Ranking suggestions for black-box interactive translation prediction systems with multilayer perceptrons

Daniel Torregrosa dtorregrosa@dlsi.ua.es
Juan Antonio Pérez-Ortiz japerez@dlsi.ua.es
Mikel L. Forcada mlf@dlsi.ua.es
Departament de Llenguatges i Sistemes Informàtics
Universitat d'Alacant, E-03690 Sant Vicent del Raspeig, Spain

Abstract

The objective of interactive translation prediction (ITP), a paradigm of computer-aided translation, is to assist professional translators by offering context-based computer-generated suggestions as they type. While most state-of-the-art ITP systems are tightly coupled to a machine translation (MT) system (often created ad-hoc for this purpose), our proposal follows a *resource-agnostic* approach, one that does not need access to the inner workings of the bilingual resources (MT systems or any other bilingual resources) used to generate the suggestions, thus allowing to include new resources almost seamlessly. As we do not expect the user to tolerate more than a few proposals each time, the set of potential suggestions need to be filtered and ranked; the *resource-agnostic* approach has been evaluated before using a set of intuitive length-based and position-based heuristics designed to determine which suggestions to show, achieving promising results. In this paper, we propose a more principled suggestion ranking approach using a regressor (a multilayer perceptron) that achieves significantly better results.

1 Introduction

Translation technologies are frequently used to assist professional translators. Common approaches use machine translation (MT) (Hutchins and Somers, 1992) or translation memories (Somers, 2003, Chapter 3) to produce a first (and usually inadequate) prototype of the translation which is then edited by the professional translator in order to produce a target-language text that is adequate for publishing. In both scenarios, the suggestion may be considered by the professional translators as a source of inspiration: they will assemble the final translation on some occasions by accepting and rearranging parts of the proposal, or on other occasions by introducing their own words when an appropriate equivalent is not found in the suggestion.

Our approach, described in a previous work (Pérez-Ortiz et al., 2014), follows a different paradigm known as *interactive translation prediction* (ITP), which, instead of presenting a translation proposal that gets reshaped by the target-language sentence formed in the translator's mind, focuses on offering translation suggestions as the translation is carried out. Most state-of-the-art ITP approaches obtain the suggestions by means of a modified (or tailor-made) statistical machine translation system (SMT) (Koehn, 2004) that is able to provide additional information (such as word alignments, alternative translations, and scores or probabilities for the translation). These systems are able to leverage more information from the bilingual resource than if it were used unmodified as a black-box, but doing so they inherit the common requirements of SMT,

namely, the dependency on the availability of extensive parallel corpora. It is worth noting that integrating other resources of bilingual information would be almost impossible in this kind of systems, as the ITP tool needs the additional information obtained from the underlying SMT engine.

Unlike those previously described, the approach described here is able to use any bilingual resource capable of delivering one or more translations into target language, regardless of how they are obtained and without the need to modify the resource; suggestions are created by generating all possible sub-segments of words in the source-language sentence (up to a given length) and then querying the available bilingual resources for their translations. The nature of these bilingual resources is not limited to MT systems, but they may also include translation memories, dictionaries, catalogues of bilingual phrases, or any combination of them. A neural-based machine learning algorithm trained on features extracted from the source sentence, from the current prefix of the target sentence, and from the translated sub-segments is used to rank and select which suggestions to show at each time step. Not having to rely on the inner workings of each system allows us to integrate new resources without modifying how the ITP system works; similarly, we do not need to modify the resources in any way. Both these features make it possible to use any resource the professional translator has access to in a seamless way.

These translated sub-segments are then offered to the user as the translation is being typed with the objective of saving keystrokes and, hopefully, time. In previous works we have explored the performance of our approach considering rule-based MT systems (Pérez-Ortiz et al., 2014) and in-domain and out-of-domain SMT systems (Torregrosa et al., 2014), using a naïve strategy based on a number of intuitive heuristics to rank and select a few suggestions that are shown to the user. In this paper, however, we aim to improve the strategy by using machine learning techniques. The improvement presented here is twofold: on the one hand, more coherent and principled models are introduced; on the other hand, the results we achieve are significantly better.

The remainder of the paper is organised as follows. After reviewing the state-of-the-art of ITP in Section 2, we describe our method for generating and ranking translation suggestions from bilingual resources in Section 3, emphasizing the differences between the sounder approach presented in this paper and the former heuristic ranking method. We then introduce in Section 4 our experimental set-up and show the results of its evaluation. Finally, we discuss the results and propose future lines of research in Section 5.

2 Related work

The systems which have most significantly contributed to the field of ITP are those built in the pioneering TransType project (Foster et al., 1997; Langlais et al., 2000), and its continuation, the TransType2 project (Macklovitch, 2006). These systems observe the current partial translation already typed by the user and, by exploiting an embedded statistical MT engine, propose one or more completions that are compatible with the sentence prefix entered by the user. The proposals offered may range from one or several words to a completion of the remainder of the target sentence. An automatic best-scenario evaluation with training and evaluation corpora belonging to the same domain (Barrachina et al., 2009) showed that it might theoretically be possible to use only 20–25% of the keystrokes in comparison with the unassisted translation for English–Spanish translation (both directions) and around 45% for English–French and English–German.

