It is possible to obtain a dependency parse for a sentence from its constituency parse (Gaifman, 1965) by exploiting lexicalized heads, i.e. head words of each phrase or constituent. In the absence of this information, a head percolation table is used to select the head node in each constituent structure. For example, the head of a phrase `(NP (DET The) (N box))` is the node `(N box)`. This implies an entry in the head percolation table specifying the node `N` as a head child of the node `NP`. Head percolation tables were first introduced in Magerman (1995) and implemented in Collins (1997). Head percolation tables are so called because, to extract head-dependent information from a constituency parsed treebank, the lexical items are percolated like features from the heads to their parent projections. A head percolation table consists of hand-coded rules identifying the head-child of each node. We imple-

	(a) *JOC DATA*					(b) *EUROPARL DATA*				
SYSTEM	**BLEU**	**NIS**	**MET**	**WER**	**PER**	**BLEU**	**NIS**	**MET**	**WER**	**PER**
STR (S)	31.29	6.31	63.91	61.09	47.34	28.50	7.00	57.83	57.43	44.11
CON (C)	30.64	6.34	63.82	60.72	45.99	25.64	6.55	55.26	60.77	46.82
DEP (D)	30.75	6.31	64.12	61.34	46.77	25.24	6.59	54.65	60.73	46.51
PERC (P)	29.19	6.09	62.12	62.69	48.21	25.87	6.59	55.63	60.76	46.48
S + C	**32.87**	**6.55**	**65.04**	58.70	44.93	**29.50**	**7.10**	**58.55**	56.62	**43.40**
S + D	32.69	**6.55**	64.98	58.66	44.81	29.30	7.08	58.43	56.84	43.62
S + P	32.34	6.48	64.56	59.42	45.51	29.45	**7.10**	58.54	56.73	43.43
C + D	31.24	6.41	64.40	60.28	45.76	26.32	6.69	55.56	59.97	45.90
C + P	30.99	6.36	63.84	60.47	45.81	26.37	6.62	56.05	60.41	46.40
D + P	31.40	6.41	64.41	60.28	45.87	26.57	6.74	55.83	59.53	45.62
S + C + D	32.70	6.53	64.86	58.45	**44.73**	29.29	7.09	58.48	56.70	43.41
S + C + P	32.49	6.48	64.65	58.82	45.22	29.49	**7.10**	58.50	**56.59**	43.45
S + D + P	32.62	6.51	64.82	58.72	45.07	29.39	7.09	58.49	56.80	43.65
C + D + P	31.46	6.41	64.33	59.90	45.58	26.90	6.75	56.14	59.38	45.53
S+C+D+P	32.82	**6.55**	65.03	**58.35**	44.77	29.40	7.09	58.49	56.67	43.49

Table 1: Summary of the results on (a) JOC and (b) Europarl test data

mented the algorithm described in Xia and Palmer (2001) to obtain head-dependent relations between words of a sentence. The head percolation algorithm will output the head or governor for each word in the sentence. In case the word is the head word of the sentence, it will be assigned a default value as its head.

We used this method to obtain dependency parse structures from constituency parse structures for both the source and target languages. We distinguish these structures from the dependency structures obtained directly from a dependency parser by labelling the former as **percolated dependencies**. Theoretically, these percolated dependencies are induced from constituency parses and structurally equivalent to unlabelled dependency parses (Nivre, 2006). However, experimentation in section 4 showed the percolated dependencies to be another source of information different from both constituency and dependency parses.

3 System Specifics

Before evaluating the impact of phrase pairs extracted from percolated dependencies, we describe the machine translation system and data used in our experiments followed by a brief description of the four phrase extraction methodologies.

