Introducing Mouse Actions into Interactive-Predictive Neural Machine Translation

Angel Navarro annamar8@prhlt.upv.es
Francisco Casacuberta fcn@prhlt.upv.es
Patter Recognition and Human Language Technology Research Center, Universitat Politècnica de València - Camino de Vera s/n, 46022 Valencia, Spain

Abstract

The quality of the translations generated by Machine Translation (MT) systems has highly improved through the years, but we are still far away to obtain fully automatic high-quality translations. To generate them, translators use Computer-Assisted Translation (CAT) tools, among which we find the Interactive-Predictive Machine Translation (IPMT) systems. This paper uses bandit feedback as the principal and only information needed to generate new predictions that correct the previous translations. Furthermore, the application of bandit feedback reduces the number of words that the translator needs to type in an IPMT session. In conclusion, this technique saves valuable time, and effort for translators. Moreover, its performance improves improves with the future advances in MT, so we recommend its application in the actuals IPMT systems.

1 Introduction

In recent years there had been a large number of advances in the Machine Translation (MT) field that has led to a significant improvement in the quality of the translations. Currently, even with all the new advances, the MT systems are still not able to generate perfect ready to use translations (Toral, 2020). Indeed, MT systems usually require human post-editing in order to achieve perfect translations.

The Computer-Assisted Translation (CAT) tools aim to generate high-quality translations using the knowledge and experience of professional translators while reducing the effort that they need to do. There is a large variety of CAT tools approaches, among which we focus on the Interactive-Predictive Machine Translation (IPMT) systems.

Some of the recent projects in this field are TransType (Langlais et al., 2000; Esteban et al., 2004; Cubel et al., 2003), Matecat (Federico et al., 2014), CasMacat (Alabau et al., 2014, 2013; Sanchis-Trilles et al., 2014) and MMPE (Herbig et al., 2020). They aim to create a workbench with an array of innovative features that were not available in other tools when they started. IPMT is one of the main paradigms that include these projects, where an expert translator provides feedback to the system, typically using the keyboard and mouse, to generate new predictions that correct previous errors.

There are two main IPMT approaches, both use usually the keyboard and mouse as the main feedback interface, but the validation process changes between prefix (Foster et al., 1997) and segments (Peris et al., 2017; Domingo et al., 2017). In this project, we use the validation by prefix approach. Figure 1 illustrates a conventional IPMT session. Initially, the user is provided with a source sentence x to be translated. At iteration 0, the IPMT system generates the first

SOURCE (x):		Una versión traducida de un texto.
REFERENCE (y):		A translated version of a text.
ITER-0	(**p**)	( )
	($\hat{s}_h$)	*A written version of a story.*
ITER-1	(**p**)	A
	(s_t)	*written version of a story.*
	(k)	translated
	($\hat{s}_h$)	*version of a text.*
ITER-2	(**p**)	A translated version of a text.
	(s_t)	( )
	(k)	(#)
	($\hat{s}_h$)	( )
FINAL	($p \equiv y$)	A translated version of a text.

Figure 1: Example of a conventional IPMT session to translate a sentence from Spanish to English. Non-validated hypotheses are displayed in italics, and accepted prefixes are printed in normal font.

hypothesis $\hat{s}_h$. At the next iteration, the user moves the cursor to the first error of the sentence, validanting the prefix **p**, and corrects the next word typing k. With this new information, the IPMT system searches the suffix $\hat{s}_h$ with the highest probability for the validated prefix **p**. This process continues until the whole sentence is validated and the user introduces the special token '#'.

IPMT aims to reduce the effort that the experts have to made in their translation sessions while preserving high-quality translations. Indeed, in Figure 1, the user has translated correctly the source sentence performing only three actions. Normally, in a regular post-editing system, the translator would have needed to perform five actions: two mouse movements, two word strokes, and the sentence validation.

In this paper, we reduce the effort done by the user taking into account bandit feedback. The system only needs the error position to correct the sentence, information that can be provided by the user easily with an interface like a mouse. For this reason, and to simplify, we are going to suppose that the feedback is provided with the mouse, although any other interface capable to provide a sentence position or make a click could be useful.

2 Related Work

The reduction of the effort needed in the translation process is a problem that has been thoroughly studied, resulting in a large variety of approaches. Some projects have investigated which information and display are more useful to the users, like showing the word alignment information (Brown et al., 1993), setting a maximum length for the predictions displayed (Alabau et al., 2012) or just using touch-based actions (Wang et al., 2020).

