

Improving Interactive Machine Translation via Mouse Actions

Germán Sanchis-Trilles and **Daniel Ortiz-Martínez** and **Jorge Civera**

Instituto Tecnológico de Informática
Universidad Politécnica de Valencia
{gsanchis,dortiz,jorcisai}@iti.upv.es

Francisco Casacuberta and **Enrique Vidal**

Departamento de Sistemas Informáticos y Computación
Universidad Politécnica de Valencia
{fcn,evidal}@dsic.upv.es

Hieu Hoang

University of Edinburgh
hhoang@sms.ed.ac.uk

Abstract

Although Machine Translation (MT) is a very active research field which is receiving an increasing amount of attention from the research community, the results that current MT systems are capable of producing are still quite far away from perfection. Because of this, and in order to build systems that yield correct translations, human knowledge must be integrated into the translation process, which will be carried out in our case in an Interactive-Predictive (IP) framework. In this paper, we show that considering Mouse Actions as a significant information source for the underlying system improves the productivity of the human translator involved. In addition, we also show that the initial translations that the MT system provides can be quickly improved by an expert by only performing additional Mouse Actions. In this work, we will be using word graphs as an efficient interface between a phrase-based MT system and the IP engine.

1 Introduction

Information technology advances in modern society have led to the need of more efficient methods of translation. It is important to remark that current MT systems are not able to produce ready-to-use texts (Kay, 1997; Hutchins, 1999; Arnold, 2003). Indeed, MT systems are usually limited to specific semantic domains and the translations provided re-

quire human post-editing in order to achieve a correct high-quality translation.

A way of taking advantage of MT systems is to combine them with the knowledge of a human translator, constituting the so-called Computer-Assisted Translation (CAT) paradigm. CAT offers different approaches in order to benefit from the synergy between humans and MT systems.

An important contribution to interactive CAT technology was carried out around the TransType (TT) project (Langlais et al., 2002; Foster et al., 2002; Foster, 2002; Och et al., 2003). This project entailed an interesting focus shift in which interaction directly aimed at the production of the target text, rather than at the disambiguation of the source text, as in former interactive systems. The idea proposed was to embed data driven MT techniques within the interactive translation environment.

Following these TT ideas, (Barrachina and others, 2008) propose the usage of fully-fledged statistical MT (SMT) systems to produce full target sentence hypotheses, or portions thereof, which can be partially or completely accepted and amended by a human translator. Each partial correct text segment is then used by the SMT system as additional information to achieve further, hopefully improved suggestions. In this paper, we also focus on the interactive and predictive, statistical MT (IMT) approach to CAT. The IMT paradigm fits well within the *Interactive Pattern Recognition* framework introduced in (Vidal and others, 2007).

SOURCE (x):		Para encender la impresora:
REFERENCE (y):		To power on the printer:
ITER-0	(p) (<i>\hat{s}_h</i>)	() <i>To switch on:</i>
ITER-1	(p) (<i>s_l</i>) (<i>k</i>) (<i>\hat{s}_h</i>)	To <i>switch on:</i> power <i>on the printer:</i>
ITER-2	(p) (<i>s_l</i>) (<i>k</i>) (<i>\hat{s}_h</i>)	To power on the printer: () (#) ()
FINAL	(p \equiv y)	To power on the printer:

Figure 1: IMT session to translate a Spanish sentence into English. Non-validated hypotheses are displayed in italics, whereas accepted prefixes are printed in normal font.

Figure 1 illustrates a typical IMT session. Initially, the user is given an input sentence \mathbf{x} to be translated. The reference \mathbf{y} provided is the translation that the user would like to achieve at the end of the IMT session. At iteration 0, the user does not supply any correct text prefix to the system, for this reason \mathbf{p} is shown as empty. Therefore, the IMT system has to provide an initial complete translation s_h , as it were a conventional SMT system. At the next iteration, the user validates a prefix \mathbf{p} as correct by positioning the cursor in a certain position of s_h . In this case, after the words “*To print a*”. Implicitly, he is also marking the rest of the sentence, the suffix s_l , as potentially incorrect. Next, he introduces a new word k , which is assumed to be different from the first word s_{l_1} in the suffix s_l which was not validated, $k \neq s_{l_1}$. This being done, the system suggests a new suffix hypothesis \hat{s}_h , subject to $\hat{s}_{h_1} = k$. Again, the user validates a new prefix, introduces a new word and so forth. The process continues until the whole sentence is correct that is validated introducing the special word “#”.