3.1 Tools and Resources

As described in the previous section, we develop four French–English PB-SMT systems for our experiments: STR, CON, DEP, and PERC. We use two different datasets. We obtain results on a small parallel corpora of approximately 7,700 parallel sentences—the JOC English–French parallel corpus (Chiao et al., 2006) [7,723 train + 400 dev + 599 test sentences]—and a larger set of 100,000 parallel sentences extracted from the freely available Europarl corpus (Koehn, 2005) [100,000 train + 1,889 dev + 2,000 test sentences]. Experimenting on the JOC corpus allows us compare our results directly with those of Hearne et al. (2008), while at the same time we successfully scale up their experiments by almost 13 times.

We also used an open source tree aligner (Zhechev, 2009) to obtain subtree-alignments for the linguistic chunks CON, DEP, and PERC. The tree aligner works by performing a greedy search on all possible alignments between the tree pair nodes and scores using lexical probabilities to select the highest scoring alignment hypothesis. Constituency parse trees were obtained by using the Berkeley parser (Petrov et al., 2006) for both the French and English sides, and dependency parse trees were obtained from the English and French versions of the

Syntex parser (Bourigault et al., 2005). The dependency structures were converted into bracketed format to enable using the tree aligner.

We used GIZA++ (Och and Ney, 2003) for word alignment, SRILM (Stolcke, 2002) for building a 5-gram language model, Minimum Error Rate Training (Och, 2003) for tuning, and the Moses decoder (Koehn et al., 2007) in each of our systems. Thus the only difference between each of the four systems is in the phrase table used in the translation model.

3.2 Phrase Extraction

We explore four different types of phrase pairs in this paper. The first type we term 'non-linguistic' in that phrase pairs are extracted by carrying out string-based union and intersection of source-to-target and target-to-source GIZA++ word alignments (Koehn et al., 2003). The resulting phrases are mere sequences of aligned words occurring together and have no *a priori* syntactic motivation (cf. footnote 1). We label these as STR.

The remaining three phrase-pair inductions are syntactically motivated in that they are produced by first monolingually parsing both the source and target sides. These parse trees are then node-aligned using the statistical tree aligner described above, which also uses information from the GIZA++ word alignment probabilities. The phrase pairs are then extracted by obtaining the surface-level chunks from the aligned subtrees. CON and DEP phrase pairs are induced from the parse trees obtained using off-the-shelf source and target language parsers. Finally, PERC phrase pairs are induced from another dependency-annotated structure which is obtained by applying head percolation features on the phrase structure parse trees used to produce CON phrases. Note that PERC annotations do not require the availability of a dependency-structure parser. Hence, the phrase extraction techniques used to obtain CON, DEP, and PERC chunks differ in only their source of parse trees, i.e. the type of parser and heuristics used to obtain the corresponding parse trees. All other steps in the process remain the same.

After each of the four types of aligned phrases are extracted, they are scored (estimating translation probabilities) using the same algorithm (as defined in Moses system) to build four translation tables.

4 Experimental Analysis

For the purposes of our experiments, we create 15 possible combinations of translation tables from the four types of phrase extractions, namely STR, CON, DEP, and PERC. The combining of two or more systems is carried out by merging the individual phrase tables and re-estimating the phrase translation scores as defined in Moses. For example, the translation table of the system C+D+P is computed by concatenating the extracted phrase tables CON, DEP, and PERC and then re-estimating the probabilities. Each of the 15 configurations were run on both the JOC and Europarl datasets in the French–English translation direction. The results are jointly displayed in Table 1. We evaluate the MT system performance using five evaluation metrics. These are BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), WER (Word Error Rate) and PER (Position-independent WER).

4.1 System Evaluation

What is quite clear from analysing the results on both the JOC and Europarl corpora is the very strong baseline performance of the STR system. For the pairwise comparison, any system combination omitting STR-induced phrase pairs underperforms. Note that in their experiments, both Groves and Way (2005) and Tinsley et al. (2007) acknowledge, as we do here, that *n*-gram-induced phrase pairs are required for both improved translation performance and coverage.