Other approaches reduce the effort that the user has to do more directly: using confidence measures to reduce the number of words to check (González-Rubio et al., 2010), autocompleting the predictions typed by the user (Barrachina et al., 2009), or adding new input information to the system reduces the human effort of generating a new prediction (Sanchis-Trilles et al., 2008a).

There are also projects like Lam et al. (2018, 2019) that investigated how to reduce the human effort in an IPMT system using Reinforcement Learning. This technique lets them use new kinds of feedback to the system that they use as a reward to adjust the parameters of the model and obtain better translations.

In the paper, we take the approach introduced by Sanchis-Trilles et al. (2008a,b), demonstrating that with only the error position, the Interactive-Predictive Statistical Machine Translation (IPSMT) systems are capable of correct their translations. We apply and implement this technique on an Interactive-Predictive Neural Machine Translation (IPNMT) system, obtaining a higher reduction in the human effort.

3 Interactive-Predictive Neural MT

In this section, we see briefly the IPNMT framework. First of all, we have to see the general framework of the Neural Machine Translation (NMT) models that we use to understand how the translations are created and how we later add human feedback to the equation. This framework was introduced by Castaño and Casacuberta (1997) and has demonstrated its power in the last years (Cho et al., 2014; Klein et al., 2017). Given a sentence $x_1^J = x_1, ..., x_J$ from the source language X, to find the sentence $\hat{y}_1^{\hat{I}} = \hat{y}_1, ..., \hat{y}_{\hat{I}}$ from the target language Y, that has the highest probability of being the translation of x_1^J, the fundamental equation of the statistical approach to NMT would be:

$$\hat{y}_1^{\hat{I}} = \arg\max_{I, y_1^I} \Pr(y_1^I \mid x_1^J) \approx \arg\max_{I, y_1^I} \prod_{i=1}^{I} \mathrm{p}(y_i \mid y_1^{i-1}, x_1^J; \widehat{\Theta}) \tag{1}$$

where $\Pr(y_i|y_1^{i-1}, x_1^J)$ and $\mathrm{p}(y_i|y_1^{i-1}, x_1^J)$, are the probability distribution and the probability that assigns the neural model to the next word given the source sentence and the previous words so far. $\hat{\Theta}$ are the parameters of the neural model which are obtained from trying to minimize the minus log-likelihood on a set of parallel corpus (Shen et al., 2016).

The IPNMT framework adds the feedback generated by the human to Equation (1) to help with the translation process. When the expert translator finds an error in position p, moves the cursor and types the correct word, producing the feedback $f_1^p = f_1, ..., f_p$ where f_p is the word that the user has typed to correct the error. We add the feedback with the last generated hypothesis to Equation (1):

$$\hat{y}_1^{\hat{I}} = \arg\max_{I, y_1^I} \Pr(y_1^I \mid x_1^J, \bar{y}_1^{\bar{I}}, f_1^p) = \arg\max_{I, y_1^I} \prod_{i=1}^{I} \Pr(y_i \mid y_1^{i-1}, x_1^J, \bar{y}_1^{\bar{I}}, f_1^p) \tag{2}$$

subject to

$$1 \leq i < p \qquad f_i = y_i = \bar{y}_i$$
$$f_p = y_p \neq \bar{y}_p$$

where $\bar{y}_1^{\bar{I}} = \bar{y}_1, ..., \bar{y}_{\bar{I}}$ is the previous hypothesis, f_1^p is the feedback provided, and p is the length of the feedback. With the constraints $1 \leq i < p$ $f_i = y_i = \bar{y}_i$ and $f_p = y_p \neq \bar{y}_p$, we assure that the feedback that the expert has provided appears in the hypothesis generated by the system. As the user corrects and validates the translation from left to right, this equation can be seen as obtaining the most probable suffix for the prefix provided.

4 Enriching User-Machine Interaction

Until now, the only interface that we have explored to IPMT is the combination of keyboard and mouse. The IPMT system provides a translation, and the user corrects it by placing the cursor before the first error and typing the correct word.

In this paper, we retake the work introduced by Sanchis-Trilles et al. (2008a). We use the mouse as an interface for the user-machine interaction to provide the IPMT system the

information about the position of the first error. First of all, we have to consider the two different classes of actions that can be performed with the mouse, *non-explicit* Mouse Actions (MAs) and *interaction-explicit* MAs.