A number of projects continued the research where TransType2 had left off. Caitra (Koehn, 2009) is an ITP tool which uses both the phrase table and the decoder of a statistical MT system to generate suggestions. Researchers at the Universitat Politècnica de València have also made significant improvements to a TransType2-style system such as allowing users to accept discontinuous segments of the suggested translation (Domingo et al., 2016). The CASMACAT

project (Koehn et al., 2015) followed the same line of research, improving ITP using active and on-line learning (Alabau et al., 2014). Commercial translation memory systems also integrate some form of ITP as one of their basic features (see, for example, SDL Trados AutoSuggest 2.0[1]), and new translation tools such as Lilt (Green et al., 2014) focus on delivering ITP on an user-friendly web interface.

3 Method

As already described in other articles (Pérez-Ortiz et al., 2014; Torregrosa et al., 2014), our method starts by generating all possible whole-word sub-segments of the source-language sentence S of lengths $l \in [1, L]$, where L is the maximum source sub-segment length measured in words.[2] The resulting sub-segments are then translated by means of a bilingual resource (or a combination of bilingual resources). The set of *potential proposals* P^S for sentence S is made up of suggestions p, which in turn are made up of a target-language sub-segment t_p and the starting b_p and ending e_p positions of the corresponding sub-segment in S.

These suggestions are then offered as the translation T is being typed.[3] Let $\Pr(x)$ be the character-level set of prefixes of a string x, T_k the k-th word of T, and $\hat{T} = T_1...T_{k-1}\hat{w}$ the partially translated sentence where $\hat{w} \in \Pr(T_k)$ is the currently typed prefix of T_k; we define the set of *compatible suggestions* $P^S_{\text{compatible}}$ as

$$P^S_{\text{compatible}}(\hat{w}) = \{p \in P^S : \hat{w} \in \Pr(t_p)\}$$

For example, given $S =$*"Mi sastre está sano"*, with $L = 2$, the set of potential proposals will be $P^S = \{$ *"My"*, *"My tailor"*, *"tailor"*, *"tailor is"*, *"is"*, *"is healthy"*, *"healthy"*$\}$; with $\hat{T} =$*"My t"* and $\hat{w} =$*"t"*, the set of compatible suggestions would be $P^S_{\text{compatible}} = \{$ *"tailor"*, *"tailor is"*$\}$.

As studied in (Pérez-Ortiz et al., 2014), the number of compatible suggestions depends not only on the value of L, but also on the specific word prefix; for example, when users type the letter *d* when translating a long sentence into Spanish, they will probably obtain a significant number of suggestions starting with *de*[4] originating from sub-segments located in different source positions. Obviously, only suggestions originating from the part of the source sentence currently being translated may be useful, but this position is difficult to determine unambiguously. As we do not expect users to tolerate a long list of suggestions, more elaborated strategies are needed both to rank suggestions and to reduce the list to a manageable size.

3.1 Previous approach

We have already proposed (Pérez-Ortiz et al., 2014) a naive way of ranking suggestions, based on the following assumptions:

- the source-language sentence S and the target-language sentence T have similar lengths, and translation is mainly monotonous; useful suggestions for the n-th word of T will be generated from sub-segments close to the n-th word of S;

[1] `http://www.translationzone.com/products/trados-studio/autosuggest/`

[2] Suitable values for L will depend on the bilingual resource: on the one hand, we expect higher values of L to be useful for high-quality MT systems, such as those translating between closely related languages, since adequate translations may stretch to a relatively large number of words; on the other hand, L should be kept small for low-quality MT systems whose translations quickly deteriorate as the length of the input sub-segment increases; of course, L will be small for resources such as dictionaries.

[3] While T is fixed during automatic tests, professional translators may change their minds during the process.

[4] The preposition *de* ('of') is one of the most frequent words in Spanish texts.

- long suggestions are seldom useful,[5] but when used there is a significant effort reduction;
- short suggestions are usually compatible,[6] but do not save too much effort when used.

We therefore devised the following selection scheme: when the user translates the k-th word, the shortest and longest (measured in number of words) suggestions originated from the closest position in $P^S_{\text{compatible}}$ are offered, followed by the shortest and longest of the second closest position, or, if no other position generated compatible suggestions, the second shortest and the second longest suggestions from the previous position, and so on, up to a maximum number of suggestions M.[7]

This heuristic performed remarkably well in spite of the simplicity of the approach: during a conducted preliminary test using $M = 4$ with human translators (Pérez-Ortiz et al., 2014), savings in the range of 25–65% keystrokes (depending on the language pair) were achieved, without any explicit complaints from users about being offered too many suggestions.

3.2 Neural network model

The approach discussed in the preceding section can still be improved: on the one hand, more rigorous and principled models rather than intuitive heuristics can be used; on the other hand, if we get a better ranking of suggestions we can reduce the number of suggestions offered (reducing the cognitive effort used for reading and selecting suggestions), the number of keystrokes or both. Consequently, we propose to replace the previous intuitive heuristics with a ranker based on a multilayer perceptron (Duda et al., 2000, Chapter 6).Four different systems will be trained, *Full feature set with usable suggestions*, *Full feature set with winning suggestions*, *Reduced feature set with usable suggestions*, and *Reduced feature set with winning suggestions*, depending on the set of features and the kind of suggestions they will learn to identify:

- *Full feature set*: the full feature set that will be discussed in Subsection 3.3;
- *Reduced feature set*: a reduced feature set consisting of only two features, the length of the suggestion and the normalized distance, as will be discussed in Subsection 3.3. This that means that the model has access to similar information to that available to the previous intuitive heuristic method.
- *Winning suggestions*: the set of *winning suggestions*, those that get chosen during the automatic evaluation procedure described in Section 4. Winning suggestions are given a score of 1, and the rest get a score of 0.[8]
- *Usable suggestions*: the set of *usable suggestions*, namely those which, for the current partially translated sentence $\hat{T}$, could be used for advancing the translation, but are not necessarily are part of the suboptimal sequence of actions the greedy automatic evaluation procedure executes. For instance, given $S =$"Un coche rojo", $T =$"A red car", $\hat{T} =$"A r", $p_1 =$"red" and $p_2 =$"red car" both are deemed as usable, even when p_2 would be more advantageous. Usable suggestions are given a score of 1, and the rest get a score of 0.