Working on the JOC corpus allowed us to directly compare our novel phrase induction method against the work of Hearne et al. (2008). While we could not improve upon their results (when substituting D with P in any system in Table 1 (a)) for the JOC corpus, running experiments on the 13 times larger Europarl data set showed clear performance gains (a relative increase of as high as 2.49% in BLEU when replacing D with P in any system in Table 1 (b)) over their method when the PERC phrases were utilised. Even if our method did not outperform theirs, our method would still be of use if no separate dependency parser was available for either the source or target language or both.

While the best-performing system combination on both tasks was where STR and CON phrases were

merged, for almost all metrics, the lowest WER rates were observed when PERC chunks were included. In addition, there are quite a few sentences (when computing sentence-level WER scores for each of the four base systems, PERC ranked 2nd best with nearly 25% sentences on both datasets) where PERC performs better than any other system, as in (1):

(1) *Source*: La commission entend-elle garantir plus de transparence à cet égard?

Ref: Does the commission intend to seek more transparency in this area?

STR: Will the commission ensure that more than transparency in this respect?

CON: The commission will the commission ensure greater transparency in this respect?

DEP: The commission will the commission ensure greater transparency in this respect?

PERC: Does the commission intend to ensure greater transparency in this regard?

Note that the propensity of the baseline STR model to omit the verb can be seen to good effect here. Both CON and DEP phrases repeat the translation of the subject NP. In contrast, the translation using PERC phrases is both fluent and accurate, despite not mimicking exactly the reference translation. The lexical differences between the outputs and the reference translation leads us to speculate that the gains from PERC are not accurately reflected in the automatic evaluation scores.

Therefore we also performed manual evaluation on a random selection of 100 sentences from the Europarl testset. A human annotator was shown pairs of sentences along with the source and reference translations and asked to grade whether one system was better than the other or if they were of equal calibre. While PERC and CON systems performed better than the other on the same number of sentences (27%), PERC performed 5% better than DEP. When comparing systems S+C and S+C+P (where the automatic evaluation score differences were not statistically significant), the former system was 11% better. However there were a number of sentences (around 30%) in which PERC was responsible for an output's superior quality. Although no pattern was immediately discernible, a more thorough analysis of these sentence types is left for future work.

We conducted a range of other tests in order to evaluate the uniqueness (degree of difference from other phrase extractions) and usefulness (contribution to MT system performance) of PERC chunks, as described in the next two sections.

4.2 Uniqueness Test

Phrase Types	Common to Both	Uniq. Alig. in 1st type	Uniq. Alig. in 2nd type
S & C	144,671	2,000,942	518,464
S & D	128,760	2,016,853	454,771
S & P	127,531	2,018,082	437,480
C & D	391,804	271,332	191,728
C & P	492,083	171,053	72,929
D & P	369,974	213,558	195,038

Table 2: No. common and unique alignments (phrase pairs) for each method: Europarl data

The total number of entries in each of the four phrase tables (Europarl data) are STR:2,145,614 CON:663,136 DEP:583,532 and PERC:565,012. We can see that the CON t-table is just 31% of the size of the full STR t-table, with DEP just 27% and PERC even smaller at just 26% of the size.

Table 2 provides clear evidence of the differences between the types of chunks produced by each of the four methods. It is interesting that despite the huge size of the STR phrase table, there is very little overlap with any of the other methods; the largest overlap with STR is using CON phrases, but this amounts to only 6% of the STR phrase table being also derived via CON, and only 22% of the CON phrase table being also derived via STR.