4.1 Non-Explicit MA

In conventional IPMT systems, before the user types any word, he has to move the cursor to the position where he wants to make the correction. With the cursor movement, the user is already providing valuable information to the system that we can use. He validates all the previous words and tags the next as incorrect. Just with this information, the system can generate a new hypothesis, in which the prefix remains unaltered, and the suffix changes for the following hypothesis with the higher probability that starts by a different word. This action does not suppose an extra cost for the translator, it is automatically performed when the mouse already needs to be moved to perform a correction. This process does not assure that the new suffix is correct but in the worst scenario, the user behaves as in a conventional IPMT system. In Equation (2) we calculate the best hypothesis using the feedback that the user provides to the system $f_1^p = f_1, ..., f_p$ where f_p is the word that the user types to correct the error. In this new situation, the user does not provide the correct word in position p, but we know that it has to be different from the used in the previous hypothesis y_p. This situation can be expressed as follows:

$$\hat{y}_1^{\hat{I}} = \arg\max_{I, y_1^I} \Pr(y_1^I \mid x_1^J, \bar{y}_1^{\bar{I}}, f_1^p) = \arg\max_{I, y_1^I} \prod_{i=1}^{I} \Pr(y_i \mid y_1^{i-1}, x_1^J, \bar{y}_1^{\bar{I}}, f_1^p) \tag{3}$$

subject to

$$1 \leq i < p \qquad f_i = y_i = \bar{y}_i$$

$$y_p : \exists\, y_p \hat{y}_{p+1}^{\hat{I}}$$

$$y_p \hat{y}_{p+1}^{\hat{I}} = \arg\max_{\substack{I', y'_p, {y'}_{p+1}^{I'} \\ y'_p \neq \bar{y}_p}} \Pr({y'}_p, {y'}_{p+1}^{I'} \mid x_1^J, y_1^{p-1}) \tag{4}$$

where y_p is the word that the system is trying to correct. To assure that the new word at position p from the suffix is different from the one used in the previous hypothesis y_p we add the constraint $y'_p \neq \bar{y}_p$ to Equation (4) that is responsible for the generation of new suffixes. $y_p \hat{y}_{p+1}^{\hat{I}}$ is the suffix with the highest probability given the source sentence and the prefix that the user has validated.

4.2 Interaction-Explicit MA

The non-explicit MAs does not suppose an extra cost for the translator. In a conventional IPMT system, the user needs to move the cursor to the correct position in order to change a word. Once the user has moved the cursor to the correct position and the system has performed a non-explicit MA, if the translation still has an error in the same position the user can perform an interaction-explicit MA. This kind of MA needs that the user explicitly executes the action of asking for a new suffix, for this reason, the interaction-explicit MAs suppose a little extra cost that can save the user the effort of typing the correct word. In the end, is the user who has to decide which kind of action performs depending on his experience.

In this project, we have used the mouse as an interface to provide to the system the position of the error, and the action of performing an interaction-explicit MA. Note that the interface used could be different, e.g. using a touch screen, or typing some special key such as `F1` or `Tab`. However, it is explained with the mouse because we found it more intuitive and understandable.

SOURCE (x):		Escriba aquí la traducción.
REFERENCE (y):		Write the translation here.
ITER-0	(**p**)	( )
	($\hat{s}_h$)	‖ *Write there the translation.*
ITER-1	(**p**)	Write
	(s_t)	‖ *there the translation.*
	($\hat{s}_h$)	*here the translation.*
ITER-2	(**p**)	Write
	(s_t)	‖ *here the translation.*
	($\hat{s}_h$)	*the translation here.*
ITER-3	(**p**)	Write the translation here.
	(s_t)	( )
	(k)	(#)
	($\hat{s}_h$)	( )
FINAL	(**p** ≡ **y**)	Write the translation here.

Figure 2: Example of an IPMT session with non-explicit and interaction-explicit MAs. At iteration 0, the user moves the cursor before '*there*', and the system provides a new suffix. At iteration 1, before manually correcting the word, the user performs an interactive-explicit MA. At iteration 3, the user validates the translation. Non-validated hypotheses are displayed in italics, and accepted prefixes are in normal font. The MAs are indicated by the symbol '‖'.

Each time we perform an MA for the same position p, we obtain a new word that we do not want to get in the new suffix. The following equation solves this problem by keeping track of the k previous hypotheses, where k is the number of MAs performed in the same position:

$$\hat{y}_1^{\hat{I}} = \arg\max_{I, y_1^I} \Pr(y_1^I \mid x_1^J, \bar{y}_1^{\bar{I}}, f_1^p, k) = \arg\max_{I, y_1^I} \prod_{i=1}^{I} \Pr(y_i \mid y_1^{i-1}, x_1^J, \bar{y}_1^{\bar{I}}, f_1^p, k) \quad (5)$$