[5] Our automatic testing strategy (see Section 4) only accepts suggestions that exactly match the provided reference. However, professional translators may accept suggestions that slightly differ from their initial translation, editing the suggestion or even adapting their planned translation. However, we lack a formal model that reflects this behaviour; devising one is a requisite for conveying a more human-like automatic evaluation.

[6] Usually, short words such as determinants, prepositions, and non-ambiguous nouns are correctly translated even by low quality resources.

[7] A system that reversed this order picking the longest and shortest suggestions, a second one picking first the longest of each position and a third one picking first the shortest were also tested, but performed worse.

[8] During training, the desired output values will be 0 and 1, but during testing the network output will be a real value between 0 and 1 that correlates with the goodness of the suggestion.

3.3 Features

We will use features $f_i, 1 \leq i \leq 30$ for each p. As multilayer perceptrons cannot work with nominal features, those will be transformed into multiple one-hot-encoded binary features and then treated as numeric. Likewise, binary features are treated as numeric. The description of this transformation will not be included in the definition of the features to avoid overcomplicating it; when applied, the actual number of features grows to 79.

Length of the suggestion The length of the span the suggestion is generated from $f_1 = e_p - b_p$ and the length of the translated sub-segment $f_2 = |t_p|$, both at the word and the character (f_3, f_4) level. As discussed in Subsection 3.1, long suggestions are seldom used under our testing conditions.

Position of the suggestion A set of features that relate to the position where each suggestion comes from and where it is (potentially) offered. The features include the absolute position in the source sentence $f_5 = b_p$ and the position being currently worked on in the target sentence $f_6 = k$, the position normalized by the length of the sentence $f_7 = b_p/|S|$ and $f_8 = k/|T|$,[9] the distance between source and target positions, both with absolute positions $f_9 = k - b_p$ and their normalized counterparts $f_{10} = k/|T| - b_p/|S|$, and their position ratios $f_{11} = k/b_p$. These features help to determine how far we are from the suggestion source: long sentences will have potentially higher values for the differences, but lower values on the ratio; the opposite stands for short sentences. We also define ($f_{12}...f_{18}$), the equivalent feature set for character level positions and distances.

Position and length A set of 3 nominalized features (f_{19}, f_{20}, f_{21}) that relate to the position and the length of the suggestion. Each feature takes a value in the $\{\mathrm{short}, \mathrm{long}\} \times \{\mathrm{close}, \mathrm{far}\}$ set. We classify a suggestion as short if it has 2 or less words (f_2) or 10 or less characters (f_4), and long otherwise. We classify a suggestion as close if it is 3 or less words away (f_9), 20 or less characters away (f_{15}) or the ratio position (f_{11}) is lower than 1.2; far otherwise. The feature f_{19} uses word-level length and distance, f_{20} uses character-level length and distance, and f_{21} uses word-level length and ratio.

Distance distribution Given a training set, we compute the average $\overline{x}$ and standard deviation σ values for the distribution of normalized distances and position ratios for the *winning suggestions* set (as described in Subsection 3.2), we define four features:

- $f_{22} = (f_{10} - \overline{x})/\sigma$
- a nominal feature f_{23} that has 4 different classes depending on the relationship between f_{10} and the distribution: less than half σ away from $\overline{x}$, a full σ away, 2σ away, or further;
- two more features (f_{24} and f_{25}) similar to the two above, but replacing f_{10} with f_{11}, using their respective average and deviation.

Starting letter of the suggestion As discussed in Section 3, the starting letter of a word is related to the size of the set of compatible suggestions $P^S_{\mathrm{compatible}}$. The nominal feature f_{26} takes the value of the first letter of the suggestion (ignoring the capitalization) if it belongs to the English alphabet, and replaced with a generic $other$ token otherwise.

Last action taken The binary feature f_{27} represents the action we took for the previous word T_{k-1}: whether we typed it or it was part of an accepted suggestion.

[9] During testing, when the length of T cannot be known, we assume $|T| = |S|$.