As the reader could devise from the IMT session described above, IMT aims at reducing the effort and increasing the productivity of translators, while preserving high-quality translation. For instance, in Figure 1, only three interactions were necessary in order to achieve the reference translation.

In this paper, we will show how Mouse Actions performed by the human expert can be taken advantage of in order to further reduce this effort.

2 Statistical interactive-predictive MT

In this section we will briefly describe the statistical framework of IMT. IMT can be seen as an evolution of the SMT framework, which has proved to be an efficient framework for building state-of-the-art MT systems with little human effort, whenever adequate corpora are available (Hutchings and Somers, 1992). The fundamental equation of the statistical approach to MT is

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} Pr(\mathbf{y} | \mathbf{x}) \quad (1)$$

$$= \underset{\mathbf{y}}{\operatorname{argmax}} Pr(\mathbf{x} | \mathbf{y}) Pr(\mathbf{y}) \quad (2)$$

where $Pr(\mathbf{x} | \mathbf{y})$ is the *translation model* modelling the correlation between source and target sentence and $Pr(\mathbf{y})$ is the *language model* representing the well-formedness of the candidate translation \mathbf{y} .

In practise, the direct modelling of the posterior probability $Pr(\mathbf{y} | \mathbf{x})$ has been widely adopted. To this purpose, different authors (Papineni et al., 1998; Och and Ney, 2002) propose the use of the so-called log-linear models, where the decision rule is given by the expression

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}} \sum_{m=1}^M \lambda_m h_m(\mathbf{x}, \mathbf{y}) \quad (3)$$

where $h_m(\mathbf{x}, \mathbf{y})$ is a score function representing an important feature for the translation of \mathbf{x} into \mathbf{y} , M is the number of models (or features) and λ_m are the weights of the log-linear combination.

One of the most popular instantiations of log-linear models is that including phrase-based (PB) models (Zens et al., 2002; Koehn et al., 2003). Phrase-based models allow to capture contextual information to learn translations for whole phrases instead of single words. The basic idea of phrase-based translation is to segment the source sentence into phrases, then to translate each source phrase into a target phrase, and finally to reorder the translated target phrases in order to compose the target sentence. Phrase-based models were employed throughout this work.

In log-linear models, the maximisation problem stated in Eq. 3 is solved by means of the beam search algorithm¹ which was initially introduced in (Lowe, 1976) for its application in the field of speech recognition. The beam search algorithm attempts to generate partial solutions, called *hypotheses*, until a complete sentence is found; these hypotheses are stored in a stack and ordered by their *score*. Such a score is given by the log-linear combination of feature functions.

However, Eq. 1 needs to be modified according to the IMT scenario in order to take into account part of the target sentence that is already translated, that is \mathbf{p} and k

$$\hat{s}_h = \operatorname{argmax}_{s_h} Pr(s_h | \mathbf{x}, \mathbf{p}, k) \quad (4)$$

where the maximisation problem is defined over the suffix s_h . This allows us to rewrite Eq. 4, by decomposing the right side appropriately and eliminating constant terms, achieving the equivalent criterion

$$\hat{s}_h = \operatorname{argmax}_{s_h} Pr(\mathbf{p}, k, s_h | \mathbf{x}). \quad (5)$$

An example of the intuition behind these variables can be seen in Figure 1.

Note that, since $(\mathbf{p} k s_h) = \mathbf{y}$, Eq. 5 is very similar to Eq. 1. The main difference is that the argmax search is now performed over the set of suffixes s_h that complete $(\mathbf{p} k)$ instead of complete sentences (\mathbf{y} in Eq. 1). This implies that we can use the same models if the search procedures are adequately modified (Barrachina and others, 2008).

¹Also known as stack decoding algorithm.