subject to

$$1 \leq i < p \qquad f_i = y_i = \bar{y}_i$$

$$y_p : \exists\, y_p^{(k)} \hat{y}_{p+1}^{\hat{I}}$$

$$y_p^{(k)} \hat{y}_{p+1}^{\hat{I}} = \arg\max_{\substack{I', y'_p, {y'}_{p+1}^{I'} \\ y'_p \notin \{\bar{y}_p, y_p^{(1)}, \ldots, y_p^{(k-1)}\}}} \Pr(y'_p, {y'}_{p+1}^{I'} \mid x_1^J, y_1^{p-1}) \quad (6)$$

where $y_p^{(k)}$ is the word that occupies the position p of the new hypothesis when the user performs the k_{th} MA. $y_p^{(l)}$ $l < k$ are the words that have been generated before the user performs the k_{th} MA, and $\bar{y}$ is the first hypothesis generated before performing any MA in position p.

We can see an example of a conventional IPMT session where the user performs a non-explicit MA and an interactive-explicit MA in Figure 2. At iteration 0 the system provides to the user the translation, and the cursor stays at the start of the sentence. At iteration 1 the user moves the cursor to the first error, validating the prefix (p) and performing a non-explicit MA. The system automatically generates a new suffix $\hat{s}_h$ that the user has to check in the next iterations. At iteration 2, the translation is still incorrect and the user decides to perform an interactive-explicit MA to correct it. The system generates a new suffix that can not start with the words '*there*' or '*here*'. Finally, at iteration 3, the user does not see any error and validates all the sentence.

5 Experimental Setup

5.1 System Evaluation

In this article, we report our results using different metrics to measure the human effort performed in an IPMT session, differentiating between the keystrokes and the mouse actions performed. We report the effort done by the user in Word Stroke Ratio (WSR), Mouse Action Ratio (MAR), character MAR (cMAR), and useful MAR (uMAR) that gives us a reference of the mouse actions performed and the quality of them.

WSR, introduced by Tomás and Casacuberta (2006), is computed as the number of words that the user needs to type to generate the reference translation, normalized by the total number of words in the sentence. In this context, a word stroke is interpreted as a single action. Moreover, it is assumed to have a constant cost.

MAR, cMAR and uMAR were introduced by Sanchis-Trilles et al. (2008b) when they first considered the mouse actions as significant information to IPMT systems. MAR is computed as the number of MAs that the user needs to perform in order to generate the reference translation, normalized by the total number of words in the sentence. The cMAR is calculated normalizing by the total number of characters. Non-explicit and Interaction-explicit MAs have the same cost.

Lastly, uMAR indicates the amount of MAs that are useful to achieve the translation that the user has in mind i.e. the MAs that actually ending changing correctly the first word of the suffix. Formally, uMAR is defined as follows:

$$\text{uMAR} = \frac{\text{MAC} - n\text{WSC}}{\text{MAC}} \tag{7}$$

where Mouse Action Count (MAC) is the total number of MAs performed, Word Stroke Count (WSC) is the number of words typed and n is the maximum amount of MA allowed before the user types in a word. Note that in order to perform a word-stroke the user previously must have performed n MAs, so in Equation (7), we are removing from the total count of MAs those that were not useful and did not help to find the correct word.

5.2 Corpora

We conduct our experiments on the domain Europarl (Koehn, 2005). The Europarl corpus is built from the Proceedings of the European Parliament, which exists in all official languages of the European Union, and is publicly available on the internet. We use the pair of languages Deutch-English (De-En), Spanish-English (Es-En) and French-English (Fr-En) in both directions in all our experiments. Their characteristics are described in Table 1. All the corpora have been cleaned, lower-cased and tokenized using the scripts included in the toolkit Moses, developed by Koehn et al. (2007). Once we have them tokenized, we have applied the subword subdivision BPE, described in Sennrich et al. (2016), with a maximum of 32000 merges.

		De-En		Es-En		Fr-En	
Training	Sentences	751K		730K		688K	
	Avg. Length	20	21	21	20	22	20
	Run. Words	15M	16M	15M	15M	15M	14M
	Vocabulary	195K	65K	102K	64K	80K	61K
Dev.	Sentences	2000		2000		2000	
	Avg. Length	27	29	30	29	33	29
	Run. Words	55K	59K	60K	59K	67K	59K
Test	Sentences	2000		2000		2000	
	Avg. Length	27	29	30	29	33	29
	Run. Words	54K	58K	67K	58K	66K	58K

Table 1: Characteristics of the Europarl corpus. K and M stands for thousands and millions.

5.3 User Simulation

Our experiments have not used real humans to translate the source sentences interactively because it would have been costly and slow. Instead, we have simulated the expected behaviour of professional translators.