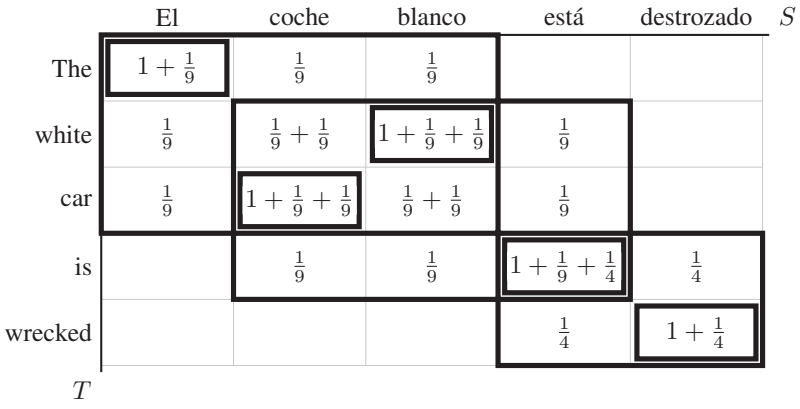

Figure 1: *Alignment strengths* for S =*'El coche blanco está destrozado'*, T =*'The white car is wrecked'*, $P^S_{\text{compatible}}$ = { 'The', 'The white', 'The white car', 'car', 'white car', 'white car is','white', 'white is', 'white is wrecked', 'is', 'is wrecked', 'wrecked'}. Each suggestion is given an *alignment strength* of 1, that gets evenly split along the surface of the suggestion. Some suggestions like 'white is wrecked' cannot be aligned. Each position has a label with the *alignment strength* over it; positions with more *alignment strength* are more likely to be aligned.

Relationship to the last used suggestion The nominal feature f_{28} represents the relationship of the current suggestion p with the last used one p'. It takes 5 possible values:

- p ends before p', $e_p + 1 < b_{p'}$
- p is contiguous and is placed immediately before p', $e_p + 1 = b_{p'}$
- p and p' overlap,[10] $b_p \in [b_{p'}, e_{p'}] \vee e_p \in [b_{p'}, e_{p'}]$
- p is contiguous and is placed immediately after p', $b_p - 1 = e_{p'}$
- p starts after p', $b_p - 1 > e_{p'}$

At the beginning of the translation, where no suggestion has been used, all the suggestions are deemed as belonging to the last category (starts after p').

Light alignment model The light alignment model described in (Esplà-Gomis et al., 2012) performs similarly to other state-of-the-art word-alignment methods using previously existing bilingual resources without needing any training procedure. The model relies on a intuitive idea: each sub-segment that contains S_j (the j-th word of S) whose translation covers T_y (the y-th word of T), and vice-versa, increases the likelihood (measured in *alignment strength*) of S_j and T_y to be aligned. An example of how the model works is shown in Figure 1.

In the same way as the ITP system described here, all possible whole-word sub-segments of S and T up to a given length are generated, and then translated using a bilingual resource (MT in (Esplà-Gomis et al., 2012)), although we cannot use the sub-segments that are the product of translating sub-segments of T: T only becomes available during the process as the user is

[10]This would mean a given part of S participates on the generation of different parts of T, which, in general terms, is unlikely to be desirable.

typing it (which means nothing at all is available at the start); translating these sub-segments as the translator types could degrade the performance of the system to the point that it coud not be effectively used, specially if working with complex MT systems, when the user computer has low processing power or on-line.

We take the original idea one step further: suggestions that cover an area with high *alignment strength* are more likely to be aligned; hence those covering the position currently being worked on (k) are more likely to be used. To this end, we analyze the set of suggestions that overlap with the end of the typed prefix $\hat{T}$. Let Pr and Suf be the character-level set of prefixes and suffixes of a given string, we define $\mathrm{extender}$, the set of suggestions that overlap with and extend the end of the current typed prefix $\hat{T}$

$$\mathrm{extender}(\hat{T}) = \{p \in P^S : \mathrm{Pr}(t_p) \cap \mathrm{Suf}(\hat{T}) \neq \varnothing\}$$

As discussed by Esplà-Gomis et al. (2012), we operate on the idea that sub-segment alignment applies *alignment pressure*: the larger the surface covered, the weaker the confidence in the alignment. Each sub-segment pair is given an *alignment strength* of 1 unit. This strength is split evenly along the *surface* of the suggestion as measured in *square words*. So, the force exerted on each position is

$$v_p = \frac{1}{(e_p - b_p)|t_p|}$$

To estimate which position j of S is being worked on, we look at how much *alignment strength* is exerted on it. For this mean, we define

$$W(j, \hat{T}) = \sum_{p \in \mathrm{extender}(\hat{T})} \begin{cases} v_p & \text{if } j \in [b_p, e_p] \\ 0 & \text{otherwise} \end{cases}$$

As discussed, we interpret high *alignment strength* as high confidence in an alignment; positions (j) of S with higher pressure for the position k of T currently being worked on are more likely to be aligned. Hence, suggestions covering k whose area collects more *alignment strength* have a higher probability of being direct translation of the segment of S being currently translated. Moreover, we do not want to realign already used suggestions: for this mean we define P^S_{accepted}, the set of accepted suggestions. Having this in consideration, we add the combined *alignment strength* pressing the area under each suggestion:

$$f_{29} = \sum_{j \in \mathrm{span}(p)} \begin{cases} W(j, \hat{T}) & \text{if } p \notin P^S_{\mathrm{accepted}} \\ 0 & \text{otherwise} \end{cases} \tag{1}$$

Conversely, we can exploit this alignment model to discredit suggestions originating in positions j that may have been already covered in T. To this end, we calculate the *alignment strength* over the already translated part of the sentence $\overline{T} = T_1 ... T_{k-1}$. Let $\mathrm{occurrences}(t_p, T)$ be the function that returns the number of times t_p appears as a subsequence (substring) of T, we define the set of suggestions that overlap with $\overline{T}$,

$$\mathrm{overlap}(\overline{T}) = \{p \in P^S : \mathrm{occurrences}(t_p, \overline{T}) > 0\}$$

