Wrapper Syntax for Example-Based Machine Translation

Karolina Owczarzak Bart Mellebeek Declan Groves Josef Van Genabith Andy Way
National Centre for Language Technology
School of Computing, Dublin City University, Dublin 9
{owczarzak,mellebeek,dgroves,josef,away}@computing.dcu.ie

Abstract

TransBooster is a wrapper technology designed to improve the performance of wide-coverage machine translation systems. Using linguistically motivated syntactic information, it automatically decomposes source language sentences into shorter and syntactically simpler chunks, and recomposes their translation to form target language sentences. This generally improves both the word order and lexical selection of the translation. To date, TransBooster has been successfully applied to rule-based MT, statistical MT, and multi-engine MT. This paper presents the application of TransBooster to Example-Based Machine Translation. In an experiment conducted on test sets extracted from Europarl and the Penn II Treebank we show that our method can raise the BLEU score up to 3.8% relative to the EBMT baseline. We also conduct a manual evaluation, showing that TransBooster-enhanced EBMT produces a better output in terms of fluency than the baseline EBMT in 55% of the cases and in terms of accuracy in 53% of the cases.

1 Introduction

Almost all research in Machine Translation (MT) carried out today is corpus-based, and one of the most promising research directions in that area is Example-Based Machine Translation (EBMT). EBMT models have recently achieved considerable improvements in translation quality; however, like other statistical MT systems, they still face difficulty when it comes to modelling long-distance dependencies or differences in word order between source and target languages.

Our approach uses TransBooster, a wrapper technology designed to improve the output of wide-coverage MT systems (Mellebeek et al., 2005a) by exploiting the fact that both rule-based and statistical MT systems tend to perform better when translating shorter sentences than longer ones. TransBooster decomposes source language sentences into shorter, syntactically simpler 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 and statistical MT systems (Mellebeek et al., 2005b, Mellebeek et al., 2006), as well as in experiments with multi-engine MT (Mellebeek et al., this volume). In this paper we apply the TransBooster wrapper technology to an Example-Based MT system. Even though we see a relative improvement of 3.8% in BLEU and 0.5% in NIST scores over the baseline EBMT system, we argue that these metrics are not able to reflect fully the improvement our method introduces. In a preliminary manual evaluation, we show that TransBooster helps obtain better translation fluency in 55% of the cases, and accuracy in 53% of the cases. This paper is organized as follows: Section 2 describes background research in EBMT; Section 3 presents the architecture of TransBooster; Section 4 describes the experimental setup; Section 5 gives the results of the experiment; Section 6 concludes.

2 Related research

Example-based MT is based on bitexts, i.e. a set of sentences in one language aligned with their translations in another. Taking a corpus of source–target aligned sentence pairs, EBMT models of translation perform three distinct processes in order to transform a new input string into a target language translation:

1. Searching the source side of the bitext for ‘close’ matches of sentences and subsentential strings and their translations.

2. Determining the sub-sentential translation links in those retrieved examples.

3. Recombining relevant parts of the target translation links to derive the translation.

In order to determine a similarity metric during the search for relevant matches, word co-occurrence, part-of-speech labels, generalised templates and bilingual dictionaries are often used. The recombination process depends on the nature of the examples used in the first place, which may include aligning phrase-structure trees (Hearne and Way, 2003) or dependency trees (Watanabe et al., 2003), or using placeables (Brown, 1999) as indicators of chunk boundaries.

2.1 Marker-based EBMT

One approach in EBMT is to use a set of closed-class words to segment aligned source and target sentences and to derive an additional set of lexical and phrasal resources. This approach is based on the ‘Marker Hypothesis’ (Green, 1979), a universal psycholinguistic constraint which posits that languages are ‘marked’ for syntactic structure at surface level by a closed set of specific lexemes and morphemes. In a preprocessing stage, the source–target aligned sentences are segmented at each new occurrence of a marker word (e.g. determiners, quantifiers, conjunctions etc.), and together with cognate matches and mutual information scores, aligned marker chunks are derived.