When the simulated user receives a new prediction from the IPMT system, they search for the first error of the translation, comparing the words and position from the hypothesis and the reference. Then, when the user has found an error, they perform a non-explicit MA if the mouse is not in the correct position or an interaction-explicit MA. The simulated user performs a maximum of n MAs for the same position, where n is a value set at the start of the experiment. If the error is not corrected once the user performs all the possible actions, they type the correct word looking at the reference. We repeat this process until the simulated user translates all the sentence correctly.

5.4 Model Architecture

We built our NMT models using NMT-Keras (Álvaro Peris and Casacuberta, 2018). We have tested the experiments using a Recurrent Neural Network (RNN) and a Transformer. All the systems used Adam (Kingma and Ba, 2017) as the learning algorithm, with a learning rate of 0.0002. We clipped the L_2 norm of the gradient to 5. The batch size was set to 30 and the beam size to 6.

The RNN-based NMT system used was an encoder-decoder architecture with an attention model (Chorowski et al., 2015) and LSTM cells (Hochreiter and Schmidhuber, 1997). The dimensions of the encoder, decoder, attention model and word embeddings were set to 512. We used a single hidden layer of the encoder and the decoder.

The Transformer (Vaswani et al., 2017) model used a word embedding and dimension size of 512. The hidden and output dimensions of the feed-forward layers were set to 2048 and 512. Each multi-head attention layer had 8 heads, and we stacked 6 layers of encoder and decoder.

Table 2 shows the translation performance in terms of BLEU of RNN-based and Transformer neural models.

	BLEU (↑)	
	RNN	Transformer
De-En	27.8	28.8
En-De	21.8	19.2
Es-En	32.1	32.1
En-Es	31.7	31.4
Fr-En	30.9	31.1
En-Fr	33.0	32.3

Table 2: Translation quality for the Europarl task in terms of BLEU for RNN and Transformer.

5.5 Experimental Results

The results of both models are displayed in Tables 3 and 4. There, we compare the results obtained from a conventional IPMT system, with the addition to the system of the non-explicit MAs, and the interaction-explicit MAs with a maximum of 4 explicit actions per position. By just adding the non-explicit MAs to the system, on average, the user reduces his effort by 27.45%. The models are good enough that the correct word is the second most probably from the error position. And if we take account of the interactive-explicit MAs, the reduction is 55.9%. Note how with the non-explicit MAs the MAR values remains almost identical because the non-explicit MAs does not suppose an extra cost. The differences in values are special cases where the system predicted a correct sentence different to the obtained by typing the correct word.

	baseline		non-explicit			interaction-explicit		
	MAR (↓)	WSR (↓)	MAR (↓)	WSR (↓)	WSR rel. (↑)	MAR (↓)	WSR (↓)	WSR rel. (↑)
De-En	44.2	42.2	46.0	**31.0***	26.5	145.8	**19.2***	54.6
En-De	46.9	45.0	49.0	**34.0***	24.3	162.0	**22.7***	49.6
Es-En	41.0	38.7	42.6	**27.6**	28.6	131.2	**16.9**	56.4
En-Es	41.2	39.3	43.1	**28.8**	26.9	136.2	**17.9**	54.5
Fr-En	42.0	39.6	43.6	**28.7***	27.6	135.9	**17.6***	55.5
En-Fr	38.4	36.5	40.0	**26.2**	28.2	123.1	**15.5**	57.5

Table 3: Experimental results with RNN in the Europarl corpus when considering non-explicit and interaction-explicit MAs. Systems significantly differents from the Transformers systems are indicated with a *.

	baseline		non-explicit			interaction-explicit		
	MAR (↓)	WSR (↓)	MAR (↓)	WSR (↓)	WSR rel. (↑)	MAR (↓)	WSR (↓)	WSR rel. (↑)
De-En	42.5	40.5	44.3	**29.1***	28.2	136.7	**17.5***	56.7
En-De	49.7	47.8	51.8	**36.2***	24.3	173.1	**24.5***	48.8
Es-En	40.5	38.2	42.2	**27.0**	29.3	127.9	**16.3**	57.4
En-Es	41.4	39.6	43.3	**28.7**	27.6	135.9	**17.8**	55.1
Fr-En	41.2	38.9	42.9	**27.3***	29.9	129.6	**16.4***	58.0
En-Fr	38.1	36.2	39.7	**25.7**	29.0	121.2	**15.3**	57.7

Table 4: Experimental results with Transformer in the Europarl corpus when considering non-explicit and interaction-explicit MAs. Systems significantly differents from the RNN systems are indicated with a *.