As the target text of each suggestion $\bar{t}_p$ may appear more than once in $\hat{T}$, we are unsure of which alignment is the correct one. For addressing this problem, we split the *alignment strength* between the different matches,

$$u_p = \frac{v_p}{\mathrm{occurrences}(t_p, \overline{T})}$$

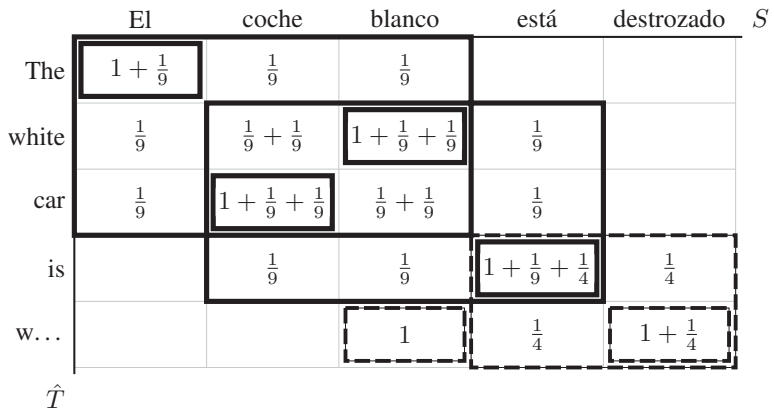

Figure 2: *Alignment strengths* for S ='*El coche blanco está destrozado*', T ='*The white car is wrecked*', $\hat{T}$ ='*The white car is w...*', $P^S_{\text{compatible}}$ = { 'The', 'The white', 'The white car', 'car','white car', 'white car is', 'white', 'white is', 'white is wrecked', 'is', 'is wrecked', 'wrecked'}. Rectangles with solid outlines represent suggestions in overlap, those with dashed outlines represent suggestions in extender. Assuming 'white' $\notin P^S_{\text{accepted}}$ ('white' has not been used when typing $\hat{T}$), 'white' has $f_{26} = 1$, eq. 1 and $f_{27} = 1 - (1 + 6/9)$, eq. 2, 'wrecked' has $f_{26} = f_{27} = 1 + 1/4$

We define the past *alignment strength* function, that includes those suggestions that could have been used for $\overline{T}$, even if those were never offered or used by the translator:

$$W_{\text{past}}(j, \overline{T}) = \sum_{p \in \text{overlap}(\overline{T})} \begin{cases} u_p & \text{if} j \in [b_p, e_p] \\ 0 & \text{otherwise} \end{cases}$$

We define a second feature that discredits suggestions originating from positions that have evidence (in the form of suggestions that overlap with $\overline{T}$) of having already been covered in the translation:

$$f_{30} = \sum_{j \in \text{span}(p)} \begin{cases} W_{\text{past}}(j, \overline{T}) & \text{if } p \notin P^S_{\text{accepted}} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

An example of how this both features work is discussed in Figure 2.

4 Experimental setup and results

As explained in Section 3.2, four different configurations are trained. Training, development and test data are extracted from a corpus with 15,000 English–Spanish sentences extracted from Europarl (Koehn, 2005) version 7, a collection of proceedings from the European Parliament; 11,000 sentences were used as training set, 1,000 as development or validation set and the remaining 3,000 as test set. As a bilingual MT engine capable of providing the translation of the sub-segments we used the free/open-source statistical MT system Moses (Koehn et al., 2007) trained over 155,760 independent sentences from the same corpus, following the standard procedure for training a baseline system.[11] The evaluation was conducted for the translation of texts from English to Spanish.

[11]The corpora is available at `http://www.statmt.org/europarl`. The sentences match those we used in a previous paper (Torregrosa et al., 2014).

To generate the training set, using a source sentence and a reference, an automatic system iteratively considers the first letter of each word and evaluates all the possible suggestions; those that fully match the following words of the reference are tagged as *usable* in that context, and those that would be part of the greedy suboptimal sequence of actions that leads to the best performance are tagged as *winning* in that context.

In order to measure the performance of each configuration, we use the keystroke ratio (KSR), that is, the ratio between the actual number of keys pressed for typing the translation and the length in characters of the translation, as described by Langlais et al. (2000). Accepting a suggestion, no matter its rank, costs one keystroke. It is worth noting that this metric does not measure the time or effort needed to read the suggestions, and does not penalize the offering of inappropriate suggestions in any way. The automatic evaluation system[12] is similar to the one described in the previous work (Pérez-Ortiz et al., 2014): it tries to emulate the behaviour of a user that, given a source sentence S, mentally generates a translation T,[13] then proceeds to type it, reading every offered suggestion and accepting the longest one that matches exactly T, if any. Suggestions need to be full-word translations: if T_k is "thesaurus", the suggestion "the" will not be accepted. The system types the first letter of the next word in T, evaluates all the suggestions, and either chooses the longest one that matches with T or types the rest of the word,[14] until T is completed.[15] The models have been tested with different limits for the maximum number of suggestions $M \in [1, 8]$, sorted in descending order according to the multilayer perceptron output value.

All the multilayer perceptron configurations have three layers: input, hidden and output. Those trained with the reduced feature set only have two input neurons; perceptrons dealing with the full feature set have 75 input units. We will explore different values for the number of units in the hidden layer. All configurations have just one output neuron, and are fully connected. All the neurons but the output neuron have logistic activation functions; as we are estimating a regression function, the output neuron has the identity activation function.