The largest overlap in pure numerical terms is between CON and PERC; 74% of the CON phrase table are common with PERC, whereas 87% of the PERC phrase pairs are common with CON. Given that the PERC phrases are derived from the CON

TABLE	JOC	EP
STR (S)	2090	3423
CON (C)	95	419
DEP (D)	111	402
PERC (P)	236	385
S & C	44	287
S & D	87	280
S & P	61	275
C & D	301	330
C & P	91	364
D & P	31	305
S & C & D	196	222
S & C & P	73	259
S & D & P	8	220
C & D & P	780	322
ALL	1261	238
NONE	656	4017

Table 3: Analysis of which phrases the decoder uses in decoding the test data, when trained on the S+C+D+P translation model

trees, one might have expected these two to have the biggest intersection. However, surprisingly, the output (translated sentences produced by CON and PERC systems) has a 30% overlap only. Therefore, it seems that despite a huge overlap in the phrase table configurations, the systems are different enough to produce different translations. We leave for future work an investigation into any bias here. By using two different constituency parsers to produce two sets of PERC chunks, we plan to study their correlation as a measure of bias.

For each of the four phrase extraction methods, the average number of phrase pairs per sentence and the highest number of phrase pairs in a sentence were computed as follows: JOC corpus– (STR: 35.37 (134), CON: 17.62 (71), DEP: 17.82 (71), PERC: 8.45 (53)) and Europarl corpus– (STR: 20.33 (45), CON: 10.82 (27), DEP: 10.67 (27), PERC: 10.66 (26)). Similar performance is seen between the three non-STR methods on Europarl, whereas on JOC our PERC model produces a sizeable number of fewer alignments. The smaller number of phrase pair alignments might very well prove useful for systems with a smaller footprint requiring smaller t-tables (Sanchez-Martinez and Way, 2009).

Having investigated the differences between the chunking methods, the next, more important step is to evaluate whether these unique chunks are of use in PB-SMT.

4.3 Usefulness Test

Moses (Koehn et al., 2007) can be run in 'trace' mode (-t switch) in order to investigate what particular phrases are being selected to derive the translation at any particular time.

In Table 1, we demonstrated that all four sets of phrase pairs could be combined in one phrase table in what we called the 'S+C+D+P' system. In order to translate the Europarl test set of 2,000 sentences, 11,748 phrases were found to be of use. These comprised 5204 STR phrases (of which 3423 were unique, i.e. not produced by any of the other three phrase tables), 2441 CON (419), 2319 DEP (402), and 2368 PERC (385). When it came to a pairwise comparison, the biggest overlap was between CON and PERC. As with our finding regarding Table 2, we will investigate in further work whether there was any bias between these two phrase induction methods. In case of the JOC corpus, for a test set of 599 sentences, 6,121 phrases were found to be of use. These comprised 3820 STR (2090 unique), 2841 CON (95), 2775 DEP (111), and 2541 PERC (236). Note, however, that for the JOC corpus, we found the biggest overlap to be between the CON and DEP phrase tables. As far as triples are concerned, by far the greatest overlap was between CON, DEP and PERC, with an intersection of 780 phrase pairs (the next nearest was just 196). Overall, 1261 phrase pairs were found by each of the four methods. The details for both corpora can be found in Table 3.

In future work, we plan to extract each of these resources as separate phrase tables in the log-linear framework, as it should be the case that where a set of phrase pairs has been verified by all four methods, these can be considered to be of high quality, and worthy of a large weight in the combination of translation resources.

With respect to actual system performance, Figure 1 shows that adding PERC chunks to any system shows a general trend towards boosting scores for BLEU. While we do not include similar graphs for the other automatic evaluation metrics, this tendency

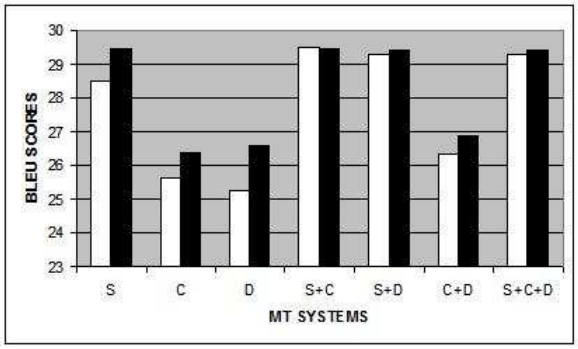

Figure 1: Bar graph to show that adding PERC chunks (black bar) to any system (white bar) generally boosts the BLEU score: Europarl data

is confirmed across all evaluation metrics used in our experiments for both corpora.