3 Phrase-based IMT

The phrase-based approach presented above can be easily adapted for its use in an IMT scenario. The most important modification is to rely on a word graph that represents possible translations of the given source sentence. The use of word graphs in IMT has been studied in (Barrachina and others, 2008) in combination with two different translation techniques, namely, the Alignment Templates technique (Och et al., 1999; Och and Ney, 2004), and the Stochastic Finite State Transducers technique (Casacuberta and Vidal, 2007).

3.1 Generation of word graphs

A word graph is a weighted directed acyclic graph, in which each node represents a partial translation hypothesis and each edge is labelled with a word of the target sentence and is weighted according to the scores given by an SMT model (see (Ueffing et al., 2002) for more details). In (Och et al., 2003), the use of a word graph is proposed as interface between an alignment-template SMT model and the IMT engine. Analogously, in this work we will be using a word graph built during the search procedure performed on a PB SMT model.

During the search process performed by the above mentioned beam search algorithm, it is possible to create a *segment graph*. In such a graph, each node represents a state of the SMT model, and each edge a weighted transition between states labelled with a sequence of target words. Whenever a hypothesis is extended, we add a new edge connecting the state of that hypothesis with the state of the extended hypothesis. The new edge is labelled with the sequence of target words that has been incorporated to the extended hypothesis and is weighted appropriately by means of the score given by the SMT model.

Once the segment graph is generated, it can be easily converted into a word graph by the introduction of artificial states for the words that compose the target phrases associated to the edges.

3.2 IMT using word graphs

During the process of IMT for a given source sentence, the system makes use of the word graph generated for that sentence in order to complete the prefixes accepted by the human translator. Specifically,

SOURCE (x):		Para encender la impresora:
REFERENCE (y):		To power on the printer:
ITER-0	(p) (\hat{s}_h)	() To switch on:
ITER-1	(p) (s_l) (\hat{s}_h)	To switch on: power on the printer:
ITER-2	(p) (s_l) (k) (\hat{s}_h)	To power on the printer: () (#) ()
FINAL	($p \equiv y$)	To power on the printer:

Figure 2: Example of non-explicit positioning MA which solves an error of a missing word. In this case, the system produces the correct suffix s_h immediately after the user validates a prefix p , implicitly indicating that we want the suffix to be changed, without need of any further action. In **ITER-1**, character | indicates the position where a MA was performed, s_l is the suffix which was rejected by that MA, and \hat{s}_h is the new suffix that the system suggests after observing that s_l is to be considered incorrect. Character # is a special character introduced by the user to indicate that the hypothesis is to be accepted.

the system finds the best path in the word graph associated with a given prefix so that it is able to complete the target sentence, being capable of providing several completion suggestions for each prefix.

A common problem in IMT arises when the user sets a prefix which cannot be found in the word graph, since in such a situation the system is unable to find a path through the word graph and provide an appropriate suffix. The common procedure to face this problem is to perform a tolerant search in the word graph. This tolerant search uses the well known concept of Levenshtein distance in order to obtain the most similar string for the given prefix (see (Och et al., 2003) for more details).

4 Enriching user-machine interaction

Although the IMT paradigm has proved to offer interesting benefits to potential users, one aspect that has not been reconsidered as of yet is the user-machine interface. Hence, in traditional IMT the system only received feedback whenever the user typed in a new word. In this work, we show how to enrich user-machine interaction by introducing Mouse Actions (MA) as an additional information source for the system. By doing so, we will consider two types of MAs, i.e. *non-explicit* (or *positioning*) MAs and *interaction-explicit* MAs.

4.1 Non-explicit positioning MAs

Before typing in a new word in order to correct a hypothesis, the user needs to position the cursor in the place where he wants to type such a word. In this work, we will assume that this is done by performing a MA, although the same idea presented can also be applied when this is done by some other means. It is important to point out that, by doing so, the user is already providing some very useful information to the system: he is validating a prefix up to the position where he positioned the cursor, and, in addition, he is signalling that whatever word is located after the cursor is to be considered incorrect. Hence, the system can already capture this fact and provide a new translation hypothesis, in which the prefix remains unchanged and the suffix is replaced by a new one in which the first word is different to the first word of the previous suffix. We are aware that this does not mean that the new suffix will be correct, but given that we know that the first word in the previous suffix was incorrect, the worst thing which can happen is that the the first word of the new suffix is incorrect as well. However, if the new suffix happens to be correct, the user will happily find that he does not need to correct that word any more.