In order to describe this resource creation in more detail, consider the English–Spanish example in (1):

```
(1) You click on the red
button to view the effect of
the selection.
-> Usted cliquea en el botón
rojo para ver el efecto de la
selección.
```

The first stage involves automatically tagging each closed-class word in (1) with its marker tag, as in (2):

```
(2) <PRON> You click <PREP>
on <DET> the red button
<PREP> to view <DET> the
effect <PREP> of <DET> the
selection.
-> <PRON> Usted cliquea
<PREP> en <DET> el botón rojo
<PREP> para ver <DET> el
efecto <PREP> de <DET> la
selección.
```

Taking into account marker tag information (label, and relative sentence position), and lexical similarity (via mutual information), the marker chunks in (3) are automatically generated from the marker-tagged strings in (2):

```
(3) a. <PRON> You click :
<PRON> Usted cliquea
b. <PREP> on the red button :
<PREP> en el botón rojo
c. <PREP> to view : <PREP>
para ver
d. <DET> the effect : <DET>
el efecto
e. <PREP> of the selection :
<PREP> de la selección
```

In our experiments our marker set consisted of determiners, prepositions, conjunctions, personal pronouns, possessive pronouns, quantifiers and wh-adverbs, following (Gough, 2005; and Gough and Way, 2004). We also made use of auxiliary verbs, such as *has* and *is* in English and their Spanish counterparts *ha* and *es*, in addition to punctuation, which acted as chunk-final, rather than chunk-initial markers.

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 an 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 (Bikel, 2002) in Section 5. Essentially, each TransBooster run 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:

