Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

A New Input Method for Human Translators: Integrating Machine Translation Effectively and Imperceptibly

Guoping Huang, Jiajun Zhang, Yu Zhou and Chengqing Zong
National Laboratory of Pattern Recognition, CASIA, Beijing, P.R. China
{guoping.huang, jjzhang, yzhou, cqzong}@nlpr.ia.ac.cn

Abstract

Computer-aided translation (CAT) system is the most popular tool which helps human translators perform language translation efficiently. To further improve the efficiency, there is an increasing interest in applying the machine translation (MT) technology to upgrade CAT. Post-editing is a standard approach: human translators generate the translation by correcting MT outputs. In this paper, we propose a novel approach deeply integrating MT into CAT systems: a well-designed input method which makes full use of the knowledge adopted by MT systems, such as translation rules, decoding hypotheses and n-best translation lists. Our proposed approach allows human translators to focus on choosing better translation results with less time rather than just complete translation themselves. The extensive experiments demonstrate that our method saves more than 14% time and over 33% keystrokes, and it improves the translation quality as well by more than 3 absolute BLEU scores compared with the strong baseline, i.e., post-editing using Google Pinyin.

1 Introduction

Computer-aided translation (CAT) is a form of language translation in which a human translator uses a software to perform and facilitate the translation process. To further improve the translation efficiency, incorporating the technology of machine translation (MT), especially statistical machine translation (SMT), into the CAT tools has drawn more and more attention. Researchers have proposed many approaches, which can be divided into two types. One aims at developing interactive machine translation (IMT) systems (Foster, 2002; Barrachina et al., 2009). The other focuses on designing good post-editing (PE) systems (Koehn, 2009a, 2009b, 2012a; Carl et al., 2011; Zhechev, 2012; Green et al., 2014; Koehn et al., 2014).

In IMT, the core idea is to help an MT system dynamically generate the acceptable translation in a left to right manner through a series of timely interactions between human translators and the MT system. Human translators are required to observe the MT output and carefully revise the output if necessary. As far as we know, IMT is not accepted and used in any commercial CAT tools because of the high workload of the translation process.

In practice, post-editing is a standard and widely-used approach to apply the MT technology to upgrade the CAT system: human translators generate the translation by correcting the MT outputs. If the raw MT output is good enough, it will take little time for human translators to achieve the final acceptable translation. Considerable evidence has shown that human translators are more productive and the translation results are more accurate when post-editing is adopted (Carl et al., 2011; Koehn, 2012; Zhechev, 2012). In the real world, there are a number of CAT tools supporting post-editing, such as SDL Trados and MemoQ.

However, post-editing is far from perfect. There are two main challenges for post-editing in practice. For one thing, the low-quality of the MT results often makes a human translator puzzled and headachy to edit, and he/she would rather ignore the MT results and start translating from scratch. For another, many target languages (e.g., Chinese and Japanese) are written in complex character sets. In order to type these complex characters into the computer, human translators have to use a specially developed input method, such as Google Pinyin, which allows the users to input Chinese characters by entering phonetic spellings. In Chinese, a phonetic spelling usually matches dozens of Chinese characters. It leads to many more editing operations.

Fortunately, even if the MT output translation is terrible and is ignored by human translators in post-editing, it contains some perfect fragments. Therefore, it raises the question of how to take advantage of such fragments. Moreover, it is a fact that all the human translators translating other language texts into Chinese (or other languages with complex characters) need an input method. It inspires us to think why not integrate the MT technology into the input method so as to speed up the human translation process? Why not avoid the intensive interaction between human translators and the low-quality MT outputs by applying only the reliable information of the MT system imperceptibly into the input method?

