TransBooster: Boosting the Performance of Wide-Coverage Machine Translation Systems

Bart Mellebeek, Anna Khasin, Josef Van Genabith and Andy Way

National Centre for Language Technology
School of Computing
Dublin City University, Dublin 9, Ireland.
{mellebeek,akhasin,josef,away}@computing.dcu.ie

Abstract. We propose the design, implementation and evaluation of a novel and modular approach to boost the translation performance of existing, wide-coverage, freely available machine translation systems based on reliable and fast automatic decomposition of the translation input and corresponding composition of translation output. We provide details of our method, and experimental results compared to the MT systems SYSTRAN and Logomedia. While many avenues for further experimentation remain, to date we fall just behind the baseline systems on the full 800-sentence testset, but in certain cases our method causes the translation quality obtained via the MT systems to improve.

1. Introduction

A significant number of freely available, commercial, wide-coverage machine translation (MT) systems populate the market, including SDL International's Enterprise Translation Server, Reverso by Softissimo, Logomedia, Promt and perhaps best known BabelFish by AltaVista.

Some of these systems are based on 1st generation 'word-for-word' direct translation technology, and share a number of characteristics: (i) they are designed to translate wide-coverage, general language material; (ii) they are robust; and (iii) they perform comparatively limited linguistic analysis. These points are interrelated and bear further elaboration. Detailed automatic linguistic analysis of translation input is potentially costly (both in terms of processing time and required lingware such as computational grammars etc.), and in the past has often been inversely related to coverage and robustness. In other words, the more detailed the linguistic analysis, the smaller the coverage of the system, and conversely, the wider the coverage, the less detailed the linguistic analysis. This has led commercial, wide-coverage MT systems to concentrate on 'linguistics-lite', robust design principles. In order to analyse translation input, they often consider only a limited linguistic context. A consequence of this is that existing commercial systems are much stronger when translating shorter sentences than they are on longer, more complex input. The reason behind this is simple: the longer the input sentence to be translated, the more likely that the automatic translation system will be led astray by the complexities in the source and target languages.

We contend that better performance in terms of output quality can be achieved than these systems can obtain by processing the texts that they are required to translate at any one time into smaller chunks.

Consider the example in (1):

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

A reasonable translation into German (established by a human translator) is:

(2) Der Vorsitzende, ein langfristiger Rivale von Bill Gates, mag schnelle und vertrauliche Abkommen.

However, the translation produced by the *BabelFish* MT system is *(3)*:

(3) Der Vorsitzende, ein langfristiger Rivale von Bill Gates, Gleiche fasten und vertrauliche Abkommen.

This involves a significant distortion of the input proposition, almost to the point of rendering it unrecognisable. The problem is that the English verb *likes* is mistranslated as a noun (*Gleiche*) and the adjective *fast* is completely misrecognised as a verb *fasten* ('to fast') .

Contrast what happens if you feed *BabelFish* the shorter sentences in (4):

(4) a. *The chairman likes deals* → *Der Vorsitzende mag Abkommen*
b. *The chairman likes fast deals* → *Der Vorsitzende mag schnelle Abkommen*

Both German strings in (4) are perfectly acceptable translations of the English input and constitute no errors.

This small set of translation examples is indicative of a general trend: that commercially available, wide-coverage MT systems tend to be much better at translating short and simple input. They perform much worse on longer strings, as the extra context provided gives ample opportunity for mistakes to be made.

In this paper, we present our method which takes long input strings from the Penn-II Treebank and breaks them down recursively into smaller and simpler constituents, and translates those shorter parts individually. At the same time, we keep track of where those individual parts fit into the overall translation in order to stitch together the translation result for the entire input string. It is important to note that throughout the process, the MT engine *itself* does all the translation: we are essentially helping the system work to the best of its ability so as to generate better translations than would otherwise have been produced to the benefit of the end user. We use *SYSTRAN*[1] because of its widespread use in the industry and *Logomedia*[2] since it was deemed the better of the three online MT systems tested in (Way & Gough, 2003). Accordingly, we have set ourselves a rather challenging task: we anticipate that the poorer the MT engine, the larger the increase in translation quality to be seen from incorporating the method described here.