5 Conclusion and Future Work

While producing smaller translation models and believed to contain more useful (syntax-aware) phrases than the standard string-based extraction, the syntax-based extractions may perform worse than the PB-SMT string-based baseline, especially as the amount of training data increases (cf. (Zollmann et al., 2008)).[2] However, it has been observed by many researchers that rather than replacing one with the other, combining both types of induced phrases into one translation model significantly improves the translation accuracy. Thus we can supplement SMT phrases with syntax-aware phrases.

Most system development today uses one particular approach to generate phrase pairs for us in translation, namely that of Koehn et al. (2003) (or perhaps more accurately, using the word- and phrase-alignment scripts in Moses (Koehn et al., 2007)).

However, some researchers have pointed out that system performance can be increased when chunks induced by other methods (EBMT (Groves and Way, 2005); constituency parsers (Tinsley et al., 2007); dependency parsers (Hearne et al., 2008)) are added to the SMT phrase table.

[2] cf. also (Lopez, 2009), who argues that due the lack of systematicity in MT system development, it is extremely difficult to compare systems purporting to be of different types, and nigh on impossible to pinpoint exactly to which component any gains in performance might accurately be attributed.

The point is: adherence to one approach may lead to sub-optimal system performance; if any one phrase pair induced by some other method proves to be useful, then ignoring other approaches will cause translation performance to deteriorate, even when the data size is increased.

Accordingly, in this paper we invesigated whether phrase pairs induced via head percolation (Magerman, 1995) might prove useful in PB-SMT. In a number of experiments, we showed that the number of chunks, and their content, was different for each of the four methods: STR, CON, DEP, and PERC. Furthermore, we showed that system perfomance improved significantly when PERC phrases were added to the phrase table of any other system. This was validated on two tasks for French–English: a small (JOC) and a larger (Europarl) dataset. Working on the JOC corpus allowed us to directly compare our novel phrase induction method against the work of (Hearne et al., 2008). While we could not improve upon their results for the JOC corpus, running experiments on the far larger Europarl data set showed clear performance gains over their method (dependencies using a parser) when the PERC phrases were utilised. In any case, our method would be useful in language pairs for which no separate dependency parser was available.

It was also discovered through automatic evaluation measures that the 'S+C' system gave the best performance. However lack of statistical significance in the results and manual evaluation leads us to believe that PERC is useful enough to grant further investigation. Perhaps, better ways of combining the individual phrase tables need to be studied.

As regards further work, we plan to conduct a more in-depth manual analysis to discover exactly what are the individual contributions of each of the phrase induction methods to translation quality. In addition to exploring some of the issues raised in section 4 we also intend to verify our results on larger data sizes, more domains and different language pairs.

Acknowledgements

Thanks to Sylwia Ozdowska for helping us with manual evaluation and John Tinsley for providing us with initial data. This work is supported by Science

Foundation Ireland (grant number: 07/CE/I1142).

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. In *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at ACL '05*, Ann Arbor, Michigan, pp.65–72.

Didier Bourigault, Cécile Fabre, C. Frérot, M. P. Jacques, and Sylwia Ozdowska. 2005. Syntex, analyseur syntaxique de corpus. In *TALN '05*, Dourdan, France.

David Chiang. 2005. A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of ACL '05*, Ann Arbor, Michigan, pp.263–270.

Yun-Chuang Chiao, Olivier Kraif, Dominique Laurent, Thi Nguyen, N. Semmar, F. Stuck, Jean Véronis, and W. Zaghouani. 2006. Evaluation of Multilingual Text Alignment Systems: the ARCADE II Project. In *Proceedings of LREC '06*, Genoa, Italy, pp.1975–1978.