For the training procedure, we use backpropagation with mean squared error (MSE) as the error function to optimize, and no momentum or regularization of any kind. As neural networks have trouble dealing with local minima (Gori and Tesi, 1992), each perceptron has been trained five times with different random initializations. Table 1 shows that there is a strong correlation between MSE and KSR; as a result of this we can presume that the systems with lower minimum squared error MSE will do a better job ranking the suggestions, so we select the model that achieves the lowest MSE out of the 5 different initializations. While the weights were updated after computing the error of every event in the training set, the decision to stop the training (also known as convergence condition) is based on the validation set, in order to minimise the risk of overfitting. We will use these algorithms as implemented in the free open source neural network library FANN (Nissen, 2003).

As hyperparameters, we explored different values for the learning rate and the number of neurons in the hidden layer. All the configurations were tested with learning rates $\alpha \in \{10^{-2}, 10^{-3}, 10^{-4}\}$. The configurations using the reduced feature set were trained and tested with $N \in \{2, 4, 8, 16\}$ neurons in the hidden layer; the ones using the full feature set used

[12] While human evaluation is preferable, it is too expensive for systematically testing every model.

[13] During automatic evaluation, a reference translation is provided.

[14] This differs from the previously described approach, where the suggestions were evaluated after every keypress, rather than only evaluating them at the start of each word.

[15] This evaluation model assumes a *greedy* left-to-right longest-match coverage and, as such, produces a suboptimal solution. Experiments using an optimal coverage strategy that find the global optimum sequence of actions rather than the suboptimal one achieved by the the greedy left-to-right longest match strategy, have been conducted, achieving keystroke ratio improvements under 1%. As such experiments are much more computationally expensive, the left-to-right longest-match strategy is used.

Configuration \ M	1	2	3	4	5	6	7	8
Full set, Usable	0.87	0.88	0.88	0.88	0.88	0.87	0.87	0.86
Full set, winning	0.92	0.95	0.96	0.96	0.96	0.95	0.95	0.95
Reduced set, Usable	0.91	0.94	0.93	0.93	0.93	0.92	0.91	0.91
Reduced set, winning	0.90	0.90	0.89	0.87	0.87	0.86	0.85	0.84

Table 1: Pearson correlation coefficients between minimum squared error (MSE) and keystroke ration (KSR) for each maximum number of suggestions M and configuration. Every configuration shows strong positive correlation between MSE and KSR.

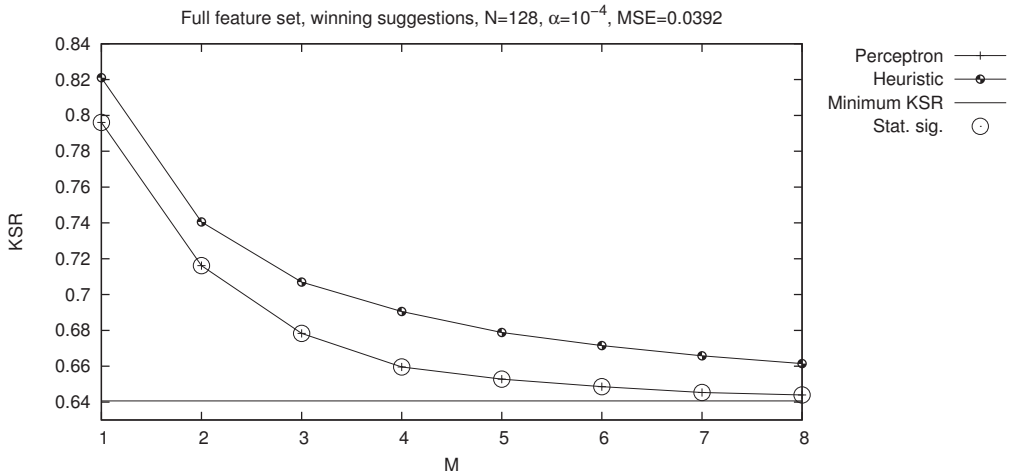

Figure 3: KSR for the multilayer perceptron hyperparameter configuration trained using the *full feature set* and the *winning suggestions* set that got the best MSE. The system beats the baseline in a statistically significant way for every tested value of the maximum number of suggestions M.

$N \in \{2, 8, 32, 128\}$ neurons in the hidden layer.

We will also use the previous intuitive heuristic approach, as described in Subsection 3.1 (referred to as *heuristic* onwards) as a baseline. Paired bootstrap resampling (Koehn, 2004) is performed between the different models (including the *heuristic* approach) using 1000 iterations and $p \leq 0.05$; the best statistically significant KSR values achieved will be denoted with a circle.

Results show that the perceptrons trained with the *winning suggestions* and full feature set (Figure 3) perform notably better than the ones trained with the reduced set (Figure 4). In every figure, the line "Minimum KSR" denotes the best KSR we can achieve offering all the suggestions using the current methodology, and circled points denote the best statistically significant KSR. While several systems trained with the full feature set performed statistically significantly better than the heuristic approach, no reduced feature set system manages to outperform the *heuristic* baseline with $M = [1, 2]$. Those trained with the *usable suggestions* perform similarly to the ones trained with the *winning suggestions*: the best performing systems are not statistically significantly better or worse than the best ones using *winning suggestions* for most values of the maximum number of suggestions M, but, overall, less configurations manage to beat the baseline. Figure 5 compares the performance of the best system for each configuration.