The remainder of this paper is organised as follows: in section 2, we provide details of related research. In section 3, we illustrate the approach we have taken to date. Section 4 shows a worked-out sample sentence. Section 5 includes the results of a number of experiments we have carried out on a testset of 800 sentences randomly extracted from Section 23 of the Penn-II Treebank, and translated by native speakers into Spanish. We provide both automatic and human evaluations of translation quality, using *SYSTRAN* and *Logomedia* as baseline systems. In section 6, we note a number of possible improvements we wish to carry out in further research, and finally we conclude.

2. Related Research

The use of on-line systems is the biggest growth area in the use of MT: people are translating web pages (an area where MT provides *the* solution, as human translation of pages which need to be continually updated is not feasible) or communicating with one another in their own languages via email, using on-line MT systems as the translation engine.

Surprisingly, however, we are aware of very little research that has been carried out to try and investigate (a) how such systems work, and (b) how their obvious faults might be improved. The main point, of course, is that engines such as BabelFish are 'black box' systems, where any lexical and structural rules are hidden from the user; the only way to figure out how the system is working is to compare the input strings against the generated translations.

(Pérez-Ortiz & Forcada, 2001) demonstrate a laboratory experiment they created in order to show students new to MT that these on-line systems are rather more sophisticated than what they term a 'Model 0' MT system, a basic word-for-word version of these on-line engines. In so doing the students infer that by iteratively providing the MT system with more and more context, certain rule-based processing is apparent.

As to seeking to improve on the output generated by such systems, the only previous (yet unpublished) research that we know of took place at the University of Leuven in the late-80s. Researchers experimented with a pre-processing system named 'Tarzan' in which a human translator identified certain clearly defined syntactic units in the input sentence which could be replaced by a syntactically similar

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

placeholder for the purposes of simplifying the task of MT.

3. Our Current Approach

3.1. Overview

In the first phase of this project, we have used the pre-parsed sentences in the WSJ section of the Penn-II Treebank (Marcus et al., 1994) as input to our decomposition algorithm. A further line of research will involve the possible use of statistical parsing techniques to produce this input from previously unseen data.

In order to prepare a Penn-II input sentence for translation with TransBooster, the tree for that string is flattened into a simpler structure consisting of a pivot (meaningful head) and a number of satellites (arguments and adjuncts). The satellites are then replaced by syntactically similar strings, the translations of which are known in advance, as in (5)

(5) *SL:* $[SAT_1]$ ... $[SAT_l]$ *pivot* $[SAT_{l+1}]$ ... $[SAT_{l+r}]$

l = number of satellites to left of pivot r = number of satellites to right of pivot

The string SL is then submitted to the client MT system which outputs TL.

(6) *TL:* $[SAT'_1]$ ... $[SAT'_l]$ *pivot'* $[SAT'_{l+1}]$ ... $[SAT'_{l+r}]$

Note that the position of the translation SAT'_i does not necessarily have to be identical to the position of the constituent SAT_i in the source. We proceed to retrieve the translation of the pivot as well as the placement of each of the satellites. This process is applied recursively to each satellite found, after which the retrieved partial translations are recombined to yield the final target string corresponding to the input sentence.

We will extend each point of this process in the subsequent sections and illustrate it with an example.

3.2. Flattening Penn-II trees into TransBooster trees

Consider the Penn-II tree of example (1):

(7) *(S (NP-SBJ (NP (DT the) (NN chairman)) (, ,) (NP (NP (DT a) (JJ long-time) (NN rival)) (PP (IN of) (NP (NNP Bill) (NNP Gates)))) (, ,)) (VP (VBZ likes) (NP (ADJP (JJ fast) (CC and) (JJ confidential)) (NNS deals))))*

After finding the pivot '*likes*' (explained in section 3.3) and replacing the arguments '*the chairman, a long-time rival of Bill Gates*' and '*fast and confidential deals*' by adequate substitution variables (explained in section 3.5), we obtain the following flattened structure:

(8) *(S (NP-SBJ The man) (VBZ likes) (NP dogs))*

3.3. Finding the Pivot

The pivot is most often the head terminal of the Penn-II node currently being examined. In certain cases in English, in addition to the head, some of its rightmost neighbours are used in the construction of the pivot, where we consider it too dangerous to translate either part out of context. An obvious example is the use of auxiliaries, as is shown in (9).