An example of such behaviour can be seen in Figure 2. In this example, the SMT system first provides a translation which the user does not

like. Hence, he positions the cursor before word “*postscript*”, with the purpose of typing in “*lists*”. By doing so, he is validating the prefix “*To print a*”, and signalling that he wants “*postscript*” to be replaced. Before typing in anything, the system realises that he is going to change the word located after the cursor, and replaces the suffix by another one, which is the one the user had in mind in the first place. Finally, the user only has to accept the final translation.

We are naming this kind of MA *non-explicit* because it does not require any additional action from the user: he has already performed a MA in order to position the cursor at the place he wants, and we are taking advantage of this fact to suggest a new suffix hypothesis.

Since the user needs to position the cursor before typing in a new word, it is important to point out that any improvement achieved by introducing non-explicit MAs does not require any further effort from the user, and hence is considered to have no cost.

Hence, we are now considering two different situations: the first one, the traditional IMT framework, in which the system needs to find a suffix according to Eq. 5, and a new one, in which the system needs to find a suffix in which the first word does not need to be a given k , but needs to be *different* to a given s_{l_1} . This constraint can be expressed by the following equation:

$$\hat{s}_h = \operatorname{argmax}_{s_h: s_{h_1} \neq s_{l_1}} Pr(\mathbf{p}, s_h | \mathbf{x}, s_l) \quad (6)$$

where s_l is the suffix generated in the previous iteration, already discarded by the user, and s_{l_1} is the first word in s_l . k is omitted in this formula because the user did not type any word at all.

4.2 Interaction-explicit MAs

If the system is efficient and provides suggestions which are good enough, one could easily picture a situation in which the expert would ask the system to replace a given suffix, without typing in any word. We will be modelling this as another kind of MA, *interaction-explicit* MA, since the user needs to indicate *explicitly* that he wants a given suffix to be replaced, in contrast to the non-explicit positioning MA. However, if the underlying MT engine providing the suffixes is powerful enough, the user would

quickly realise that performing a MA is less costly than introducing a whole new word, and would take advantage of this fact by systematically clicking before introducing any new word. In this case, as well, we assume that the user clicks before an incorrect word, hence demanding a new suffix whose first word is different, but by doing so he is adopting a more participative and interactive attitude, which was not demanded in the case of non-explicit positioning MAs. An example of such an explicit MA correcting an error can be seen in Figure 3

In this case, however, there is a cost associated to this kind of MAs, since the user does need to perform additional actions, which may or may not be beneficial. It is very possible that, even after asking for several new hypothesis, the user will even though need to introduce the word he had in mind, hence wasting the additional MAs he had performed.

If we allow the user to perform n MAs before introducing a word, this problem can be formalised in an analogous way as in the case of non-explicit MAs as follows:

$$\hat{s}_h = \operatorname{argmax}_{s_h: s_{h_1} \neq s_{l_1}^i \forall i \in \{1..n\}} Pr(\mathbf{p}, s_h | \mathbf{x}, s_l^1, s_l^2, \dots, s_l^n) \quad (7)$$

where $s_{l_1}^i$ is the first word of the i -th suffix discarded and $s_l^1, s_l^2, \dots, s_l^n$ is the set of all n suffixes discarded.

Note that this kind of MA could also be implemented with some other kind of interface, e.g. by typing some special key such as F1 or Tab. However, the experimental results would not differ, and in our user interface we found it more intuitive to implement it as a MA.

5 Experimental setup

5.1 System evaluation

Automatic evaluation of results is a difficult problem in MT. In fact, it has evolved to a research field with own identity. This is due to the fact that, given an input sentence, a large amount of correct *and* different output sentences may exist. Hence, there is no sentence which can be considered ground truth, as is the case in speech or text recognition. By extension, this problem is also applicable to IMT.