To achieve these goals, we propose in this paper a novel approach deeply integrating MT into the CAT system: a well-designed input method named CoCat which makes full use of the knowledge adopted by the MT systems, such as

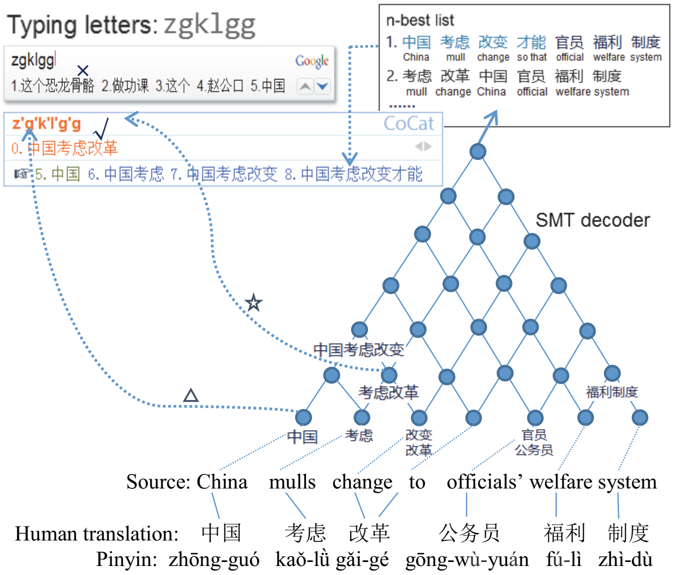


Figure 1: The overview of CoCat input method. In the SMT decoder, each blue node refers to a decoding span. The Chinese phrases below the nodes refer to the phrase translation rules and hypotheses during translation decoding.

translation rules, decoding hypotheses and n-best translation lists. First, we analyze and extract the useful information of the MT system, and transform them into features. Then, we extend the standard input method and design a log-linear model to incorporate multiple sources of features. Finally, we design an n-gram prediction model for the input method to further facilitate the human translators. Our proposed input method CoCat works well in both scenarios: translating from scratch and post-editing. Figure 1 demonstrates how the approach works in translation from English to Chinese. If the human translator adopts the standard Google Pinyin to perform translation from scratch, the abbreviated Chinese typing letters "*zgklgg*" (the acronym in Chinese Pinyin) cannot elicit the correct translation (left-top in Figure 1). Instead, our proposed input method CoCat can correctly decode the abbreviated letters "*zgklgg*" into the desired translation which is provided by the SMT system (right part in Figure 1).

In the experiments, our proposed input method has achieved remarkable results on professional human translation test. This paper makes the following contributions:

(1) The well designed input method CoCat can take full advantage of the useful information of the MT system. CoCat is the first input method to exploit depth information used by MT, such as translation rules and decoding hypotheses.

(2) CoCat can not only help human translators significantly save time and keystrokes, but also substantially improve the final translation quality.

(3) CoCat can collaborate with other CAT technologies, for example, post-editing. When CoCat is integrated with post-editing, it can further speed up the translation process.

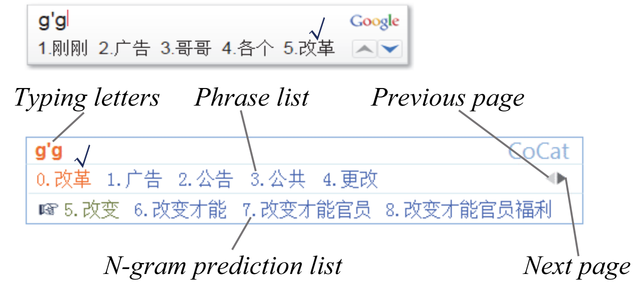


Figure 2: The comparison between Google Pinyin interface and CoCat interface.

2 CoCat Input Method

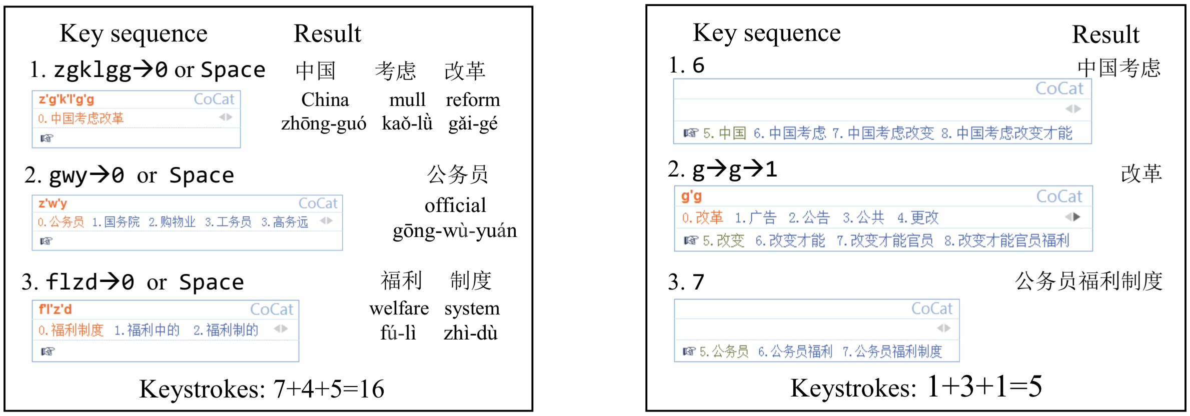
In this paper, we focus on English-to-Chinese translation, which is a typical case for translating texts into languages with non-alphabetic characters.

