A Syntactic Skeleton for Statistical Machine Translation

Bart Mellebeek, Karolina Owczarzak, Declan Groves, Josef Van Genabith and Andy Way

Dublin City University, National Centre for Language Technology, Dublin 9, Ireland

{mellebeek | owczarzak | dgroves | josef | away}@computing.dcu.ie

Abstract

We present a method for improving statistical machine translation performance by using linguistically motivated syntactic information. Our algorithm recursively decomposes source language sentences into syntactically simpler and shorter chunks, and recomposes their translation to form target language sentences. This improves both the word order and lexical selection of the translation. We report statistically significant relative improvements of 3.3% BLEU score in an experiment (English→Spanish) carried out on an 800-sentence test set extracted from the Europarl corpus.

1 Introduction

Almost all research in MT being carried out today is corpus-based, with by far the most dominant paradigm being phrase-based Statistical Machine Translation (SMT). Phrase-based models have recently achieved considerable improvements in translation quality; however, they still face difficulty when it comes to modeling long-distance dependencies or differences in word order between source and target languages. An obvious way to help overcome these obstacles is to try to add a syntactic level to the models. While a number of attempts have been made to incorporate syntactic knowledge into phrase-based SMT, this has led to little improvement in translation and the loss of language-independence for the systems.

Our novel approach uses TransBooster, a wrapper technology designed to improve the output of wide-coverage MT systems (Mellebeek, Khasin, Van Genabith, & Way, 2005) by exploiting the fact that both rule-based and statistical MT systems tend to perform better at translating shorter sentences than longer ones. TransBooster decomposes source language sentences into syntactically simpler and shorter chunks, sends the chunks to a baseline MT system and recomposes the translated output into target language sentences. It has already proved successful in experiments with rule-based MT systems (Mellebeek, Khasin, Owczarzak, Van Genabith, & Way, 2005). In this paper we apply the TransBooster wrapper technology to a state-of-the-art phrase-based English → Spanish SMT model constructed with Pharaoh (Koehn, 2004) and we report a statistically significant improvement in BLEU and NIST score.

The paper is organised as follows. In section 2, we give a short overview of the most relevant methods that incorporate syntactic knowledge in SMT models. We explain our approach in section 3 and demonstrate it with a worked example. Sections 4 and 5 contain the description, results and analysis of our experiments. We summarize our findings in section 6.

2 Related Research

One of the major difficulties SMT faces is its inability to model long-distance dependencies and correct word order for many language pairs. In this respect, phrase-based SMT systems fare much better than word-based systems, but are still far from

perfect. As reported by (Koehn, Och, & Marcu, 2003), even increasing phrase length above three words does not lead to significant improvements due to data sparseness. Therefore, SMT is usually most accurate in very localised environments and for language pairs that do not differ too much in the systematic ordering of constituents. A number of more recent MT models attempt to remedy the shortcomings of SMT by introducing a degree of syntactic information into the process. Generally, MT models that do incorporate syntax do so in a limited fashion, by using syntax on either the source or target side but not on both. (Yamada & Knight, 2001, 2002; Charniak, Knight, & Yamada, 2003; Burbank et al., 2005)

It should also be noted that, to date, approaches which have attempted to incorporate more syntactic modeling into SMT have on the whole not yet resulted in significant improvements. Previous approaches include the tree-to-string manipulation model of (Yamada & Knight, 2001, 2002), and the attempt to marry PCFG language models to SMT (Charniak et al., 2003). Furthermore, the *post hoc* reranking approach of (Koehn et al., 2003) actually demonstrated that adding syntax harmed the quality of their SMT system.

(Chiang, 2005) presents an SMT model that uses hierarchical phrase probabilities. This allows for the correct treatment of higher-level dependencies, such as the different ordering of NP-modifying relative clauses in Chinese and English. In an experiment on Mandarin to English translation, (Chiang, 2005) reports an relative increase of 7.5% in the BLEU score over a baseline Pharaoh model. Although this method can deal successfully with certain notoriously problematic tasks and is language-independent, it induces a grammar from a parallel text without relying on any linguistic annotations or assumptions. It therefore does not make use of *linguistically motivated syntax*, in contrast to TransBooster.

