TransAhead: A Computer-Assisted Translation and Writing Tool

[*]Chung-chi Huang [+]Ping-che Yang [**]Keh-jiann Chen [++]Jason S. Chang

[*]ISA, NTHU, HsinChu, Taiwan, R.O.C. [**]IIS, Academia Sinica, Taipei, Taiwan, R.O.C.
[+]III, Taipei, Taiwan, R.O.C. [++]CS, NTHU, HsinChu, Taiwan, R.O.C.
{[*]u901571,[+]maciaclark,[++]jason.jschang}@gmail.com;[**]kchen@iis.sinica.edu.tw

Abstract

We introduce a method for learning to predict text completion given a source text and partial translation. In our approach, predictions are offered aimed at alleviating users' burden on lexical and grammar choices, and improving productivity. The method involves learning syntax-based phraseology and translation equivalents. At run-time, the source and its translation prefix are sliced into ngrams to generate and rank completion candidates, which are then displayed to users. We present a prototype writing assistant, TransAhead, that applies the method to computer-assisted translation and language learning. The preliminary results show that the method has great potentials in CAT and CALL with significant improvement in translation quality across users.

1 Introduction

More and more language workers and learners use the MT systems on the Web for information gathering and language learning. However, web translation systems typically offer top-1 translations (which are usually far from perfect) and hardly interact with the user.

Text translation could be achieved more interactively and effectively if a system considered translation as a collaborative between the machine generating suggestions and the user accepting or overriding on those suggestions, with the system adapting to the user's action.

Consider the source sentence "我們在完成這筆交易上扮演重要角色" (We play an important role in closing this deal). The best man-machine interaction is probably not the one used by typical existing MT systems. A good working environment might be a translation assistant that offers suggestions and gives the user direct control over the target text.

We present a system, TransAhead[1], that learns to predict and suggest lexical translations (and their grammatical patterns) likely to follow the ongoing translation of a source text, and adapts to the user's choices. Example responses of TransAhead to the source sentence "我們在完成這筆交易上扮演重要角色" and two partial translations are shown in Figure 1. The responses include text and grammatical patterns (in all-cap labels representing parts-of-speech). TransAhead determines and displays the probable subsequent grammatical constructions and partial translations in the form of parts-of-speech and words (e.g., "IN[*in*] VBG[*close*,…]" for keywords "play role" where lexical items in square brackets are lemmas of potential translations) in a pop-up. TransAhead learns these constructs and translations during training.

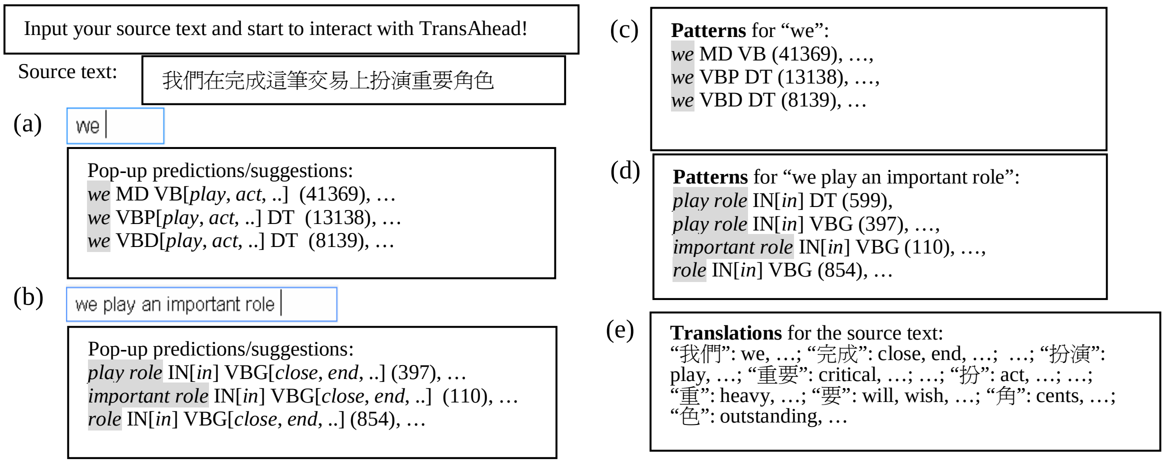
At run-time, TransAhead starts with a source sentence, and iterates with the user, making predictions on the grammar patterns and lexical translations, while adapting to the user's translation choices to resolve ambiguities in the source sentence related to word segmentation and word sense. In our prototype, TransAhead mediates between users and suggestion modules to translation quality and productivity.