For a human translator, the translation process using CAT tools can be divided into three steps: (1) the human translator generates the initial translation in mind after analyzing the source sentence; (2) he/she uses an input method to type the initial translation ("Human translation" in Figure 1) into the text input box; (3) the translator repeatedly revises the entered target sentence until the translation is satisfactory.

After applying MT to CAT tools, the MT result will be evaluated and used in three ways according to its quality: (1) the MT output is perfect, and it happens to be the one the human translator expects; (2) the MT output is good but not perfect, and the human translator needs to do some minor modifications with post-editing; (3) the MT output is too bad, and the human translator will ignore it.

Usually, MT results greatly facilitate human translators when using post-editing. However, in most cases, the MT output is not good enough. There are only a few perfect fragments, and tedious modifications are required.

Based on the analysis above, we propose a well-designed input method called CoCat that provides a novel interactive approach by making full use of those perfect fragments. The interface of the CoCat is shown in Figure 2 with five phrases per page. This CoCat input method includes two novel CAT-oriented models: phrase generation model and n-gram prediction model. Figure 1 and Figure 2 illustrate the superiority of the two models compared with traditional input methods in three ways: (1) the phrase generation model generates new target phrases which the existing input methods cannot obtain. For example, typing letters "*zgklgg*" using Google Pinyin cannot lead to the correct phrase "中国考虑改革(China mulls to change)" as shown in Figure 1. (2) the phrase generation model can re-rank the target phrase list with additional features induced from the MT systems. For example, "改革(change)" is ranked first by CoCat among all the candidates for the typing letters "*gg*", while Google Pinyin will not bring the desired result as shown in Figure 2. (3) the n-gram prediction model can offer a list of translation suggestions. For example, it provides four potential phrases from index 5 "改变(change)" to 8 "改变才能官员福利" as shown in Figure 2.



(a) The key sequence without n-gram prediction model (b) The key sequence with n-gram prediction model

Figure 3: The comparison of key sequences without/with n-gram prediction model. The text is broken into chunks with the hypothesis that human translators prefer to directly select longer correct predictions.

As a summary, the proposed CoCat input method will improve the human translation process in two aspects: (1) human translators do not need to evaluate the MT output; (2) the new input method automatically speeds up the translation process by providing suggestions in a friendly way.

2.1 Phrase Generation Model

The user types phonetic spellings, which are called typing letters in this paper, to input a Chinese phrase. For example, to enter the target phrase "中国考虑改革(China mulls to change)", the translator usually types Chinese Pinyin "*zhongguokaolvgaige*". The input method automatically segments the typing letters into "*zhong'guo'kao'lv'gai'ge*" and finally converts it into a large candidate set of Chinese character strings, including "中国考虑改革", whose scores may rank low among other candidates. In Chinese, a phonetic spelling usually matches dozens of characters. So, it is very difficult for input methods to select the correct corresponding Chinese characters automatically.

Our CoCat input method tries to reduce the number of the typing letters and make the expected phrase rank as high as possible. For example, if we type the shorter "*zgklgg*", the correct result "中国考虑改革" can still rank first according to the context with the aid of the MT system in Figure 1. The smaller number of the typing letters, the faster the translating, the more time left for human translators to think, and the better the translation. However, all the existing Pinyin input methods, such as Google Pinyin, cannot decode "*zgklgg*" correctly.