3 TransBooster: Architecture

TransBooster uses a chunking algorithm to divide input strings into smaller and simpler constituents, sends those constituents in a minimal necessary context to a baseline MT system and recomposes the MT output chunks to obtain the overall translation of the original input string.

Our approach presupposes the existence of some sort of syntactic analysis of the input sentence. We report experiments on human parse-annotated sentences (the Penn II Treebank (Marcus et al., 1994)) and on the output of a state-of-the-art statistical parser (Charniak, 2000) in section 5.

Essentially, each TransBooster cycle from a parsed input string to a translated output string consists of the following 5 steps:

1. Finding the Pivot.
2. Locating Arguments and Adjuncts ('Satellites') in the source language.
3. Creating and Translating Skeletons and Substitution Variables.
4. Translating Satellites.
5. Combining the translation of Satellites into the output string.

We briefly explain each of these steps by processing the following simple example sentence.

(1) The chairman, a long-time rival of Bill Gates, *likes* **fast** and confidential deals.

BabelFish (English → Spanish) translates (1) as (2):

(2) El presidente, rival de largo plazo de Bill Gates, *gustos* **ayuna** y los repartos confidenciales.

Since the system has wrongly identified **fast** as the main verb (**'ayunar' = 'to fast'**) and has translated *likes* as a noun (*'gustos' = 'tastes'*), it is almost impossible to understand the output. The following sections will show how TransBooster interacts with the baseline MT system to help it improve its own translations.

3.1 Decomposition of Input

In a first step, the input sentence is decomposed into a number of syntactically meaningful chunks as in (3).

(3) $[ARG_1]$ $[ADJ_1]\ldots[ARG_L]$ $[ADJ_l]$ **pivot** $[ARG_{L+1}]$ $[ADJ_{l+1}]\ldots[ARG_{L+R}]$ $[ADJ_{l+r}]$

where **pivot** = the nucleus of the sentence, ARG = argument, ADJ = adjunct, {l,r} = number of ADJs to left/right of **pivot**, and {L,R} = number of ARGs to left/right of **pivot**.

The pivot is the part of the string that must remain unaltered during decomposition in order to ensure a correct translation. In order to determine the pivot, we compute the head of the local tree by adapting the head-lexicalised grammar annotation scheme of (Magerman, 1995). In certain cases, we derive a 'complex pivot' consisting of this head terminal together with some of its neighbours, e.g. phrasal verbs or strings of auxiliaries. In the case of the example sentence (1), the pivot is 'likes'.

During the decomposition, it is essential to be able to distinguish between arguments (required elements) and adjuncts (optional material), as adjuncts can safely be omitted from the simplified string that we submit to the MT system. The procedure used for argument/adjunct location is an adapted version of Hockenmaier's algorithm for CCG (Hockenmaier, 2003). The result of this first step on a the example sentence (1) can be seen in (4).

(4) [The chairman, a long-time rival of Bill Gates,]$_{ARG_1}$ [likes]$_{pivot}$ [fast and confidential deals]$_{ARG_2}$.

3.2 Skeletons and Substitution Variables

In a next step, we replace the arguments by similar but simpler strings, which we call 'Substitution Variables'. The purpose of Substitution Variables is: (i) to help to reduce the complexity of the original arguments, which often leads to an improved translation of the pivot; (ii) to help keep track of the location of the translation of the arguments in target. In choosing an optimal Substitution Variable for a constituent, there exists a trade-off between accuracy and retrievability. 'Static' or previously defined Substitution Variables (e.g. 'cars' to replace the NP 'fast and confidential deals') are easy to track in target, since their translation by a specific MT engine is known in advance, but they might distort the translation of the pivot because of syntactic/semantic differences with the original constituent. 'Dynamic' Substitution Variables comprise the real heads of the constituent (e.g. 'deals' to replace the NP 'fast and confidential deals') guarantee a maximum similarity, but are more difficult to track in target. Our algorithm employs Dynamic Substitution Variables first and backs off to Static Substitution Variables if problems occur. By replacing the arguments by their Substitution Variables and leaving out the adjuncts in (1), we obtain the skeleton in (5)