2 Related Work

Computer Assisted Translation (CAT) has been an area of active research. We focus on offering suggestions during the translation process with an

[1] http://140.114.214.80/theSite/TransAhead/ (Chrome only)

2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pages 352–356,
Montréal, Canada, June 3-8, 2012. ©2012 Association for Computational Linguistics



Input your source text and start to interact with TransAhead!

Source text: 我們在完成這筆交易上扮演重要角色

(a) we

Pop-up predictions/suggestions:
we MD VB[*play*, *act*, ..] (41369), …
we VBP[*play*, *act*, ..] DT (13138), …
we VBD[*play*, *act*, ..] DT (8139), …

(b) we play an important role

Pop-up predictions/suggestions:
play role IN[*in*] VBG[*close*, *end*, ..] (397), …
important role IN[*in*] VBG[*close*, *end*, ..] (110), …
role IN[*in*] VBG[*close*, *end*, ..] (854), …

(c) **Patterns** for "we":
we MD VB (41369), …,
we VBP DT (13138), …,
we VBD DT (8139), …

(d) **Patterns** for "we play an important role":
play role IN[*in*] DT (599),
play role IN[*in*] VBG (397), …,
important role IN[*in*] VBG (110), …,
role IN[*in*] VBG (854), …

(e) **Translations** for the source text:
"我們": we, …; "完成": close, end, …; …; "扮演": play, …; "重要": critical, …; …; "扮": act, …; …; "重": heavy, …; "要": will, wish, …; "角": cents, …; "色": outstanding, …

Figure 1. Example TransAhead responses to a source text under the translation (a) "we" and (b) "we play an important role". Note that the grammar/text predictions of (a) and (b) are not placed directly under the caret (current input focus) for space limit. (c) and (d) depict predominant grammar constructs which follow and (e) summarizes the confident translations of the source's character-based ngrams. The frequency of grammar pattern is shown in round brackets while the history (i.e., keyword) based on the user input is shown in shades.

emphasis on language learning. Specifically, our goal is to build a translation assistant to help translator (or learner-translator) with inline grammar help and translation. Unlike recent research focusing on professional (e.g., Brown and Nirenburg, 1990), we target on both professional and student translators.

More recently, interactive MT (IMT) systems have begun to shift the user's role from post-editing machine output to collaborating with the machine to produce the target text. Foster et al (2000) describe TransType, a pioneering system that supports next word predictions. Along the similar line, Koehn (2009) develops caitra which predicts and displays phrasal translation suggestions one phrase at a time. The main difference between their systems and TransAhead is that we also display grammar patterns to provide the general patterns of predicted translations so a student translator can learn and become more proficient.

Recent work has been done on using fully-fledged statistical MT systems to produce target hypotheses completing user-validated translation prefix in IMT paradigm. Barrachina et al. (2008) investigate the applicability of different MT kernels within IMT framework. Nepveu et al. (2004) and Ortiz-Martinez et al. (2011) further exploit user feedbacks for better IMT systems and user experience. Instead of triggered by user correction, our method is triggered by word delimiter and assists both translation and learning the target language.

In contrast to the previous CAT research, we present a writing assistant that suggests grammar constructs as well as lexical translations following users' partial translation, aiming to provide users with choice to ease mental burden and enhance performance.

3 The TransAhead System

3.1 Problem Statement

We focus on predicting a set of grammar patterns with lexical translations likely to follow the current partial target translation of a source text. The predictions will be examined by a human user directly. Not to overwhelm the user, our goal is to return a reasonable-sized set of predictions that contain suitable word choices and grammatical patterns to choose and learn from. Formally,

Problem Statement: We are given a target-language reference corpus C_t, a parallel corpus C_{st}, a source-language text S, and its translation prefix T_p. Our goal is to provide a set of predictions based on C_t and C_{st} likely to further translate S in terms of grammar and text. For this, we transform S and T_p into sets of ngrams such that the predominant grammar constructs with suitable translation options following T_p are likely to be acquired.

3.2 Learning to Find Pattern and Translation

In the training stage, we find and store syntax-based phraseological tendencies and translation pairs. These patterns and translations are intended to be used in a real-time system to respond to user input speedily.

First, we part of speech tag sentences in C_t. Using common phrase patterns (e.g., the **possessive noun** *one's* in "make up *one's* mind") seen in grammar books, we resort to parts-of-speech (POS) for syntactic generalization. Then, we build up inverted files of the words in C_t for the next stage (i.e., pattern grammar generation). Apart from sentence and position information, a word's lemma and POS are also recorded.