For a given segmented Pinyin $y_1^n = y_1 y_2 \dots y_n$, the goal of conversion from Pinyin to Chinese characters is to find the most probable Chinese characters $h_1^n = h_1 h_2 \dots h_n$ from the candidates set H by maximizing $\Pr(h_1^n|y_1^n)$. Usually, y and h have the same length, y_i is a syllable of a Chinese character, and h_i is one of the characters that y_i responds to. To better integrate MT system, we design a new phrase generation model for CoCat using the log-linear model:

$$\hat{H}(y_1^n, \lambda_1^M) = \arg\max_H \{\sum_{m=1}^{M} \lambda_m f_m(h_1^n, y_1^n)\}$$

in which, λ_m denotes the weight of the corresponding feature, and M denotes the number of feature functions. The feature function set $\{f_m(h_1^n, y_1^n)\}$ includes typical features employed by the input methods, such as the word frequency, log-probabilities for the typing model $\Pr(y_1^n|h_1^n)$ and the language model $\Pr(h_1^n)$. In addition, the following three features which are induced from the MT systems are employed by the log-linear model:

(1) The feature function indicating whether the candidate is included in the translation rules corresponding to the current source sentence.

(2) The feature function indicating whether the candidate is included in the hypotheses during MT decoding.

(3) The feature function indicating whether the candidate is included in the n-best list of the MT result.

Decoding is performed by the CYK algorithm (Kasami, 1965; Younger, 1967), and a beam-search algorithm is employed to speed up the decoding.

To complete the translation task using CoCat, we need to type the key sequence as shown in Figure 3(a). We can get the correct result by typing letters "*zgklgg*" when we translate the sentence in Figure 1. It is because that the candidate substring "中国" is included in the MT translation rules (Δ in Figure 1) and another candidate substring "考虑改革" is contained in the MT translation hypotheses (☆ in Figure 1). "中国考虑改革" will be rewarded during pruning and re-ranking just as the step 1 in Figure 3(a) shows. We can achieve the same effect on other languages in a similar way.

2.2 N-gram Prediction Model

To further reduce the number of keystrokes during the whole human translation process, we propose an n-gram prediction model for the input method CoCat.

Consider the number of predictions, W. A bigger W may save more keystrokes, but too many predictions will impose an additional burden on translators' decision-making and selecting through a numeric key (plus 5 phrases). So, we choose $W = 4$ as the default setting.

Given an MT n-best list $O = \{O_i | 0 < i \leq |O|\}$, the i-th machine translation candidate is $o_i = o_{i1}o_{i2} \dots o_{i|o_i|}$, |O| denotes the size of the n-best list, $|o_i|$ denotes the word number of o_i, and o_{ij} denotes the j-th word in o_i. Suppose the desired human translation is $t_1^m = t_1 t_2 \dots t_m$. The algorithm of generating n-gram predictions is given as follows:

(1) Generate the initial W n-gram predictions $\{p_l | p_l = o_{11} \dots o_{1l}\ (1 \leq l \leq W)\ \}$ based on the best MT candidate o_1 before typing.

(2) The translator can press a certain numeric key to select the corresponding prediction, or ignore all the predictions and just continue typing his own translation.

(3) Once the translator completes typing the word t_j of the human translation t_1^m, we get the current prefix of the human translation $t_1^j = t_1 t_2 \dots t_j$, then we find its maximum suffix matching, o_i, in the MT n-best list using the maximum suffix matching algorithm.

(4) If there exists o_i in the step (3), we continue to use the maximum suffix matching algorithm to find k that satisfies $o_{ik} = t_j$. The system will dynamically update predictions $\{p_l | p_l = o_{i(k+1)} o_{i(k+2)} \dots o_{i(k+l)}\ (1 \leq l \leq W)\}$ based on o_i. If there is no such o_i, the n-gram prediction list is empty.

(5) Go to step (2) until the translator completes the whole translation process.