(9) *(VP (MD might) (VP (VB have) (S (NP-SBJ (-NONE- *-2)) (VP (TO to) (VP (VB buy) (NP (NP (DT a) (JJ large) (NN quantity)) (PP (IN of) (NP (NN sugar)))))))))*

Here the found pivot is '*might have to buy*'.

Another example would be an ADJP whose head dominates a PP, as in (10).

(10) *(ADJP (JJ close) (PP (TO to) (NP (DT the) (NN utility) (NN industry))))*

Here the found pivot is '*close to*'.

In the initial experiments presented here, only contiguous pivots have been considered. In ongoing work, we intend to incorporate non-contiguous pivots in both source and target languages. Phrasal verbs and verbs with auxiliaries can be non-contiguous pivots in the presence of intervening material.

One of the pivot search parameters is the maximum length L of the pivot. If a head node N with L words or less in its coverage is arrived at during pivot search, the node N in its entirety is taken to be the pivot. If, on the other hand, the head node N contains too many leaf nodes (>L), we consider the head node N' of node N

to be a pivot candidate, and so on, until a head with L words or less in its coverage is found. This parameter allows us to experiment with varying maximum pivot lengths. Until now, the best results have been achieved for L = 4.

3.4. Finding Arguments and Adjuncts in the Source Language

We have explained how the strings submitted to the MT system comprise pivots, arguments, and adjuncts. We broaden the traditional notion of the term 'argument' to those nodes that are required for the correct (or, at any rate, safe) translation of the parent node. The distinction between arguments and adjuncts is essential, since nodes labelled as adjuncts can be safely omitted in the SL string that we submit to the client MT system.

For example, in (1) a substitution of the arguments '*the chairman, a long-time rival of Bill Gates*' and '*fast and confidential deals*' has to be present in the string submitted to the client MT system in order to retrieve a correct translation of pivot '*likes*'. On the other hand, when treating '*the chairman, a long-time rival of Bill Gates*', the apposition '*a long-time rival*' can be safely left out in the string submitted to the MT system. This is shown in more detail in the example in section 4. The omission of adjuncts is a simple and safe method to reduce the complexity of the SL candidate strings. Additional strategies for reducing the complexity of a sentence involve substituting simpler but syntactically similar elements for constituents (as explained in the following section) and are more hazardous.

In the current implementation, in cases of doubt we have veered in favour of labelling nodes as arguments. We continue to experiment to see whether better translations can be obtained by labelling nodes as arguments only when we can be (reasonably) sure that they are indeed required by the local head.

The procedure used for argument/adjunct location is an adapted version of Hockenmaier's algorithm for CCG (Hockenmaier, 2003). The nodes we label as arguments include all the nodes Hockenmaier labels as arguments together with some of the nodes (e.g. VP children of S where S is headed by a modal verb; quantitative adjectives) which she describes as adjuncts. In ongoing research, we wish to compare this procedure with the annotation of Penn-II nodes with LFG functional information (Cahill et al., 2004).

3.5. Substitution Variables and Skeletons

When trying to find an appropriate substitution variable for a satellite, we have to take into account a trade-off between accuracy and retrievability. On the one hand, non-word strings and certain acronyms are easy to retrieve because their translation is known in advance, but often they don't have the necessary syntactic properties to ensure a correct translation of the pivot. On the other hand, substitution variables that comprise the real head of the satellite that they substitute for are very accurate and will only in rare cases distort the translation of the pivot, but their translation is much more difficult to retrieve.

To confirm the low accuracy of non-word string substitution variables, we experimented with different kinds of substitution variables for the most frequent verb subcategorisation frames in the Penn-II Treebank (Cahill et al., 2004). We chose verbs belonging to the 10 most frequent subcategorization frames (8 in the active voice and 2 in the passive voice), so as to be able to handle the most frequently occurring syntactic contexts, and extracted the sentences in the Treebank which contained those verbs. This gave us 6559 frame-verb lemma pairs, for each of which we made test sentences with dummy arguments in the future and past tense. We replaced the arguments in these sentences with different kinds of substitution variables, ranging from non-word strings to syntactically similar constituents, and had these sentences translated by 4 MT systems (SYSTRAN, Logomedia, Promt[3], and SDL[4]) into Spanish and German. We used a string comparison script to automatically check the 262360 obtained translations for the correctness of the location of the arguments in the target language and for the quality of the pivot translation. Although the results are dependent on the sub-

[3] http://www.online-translator.com/default.asp?lang=en

[4] http://www.freetranslation.com/

categorisation frame used, MT system and language pair, the overall results of the experiment confirmed our expectation that substitution variables with a syntactic structure similar to the one of the substituted constituent outperform simple non-word strings.