In this paper, we will be reporting our results as measured by *Word Stroke Ratio* (WSR) (Barrachina

	SOURCE (x):	Seleccione el tipo de instalación.
	REFERENCE (y):	Select the type of installation.
ITER-0	(p) (ŝ _h)	() Select the installation wizard.
ITER-1	(p) (s _i) (ŝ _h)	Select the installation wizard. install script.
ITER-2	(p) (k) (ŝ _h)	Select the type installation wizard.
ITER-3	(p) (s _i) (ŝ _h)	Select the type installation wizard. of installation.
ITER-4	(p) (s _i) (k) (ŝ _h)	Select the type of installation. () (#) ()
FINAL	(p ≡ y)	Select the type of installation.

Figure 3: Example of explicit interactive MA which corrects an erroneous suffix. In this case, a non-explicit MA is performed in **ITER-1** with no success. Hence, the user introduces word “*type*” in **ITER-2**, which leaves the cursor position located immediately after word “*type*”. In this situation the user would not need to perform a MA to reposition the cursor and continue typing in order to further correct the remaining errors. However, since he has learnt the potential benefit of MAs, he performs an interaction-explicit MA in order to ask for a new suffix hypothesis, which happens to correct the error.

and others, 2008), which is computed as the quotient between the number of word-strokes a user would need to perform in order to achieve the translation he has in mind and the total number of words in the sentence. In this context, a word-stroke is interpreted as a single action, in which the user types a complete word, and is assumed to have constant cost. Moreover, each word-stroke also takes into account the cost incurred by the user when reading the new suffix provided by the system.

In the present work, we decided to use WSR instead of *Key Stroke Ratio* (KSR), which is used in other works on IMT such as (Och et al., 2003). The reason for this is that KSR is clearly an optimistic measure, since in such a scenario the user is often overwhelmed by receiving a great amount of translation options, as much as one per key stroke, and it is not taken into account the time the user would need to read all those hypotheses.

In addition, and because we are also introducing MAs as a new action, we will also present results in terms of *Mouse Action Ratio* (MAR), which is the quotient between the amount of explicit MAs per-

formed and the number of words of the final translation. Hence, the purpose is to elicit the number of times the user needed to request a new translation (i.e. performed a MA), on a per word basis.

Lastly, we will also present results in terms of uMAR (useful MAR), which indicates the amount of MAs which were *useful*, i.e. the MAs that actually produced a change in the first word of the suffix and such word was accepted. Formally, uMAR is defined as follows:

$$uMAR = \frac{MAC - n \cdot WSC}{MAC} \quad (8)$$

where *MAC* stands for “Mouse Action Count”, *WSC* for “Word Stroke Count” and *n* is the maximum amount of MAs allowed before the user types in a word. Note that $MAC - n \cdot WSC$ is the amount of MAs that were useful since *WSC* is the amount of word-strokes the user performed even though he had already performed *n* MAs.

Since we will only use single-reference WSR and MAR, the results presented here are clearly pessimistic. In fact, it is relatively common to have the underlying SMT system provide a perfectly correct

Table 1: Characteristics of Europarl for each of the sub-corpora. OoV stands for “Out of Vocabulary” words, Dev. for Development, K for thousands of elements and M for millions of elements.

		De	En	Es	En	Fr	En
Training	Sentences	751K		731K		688K	
	Run. words	15.3M	16.1M	15.7M	15.2M	15.6M	13.8M
	Avg. len.	20.3	21.4	21.5	20.8	22.7	20.1
	Voc.	195K	66K	103K	64K	80K	62K
Dev.	Sentences	2000		2000		2000	
	Run. words	55K	59K	61K	59K	67K	59K
	Avg. len.	27.6	29.3	30.3	29.3	33.6	29.3
	OoV	432	125	208	127	144	138
Test	Sentences	2000		2000		2000	
	Run. words	54K	58K	60K	58K	66K	58K
	Avg. len.	27.1	29.0	30.2	29.0	33.1	29.3
	OoV	377	127	207	125	139	133

translation, which is “corrected” by the IMT procedure into another equivalent translation, increasing WSR and MAR significantly by doing so.

5.2 Corpora

Our experiments were carried out on the Europarl (Koehn, 2005) corpus, which is a corpus widely used in SMT and that has been used in several MT evaluation campaigns. Moreover, we performed our experiments on the partition established for the Workshop on Statistical Machine Translation of the NAACL 2006 (Koehn and Monz, 2006). The Europarl corpus (Koehn, 2005) is built from the proceedings of the European Parliament. Here, we will focus on the German–English, Spanish–English and French–English tasks, since these were the language pairs selected for the cited workshop. The corpus is divided into three separate sets: one for training, one for development, and one for test. The characteristics of the corpus can be seen in Table 1.