If we enable the n-gram prediction model, the key sequence would be further optimized as shown in Figure 3(b). In step 1, the correct initial prediction “中国考虑(China mulls)” is produced when we start translating as shown in Figure 3, and we can press the numeric key 6 to choose it directly. In step 3, the n-gram prediction model generates the correct prediction “公务员福利制度(officials’ welfare system)”, and we can press the numeric key 7 to select the suitable prediction. As shown in Figure 3, using the model, we can save the keystrokes by:

$$\frac{16-5}{16} \times 100\% = 68.75\%.$$

In this way, CoCat input method provides effective interactions and reduces the keystrokes as many as possible even if the MT results are not good enough.

3 Experiments

We conduct the experiments to test the performance of our CoCat input method in improving the productivity of the human translators. To have a comprehensive understanding, we measure the human productivity from three perspectives: translation time, keystrokes and translation quality.

3.1 Experimental Setup

All the experiments are conducted on our CoTrans Translator platform, which is an in-house developed CAT tool integrated with a typical phrase-based MT system (Xiong et al., 2006). This CAT tool supports translation among many languages. We test our method on English-to-Chinese translation. The integrated MT system is trained on about 10,000,000 parallel sentence pairs of English-Chinese news, and it is tuned on 1,000 parallel sentence pairs using ZMERT (Zaidan, 2009) with the objective to optimize TER (Snover and Dorr, 2006). This tuning set, which was translated into Chinese by professional translators, was chosen from Chinese news (prior to March 2014) of China Daily. All the knowledge contained in this MT system is utilized in our proposed CoCat input method. The statistical significance test is performed by the re-sampling approach (Koehn 2004).

		English-Chinese	
#translators		12	
male/female		6/6	
Total	#source words		3,918
	time (sec/word)		3.10
M_1	#source words		990
	time (sec/word)		3.56
M_2	#source words		983
	time (sec/word)		2.95
M_3	#source words		969
	time (sec/word)		2.8
M_4	#source words		976
	time (sec/word)		3.07

Table 1: The statistics of the 4 groups of test subset data $M_1/M_2/M_3/M_4$.

	A	B	C	D
Google	M_1	M_4	M_3	M_2
CoCat	M_2	M_1	M_4	M_3
PE+Google	M_3	M_2	M_1	M_4
PE+CoCat	M_4	M_3	M_2	M_1

Table 2: The permutation of assignments. Translation subsets M_1-M_4 are assigned to the human translator groups A-D under varying types of assistance.

CoTrans Translator platform tracks every key stroke and mouse click of the user and generates user interaction log which allows us to analyze the users’ translation time, keystrokes and translation quality in detail afterwards.

Next, we will introduce the participanting practationers and the experimental data.

Professional Translation Practitioners

Following the convention, we recruited 12 professional translators for our study. We divided the 12 translators into 4 groups evenly (A/B/C/D). Each translator translated the same set of sentences from English to Chinese. All of the professional translators are Chinese native speakers.

Experimental Data

We choose 160 sentences, $S = \{s_i | i = 1,2, \dots, 160\}$, from China news (prior to December 2014) of China Daily as the test set for human translators. This test set contains 3,918 English words. Each sentence ranges from 23 to 26 words.

The professional translators were asked to translate the text with four different assistant tools: (1) Google Pinyin (“Google”); (2) CoCat input method (“CoCat”); (3)post-editing with Google Pinyin (“PE+Google”); (4) post-editing

	A			B			C			D			total		
	time (s)	keystrokes	quality (BLEU)	time (s)	keystrokes	quality (BLEU)	time (s)	keystrokes	quality (BLEU)	time (s)	keystrokes	quality (BLEU)	time (s)	keystrokes	quality (BLEU)
Google	114.68	209.83	68.17	110.67	236.78	72.25	80.39	168.65	75.96	100.30	184.30	71.57	102.38	204.26	72.12
CoCat	89.61** (21.86%↓)	138.41** (34.04%↓)	76.31** (8.14↑)	98.05** (11.40%↓)	168.13** (28.99%↓)	80.42** (8.17↑)	68.05** (15.35%↓)	93.94** (44.30%↓)	86.06** (10.10↑)	71.56** (28.65%↓)	124.33** (32.50%↓)	82.84** (11.27↑)	**84.03**** **(17.89%↓)**	**134.85**** **(33.98%↓)**	**81.29**** **(9.17↑)**
PE+Google	64.7	100.66	78.49	52.93	92.24	80.74	83.25	158.13	77.02	71.78	121.81	77.72	66.59	115.75	78.79
PE+CoCat	52.03** (19.58%↓)	59.36** (41.03%↓)	81.53** (3.04↑)	48.34** (8.68%↓)	63.44** (31.22%↓)	85.32** (4.58↑)	65.43** (21.41%↓)	80.77** (48.92%↓)	84.05** (7.03↑)	66.90** (6.80%↓)	82.11** (32.60%↓)	72.76 (4.96↓)	**56.63**** **(14.97%↓)**	**69.865**** **(39.64%↓)**	**81.98**** **(3.19↑)**