Subsequently, we use the procedure in Figure 2 to generate grammar patterns following any given sequence of words, either contiguous or skipped.

```
procedure PatternFinding($query$,$N$,$C_t$)
(1)  $interInvList$=findInvertedFile($w_1$ of $query$)
     for each word $w_i$ in $query$ except for $w_1$
(2)    $InvList$=findInvertedFile($w_i$)
(3a)   $newInterInvList$= $\phi$ ; $i$=1; $j$=1
(3b)   while $i$<=length($interInvList$) and $j$<=lengh($InvList$)
(3c)     if $interInvList$[$i$].SentNo==$InvList$[$j$].SentNo
(3d)       Insert($newInterInvList$, $interInvList$[$i$],$InvList$[$j$])
         else
(3e)       Move $i$,$j$ accordingly
(3f)   $interInvList$=$newInterInvList$
(4) $Usage$= $\phi$
    for each $element$ in $interInvList$
(5)    $Usage$+={PatternGrammarGeneration($element$,$C_t$)}
(6) Sort patterns in $Usage$ in descending order of frequency
(7) return the $N$ patterns in $Usage$ with highest frequency
```

Figure 2. Automatically generating pattern grammar.

The algorithm first identifies the sentences containing the given sequence of words, *query*. Iteratively, Step (3) performs an AND operation on the inverted file, *InvList*, of the current word w_i and *interInvList*, a previous intersected results.

After that, we analyze *query*'s syntax-based phraseology (Step (5)). For each *element* of the form ([wordPosi(w_1),…, wordPosi(w_n)], *sentence number*) denoting the positions of *query*'s words in the *sentence*, we generate grammar pattern involving replacing words in the sentence with POS tags and words in wordPosi(w_i) with lemmas, and extracting fixed-window [2] segments surrounding *query* from the transformed sentence. The result is a set of grammatical patterns (i.e., syntax-based phraseology) for the query. The procedure finally returns top N predominant syntactic patterns of the query. Such patterns characterizing the query's word usages in the spirit of pattern grammar in (Hunston and Francis, 2000) and are collected across the target language.

In the fourth and final stage, we exploit C_{st} for bilingual phrase acquisition, rather than a manual dictionary, to achieve better translation coverage and variety. We obtain phrase pairs through a number of steps, namely, leveraging IBM models for bidirectional word alignments, grow-diagonal-final heuristics to extract phrasal equivalences (Koehn et al., 2003).

3.3 Run-Time Grammar and Text Prediction

Once translation equivalents and phraseological tendencies are learned, they are stored for run-time reference. TransAhead then predicts/suggests the following grammar and text of a translation prefix given the source text using the procedure in Figure 3.

```
procedure MakePrediction($S$,$T_p$)
(1) Assign sliceNgram($S$) to {$s_i$}
(2) Assign sliceNgramWithPivot($T_p$) to {$t_j$}
(3) $TransOptions$=findTranslation({$s_i$},$T_p$)
(4) $GramOptions$=findPattern({$t_j$})
(5) Evaluate translation options in $TransOptions$
        and incorporate them into $GramOptions$
(6) Return $GramOptions$
```

Figure 3. Predicting pattern grammar and translations at run-time.

We first slice the source text S into character-level ngrams, represented by $\{s_i\}$. We also find the word-level ngrams of the translation prefix T_p. But this time we concentrate on the ngrams, may skipped, ending with the last word of T_p (i.e., pivoted on the last word) since these ngrams are most related to the subsequent grammar patterns.

Step (3) and (4) retrieve translations and patterns learned from Section 3.2. Step (3) acquires the target-language active vocabulary that may be used to translate the source. To alleviate the word boundary issue in MT (Ma et al. (2007)), the word boundary in our system is loosely decided. Initially, TransAhead non-deterministically segments the source text using character ngrams for translations and proceeds with collaborations with the user to obtain the segmentation for MT and to complete the translation. Note that T_p may reflect some translated segments, reducing the size of the active vocabulary, and that a user vocabulary of preference (due to users' domain knowledge or

[2] Inspired by (Gamon and Leacock, 2010).

errors of the system) may be exploited for better system performance. In addition, Step (4) extracts patterns preceding with the history ngrams of $\{t_j\}$.

In Step (5), we first evaluate and rank the translation candidates using linear combination:

$$\lambda_1 \times \left(P_1\left(t \mid s_i\right) + P_1\left(s_i \mid t\right)\right) + \lambda_2 \times P_2\left(t \mid T_p\right)$$

where λ_i is combination weight, P_1 and P_2 are translation and language model respectively, and t is one of the translation candidates under S and T_p. Subsequently, we incorporate the lemmatized translation candidates according to their ranks into suitable grammar constituents in *GramOptions*. For example, we would include “close” in pattern “*play role* IN[*in*] VBG” as “*play role* IN[*in*] VBG[*close*]”.

At last, the algorithm returns the representative grammar patterns with confident translations expected to follow the ongoing translation and further translate the source. This algorithm will be triggered by word delimiter to provide an interactive CAT and CALL environment. Figure 1 shows example responses of our working prototype.