Instead of taking an extreme position in the trade-off between accuracy and retrievability, we have chosen to adopt a middle course: in order to find the position of the satellites in the target language, we replace each of them with a substitution variable with a syntactic structure similar (but not identical) to the satellites it replaces and whose possible translations are known beforehand in most cases. For example, a simple NP as '*the man*' can replace certain NPs in singular, or simple clauses such as '*that the man was sleeping*' can be substituted for a more complex SBAR.

Substitution variables are not only used to find the location of the translated satellite in the target language. Their second function is to embed the pivot in a simplified context which we hope leads to an improvement in its translation. We call the string consisting of the pivot and its arguments, replaced by substitution variables, the 'argument skeleton'.[5] For example, the sentence in (1) takes as an argument skeleton the string in (11):

(11) The man likes dogs.

We retrieve the translation of the pivot by submitting this skeleton to the MT system and subtracting the known translations of the substitution variables. For example, translating the argument skeleton in (11) yields

(12) Der Mann mag Hunde.

If we subtract the known translations '*Der Mann*' and '*Hunde*', we obtain the translation '*mag*' for the pivot '*likes*'.

As a safeguard, we verify that the retrieved translation of the pivot is present in the translation of a 'pivot skeleton', which consists of the original source language string from which all adjuncts have been previously stripped. If our candidate translation of the pivot is not found in the translation of the pivot skeleton, the algorithm backs off to allow the MT system to translate the entire current node as is. Consider for example the pivot skeleton of the sentence in (1):

(13) *pivot skeleton = 'The chairman likes deals.'* → *'Der Vorsitzende mag Abkommen.'*

The found pivot translation '*mag*' is present in the translation of the pivot skeleton, so we continue the process and focus now on the translation of the satellites.

If the translation of the substitution variables cannot be found in the target language, the same order of arguments and adjuncts is assumed as in the source language. This is obviously very simplistic, and a modicum of linguistic knowledge about how the target language relates to the source would improve the target word order in those cases. This remains an avenue for further investigation.

3.6. Translation of Satellites

A satellite is considered to be ready for translation if the number of its leaf nodes is less than a predefined threshold N (the optimal N is established empirically and may vary according to the MT system and language pair). The satellites are then translated in a predefined template (derived based on the syntactic context of each satellite in its parsed and tagged Penn-II representation), and inserted where their replacements appear in the appropriate skeleton. If the number of leaf nodes of the satellite exceeds the threshold, the process is repeated recursively for the satellite in question.

In our example, in order to retrieve the correct translation of '*fast and confidential deals*', we have to insert this constituent into a template that will force it to be interpreted as a direct object. One of these templates might be the string '*The man sees*', which in a majority of cases will translate into the string '*Der Mann sieht*', as in (14):

(14) ***[The man sees]*** *fast and confidential deals.* ***[Der Mann sieht]*** *schnelle und vertrauliche Abkommen.*

[5] In a similar way 'adjunct skeletons' comprising the argument skeleton together with the substitution variables for the adjuncts inserted in the appropriate positions are used to retrieve the position of all the adjuncts.

In case the translation of the template gets distorted and cannot be retrieved, the satellite is translated without context.

3.7. Deriving the Translation

It is easy to see how the above described process can be applied recursively. If a node contains fewer or the same number of leaf nodes than the predefined threshold N mentioned in the previous section, that node is translated in its entirety (embedded in a context template that mimicks the original syntactic environment, if necessary). At this moment, we obtain the best results for N = 4. If the node contains more than N leaf nodes, we apply our decomposition process to each of its satellites, and so on, until all found satellites are considered small enough for translation. In the final recomposition step the translations of the pivots and satellites are recombined to yield the translation of the original input sentence.