5.3 Experimental results

As a first step, we built a SMT system for each of the language pairs cited in the previous subsection. This was done by means of the Moses toolkit (Koehn and others, 2007), which is a complete system for building Phrase-Based SMT models. This toolkit involves the estimation from the training set of four different translation models, which are in turn com-

Table 2: WSR improvement when considering non-explicit MAs. “rel.” indicates the relative improvement. All results are given in %.

pair	baseline	non-explicit	rel.
Es–En	63.0±0.9	59.2±0.9	6.0±1.4
En–Es	63.8±0.9	60.5±1.0	5.2±1.6
De–En	71.6±0.8	69.0±0.9	3.6±1.3
En–De	75.9±0.8	73.5±0.9	3.2±1.2
Fr–En	62.9±0.9	59.2±1.0	5.9±1.6
En–Fr	63.4±0.9	60.0±0.9	5.4±1.4

binated in a log-linear fashion by adjusting a weight for each of them by means of the MERT (Och, 2003) procedure, optimising the BLEU (Papineni et al., 2002) score obtained on the development partition.

This being done, word graphs were generated for the IMT system. For this purpose, we used a multi-stack phrase-based decoder which will be distributed in the near future together with the Thot toolkit (Ortiz-Martínez et al., 2005). We discarded the use of the Moses decoder because preliminary experiments performed with it revealed that the decoder by (Ortiz-Martínez et al., 2005) performs clearly better when used to generate word graphs for use in IMT. In addition, we performed an experimental comparison in regular SMT with the Europarl corpus, and found that the performance difference was negligible. The decoder was set to only consider monotonic translation, since in real IMT scenarios considering non-monotonic translation leads to excessive waiting time for the user.

Finally, the word graphs obtained were used within the IMT procedure to produce the reference translation contained in the test set, measuring WSR and MAR. The results of such a setup can be seen in Table 2. As a baseline system, we report the traditional IMT framework, in which no MA is taken into account. Then, we introduced non-explicit MAs, obtaining an average improvement in WSR of about 3.2% (4.9% relative). The table also shows the confidence intervals at a confidence level of 95%. These intervals were computed following the bootstrap technique described in (Koehn, 2004). Since the confidence intervals do not overlap, it can be stated that the improvements obtained are statistically significant.