Table 3: Translation time, keystrokes and translation quality. The numbers in parentheses represent the improvement over the corresponding previous line. Individual results vary. "**" means the scores are significantly better than the corresponding previous line with $p < 0.01$.

with CoCat input method ("PE+CoCat"). Naturally, for each human translator, he/she should translate different sentences when using different assistant tools. Thus, we splitted the test data into four subsets randomly and evenly. Table 1 shows the details about the statistics of the 4 groups of test subset data. Table 2 shows the details about the permutation of assignments inspired by the previous works (Koehn, 2009a; Green et al., 2014).

In the real world, there are many factors which may influence our experimental results, such as the different difficulties of the sentences to be translated, the tolerance of the long period of translation test and different levels of translators. To eliminate the irrelevant effects, we use the permutation of assignments in Table 2 based on the following assumptions: (1) the minor discrepancy of difficulty degrees of four test subsets can be negligible; (2) the fatigue degree difference of a particular translator in different time in one day can be negligible.

3.2 Data Cleaning

To exclude the translation irrelevant factors, such as the time spent on searching for terms and the moments of rest, we process the user interaction log as follows:

(1) Remove all the interactions which are irrelevant to the assistant tools from the timeline, such as looking up the dictionary online and searching information online.
(2) Exclude all of the time intervals lasting longer than 10 seconds between two adjacent interactions.
(3) Select the best four from the 12 human translations as references for each source sentence, and average the scores of human translations using BLEU-4 evaluation metric (Papineni et al., 2002).

3.3 Results and Analysis

We analyze the human productivity in terms of translation time, keystrokes and translation quality. To improve the robustness, we average the result values of repeated measurements. Let's take the "translation time" for example. According to the permutation of assignments in Table 2, the sentence s_i in subset M_1 has been translated by three translators in group A under the assistance of "Google". For the instance s_i, we average the three values of "translation time" given by the system and get the value $time_{s_i}^{Google}$. We compute the average translation time of a subset under the assistance of "Google" as follows:

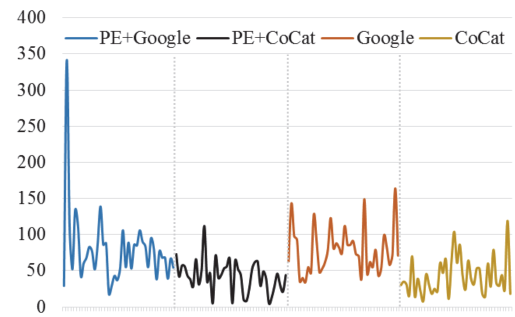


Figure 4: One translator's records on translation time. The graph plots the time spent on translation (in seconds, y-axis) against the sentence ID (x-axis).

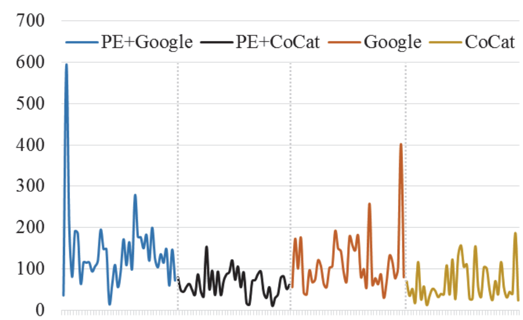


Figure 5: One translator's records on keystrokes. The graph plots the number of keystrokes spent on translation (y-axis) against the sentence ID (x-axis).