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

The commercial machine translation system Systran[1] (English to Spanish) translates (4) as (5):

```
(5) 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 an MT system to help it improve its own translations.

[1] http://www.systransoft.com/

3.1 Decomposition of input

In a first step, the input sentence is decomposed into a number of syntactically meaningful chunks as in (6):

(6) $[\mathrm{ARG}_1]$ $[\mathrm{ADJ}_1]$. . . $[\mathrm{ARG}_L]$ $[\mathrm{ADJ}_l]$ pivot $[\mathrm{ARG}_{L+1}]$ $[\mathrm{ADJ}_{l+1}]$. . . $[\mathrm{ARG}_{L+R}]$ $[\mathrm{ADJ}_{l+r}]$

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 avoid an incorrect 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 (4), 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 the example sentence (4) can be seen in (7):

(7) $[$The chairman, a long-time rival of Bill Gates,$]_{\mathrm{ARG1}}$ $[$likes$]_{\mathrm{pivot}}$ $[$fast and confidential deals$]_{\mathrm{ARG2}}$.

3.2 Skeletons and Substitution Variables

In the 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 head 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 (4), we obtain the skeleton in (8):

(8) [V_{ARG1}] . . . [V_{ARGL}] pivot [V_{ARGL+1}] . . . [V_{ARGL+R}]

Here V_{ARGi} is the simpler string substituting ARG_i. The result can be seen in (9):

(9) [The chairman]$_{VARG1}$ [likes]$_{pivot}$ [deals]$_{VARG2}$.

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 (10):

(10) El presidente tiene gusto de repartos.

This translation allows us (i) to extract the translation of the pivot (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 (10), we replace the arguments by previously defined Static Substitution Variables. E.g. in (7), 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 in (10), we interrupt the decomposition and have the entire input string (4) 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, 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 ARG2 in (7) would be the a simplified version of ARG1 followed by the pivot: *The chairman likes*, the translation of which is determined at runtime, as in (11):

(11) [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 (12):

(12) [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 the previous subsection, 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 (4), the entire decomposition and recombination leads to an improvement in translation quality compared to the original output by Systran in (5), as is shown in (13):

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

4 Experimental setup

The EBMT system used in our experiments made use of the Marker-Based methods described in Section 2.1 to extract the chunk-level lexicon (Armstrong et al., 2006). For English we used information from the CELEX lexicon to create a list of marker words used during segmentation and alignment. The marker word list for Spanish was created by merging two stop-word lists generously supplied by colleagues at the Polytechnic University of Catalunya and the University of Barcelona.

After chunking, the resulting source and target marker chunks were aligned using a best-first dynamic programming algorithm, employing chunk position, word probability, marker tag and cognate information to determine subsentential links between sentence pairs.

In addition to these chunk alignments, we used statistical techniques to extract a high quality word-level lexicon (which in turn was used during the chunk alignment process). Following the refined alignment method of (Och and Ney, 2003), we used the GIZA++ statistical word alignment tool[2] to perform source-target and target-source word alignment. The resulting 'refined' word alignment set was then passed along with the chunk database to the system decoder (for the results reported in this paper we used the Pharaoh phrase-based decoder (Koehn, 2004)).

For training the EBMT system we made use of a subsection of the English-Spanish section of the Europarl corpus (Koehn, 2005). The corpus was filtered based on sentence length (maximum sentence length set at 40 words for Spanish and English) and relative sentence length ratio (a relative sentence length ratio of 1.5 was used), resulting in 958K English-Spanish sentence pairs.

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[3]; 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 EBMT system. On the one hand, the TransBooster decomposition algorithm performs better on 'perfectly' parse-annotated sentences from the Penn II Treebank than on the output produced by a statistical parser as (Bikel, 2002), which introduces a certain amount of noise. On the other hand, the EBMT model 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

In what follows, we present results of an automatic evaluation using BLEU and NIST against the two 800-sentence test sets introduced in Section 4. We then conduct a manual evaluation of a random sample of 100 sentences from the Europarl test set, chosen from those sentences where the output of TransBooster differed from the baseline translation. Finally, we analyse the most common differences and provide a number of example translations.

[2] http://www.fjoch.com/GIZA++.html
[3] http://www.cis.upenn.edu/~treebank/

5.1 Automatic evalution

The automatic evaluation results show that TransBooster outperforms the baseline EBMT system for both test sets. The evaluation was conducted after removing punctuation from the reference and translated texts, and, in the case of Europarl test set, after removing 59 sentences containing hyphenated compounds that have been incorrectly parsed by the Bikel parser, in effect introducing sentence-level errors in TransBooster processing.

Europarl	BLEU	NIST
EBMT	0.2111	5.9243
TransBooster	0.2134	5.9342
Percent of Baseline	101%	100.2%

Table 1. Results for EBMT versus TransBooster on 741-sentence test set from Europarl.

Wall Street Journal	BLEU	NIST
EBMT	0.1098	4.9081
TransBooster	0.1140	4.9321
Percent of Baseline	103.8%	100.5%

Table 2. Results for EBMT versus TransBooster on 800-sentence test set from Penn II Treebank.

5.2 Manual evaluation