We have realized an ANOVA (ANalysis Of VAriance) with a confidence of the 95% comparing for each pair of languages the results obtained from the RNN and the Transformer to see if the models are statistically the same or not. The results are displayed in Tables 3 and 4, where we tagged with an asterisk the results that we have statistical significance that they are different.

Figure 3 shows the uMAR results versus the WSR obtainer for each maximum value of MAs up to five with the RNN and Transformer models. Each time that we increase the maximum number of MAs the number of errors fixed without typing the correct word is lower. If we look at the uMAR values obtained at each iteration we can understand how the reduction has worked. The uMAR values do not have a high variance, the value remains more or less the same for both models while increasing the maximum number of MAs, 35. Each time that we have increased the maximum number of MAs the 35% of the errors that were not corrected with the previous maximum are corrected now. Knowing how the uMAR value evolves, helps the human translator to choose between performing an interaction-explicit MA or typing directly the correct word.

5.6 Comparison Results

In the last years, this same approach was explored on Interactive-Predictive Statistical Machine Translation (IPSMT) systems and was tested in the Europarl corpora (Sanchis-Trilles et al., 2008b). In this section, we compare the results obtained in their project with the Statistical Machine Translation (SMT) models versus ours results with NMT models. We compare their results only with the Transformer because both models have obtained very similar results.

In Figure 4, we can see the comparison results obtained in the Europarl corpus with the SMT and NMT models. Taking into account the results obtained with a maximum of 5 MAs, the SMT models get a WSR relative improvement around 24%, while the NMT models obtained a relative improvement around 57%. From the uMAR results, we can see that in the SMT models

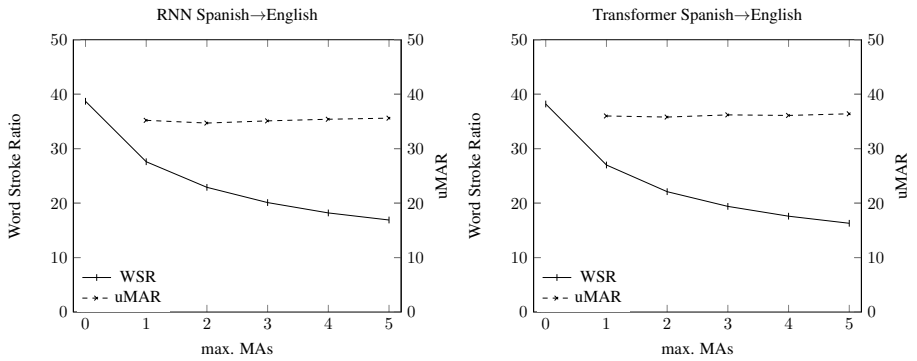

Figure 3: WSR when considering up to five maximum MAs versus uMAR with RNN and Transfromer in the Europarl corpus.

the percentage of uMAR goes from 6% to 12%, causing a lower WSR relative improvement. Meanwhile, the NMT model maintains the percentage of uMAR around 35%.

Looking at these two results we can see how the NMT models are more likely to fix an error correctly than the SMT models. Although the human interaction was simulated the same for both projects, the uMAR score that gives us the percentage of useful MAR is very different, so we can conclude that the NMT models produce better corrections with the information that we are providing.

6 Conclusions and Future Work

6.1 Conclusions

In this paper, we have implemented the use of bandit feedback to generate new predictions preserving the validated prefix. We have tested RNN and Transformer models with the Europarl corpus, and both models obtained very similar results. Both models have improved the baseline, proving that this kind of input information is useful and can reduce drastically the effort needed to correct a translation. Moreover, as the non-explicit MAs do not suppose an extra cost for the translator there are no cons to implement this approach on actual IPMT systems.

Additionally, we have compared our results with a previous work that used this same approach on SMT models, and the WSR relative improvement obtained in our experiments is greater. Proving that the NMT models obtain better results with this kind of interaction and feedback provided than the SMT models.

6.2 Future Work

In all the experiments that we have performed the user has been simulated following some basic rules. As future work, we need to test the use of mouse actions with an application where we can study the results of real humans that need to adapt to this new kind of input.

7 Acknowledgements

This work received funds from the Comunitat Valenciana under project EU-FEDER (*IDIFEDER*/2018/025), Generalitat Valenciana under project ALMAMATER (*PrometeoII*/2014/030), and Ministerio de Ciencia under project MIRANDA-DocTIUM (RTI2018-095645-B-C22).