(5) $[V_{ARG_1}]$ $\ldots[V_{ARG_L}]$ **pivot** $[V_{ARG_{L+1}}]\ldots[V_{ARG_{L+R}}]$

where V_{ARG_i} is the simpler string substituting ARG_i

The result of this second step on the worked example can be seen in (6).

(6) [The chairman]$_{V_{ARG_1}}$ [likes]$_{pivot}$ [deals]$_{V_{ARG_2}}$.

TransBooster sends this simple string to the baseline MT system, which this time is able to produce a better translation than for the original, more complex sentence, as in (7).

(7) El presidente tiene gusto de repartos.

This translation allows us (i) to extract the translation of the pivot and (ii) to determine the location of the arguments. This is possible because we determine the translations of the Substitution Variables (*the chairman*, *deals*) at runtime. If these translations are not found in (7), we replace the arguments by previously defined Static Substitution Variables. E.g. in (4), we replace *'The chairman, a long-time rival of*

Bill Gates' by *'The man'* and *'fast and confidential deals'* by *'cars'*. In case the translations of the Static Substitution Variables are not found (7), we interrupt the decomposition and have the entire input string (1) translated by the MT engine.

3.3 Translating Satellites

After finding the translation of the pivot and the location of the translation of the satellites in target, the procedure is recursively applied to each of the identified chunks *'The chairman, a long-time rival of Bill Gates'* and *'fast and confidential deals'*.

Since the chunk *'fast and confidential deals'* contains fewer words than a previously set threshold - this threshold depends on the syntactic nature of the input - it is ready to be translated by the baseline MT system. Translating individual chunks out of context is likely to produce a deficient output or lead to boundary friction phenomena, so we need to ensure that each chunk is translated in a simple context that mimics the original. As in the case of the Substitution Variables, this context can be static (a previously established template, the translation of which is known in advance) or dynamic (a simpler version of the original context).

The dynamic context for ARG_2 in (4) would be the a simplified version of ARG_1 followed by the pivot *'The chairman likes'*, the translation of which is determined at runtime, as in (8):

(8) [The chairman likes] fast and confidential deals. → [El presidente tiene gusto de] repartos rápidos y confidenciales.

An example of a static context mimicking direct object position for simple NPs would be the string *The man sees*, which most of the time in Spanish would be translated as *El hombre ve*, as in (9):

(9) [The man sees] fast and confidential deals. → [El hombre ve] repartos rápidos y confidenciales.

Since the remaining chunk *'The chairman, a long-time rival of Bill Gates'* contains more words than a previously set threshold, it is judged too complex for direct translation. The decomposition and translation procedure is now recursively applied to this chunk: it is decomposed into smaller chunks, which may or may not be suited for direct translation, and so forth.

3.4 Forming the Translation

As explained in subsection 3.3, the input decomposition procedure is recursively applied to each constituent until a certain threshold is reached. Constituents below this threshold are sent to the baseline MT system for translation. Currently, the threshold is related to the number of lexical items that each node dominates. Its optimal value depends on the syntactic environment of the constituent and the baseline MT system used. After all constituents have been decomposed and translated, they are recombined to yield the target string output to the user.

In example (1), the entire decomposition and recombination process leads to an improvement in translation quality compared to the original output by Systran in (2), as is shown in (10):

(10) El presidente, un rival de largo plazo de Bill Gates, tiene gusto de repartos rápidos y confidenciales.