4. Worked-out example

In this section, we will illustrate the entire process on the example sentence in (1)

> *'The chairman, a long-time rival of Bill Gates, likes fast and confidential deals'*

Algorithm:

```
QUEUE = {S}
While (QUEUE not empty) {
    Node N = shift QUEUE;
    If (# leaf nodes of N <= 4) {
        translate N in context;
    }
    else {
          find pivot N;
find satellites N;
          substitute satellites;
          build skeleton(s);
          translate skeleton(s);
          find translation pivot;
          if (translation pivot not OK) {
 translate N in context;
           break;
          }
          find location of translation satellites;

          add satellites to QUEUE;
 }
Recompose translations;
```

Input to algorithm =

(S (NP-SBJ (NP (DT The) (NN chairman)) (, ,) (NP (NP (DT a) (JJ long-time) (NN rival)) (PP (IN of) (NP (NNP Bill) (NNP Gates)))) (, ,)) (VP (VBZ likes) (NP (ADJP (JJ fast) (CC and) (JJ confidential)) (NNS deals))))

QUEUE = {S}

Step 1:

- S contains more than 4 leaf nodes → not ready for translation → decompose
- Find pivot S
  pivot = 'likes'
- find satellites S
  ARG1 = 'The chairman, a long-time rival of Bill Gates'
  ARG2 = 'fast and confidential deals.'
- substitute satellites
  ARG1_subst = 'The man'
  ARG2_subst = 'dogs'
- build skeleton(s)
  arg. skel = 'The man likes dogs.'
- translate skeleton(s)
  trans. arg. skel. = 'Der Mann mag Hunde.'
- find translation pivot
  trans. pivot = 'mag'
- pivot skel = 'The chairman likes deals.'
  trans pivot skel = 'Der Vorsitzende mag Abkommen.'
  'mag' is present in trans pivot skel → continue
- find location of translation satellites
  ARG1' left of pivot', ARG2' right of pivot'
- add satellites to QUEUE
  QUEUE = {ARG1, ARG2}

Step 2:

- ARG1 'The chairman, a long-time rival of Bill Gates' contains more than 4 leaf nodes → not ready for translation → decompose
- pivot = 'The chairman'
- ADJ11 = 'a long-time rival of Bill Gates'
- ...
- QUEUE = {ADJ11, ARG2}

Step 3:

- ADJ11 contains more than 4 leaf nodes → not ready for translation → decompose
- pivot = 'a long-time rival'
- ADJ111 = 'of Bill Gates'
- ...
- QUEUE = {ADJ111, ARG2}

Step 4:

- ADJ111 'of Bill Gates' contains less than 5 leaf nodes → ready for translation → translate in context
- 'The car of Bill Gates' → 'Das Auto von Bill Gates.'
- ADJ111' = 'von Bill Gates'
- QUEUE = {ARG2}

Step 5:

- ARG2 'fast and confidential deals' contains less than 5 leaf nodes → ready for translation → translate in context
- 'The man sees fast and confidential deals' → 'Der Mann sieht die schnellen und vertraulichen Abkommen.'
- ARG2' = 'die schnellen und vertraulichen Abkommen.'
- QUEUE = {}

Step 6:

- Recompose translation:
  'Der Vorsitzende, ein langfristiger Rivale von Bill Gates, mag die schnellen und vertraulichen Abkommen.'
- Original translation by Babelfish:
  'Der Vorsitzende, ein langfristiger Rivale von Bill Gates, Gleiche fasten und vertrauliche Abkommen.'

5. Results and Evaluation

The effectiveness of our algorithm is measured against an 800-sentence testset (min. 1 word, max. 54 words, ave. 19.75 words) from Section 23 of the Penn-II Treebank using a range of automatic MT evaluation metrics. (The toolkit we used, *mteval*, is obtainable from http://www.nist.gov/speech/tests/mt/resources/scoring.htm.) Given the requirements of our other research projects, the testset comprises all sentences in the PARC-700 (Riezler et al., 2002) and DCU-105 (Cahill et al., 2004) testsets for LFG. Although our approach is largely language-independent, for practical purposes we use English→, Spanish as our evaluation language pair. Groups of 200 sentences from the testset were translated by four native speakers of Spanish, each of whom was a certified translator, in order to obtain a set of reference translations for use with the automatic evaluation metrics.