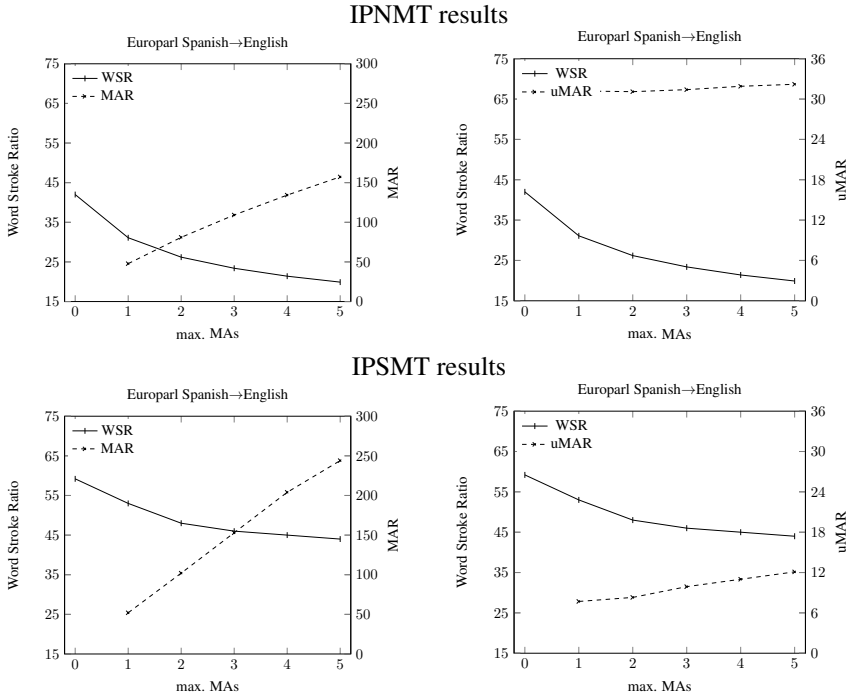

Figure 4: Comparison results with the Europarl Corpus considering up to five maximum MAs. The left column shows WSR versus MAR and in the right column shows WSR versus uMAR. Our results (up) and Sanchis-Trilles et al. (2008b) results (down)

References

Alabau, V., Bonk, R., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., González, J., Koehn, P., Leiva, L., Mesa-Lao, B., et al. (2013). Casmacat: An open source workbench for advanced computer aided translation. *The Prague Bulletin of Mathematical Linguistics*, 100(1):101–112.

Alabau, V., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., Germann, U., González-Rubio, J., Hill, R., Koehn, P., Leiva, L., Mesa-Lao, B., Ortiz-Martínez, D., Saint-Amand, H., Sanchis Trilles, G., and Tsoukala, C. (2014). CASMACAT: A computer-assisted translation workbench. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 25–28, Gothenburg, Sweden. Association for Computational Linguistics.

Alabau, V., Leiva, L. A., Ortiz-Martínez, D., and Casacuberta, F. (2012). User evaluation of interactive machine translation systems. In *Proceedings of the 16th Annual conference of the European Association for Machine Translation*, pages 20–23, Trento, Italy. European Association for Machine Translation.

Álvaro Peris and Casacuberta, F. (2018). NMT-Keras: a Very Flexible Toolkit with a Focus on Interactive NMT and Online Learning. *The Prague Bulletin of Mathematical Linguistics*, 111:113–124.

Barrachina, S., Bender, O., Casacuberta, F., Civera, J., Cubel, E., Khadivi, S., Lagarda, A., Ney, H., Tomás, J., Vidal, E., and Vilar, J.-M. (2009). Statistical approaches to computer-assisted translation. *Computational Linguistics*, 35(1):3–28.

Brown, P. F., Della Pietra, S. A., Della Pietra, V. J., and Mercer, R. L. (1993). The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311.

Castaño, A. and Casacuberta, F. (1997). A connectionist approach to machine translation. In *Fifth European Conference on Speech Communication and Technology*, pages 91–94.

Cho, K., van Merriënboer, B., Bahdanau, D., and Bengio, Y. (2014). On the properties of neural machine translation: Encoder–decoder approaches. In *Proceedings of Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 103–111, Doha, Qatar. Association for Computational Linguistics.

Chorowski, J. K., Bahdanau, D., Serdyuk, D., Cho, K., and Bengio, Y. (2015). Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Cubel, E., González, J., Lagarda, A., Casacuberta, F., Juan, A., and Vidal, E. (2003). Adapting finite-state translation to the TransType2 project. In *EAMT Workshop: Improving MT through other language technology tools: resources and tools for building MT*, pages 15–17, Budapest, Hungary. European Association for Machine Translation.

Domingo, M., Peris, A., and Casacuberta, F. (2017). Segment-based interactive-predictive machine translation. *Machine Translation*, 31(4):163–185.