The scale of improvement in translation quality is not completely reflected by n-gram measures such as BLEU and NIST, especially as the comparison is carried out against a single reference translation in both cases. In a preliminary manual evaluation, we randomly extracted 100 sentences from the Europarl test set, and compared their baseline translation with that assisted by TransBooster. This evaluation of translation quality was conducted by a native Spanish speaker fluent in English. The judge evaluated the two translations with respect to fluency and accuracy. In contrast to the generally used techniques, we used a relative scoring scale instead of the absolute one, i.e. the judge decided which of the two translation (if any) was better in terms of accuracy and which (if any) was better in terms of fluency.[4]

According to the evaluation, out of the 100 sentences, TransBooster improved the fluency of translation in 55% of the cases, and the accuracy of translation in 53% of the cases.

The improvements can be seen mainly in word order and lexical selection. Below we present examples of improved sentences and provide short analyses of the improvements achieved.

Example 1

Source: **women have decided** that they wish to work, that they wish to make their work compatible with their family life.

EBMT: **hemos decidido** su deseo de trabajar, su deseo de hacer su trabajo compatible con su vida familiar. **empresarias**

TransBooster: **mujeres han decidido** su deseo de trabajar, su deseo de hacer su trabajo compatible con su vida familiar.

Analysis: word order and lexical selection for *women have decided*

Example 2

Source: if this global warming **continues**, then **part of the territory of the eu member states** will become sea or desert.

EBMT: si esto **continúa** calentamiento global, tanto **dentro del territorio de los estados miembros** tendrán tornarse altamar o desértico

TransBooster: si esto calentamiento global **perdurará**, entonces **parte del territorio de los estados miembros de la unión europea** tendrán tornarse altamar o desértico

Analysis: word order for *continues*; lexical selection for *part of the territory of the eu member states*

[4] This relative scale was decided upon following the discussion at the SMT workshop at HLT-NAACL 2006, where the participants suggested that the relative scores would be more useful to comparing two or more MT systems, since with the typical absolute scale (1 to 5) the judges tend to choose the "safe" middle value of 3, neglecting smaller but still important differences between translations.

Example 3

Source: an **entirely** new feature **of the financial regulation** is the inclusion of rules such as those on the awarding of contracts and on financial aid.

EBMT: un nuevo rasgo es la inclusión de las normas sobre la adjudicación de contratos y de ayuda financiera **enteramente el reglamento financiero** tal como los

TransBooster: una **completamente** nueva característica **del reglamento financiero** es la inclusión de las normas sobre la adjudicación de contratos y ayuda financiera por tales como estos

Analysis: word order for *of the financial regulation* and *entirely*

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

Our current work involves conducting a fuller and more extensive manual evaluation of the results.

6 Conclusion

We have shown that example-based machine translation improves when we add a wrapper 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 that have a higher chance of being correctly translated. The resulting translations are recomposed into target language sentences.

The advantage of TransBooster over other methods is that it is universal in application, being able to work with various MT systems, and that the syntactic information it uses is linguistically motivated. We have shown in our experiment that the EBMT model coupled with TransBooster achieves an improvement of up to 3.8% in BLEU and 0.5% in NIST scores, and that the scale of the improvement is not properly reflected by n-gram based automatic evaluation. In the human evaluation, we show that TransBooster provides an improvement in fluency in 55% of the cases and in accuracy in 53% of the cases.

Acknowledgements

This work was made possible by Enterprise Ireland grant #SC/2003/0282 and an IRCSET PhD fellowship award. We would like to thank the reviewers for their insightful comments which served to improve this paper.

References

Armstrong, S., Flanagan, M., Graham, Y., Groves, D., Mellebeek, B., Morrissey, S., Stroppa, N. & Way, A. (2006). MaTrEx: Machine Translation Using Examples. *TC-STAR OpenLab on Speech Translation.* Trento, Italy.

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

Brown, R. (1999). Adding Linguistic Knowledge to a Lexical Example-based Translation System. In *Proceedings of TMI-99* (pp. 22-32). Chester, England.

Gough, N. (2005). *Example-Based Machine Translation Using the Marker Hypothesis*. PhD thesis, Dublin City University, Dublin, Ireland.

Gough, N. & Way, A. (2004). Robust Large-Scale EBMT with Marker-Based Segmentation. In *Proceedings of TMI-04* (pp. 95-104). Baltimore, MD.

Green, T. (1979). The Necessity of Syntax Markers. Two experiments with artificial languages. In *Journal of Verbal Learning and Behavior*, 18 (pp. 481-496).

Hearne, M. & Way, A. (2003). Seeing the Wood for the Trees: Data-Oriented Translation. In *Proceedings of the IX MT Summit* (pp. 165-172). New Orleans, LA.

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

Koehn, P. (2004). Pharaoh: A Beam Search Decoder for Phrase-based Statistical Machine Translation Models. In *Machine translation: From real users to research. 6th Conference of the Association for Machine Translation in the Americas (AMTA 2004)* (pp. 115-124). Georgetown University, Washington DC.

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

Koehn, P., Och, F., & Marcu, D. (2003). Statistical Phrase-based Translation. In *Proceedings of HLT-NAACL 2003* (pp. 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* (pp. 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* (pp. 114-119).

Mellebeek, B., Khasin, A., Owczarzak, K., Van Genabith, J., & Way, A. (2005a). Improving Online Machine Translation Systems. In *Proceedings of the X MT Summit* (pp. 290-297). Phuket, Thailand.

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

Mellebeek, B., Owczarzak, K., Groves, D., Van Genabith, J., & Way, A. (2006a). A Syntactic Skeleton for Statistical Machine Translation. *Proceedings of EAMT 2006* (pp. 195-202). Oslo, Norway.

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

Watanabe, H., Kurohashi, S., & Aramaki, E. (2003). Finding translation patterns from paired source and target dependency structures. In M. Carl and A.Way, editors, *Recent Advances in Example-Based Machine Translation* (pp. 397-420). Kluwer Academic Publishers, Dordrecht, The Netherlands.