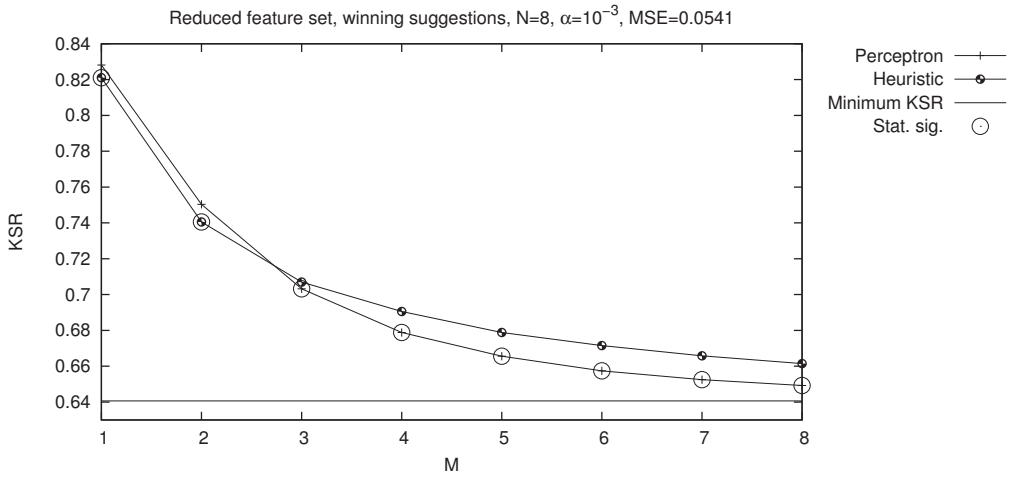

Figure 4: KSR for the multilayer perceptron hyperparameter configuration trained using the *reduced feature set* and the *winning suggestions* set that got the best MSE. The system beats the baseline in a statistically significant way for every tested value of M, but for $M = 1, 2$, where it achieves significantly worse KSR.

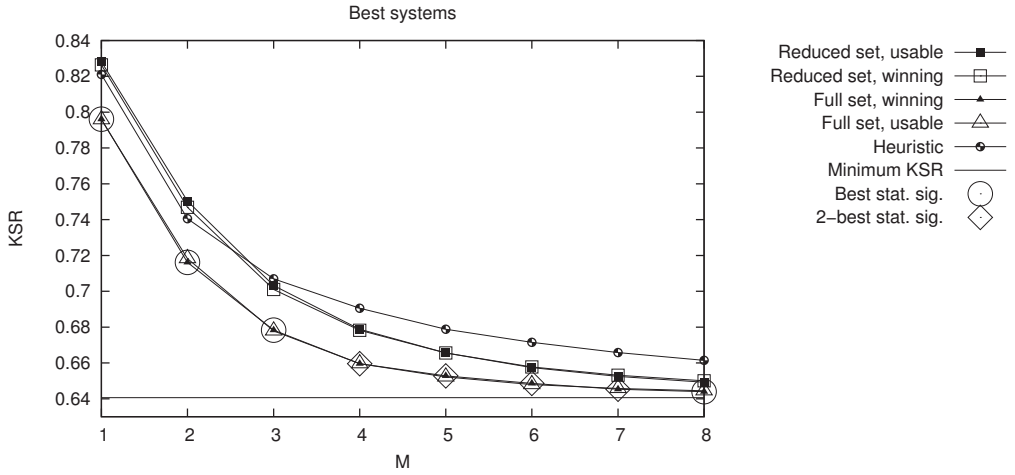

Figure 5: KSR for the best multilayer perceptron hyperparameter configuration for each training set. The best models perform similarly regardless if predicting *winning* or *usable suggestions*, even though the MSE is higher for the ones predicting usable suggestions. Best statistical significance marks those values of the maximum number of suggestions M where the model using the *Full feature set with winning suggestions* is statistically significantly better than the rest; 2-best statistical significance marks those where it is not significantly better or worse than the model using the *Full feature set with usable suggestions*, but both are statistically significantly better than the rest.

5 Conclusions and future work

Resource-agnostic ITP is a low-cost approach for computer-aided translation. We aim to provide a competitive alternative to postediting that can easily integrate any bilingual resource the user has access to.

The naïve distance-based ranking described in (Pérez-Ortiz et al., 2014) provided results comparable to glass-box ITP. We managed to significantly improve it employing a sounder machine-learned model that manages to obtain keystroke savings in the range of 25–45% when offering up to 4 suggestions to the user.

It is important to evaluate this model with more language pairs, specially those which present special interest deemed their grammatical or lexical differences. Also, selecting which features are more representative can further improve the performance of the models while reducing the processing power needed to train and test the model.

There are other ITP systems being currently developed, namely Thot. (Ortiz-Martínez and Casacuberta, 2014) Although it is addressing a different problem by using an ad-hoc SMT system, it assists the user in a similar way. A comparison of the performance achieved will be conducted to contextualize our method.

For further improving the results attained, we plan to use a distortion model as the one proposed by Al-Onaizan and Papineni (2006), that can be used to predict which source words will be translated next, integrating this information as a feature for the multilayer perceptron, hopefully complementing our light alignment model described in Section 3.3. We also plan to use a simplified language model to give a rough estimate of the likelihood of the concatenation of $\hat{T}$ and a given suggestion.