Esteban, J., Lorenzo, J., Valderrábanos, A. S., and Lapalme, G. (2004). TransType2 - an innovative computer-assisted translation system. In *Proceedings of the ACL Interactive Poster and Demonstration Sessions*, pages 94–97, Barcelona, Spain. Association for Computational Linguistics.

Federico, M., Bertoldi, N., Cettolo, M., Negri, M., Turchi, M., Trombetti, M., Cattelan, A., Farina, A., Lupinetti, D., Martines, A., Massidda, A., Schwenk, H., Barrault, L., Blain, F., Koehn, P., Buck, C., and Germann, U. (2014). The MateCat tool. In *Proceedings of the 25th International Conference on Computational Linguistics: System Demonstrations*, pages 129–132, Dublin, Ireland. Dublin City University and Association for Computational Linguistics.

Foster, G., Isabelle, P., and Plamondon, P. (1997). Target-text mediated interactive machine translation. *Machine Translation*, 12(1):175–194.

González-Rubio, J., Ortiz-Martínez, D., and Casacuberta, F. (2010). Balancing user effort and translation error in interactive machine translation via confidence measures. In *Proceedings of the Association for Computational Linguistics 2010 Conference Short Papers*, pages 173–177, Uppsala, Sweden. Association for Computational Linguistics.

Herbig, N., Düwel, T., Pal, S., Meladaki, K., Monshizadeh, M., Krüger, A., and van Genabith, J. (2020). Mmpe: A multi-modal interface for post-editing machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1691–1702.

Hochreiter, S. and Schmidhuber, J. (1997). Long short-term memory. *Neural computation*, 9(8):1735–1780.

Kingma, D. P. and Ba, J. (2017). Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Klein, G., Kim, Y., Deng, Y., Senellart, J., and Rush, A. (2017). OpenNMT: Open-source toolkit for neural machine translation. In *Proceedings of the Association for Computational Linguistics 2017, System Demonstrations*, pages 67–72, Vancouver, Canada. Association for Computational Linguistics.

Koehn, P. (2005). Europarl: A parallel corpus for statistical machine translation. In *Machine Translation summit*, volume 5, pages 79–86. Citeseer.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Lam, T. K., Kreutzer, J., and Riezler, S. (2018). A reinforcement learning approach to interactive-predictive neural machine translation. *arXiv preprint arXiv:1805.01553*.

Lam, T. K., Schamoni, S., and Riezler, S. (2019). Interactive-predictive neural machine translation through reinforcement and imitation. *arXiv preprint arXiv:1907.02326*.

Langlais, P., Foster, G., and Lapalme, G. (2000). TransType: a computer-aided translation typing system. In *ANLP-NAACL 2000 Workshop: Embedded Machine Translation Systems*, pages 46–51.

Peris, Á., Domingo, M., and Casacuberta, F. (2017). Interactive neural machine translation. *Computer Speech & Language*, 45:201–220.

Sanchis-Trilles, G., Alabau, V., Buck, C., Carl, M., Casacuberta, F., García-Martínez, M., Germann, U., González-Rubio, J., Hill, R. L., Koehn, P., et al. (2014). Interactive translation prediction versus conventional post-editing in practice: a study with the casmacat workbench. *Machine Translation*, 28(3-4):217–235.

Sanchis-Trilles, G., González, M.-T., Casacuberta, F., Vidal, E., and Civera, J. (2008a). Introducing additional input information into interactive machine translation systems. In *Proceedings of International Workshop on Machine Learning for Multimodal Interaction*, pages 284–295. Springer.

Sanchis-Trilles, G., Ortiz-Martínez, D., Civera, J., Casacuberta, F., Vidal, E., and Hoang, H. (2008b). Improving interactive machine translation via mouse actions. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 485–494, Honolulu, Hawaii. Association for Computational Linguistics.

Sennrich, R., Haddow, B., and Birch, A. (2016). Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., and Liu, Y. (2016). Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Tomás, J. and Casacuberta, F. (2006). Statistical phrase-based models for interactive computer-assisted translation. In *Proceedings of the COLING/ACL 2006 Main Conference Poster Sessions*, pages 835–841, Sydney, Australia. Association for Computational Linguistics.

Toral, A. (2020). Reassessing claims of human parity and super-human performance in machine translation at wmt 2019. *arXiv preprint arXiv:2005.05738*.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., and Polosukhin, I. (2017). Attention is all you need. *arXiv preprint arXiv:1706.03762*.

Wang, Q., Zhang, J., Liu, L., Huang, G., and Zong, C. (2020). Touch editing: A flexible one-time interaction approach for translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, pages 1–11.