The following full text is a publisher’s version.

For additional information about this publication click this link. http://hdl.handle.net/2066/129758

Please be advised that this information was generated on 2018-01-15 and may be subject to change.
Translation Assistance by Translation of L1 Fragments in an L2 Context

Maarten van Gompel & Antal van den Bosch
Centre for Language Studies
Radboud University Nijmegen
proycon@anaproy.nl

Abstract

In this paper we present new research in translation assistance. We describe a system capable of translating native language (L1) fragments to foreign language (L2) fragments in an L2 context. Practical applications of this research can be framed in the context of second language learning. The type of translation assistance system under investigation here encourages language learners to write in their target language while allowing them to fall back to their native language in case the correct word or expression is not known. These code switches are subsequently translated to L2 given the L2 context. We study the feasibility of exploiting cross-lingual context to obtain high-quality translation suggestions that improve over statistical language modelling and word-sense disambiguation baselines. A classification-based approach is presented that is indeed found to improve significantly over these baselines by making use of a contextual window spanning a small number of neighbouring words.

1 Introduction

Whereas machine translation generally concerns the translation of whole sentences or texts from one language to the other, this study focusses on the translation of native language (henceforth L1) words and phrases, i.e. smaller fragments, in a foreign language (L2) context. Despite the major efforts and improvements, automatic translation does not yet rival human-level quality. Vexing issues are morphology, word-order change and long-distance dependencies. Although there is a morpho-syntactic component in this research, our scope is more constrained; its focus is on the faithful preservation of meaning from L1 to L2, akin to the role of the translation model in Statistical Machine Translation (SMT).

The cross-lingual context in our research question may at first seem artificial, but its design explicitly aims at applications related to computer-aided language learning (Laghos and Panayiotis, 2005; Levy, 1997) and computer-aided translation (Barrachina et al., 2009). Currently, language learners need to refer to a bilingual dictionary when in doubt about a translation of a word or phrase. Yet, this problem arises in a context, not in isolation; the learner may have already translated successfully a part of the text into L2 leading up to the problematic word or phrase. Dictionaries are not the best source to look up context; they may contain example usages, but remain biased towards single words or short expressions.

The proposed application allows code switching and produces context-sensitive suggestions as writing progresses. In this research we test the feasibility of the foundation of this idea. The following examples serve to illustrate the idea and demonstrate what output the proposed translation assistance system would ideally produce. The parts in bold correspond to respectively the inserted fragment and the system translation.

- Input (L1=English, L2=Spanish): “Hoy vamos a the swimming pool.” Desired output: “Hoy vamos a la piscina.”
- Input (L1=French, L2=English): “I rentre à la maison because I am tired.” Desired output: “I return home because I am tired.”
- Input (L1=Dutch, L2=English): “Workers are facing a massive aanval op their employ-
ment and social rights."
Desired output: “Workers are facing a massive attack on their employment and social rights.”

The main research question in this research is how to disambiguate an L1 word or phrase to its L2 translation based on an L2 context, and whether such cross-lingual contextual approaches provide added value compared to baseline models that are not context informed or compared to standard language models.

2 Data preparation

Preparing the data to build training and test data for our intended translation assistance system is not trivial, as the type of interactive translation assistant we aim to develop does not exist yet. We need to generate training and test data that realistically emulates the task. We start with a parallel corpus that is tokenised for both L1 and L2. No further linguistic processing such as part-of-speech tagging or lemmatisation takes place in our experiments; adding this remains open for future research.

The parallel corpus is randomly sampled into two large and equally-sized parts. One is the basis for the training set, and the other is the basis for the test set. The reason for such a large test split shall become apparent soon.

From each of the splits (S), a phrase-translation table is constructed automatically in an unsupervised fashion. This is done using the scripts provided by the Statistical Machine Translation system Moses (Koehn et al., 2007). It invokes GIZA++ (Och and Ney, 2000) to establish statistical word alignments based on the IBM Models and subsequently extracts phrases using the grow-diag-final algorithm (Och and Ney, 2003). The result, independent for each set, will be a phrase-translation table (T) that maps phrases in L1 to L2. For each phrase-pair (f_s, f_t) this phrase-translation table holds the computed translation probabilities \( P(f_s|f_t) \) and \( P(f_t|f_s) \).