We also plan to explore the impact of simultaneously using different black-box bilingual resources. Different strategies will be evaluated in order to integrate the available resources: combining multiple black-box MT systems as described in (Jayaraman and Lavie, 2005); using confidence-based measures in order to select the most promising translations as performed by Blatz et al. (2004); predicting the best candidates for the translation of each particular sub-segment by using only source-language information, thus avoiding the need to consult every available resource, as explored by Sánchez-Martínez (2011); or letting the multilayer perceptron manage the different suggestions, devising new features if needed.

Finally, we also aim to find evaluation metrics that improve the correlation with professionals by mimicking how a professional translator would work with the tool. Currently, we only choose suggestions that perfectly fit what the professionals are going to type; a translator could however accept a partially matching suggestion and then replace the mismatching part, or even accept a suggestion that does not match the planned translation, adapting it to the new prefix.

Acknowledgments This work has been partially funded by Generalitat Valenciana through grant ACIF/2014/365 from VALi+d programme.

References

Al-Onaizan, Y. and Papineni, K. (2006). Distortion models for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th Annual Meeting of the Association for Computational Linguistics*, pages 529–536.

Alabau, V., González-Rubio, J., Ortíz-Martínez, D., Sanchis-Trilles, G., Casacuberta, F., García-Martínez, M., Mesa-Lao, B., Petersen, D. C., Dragsted, B., and Carl, M. (2014). Integrating online and active learning in a computer-assisted translation workbench. In *Proceedings of the First Workshop on Interactive and Adaptive Statistical Machine Translation, page to appear*, pages 1–8.

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.

Blatz, J., Fitzgerald, E., Foster, G., Gandrabur, S., Goutte, C., Kulesza, A., Sanchis, A., and Ueffing, N. (2004). Confidence estimation for machine translation. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 315–321, Stroudsburg, PA, USA. Association for Computational Linguistics.

Domingo, M., Peris, A., and Casacuberta, F. (2016). Interactive-predictive translation based on multiple word-segments. *Baltic Journal of Modern Computing*, 4(2):282–291.

Duda, R. O., Hart, P. E., and Stork, D. G. (2000). *Pattern Classification*. John Wiley and Sons Inc., second edition.

Esplà-Gomis, M., Sánchez-Martínez, F., and Forcada, M. L. (2012). A simple approach to use bilingual information sources for word alignment. *Procesamiento del Lenguaje Natural*, 49:93–100.

Foster, G. F., Isabelle, P., and Plamondon, P. (1997). Target-text mediated interactive machine translation. *Machine Translation*, 12(1-2):175–194.

Gori, M. and Tesi, A. (1992). On the problem of local minima in backpropagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(1):76–86.

Green, S., Chuang, J., Heer, J., and Manning, C. D. (2014). Predictive translation memory: A mixed-initiative system for human language translation. In *Proceedings of the 27th annual ACM symposium on User interface software and technology*, pages 177–187.

Hutchins, W. J. and Somers, H. L. (1992). *An introduction to machine translation*. Academic Press.

Jayaraman, S. and Lavie, A. (2005). Multi-engine machine translation guided by explicit word matching. In *Proceedings of the ACL 2005 on Interactive poster and demonstration sessions*, pages 101–104.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In *Conference on Empirical Methods on Natural Language Processing*, pages 388–395.

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

Koehn, P. (2009). A web-based interactive computer aided translation tool. In *Proceedings of the ACL-IJCNLP 2009 Software Demonstrations*, pages 17–20.

Koehn, P., Alabau, V., Carl, M., Casacuberta, F., García-Martínez, M., González-Rubio, J., Keller, F., Ortiz-Martínez, D., Sanchis-Trilles, G., Bonk, U. G. R., and and, C. B. (2015). CASMACAT: Final public report. `http://www.casmacat.eu/uploads/Deliverables/final-public-report.pdf`.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., et al. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th annual meeting of the ACL: interactive poster and demonstration sessions*, pages 177–180.

Langlais, P., Sauvé, S., Foster, G., Macklovitch, E., and Lapalme, G. (2000). Evaluation of TransType, a computer-aided translation typing system: a comparison of a theoretical-and a user-oriented evaluation procedures. In *Conference on Language Resources and Evaluation (LREC)*.

Macklovitch, E. (2006). TransType2: The last word. In *Proceedings of the 5th International Conference on Languages Resources and Evaluation (LREC 06)*, pages 167–172.

Nissen, S. (2003). Implementation of a fast artificial neural network library (FANN). Technical report, Department of Computer Science University of Copenhagen (DIKU). `http://fann.sf.net`.

Ortiz-Martínez, D. and Casacuberta, F. (2014). The new thot toolkit for fully automatic and interactive statistical machine translation. In *Proc. of the European Association for Computational Linguistics (EACL): System Demonstrations*, pages 45–48, Gothenburg, Sweden.

Pérez-Ortiz, J. A., Torregrosa, D., and Forcada, M. L. (2014). Black-box integration of heterogeneous bilingual resources into an interactive translation system. *EACL 2014 Workshop on Humans and Computer-assisted Translation*, pages 57–65.

Sánchez-Martínez, F. (2011). Choosing the best machine translation system to translate a sentence by using only source-language information. In *Proceedings of the 15th Annual Conference of the European Associtation for Machine Translation*, pages 97–104.

Somers, H. L. (2003). *Computers and Translation: A Translator's Guide*. Benjamins translation library. John Benjamins Publishing Company.

Torregrosa, D., Forcada, M. L., and Pérez-Ortiz, J. A. (2014). An open-source web-based tool for resource-agnostic interactive translation prediction. *The Prague Bulletin of Mathematical Linguistics*, 102(1):69–80.