4 Preliminary Results

In developing TransAhead, we used British National Corpus and Hong Kong Parallel Text as target-language reference corpus and parallel training corpus respectively, and deployed GENIA tagger for lemma and POS analyses.

To evaluate TransAhead in CAT and CALL, we introduced it to a class of 34 (Chinese) college freshmen learning English as foreign language. We designed TransAhead to be accessible and intuitive, so the user training tutorial took only one minute.

After the tutorial, the participants were asked to translate 15 Chinese texts from (Huang et al., 2011) (half with TransAhead assistance called experimental group, and the other without any system help whatsoever called control group). The evaluation results show that the experimental group achieved *much* better translation quality than the control group with an average BLEU score (Papineni et al., 2002) of **35.49** vs. 26.46. Admittedly, the MT system Google Translate produced translations with a higher BLEU score of 44.82.

Google Translate obviously has much more parallel training data and bilingual translation knowledge. No previous work in CAT uses Google Translate for comparison. Although there is a difference in average translation quality between the experimental TransAhead group and the Google Translate, it is not hard for us to notice the source sentences were better translated by language learners with the help of TransAhead. Take the sentence “我們在完成這筆交易上扮演重要角色” for example. A total of 90% of the participants in the experimental group produced more grammatical and fluent translations (see Figure 4) than that (“We conclude this transaction plays an important role”) by Google Translate.

1. we play(ed) a critical role in closing this/the deal.
2. we play(ed) a critical role in sealing this/the deal.
3. we play(ed) an important role in ending this/the deal.
4. we play(ed) an important role in closing this/the deal.

Figure 4. Example translations with TransAhead assistance.

Post-experiment surveys indicate that (a) the participants found Google Translate lack human-computer interaction while TransAhead is intuitive to collaborate with in translation/writing; (b) the participants found TransAhead grammar and translation predictions useful for their immediate task and for learning; (c) interactivity made the translation and language learning a fun process (like image tagging game of (von Ahn and Dabbish, 2004)) and the participants found TransAhead very recommendable and would like to use it again in future translation tasks.

5 Summary

We have introduced a method for learning to offer grammar and text predictions expected to assist the user in translation and writing. We have implemented and evaluated the method. The preliminary results are encouragingly promising. As for the further work, we intend to evaluate and improve our system further in learner productivity in terms of output quality, typing speed, and the amount of using certain keys such as *delete* and *backspace*.

Acknowledgement

This study is conducted under the “Project Digital Convergence Service Open Platform” of the Institute for Information Industry which is subsidized by the Ministry of Economy Affairs of the Republic of China.

References

S. Barrachina, O. Bender, F. Casacuberta, J. Civera, E. Cubel, S. Khadivi, A. Lagarda, H. Ney, J. Tomas, E. Vidal, and J.-M. Vilar. 2008. Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1): 3-28.

R. D. Brown and S. Nirenburg. 1990. Human-computer interaction for semantic disambiguation. In *Proceedings of COLING*, pages 42-47.

G. Foster, P. Langlais, E. Macklovitch, and G. Lapalme. 2002. TransType: text prediction for translators. In *Proceedings of ACL Demonstrations*, pages 93-94.

M. Gamon and C. Leacock. 2010. Search right and thou shalt find … using web queries for learner error detection. In *Proceedings of the NAACL Workshop*.

C.-C. Huang, M.-H. Chen, S.-T. Huang, H.-C. Liou, and J. S. Chang. 2011. GRASP: grammar- and syntax-based pattern-finder in CALL. In *Proceedings of ACL Workshop*.

S. Hunston and G. Francis. 2000. *Pattern Grammar: A Corpus-Driven Approach to the Lexical Grammar of English*. Amsterdam: John Benjamins.

P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proceedings of NAACL*.

P. Koehn. 2009. A web-based interactive computer aided translation tool. In *Proceedings of ACL*.

Y. Ma, N. Stroppa, and A. Way. 2007. Bootstrapping word alignment via word packing. In *Proceedings of ACL*.

L. Nepveu, G. Lapalme, P. Langlais, and G. Foster. 2004. Adaptive language and translation models for interactive machine translation. In *Proceedings of EMNLP*.

D. Ortiz-Martinez, L. A. Leiva, V. Alabau, I. Garcia-Varea, and F. Casacuberta. 2011. An interactive machine translation system with online learning. In *Proceedings of ACL System Demonstrations*, pages 68-73.

K. Papineni, S. Roukos, T. Ward, W.-J. Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In Proceedings of ACL, pages 311-318.

L. von Ahn and L. Dabbish. 2004. Labeling images with a computer game. In *Proceedings of CHI*.