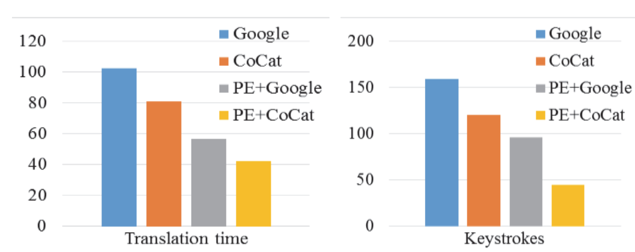


Figure 6: The comparisons of translation time and keystrokes of the four assistances applied to the sentence "CPC's discipline agency announced on Jan. 16 that Huo has been placed under investigation for suspected serious violation of party disciplines and laws".

$$time_{M_j}^{Google} = \frac{\sum_{s_i \in M_j} time_{s_i}^{Google}}{|M_j|}, j = 1,2,3,4$$

Then we calculate the average translation time of all sentences under the assistance of "Google" using the following formula:

$$time^{Google} = \frac{\sum_{i=1}^{160} time_{s_i}^{google}}{160}.$$

For keystrokes and translation quality, they are calculated in the same way.

For example, translation time and keystroke consumption on each sentence of a specific translator in group C are reported in Figure 4 and Figure 5. As we can see in the figures, CoCat helps her save about 46% time and about 41% keystrokes in the scratch mode, and save about 45% time and about 54% keystrokes in the post-editing mode.

The detailed results of all the human translators are reported in Table 3. On average, all human translators are faster and also achieve better translation quality using any of types of assistance offered. What's more, human translators are faster and also achieve better translation quality using CoCat (translating from scratch or post-editing).

For translation time and keystrokes, the figures in Table 3 show that our proposed CoCat always helps human translators significantly (with $p < 0.01$), saving more than 14% time and over 33% keystrokes compared with the strong baseline, i.e., post-editing using Google Pinyin (line 4 vs. line 3 and line 6 vs. line 5).

For translation quality, the figures in Table 3 show that CoCat can help human translators improve the translation quality significantly as well (with $p < 0.01$) by more than 3 absolute BLEU scores over the strong baseline.

Take a specific sentence as an example, such as "CPC's discipline agency announced on Jan. 16 that Huo has been placed under investigation for suspected serious violation of party discip-lines and laws", the comparison statistics of translation time and keystrokes are reported in Figure 6. CoCat can save about 21% time and about 24% keystrokes in the scratch mode, and save about 26% time and about 53% keystrokes in the post-editing mode.

Overall, the results in Table 3 indicate that post-editing consistently outperform unassisted translation. It is in line with the findings reported by Koehn (2012). Meanwhile, the post-editing well integrated with our proposed CoCat input method further improves the translation productivity.

What's more, if we focus on the comparison between "CoCat" and "PE+Google", we can find that the difference of the translation quality is very small. In the industrial world, the poor performance of the automatic translation engine is often a headache for human translators to edit the MT results. The comparison between "CoCat" and "PE+Google" tells us that we can make human translators generate better translation in less time with the aid of MT without headache.

In summary, we can draw the conclusion that the proposed new input method makes it easier for human translators to interact with MT systems effectively and imperceptibly.

4 Related Work

The goal of this paper is to improve the productivity and efficiency of human translators by fully exploiting the MT technology. The core idea is to provide human translators translation candidates effectively and friendly. There are two kinds of related work focusing on offering translation suggestions.

Koehn (2009a; Koehn et al., 2014) developed the tool Caitra which aims at providing translation suggestions to complete the target language sentence. Based on MT post-editing, their method can offer word and phrase translation candidates through interative machine translation. Green et al. (2014) made extensive modifications for the MT system and designed a new CAT interface. Their methods are tightly coupled with statistical machine translation in which only left-to-right decoding is allowed and dynamic decoding in interactive machine translation is usually time-consuming. In contrast, we integrate most of the useful knowledge of the MT system into a well designed CoCat input method that provides the translation suggestions more friendly and imperceptibly without forcing the human translators to take a view of the MT outputs. Besides using MT outputs, we are the first to exploit depth information used by MT, such as translation rules and decoding hypotheses.