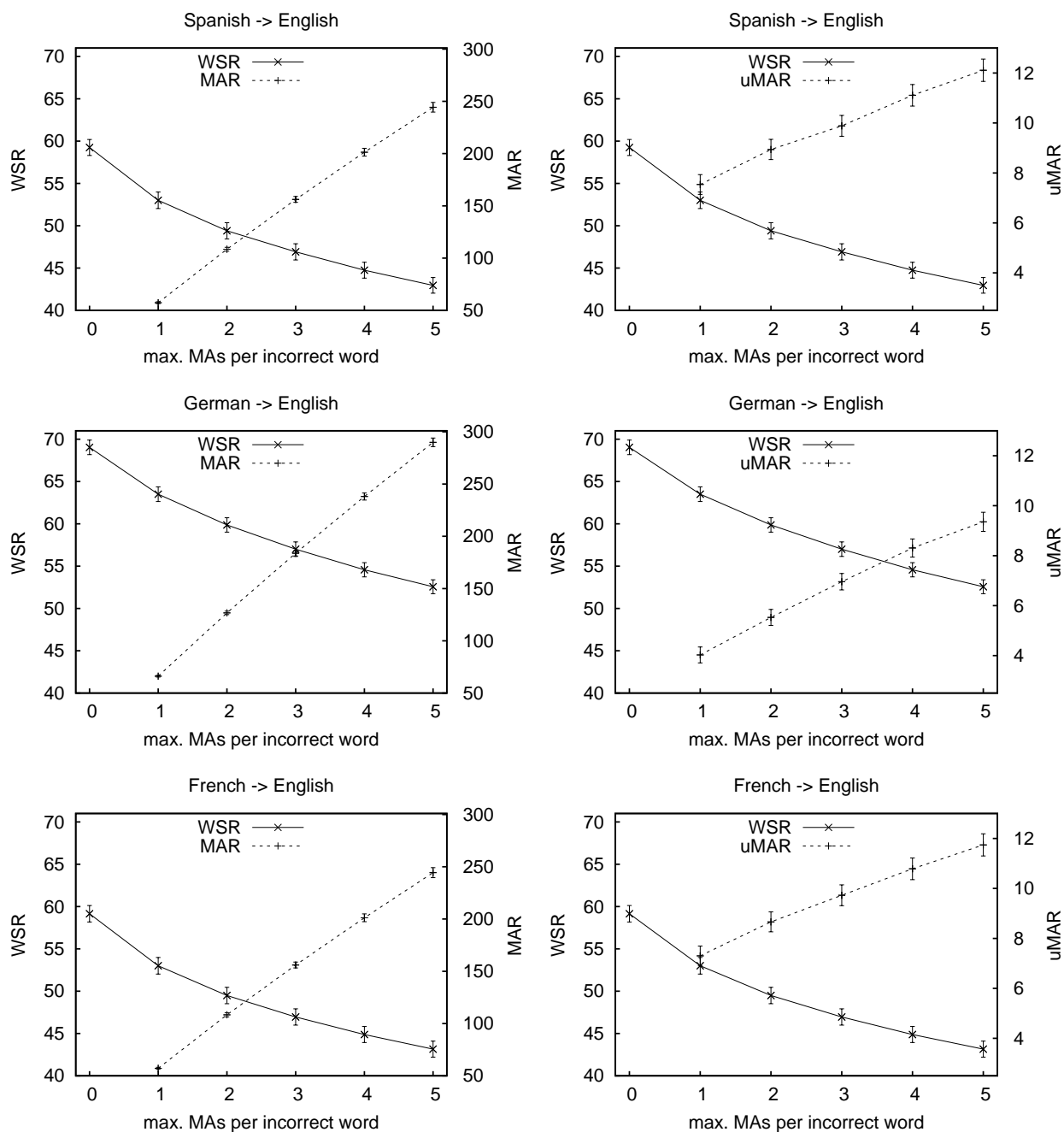


Figure 4: WSR improvement when considering one to five maximum MAs. All figures are given in %. The left column lists WSR improvement versus MAR degradation, and the right column lists WSR improvement versus uMAR. Confidence intervals at 95% confidence level following (Koehn, 2004).

Once the non-explicit MAs were considered and introduced into the system, we analysed the effect of performing up to a maximum of 5 explicit MAs. Here, we modelled the user in such a way that, in case a given word is considered incorrect, he will always ask for another translation hypothesis until

he has asked for as many different suffixes as MAs considered. The results of this setup can be seen in Figure 4. This yielded a further average improvement in WSR of about 16% (25% relative improvement) when considering a maximum of 5 explicit MAs. However, relative improvement in WSR and

uMAR increase drop significantly when increasing the maximum allowed amount of explicit MAs from 1 to 5. For this reason, it is difficult to imagine that a user would perform more than two or three MAs before actually typing in a new word. Nevertheless, just by asking twice for a new suffix before typing in the word he has in mind, the user might be saving about 15% of word-strokes.

Although the results in Figure 4 are only for the translation direction “foreign”→English, the experiments in the opposite direction (i.e. English→“foreign”) were also performed. However, the results were very similar to the ones displayed here. Because of this, and for clarity purposes, we decided to omit them and only display the direction “foreign”→English.

6 Conclusions and future work

In this paper, we have considered new input sources for IMT. By considering Mouse Actions, we have shown that a significant benefit can be obtained, in terms of word-stroke reduction, both when considering only non-explicit MAs and when considering MAs as a way of offering the user several suffix hypotheses. In addition, we have applied these ideas on a state-of-the-art SMT baseline, such as phrase-based models. To achieve this, we have first obtained a word graph for each sentence which is to be translated. Experiments were carried out on a reference corpus in SMT.