4 Experimental Setup

For our experiments, the phrase-based SMT system (English → Spanish) was constructed using the Pharaoh phrase-based SMT decoder, and the SRI Language Modeling toolkit.[1] We used an interpolated trigram language model with Kneser-Ney discounting.

The data used to train the system was taken from the English-Spanish section of the Europarl corpus (Koehn, 2005). From this data, 501K sentence pairs were randomly extracted from the designated training section of the corpus and lowercased. Sentence length was limited to a maximum of 40 words for both Spanish and English,

[1] http://www.speech.sri.com/projects/srilm/

with sentence pairs having a maximum relative sentence length ratio of 1.5. From this data we used the method of (Och & Ney, 2003) to extract phrase correspondences from GIZA++ word alignments.

For testing purposes two sets of data were used, each consisting of 800 English sentences. The first set was randomly extracted from section 23 of the WSJ section of the Penn II Treebank; the second set consists of randomly extracted sentences from the test section of the Europarl corpus, which had been parsed with (Bikel, 2002).

We decided to use two different sets of test data instead of one because we are faced with two 'out-of-domain' phenomena that have an influence on the scores, one affecting the TransBooster algorithm, the other the phrase-based SMT system.

On the one hand, the TransBooster decomposition algorithm performs better on 'perfectly' parse-annotated sentences from the Penn Treebank than on the output produced by a statistical parser as (Bikel, 2002), which introduces a certain amount of noise. On the other hand, Pharaoh was trained on data from the Europarl corpus, so it performs much better on translating Europarl data than out-of-domain Wall Street Journal text.

5 Results and Evaluation

We present results of an automatic evaluation using BLEU (Papineni, Roukos, Ward, & Zhu, 2002) and NIST (Doddington, 2002) against the 800-sentence test sets mentioned in section 4. In each case, the statistical significance of the results was tested by using the BLEU/NIST resampling toolkit described in (Zhang & Vogel, 2004).[2] We also conduct a manual evaluation of the first 200 sentences in the Europarl test set. Finally, we analyse the differences between the output of Pharaoh and TransBooster, and provide a number of example translations.

[2]http://projectile.is.cs.cmu.edu/research/public/tools/bootStrap/tutorial.htm

5.1 Automatic Evaluation

5.1.1 Europarl

English→Spanish	BLEU	NIST
Pharaoh	0.1986	5.8393
TransBooster	0.2052	5.8766
Percent. of Baseline	103.3%	100.6%

Table 1: TransBooster vs. Pharaoh: Results on the 800-sentence test set of Europarl

The comparison between TransBooster and Pharaoh on the Europarl test set is shown in Table 1. TransBooster improves on Pharaoh with a statistically significant relative improvement of 3.3% in BLEU and 0.6% in NIST score. These results shows that the TransBooster approach not only works for sentences parse-annotated by humans, as reported in (Mellebeek et al., 2005), but also for previously unseen input after parsing with a statistical parser (Bikel, 2002).

5.1.2 Wall Street Journal

English→Spanish	BLEU	NIST
Pharaoh	0.1343	5.1432
TransBooster	0.1379	5.1259
Percent. of Baseline	102.7%	99.7%

Table 2: TransBooster vs. Pharaoh: Results on the 800-sentence test set of the WSJ

The comparison between TransBooster and Pharaoh on the Wall Street Journal test set is shown in Table 2. As with Europarl, TransBooster improves on Pharaoh according to the BLEU metric, but falls slightly short of Pharaoh's NIST score. In contrast to the scores on the Europarl corpus, these results are *not* statistically significant according to a resampling test (on 2000 resampled test sets) with the toolkit described in (Zhang & Vogel, 2004).