Recently, Li (2012) and Fang (2013) also attempted to incorporate the SMT information into the Chinese Pinyin input method. In their approaches, when they developed their input methods, only the MT model scores and the fuzzy word alignment between the MT output and the human translation output are employed. However, there are two disadvantages in their approaches. On the one hand, the dynamic MT model scores are difficult to calculate and these model scores are not compatible with other features in input methods. On the other hand, the fuzzy word alignment contains much noise which would not benefit much to the input method. Instead, we design the log-linear model for the input method CoCat and integrate the translation rules, decoding hypotheses and the n-best translation list of the MT system. In addition, we propose the n-gram prediction model to further improve the efficiency of human translators.

5 Conclusion

In this paper, we have presented a novel input method CoCat which deeply integrates MT into CAT effectively and imperceptibly. This well-designed input method is modeled with a log-linear framework, and takes as features most of the useful knowledge of the MT system, such as translation rules, decoding hypotheses and n-best translation lists. Furthermore, we have proposed an n-gram prediction model that further speeds up the translation typing process.

The human translation experiments on English-to-Chinese have shown that the proposed approach can not only help human translators significantly save the time and keystrokes, but also substantially improve the final translation quality. The experiments have also shown that post-editing well integrated with our proposed approach further improves the translation productivity.

Acknowledgement

We thank anonymous reviewers for their valuable comments. The research work has been partially funded by the Natural Science Foundation of China under Grant No. 61333018 and No. 61403379 and supported by the West Light Foundation of Chinese Academy of Sciences under Grant No. LHXZ201301.

References

[Barrachina *et al*., 2009] Statistical Approaches to Computer-Assisted Translation. *Computational Linguistics*, 35(1): 3-28, 2009.

[Carl *et al*., 2011] The Process of Post-editing: A Pilot Study. *Copenhagen Studies in Language*, 41:131-142, 2011.

[Fang, 2013] Research and Implementation on Aided Translation Tools Based on Input Method. *Xiamen University*. 2013.

[Foster and Lapalme, 2002] Text Prediction for Translators. *Université de Montréal*, *2002*.

[Garay-Vitoria and Abascal, 2006] Text Prediction Systems: a Survey. *Universal Access in the Information Society*, 4(3): 188-203, 2006.

[Green *et al*., 2014] Human Effort and Machine Learnability in Computer Aided Translation. In *Proceedings of the EMNLP 2014*.

[Li, 2012] A Pinyin Input Method Editor with English-Chinese Aided Translation Function. In *2012 International Conference on Computer Science and Services System*.

[Kasami, 1965] An Efficient Recognition and Syntax Analysis Algorithm for Context-free Languages. *Technical Report AFCRL-65-758*. Air Force Cambridge Research Laboratory, Bedford, MA, 1965.

[Koehn, 2004] Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the EMNLP 2004.*

[Koehn, 2009a] A Process Study of Computer-Aided Transltion. *Machine Translation Journal*, 23(4):241-263, 2009.

[Koehn, 2009b] A Web-Based Interactive Computer Aided Translation Tool. In *Proceedings of the ACL-IJCNLP 2009*.

[Koehn, 2012] Computer-added Trasnlation. Machine Translation Marathon. 2012.

[Koehn *et al*., 2014] Refinements to Interactive Translation Prediction Based on Search Graphs. In *Proceedings of the ACL 2014*.

[Papineni *et al*., 2002] BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the ACL 2002*.

[Snover *et al*., 2006] A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of Association for Machine Translation in the Americas 2006*.

[Xiong *et al*., 2006] Maximum Entropy Based Phrase Reordering Model for Statistical Machine Translation. In *proceedings of COLING-ACL 2006*.

[Younger, 1967] Recognition and Parsing of Context-free Languages in Time n^3. *Information and Control*, 10(2):189-208, 1967.

[Zaidan, 2009] Z-MERT: A Fully Configurable Open Source Tool for Minimum Error Rate Training of Machine Translation Systems. *The Prague Bulletin of Mathematical Linguistics*. 91:79-88, 2009.

[Zhechev *et al*., 2012] Machine Translation Infrastructure and Post-editing Performance at Autodesk. In *AMTA 2012 Workshop on Post-Editing Technology and Practice*.