Tables 5.1 and 5.2 contain the test results for English→Spanish using Logomedia and SYSTRAN respectively.

MT system	Cutoff length	BLEU	NIST	GTM
Logomedia	-	0.310	7.342	0.574
TB-01	4	0.213	5.867	0.342
TB-08	4	0.309	7.322	0.566
TB-12	4	0.268	6.995	0.498

Table 5.1. Transbooster vs Logomedia

MT system	Cutoff length	BLEU	NIST	GTM
SYSTRAN	-	0.296	7.178	0.563
TB-08	4	0.290	7.104	0.549
TB-12	4	0.264	6.756	0.494

Table 5.2. Transbooster vs SYSTRAN

The baseline *Logomedia* system scored 0.31 BLEU (Papineni et al, 2002), 7.342 NIST (Doddington, 2002) and 57.4% F-Score using the GTM (Turian et al., 2003) on this testset for this language pair. The first version of TransBooster (TB-01) scored just 0.21 BLEU, 5.867 NIST, and 34.2% F-Score. Our best results (TB-08) for a 4-word cutoff length show scores of 0.309 BLEU, 7.32 NIST and 56.7% F-Score.[6] The improvements from our initial effort to these better figures are due to using enhanced substitution variables to embed translations of pivots, better pivot-finding routines and improving the addition of context in which to embed the translation of satellites.

The scores obtained by using *SYSTRAN* as our baseline system (Table 5.2) are comparable to the ones obtained by using *Logomedia*. Moreover, the scores for *Logomedia* and *SYSTRAN* on their own show *Logomedia* slightly outperforming *SYSTRAN*.

Due to our safety measure of backing off to the original translation of the sentence in case the translation of the pivot is not found in the translation of the pivot skeleton (cf. Section

[6] When the cutoff length (the number of leaf nodes below which we consider a node ready for translation) increases, all scores slightly improve. The implentation of further improvements will lead to fewer backoffs, which will make these results more meaningful.

3.5), we back off in 85% of the cases in version TB-08. Improvements in the pivot finding methods have reduced this backoff to 40% in our latest version, but have caused other errors (mainly due to a faulty substitution or context) rise to the surface, which explains the slight drop in performance. We are confident, though, that further enhancements (cf. Section 6) will lead us to improve on the baseline systems by isolating those routines which contribute positively to the automatic evaluation scores from those that cause these to deteriorate.

We also carried out a manual inspection of the translations obtained via the baseline systems and our method, and there are cases such as (15) and (16) where TransBooster's intervention caused translation quality to improve:

(15) [Source] 'Our goal is to create more programs with an individual identity,' says Paul Amos, CNN executive vice president for programming.

[LogoMedia] 'Nuestro objetivo es crear más programas con una identidad individual,' Paul Amos, Vicepresidente Ejecutivo de CNN para la programación dice.

[TransBooster] „Nuestro objetivo es crear más programas con una identidad individual , „Dice Paul Amos , Vicepresidente Ejecutivo de CNN para la programación.

The reduction of the arguments of 'says' by TransBooster forces Logomedia to keep the verb 'dice' ('says') and subject 'Paul Amos' together, which results in an improvement in word order.

(16) [Source] 'Some early selling is likely to stem from investors and portfolio managers who want to lock in this year's fat profits.'

[SYSTRAN] 'Algo temprano que vende es probable provenir a los inversionistas y a los encargados de lista que desean trabarse en beneficios gordos relativos a este año.'

[TransBooster] 'Una cierta venta temprana es para provenir probablemente a los inversionistas y a los encargados de lista que desean trabarse en beneficios gordos relativos a este año.'

The translation of *'Some early selling'* in a simplified context causes its translation by TransBooster (*'Una cierta venta temprana'*) to outperform the original translation by *SYSTRAN ('Algo temprano que vende')*

6. Improvements

We expect that improvements to the labelling of nodes as adjuncts and arguments, involving the refinement of the syntactic contexts handled, will reduce the error rate of TransBooster in two ways: firstly, arguments which are currently mislabelled as adjuncts will no longer be omitted from the (argument) skeleton; secondly, with fewer nodes defaulting to argument status, the argument skeletons will be less cluttered than they are now. This will allow the baseline MT systems to do what we think they do best, namely process a concise, syntactically simple skeleton with a reasonable expectation of a good translation. We expect further improvements from incorporating a named entity recogniser into the algorithm, either by creating one ourselves via the Penn-II tags for nouns, or by incorporating an independently developed module.