Although the input to TransBooster in this case are 'perfect' human parse-annotated sentences, we are not able to report statistically significant improvements over Pharaoh. This can be explained by the fact that the performance of phrase-based SMT systems on out-of-domain text is very poor (items are left untranslated, etc.) as

Original	Despite *an impressive number* of international studies , there is still no *clear evidence* of any direct link between violence and media consumption
Pharaoh	a pesar de los estudios internacionales , todavía no existe ninguna relación directa entre la violencia y media *un número impresionante pruebas claras* de consumo
TransBooster	pese a *un número impresionante* de estudios internacionales , todavía no hay *pruebas claras* de ninguna relación directa entre la violencia y los medios consumo
Analysis	**word order**: better placement of the translations of *'an impressive number'* and *'clear evidence'*
Original	The European Union is jointly responsible, *with the countries of origin*, for immigration and for *organising* those migration flows, which are so necessary for the development of the region.
Pharaoh	la unión europea es corresponsable de inmigración y de los flujos migratorios, que son necesarias para el desarrollo de la región, *con los países de origen*, *organizador*.
TransBooster	la unión europea es corresponsable, *con los países de origen*, de inmigración y de los flujos migratorios, que son necesarias para *organizar* el desarrollo de la región.
Analysis	**word order**: better placement of the translation of *'with the countries of origin'* and *'organising'*
Original	Presidency communication on the situation in *the Middle East*
Pharaoh	presidencia comunicación sobre la situación en *el mediterráneo*
TransBooster	presidencia comunicación sobre la situación en *el cercano oriente*
Analysis	**lexical selection**: improved translation of *'the Middle East'*
Original	*I am proud of* the fact that the Committee on Budgetary Control has been able to *agree unanimously* on a draft opinion within a very short period of time .
Pharaoh	*me alegra* el hecho de que la comisión de presupuestos ha podido *dar mi aprobación unánime* sobre un proyecto dictamen en un periodo de tiempo muy corto .
TransBooster	*estoy orgulloso* del hecho que la comisión de presupuestos *ha llevado a acuerdo unánime* sobre un proyecto dictamen en un periodo de tiempo muy corto .
Analysis	**lexical selection**: improved translation of *'I am proud of'* and *'agree unanimously'*

Table 3: Examples of improvements over Pharaoh: word order and lexical selection.

is described in (Koehn, 2005) and indicated by much lower absolute test scores in comparison to table 1. In other words, in this case it is more difficult for TransBooster to help the SMT system to improve on its own output through syntactic guidance.

5.2 Manual Evaluation

After a manual evaluation of the first 200 sentences of the Europarl test set, based on an average between accuracy and fluency, we considered 20% of these to be better when TransBooster was used, 7% being worse, and the remaining 73% adjudged to be similar.

The majority of improvements (70%) by invoking the TransBooster method on Pharaoh are caused by a better word order. This is because it is syntactic knowledge and not a linguistically limited language model, that guides the placement of the translation of the decomposed input chunks. Moreover, smaller input chunks, as produced by TransBooster and translated in a minimal context, are more likely to receive correct internal ordering from the SMT language model.

The remaining 30% of improvements resulted from a better lexical selection. This is caused not only by shortening the input, but mainly by TransBooster being able to separate the input sentences at points of least co-

hesion, namely, at major constituent boundaries. It is plausible to assume that probability links between the major consituents are weaker than inside them, due to data sparseness, so translating a phrase in the context of only the *heads* of neighbouring constituents might actually help.

Table 3 illustrates the main types of improvements with a number of examples.

6 Conclusions

We have shown that statistical machine translation improves when we add a level that incorporates syntactic information. TransBooster capitalises on the fact that MT systems generally deal better with shorter sentences, and uses syntactic annotation to decompose source language sentences into shorter, simpler chunks which have a higher chance of being correctly translated. The resulting translations are recomposed into target language sentences. The advantage of the TransBooster approach over other methods is that it is generic, being able to work with various MT systems, and that the syntactic information it uses is linguistically motivated. We show that the Pharaoh model coupled with TransBooster achieves a statistically significant relative improvement of 3.3% in BLEU score over Pharaoh alone, on English → Spanish translations of a 800-sentence test set extracted from the Europarl corpus.