Given these phrase-translation tables, we can now extract both training data and test data using the algorithm in Figure 1. In our discourse, the source language (s) corresponds to L1, the fallback language used for by the end-user for inserting fragments, whilst the target language (t) is L2.

Step 4 is effectively a filter: two thresholds can be configured to discard weak alignments, i.e. those with low probabilities, from the phrase-translation table so that only strong couplings make it into the generated set. The parameter \( \lambda_1 \) adds a constraint based on the product of the two conditional probabilities \( P(f_s|f_t) \cdot P(f_t|f_s) \), and sets a threshold that has to be surpassed. A second parameter \( \lambda_2 \) further limits the considered phrase pairs \( (f_s, f_t) \) to have the product of their conditional probabilities not not deviate more than a fraction \( \lambda_2 \) from the joint probability for the strongest possible pairing for \( f_s \), the source fragment. \( f_{\text{strongest,t}} \) in Figure 1 corresponds to the best scoring translation for a given source fragment \( f_s \). This metric thus effectively prunes weaker alternative translations in the phrase-translation table from being considered if there is a much stronger candidate. Nevertheless, it has to be noted that even with \( \lambda_1 \) and \( \lambda_2 \), the test set will include a certain amount of errors. This is due to the nature of the unsupervised method with which the phrase-translation table is constructed. For our purposes however, the test set suffices to test our hypothesis.

Figure 1: Algorithm for extracting training and test data on the basis of a phrase-translation table (T) and subset/split from a parallel corpus (S). The indentation indicates the nesting.
In our experiments, we choose fixed values for these parameters, by manual inspection and judgement of the output. The $\lambda_1$ parameter was set to 0.01 and $\lambda_2$ to 0.8. Whilst other thresholds may possibly produce cleaner sets, this is hard to evaluate as finding optimal values causes a prohibitive increase in complexity of the search space, and again this is not necessary to test our hypothesis.

The output of the algorithm in Figure 1 is a modified set of sentence pairs $(sentence'_t, sentence_t)$, in which the same sentence pair may be used multiple times with different L1 substitutions for different fragments. The final test set is created by randomly sampling the desired number of test instances.

Note that the training set and test set are constructed on their own respective and independently generated phrase-translation tables. This ensures complete independence of training and test data. Generating test data using the same phrase-translation table as the training data would introduce a bias. The fact that a phrase-translation table needs to be constructed for the test data is also the reason that the parallel corpus split from which the test data is derived has to be large enough, ensuring better quality.

We concede that our current way of testing is a mere approximation of the real-world scenario. An ideal test corpus would consist of L2 sentences with L1 fallback as crafted by L2 language learners with an L1 background. However, such corpora do not exist as yet. Nevertheless, we hope to show that our automated way of test set generation is sufficient to test the feasibility of our core hypothesis that L1 fragments can be translated to L2 using L2 context information.

3 System

We develop a classifier-based system composed of so-called “classifier experts”. Numerous classifiers are trained and each is an expert in translating a single word or phrase. In other words, for each word type or phrase type that occurs as a fragment in the training set, and which does not map to just a single translation, a classifier is trained. The classifier maps the L1 word or phrase in its L2 context to its L2 translation. Words or phrases that always map to a single translation are stored in a simple mapping table, as a classifier would have no added value in such cases. The classifiers use the IB1 algorithm (Aha et al., 1991) as implemented in TiMBL (Daelemans et al., 2009). IB1 implements $k$-nearest neighbour classification. The choice for this algorithm is motivated by the fact that it handles multiple classes with ease, but first and foremost because it has been successfully employed for word sense disambiguation in other studies (Hoste et al., 2002; Decadt et al., 2004), in particular in cross-lingual word sense disambiguation, a task closely resembling our current task (van Gompel and van den Bosch, 2013). It has also been used in machine translation studies in which local source context is used to classify source phrases into target phrases, rather than looking them up in a phrase table (Stroppa et al., 2007; Haque et al., 2011). The idea of local phrase selection with a discriminative machine learning classifier using additional local (source-language) context was introduced in parallel to Stroppa et al. (2007) by Carpuat and Wu (2007) and Giménez and Márquez (2007); cf. Haque et al. (2011) for an overview of more recent methods.