Furthermore, more elaborate variable-substitution and context-generation routines are expected to lead to a reduction in the number of cases when the translations of constituents cannot be found in the respective skeletons.

A refinement to the matching process of the translations of substition variables may include matching stems (rather than surface forms, as at present). This is expected to lead to more matches.

Handling non-contiguous pivots in the source and target will further extend the number of syntactic contexts handled adequately by TransBooster.

7. Concluding Remarks

The translation quality obtained from on-line MT systems deteriorates with longer input strings. We have presented a method where we recursively break down sentences from the Penn-II Treebank into smaller and smaller constituents, and confront the MT system with these shorter

sub-strings. We keep track of where those individual parts fit into the overall translation in order to stitch together the translation result for the entire input string. Throughout the process the commercial MT engine does all the translation *itself*: our method helps the system to improve its own output translations.

To date the quality obtained via our approach falls just below the baseline systems *SYSTRAN* and *Logomedia*, with a BLEU score of 0.268 against the best baseline *Logomedia*'s 0.310, backing off in 40% of cases. We have identified a number of research avenues which we feel will lead to further improvements, especially when we test against poorer systems. Furthermore, if we isolate those cases where our algorithm does produce better translations than the baseline systems and exclude cases where our intervention causes translation quality to deteriorate, then we expect to be able to improve the translation quality available from commercial, wide-coverage MT systems.

8. References

CAHILL, A., M. BURKE, R. O'DONOVAN, J. VAN GENABITH and A. WAY (2004) 'Long-Distance Dependency Resolution in Automatically Acquired Wide-Coverage PCFG-Based LFG Approximations'. In Proc. *42nd Annual Meeting of the Association for Computational Linguistics (ACL'04)*, Barcelona, Spain: 319–326.

DODDINGTON, George (2002). 'Automatic evaluation of machine translation quality using n-gram co-occurrence statistics'. In Proc. *Human Language Technology 2003: 3rd Meeting of the NAACL*, San Diego, CA.:128–132.

HOCKENMAIER, Julia (2003). 'Parsing with Generative models of Predicate-Argument Structure'. In Proc. *41st Annual Conference of the Association for Computational Linguistics (ACL'03)*, Sapporo, Japan: 359–366.

MARCUS, M., G. KIM, M.A. MARCINKIEWICZ, R. MACINTYRE, M. FERGUSON, K. KATZ and B. SCHASBERGER (1994). 'The Penn Treebank: Annotating Predicate Argument Structure'. In Proc. *1994 ARPA Human Language Technology Workshop*, Princeton, NJ: 110–115.

PAPINENI, K., S. ROUKOS, T. WARD and W-J. ZHU. 2002. BLEU: 'A Method for Automatic Evaluation of Machine Translation'. In Proc. *40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA.: 311–318.

PÉREZ-ORTIZ, J. & M. FORCADA (2001). 'Discovering Machine Translation Strategies: Beyond Word-for-Word Translation: a Laboratory Assignment'. *In* Proc. of the *Workshop on Teaching Machine Translation, MT Summit VIII*, Santiago de Compostela, Spain: 57–60.

RIEZLER, S., T. KING, R. KAPLAN, R. CROUCH, J. MAXWELL, and M. JOHNSON (2002). 'Parsing the Wall Street Journal using a Lexical-Functional Grammar and Discriminative Estimation Techniques'. In Proc. *40th Annual Meeting of the Association for Computational Linguistics (ACL'02)*, Philadelphia, PA.: 271–278.

TURIAN, J., L. SHEN. and D. MELAMED (2003). 'Evaluation of Machine Translation and its Evaluation'. *MT Summit IX*, New Orleans, LA.: 386–393.

WAY, A. and N. GOUGH (2003). '*wEBMT*: Developing and Validating an Example-Based Machine Translation System using the World Wide Web'. *Computational Linguistics* **29** (3): 421–457