Note that there are other systems (Esteban and others, 2004) that, for a given prefix, provide n-best lists of suffixes. However, the functionality of our system is slightly (but fundamentally) different, since the suggestions are demanded to be different in their first word, which implies that the n-best list is scanned deeper, going directly to those hypotheses that may be of interest to the user. In addition, this can be done “on demand”, which implies that the system’s response is faster and that the user is not confronted with a large list of hypotheses, which often results overwhelming.

As future work, we are planning on performing a human evaluation that assesses the appropriateness of the improvements described.

Acknowledgements

This work has been partially supported by the Spanish MEC under scholarship AP2005-4023 and under grants CONSOLIDER Ingenio-2010 CSD2007-00018, and by the EC (FEDER) and the Spanish MEC under grant TIN2006-15694-CO2-01.

References

- D. J. Arnold, 2003. *Computers and Translation: A translator’s guide*, chapter 8, pages 119–142.
- S. Barrachina et al. 2008. Statistical approaches to computer-assisted translation. *Computational Linguistics*, page In press.
- F. Casacuberta and E. Vidal. 2007. Learning finite-state models for machine translation. *Machine Learning*, 66(1):69–91.
- J. Esteban et al. 2004. Transtype2 - an innovative computer-assisted translation system. In *The Companion Volume to the Proc. ACL’04*, pages 94–97.
- G. Foster, P. Langlais, and G. Lapalme. 2002. User-friendly text prediction for translators. In *Proc. of EMNLP’02*, pages 148–155.
- G. Foster. 2002. *Text Prediction for Translators*. Ph.D. thesis, Université de Montréal.
- J. Hutchings and H. Somers. 1992. An introduction to machine translation. In *Ed. Academic Press*.
- J. Hutchins. 1999. Retrospect and prospect in computer-based translation. In *Proc. of MT Summit VII*, pages 30–44.
- M. Kay. 1997. It’s still the proper place. *Machine Translation*, 12(1-2):35–38.
- P. Koehn and C. Monz, editors. 2006. *Proc. of the Workshop on SMT*.
- P. Koehn et al. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. of the ACL’07*.
- P. Koehn, F. J. Och, and D. Marcu. 2003. Statistical phrase-based translation. In *Proc. HLT/NAACL’03*, pages 48–54.
- P. Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP’04*, pages 388–395, Barcelona, Spain.
- P. Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proc. of the MT Summit X*, pages 79–86.
- P. Langlais, G. Lapalme, and M. Loranger. 2002. Transtype: Development-evaluation cycles to boost translator’s productivity. *Machine Translation*, 15(4):77–98.
- Bruce T. Lowerre. 1976. *The harpy speech recognition system*. Ph.D. thesis, Carnegie Mellon University, Pittsburgh, PA, USA.

- F. Och and H. Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of the ACL'02*, pages 295–302.
- F.J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Comput. Linguist.*, 30(4):417–449.
- F. Och, C. Tillmann, and H. Ney. 1999. Improved alignment models for statistical machine translation. In *Proc. of EMNLP/WVLC'99*, pages 20–28.
- F.J. Och, R. Zens, and H. Ney. 2003. Efficient search for interactive statistical machine translation. In *Proc. of EACL'03*, pages 387–393.
- F.J. Och. 2003. Minimum error rate training for statistical machine translation. In *Proc. of ACL'03*, pages 160–167.
- D. Ortiz-Martínez, I. García-Varea, and F. Casacuberta. 2005. Thot: a toolkit to train phrase-based statistical translation models. In *Proc. of the MT Summit X*, pages 141–148.
- K. Papineni, S. Roukos, and T. Ward. 1998. Maximum likelihood and discriminative training of direct translation models. In *Proc. of ICASSP'98*, pages 189–192.
- K. Papineni, S. Roukos, T. Ward, and W.J. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proc. of ACL'02*.
- N. Ueffing, F. Och, and H. Ney. 2002. Generation of word graphs in statistical machine translation. In *Proc. of EMNLP'02*, pages 156–163.
- E. Vidal et al. 2007. Interactive pattern recognition. In *Proc. of MLMI'07*, pages 60–71.
- R. Zens, F.J. Och, and H. Ney. 2002. Phrase-based statistical machine translation. In *Proc. of KI'02*, pages 18–32.