References

Bikel, D. M. (2002). Design of a Multi-lingual, Parallel-processing Statistical Parsing Engine. In *Proceedings of Human Language Technology Conference (HLT 2002)* (p. 24-27). San Diego, CA.

Burbank, A., Carpuat, M., Clark, S., Dreyer, M., Fox, P., Groves, D., Hall, K., Hearne, M., Melamed, D., Shen, Y., Way, A., Wellington, B., & Wu, D. (2005). Final Report of the Johns Hopkins Summer Workshop on Statistical Machine Translation by Parsing. In *JHU Workshop 2005.* Baltimore, MD.

Charniak, E. (2000). A maximum entropy inspired parser. In *Proceedings of the First Annual Meeting of the North American Chapter of the Association for Computational Linguistics (NAACL 2000)* (p. 132-139). Seattle, WA.

Charniak, E., Knight, K., & Yamada, K. (2003). Syntax-based Language Models for Statistical Machine Translation. In *Proceedings of the Ninth Machine Translation Summit* (p. 40-46). New Orleans, LO.

Chiang, D. (2005). A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of ACL 2005* (p. 263-270). Ann Arbor, MI.

Doddington, G. (2002). Automatic Evaluation of MT Quality using N-gram Co-occurrence Statistics. *Human Language Technology*, 128-132.

Hockenmaier, J. (2003). Parsing with Generative models of Predicate-Argument Structure. In *Proceedings of the ACL 2003* (p. 359-366). Sapporo, Japan.

Koehn, P. (2004). Pharaoh: A beam search decoder for phrase-based statistical machine translation models. In *AMTA* (p. 115-124). Georgetown University, Washington DC.

Koehn, P. (2005). Europarl: A parallel Corpus for Evaluation of Machine Translation. In *MT Summit X* (p. 79-86). Phuket, Thailand.

Koehn, P., Och, F., & Marcu, D. (2003). Statistical Phrase-based Translation. In *Proceedings of HLT-NAACL 2003* (p. 127-133). Edmonton, Canada.

Magerman, D. (1995). Statistical Decision-Tree Models for Parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics* (p. 276-283). Cambridge, MA.

Marcus, M., Kim, G., Marcinkiewicz, M., MacIntyre, R., Bies, A., Ferguson, M., Katz, K., & Schasberger, B. (1994). The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the ARPA Human Language Technology workshop* (p. 114-119).

Mellebeek, B., Khasin, A., Owczarzak, K., Van Genabith, J., & Way, A. (2005). Im-

proving online Machine Translation Systems. In *Proceedings of MT Summit X* (p. 290-297). Phuket, Thailand.

Mellebeek, B., Khasin, A., Van Genabith, J., & Way, A. (2005). TransBooster: Boosting the Performance of Wide-Coverage Machine Translation Systems. In *Proceedings of the 10th Annual Conference of the European Association for Machine Translation* (p. 189-197). Budapest, Hungary.

Och, F. J., & Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics 29(1)*, 19-51.

Papineni, K., Roukos, S., Ward, T., & Zhu, W.-J. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In (p. 311-318). Philadelphia.

Yamada, K., & Knight, K. (2001). A Syntax-Based Statistical Translation Model. In *Proceedings of the 39th Annual Conference of the Association for Computational Linguistics* (p. 523-530). Toulouse, France.

Yamada, K., & Knight, K. (2002). A Decoder for Syntax-Based Statistical MT. In *Proceedings of the 40th Annual Conference of the Association for Computational Linguistics* (p. 303-310). Philadelphia, PA.

Zhang, Y., & Vogel, S. (2004). Measuring Confidence Intervals for the Machine Translation Evaluation Metrics. In *Proceedings of the Tenth Conference on Theoretical and Methodological Issues in Machine Translation* (p. 85-94). Baltimore, MD.