Michael Collins. 1997. Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of ACL/EACL '97*, Madrid, Spain, pp.16–23.

George Doddington. 2002. Automatic Evaluation of MT Quality Using N-gram Co-occurrence Statistics. In *HLT: Notebook Proceedings*, pp.128–132.

Haim Gaifman. 1965. Dependency Systems and Phrase-Structure Systems. In *Information and Control* 8:304–337.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable Inference and Training of Context-Rich Syntactic Models. In *Proceedings of COLING/ACL '06*, Sydney, Australia, pp.961–968.

Thomas Green. 1979. The Necessity of Syntax Markers. Two experiments with artificial languages. In *Journal of Verbal Learning and Behavior* 18:481–496.

Declan Groves and Andy Way. 2005. Hybrid Example-Based SMT: the Best of Both Worlds? In *Proceedings of the ACL 2005 Workshop on Building and Using Parallel Texts: Data-Driven MT and Beyond*, Ann Arbor, Michigan, pp.183–190.

Hany Hassan, Khalil Sima'an, and Andy Way. 2008. Syntactically Lexicalized Phrase-Based SMT. In *IEEE Transactions on Audio, Speech and Language Processing*, 16(7):1260–1273.

Mary Hearne, Sylwia Ozdowska, and John Tinsley. 2008. Comparing Constituency and Dependency Representations for SMT Phrase-Extraction. In *TALN '08*, Avignon, France.

Philipp Koehn. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *MT Summit X*, Phuket, Thailand, pp.79–86..

Philipp Koehn, Hieu Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, Wade Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *demonstration session of ACL '07*, Prague, Czech Republic, pp.177–180.

Philipp Koehn, Franz Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of NAACL '03*, Edmonton, Canada, pp.48–54.

Adam Lopez. 2009. Translation as Weighted Deduction. In *Proceedings of EACL '09*, Athens, Greece.

David Magerman. 1995. Statistical Decision-Tree Models for Parsing. In *Proceedings of ACL '95*, Cambridge, Massachusetts, pp.276–283.

Joakim Nivre. 2006. Inductive Dependency Parsing. BOOK *Springer Publishers*, Netherlands.

Franz Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of ACL '03*, Sapporo, Japan, pp.160–167.

Franz Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. In *Computational Linguistics*, 29(1):19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jung Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL '02*, Philadelphia, Pennsylvania, pp.311–318.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of COLING/ACL '06*, Sydney, Australia, pp.433–440.

Chris Quirk, Arul Menezes, and Colin Cherry. 2005. Dependency Treelet Translation: Syntactically-informed Phrasal SMT. In *Proceedings of ACL '05*, Ann Arbor, Michigan, pp.271–279.

Felipe Sanchez-Martinez and Andy Way. 2009. Marker-based Filtering of Biingual Phrase Pairs for SMT. In *Proceedings of EAMT '09*, Barcelona, Spain, in press.

Andreas Stolcke. 2002. SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, Denver, Colorado.

John Tinsley, Mary Hearne, and Andy Way. 2007. Exploiting Parallel Treebanks to Improve Phrase-Based Statistical Machine Translation. In *Proceedings of TLT '07*, Bergen, Norway, pp.175–187.

Fei Xia and Martha Palmer. 2001. Converting Dependency Structures to Phrase Structures. In *Proceedings of HLT '01*, San Diego, California, pp.1–5.

Ventsislav Zhechev. 2009. Unsupervised Generation of Parallel Treebank through Sub-Tree Alignment. In *Prague Bulletin of Mathematical Linguistics, Special Issue: Open Source Tools for MT.*, (91):89–98.

Andreas Zollmann, Ashish Venugopal, Franz Och, and Jay Ponte. 2008. A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. In *Proceedings of COLING '08*, Manchester, UK, pp.1145–1152.