The feature vector for the classifiers represents a local context of neighbouring words, and optionally also global context keywords in a binary-valued bag-of-words configuration. The local context consists of an $X$ number of L2 words to the left of the L1 fragment, and $Y$ words to the right.

When presented with test data, in which the L1 fragment is explicitly marked, we first check whether there is ambiguity for this L1 fragment and if a direct translation is available in our simple mapping table. If so, we are done quickly and need not rely on context information. If not, we check for the presence of a classifier expert for the offered L1 fragment; only then can we proceed by extracting the desired number of L2 local context words to the immediate left and right of this fragment and adding those to the feature vector. The classifier will return a probability distribution of the most likely translations given the context and we can replace the L1 fragment with the highest scoring L2 translation and present it back to the user.

In addition to local context features, we also experimented with global context features. These are a set of L2 contextual keywords for each L1 word/phrase and its L2 translation occurring in the same sentence, not necessarily in the immediate neighbourhood of the L1 word/phrase. The keywords are selected to be indicative for a specific

\[^1\text{http://ilk.uvt.nl/timbl}\]
translation. We used the method of extraction by Ng and Lee (1996) and encoded all keywords in a binary bag of words model. The experiments however showed that inclusion of such keywords did not make any noticeable impact on any of the results, so we restrict ourselves to mentioning this negative result.

Our full system, including the scripts for data preparation, training, and evaluation, is implemented in Python and freely available as open-source from http://github.com/proycon/colibrita/. Version tag v0.2.1 is representative for the version used in this research.

3.1 Language Model

We also implement a statistical language model as an optional component of our classifier-based system and also as a baseline to compare our system to. The language model is a trigram-based back-off language model with Kneser-Ney smoothing, computed using SRILM (Stolcke, 2002) and trained on the same training data as the translation model. No additional external data was brought in, to keep the comparison fair.

For any given hypothesis \( H \), results from the L1 to L2 classifier are combined with results from the L2 language model. We do so by normalising the class probability from the classifier (score\(_T\)(H)), which is our translation model, and the language model (score\(_{lm}\)(H)), in such a way that the highest classifier score for the alternatives under consideration is always 1.0, and the highest language model score of the sentence is always 1.0. Take score\(_T\)(H) and score\(_{lm}\)(H) to be log probabilities, the search for the best (most probable) translation hypothesis \( \hat{H} \) can then be expressed as:

\[
\hat{H} = \arg \max_{\hat{H}} (\text{score}_T(H) + \text{score}_{lm}(H)) \tag{1}
\]

If desired, the search can be parametrised with variables \( \lambda_3 \) and \( \lambda_4 \), representing the weights we want to attach to the classifier-based translation model and the language model, respectively. In the current study we simply left both weights set to one, thereby assigning equal importance to translation model and language model.

4 Evaluation

Several automated metrics exist for the evaluation of L2 system output against the L2 reference output in the test set. We first measure absolute accuracy by simply counting all output fragments that exactly match the reference fragments, as a fraction of the total amount of fragments. This measure may be too strict, so we add a more flexible word accuracy measure which takes into account partial matches at the word level. If output \( o \) is a subset of reference \( r \) then a score of \( \frac{|r|}{|o|} \) is assigned for that sentence pair. If instead, \( r \) is a subset of \( o \), then a score of \( \frac{|o|}{|r|} \) will be assigned. A perfect match will result in a score of 1 whereas a complete lack of overlap will be scored 0. The word accuracy for the entire set is then computed by taking the sum of the word accuracies per sentence pair, divided by the total number of sentence pairs.

We also compute a recall metric that measures the number of fragments that the system provided a translation for as a fraction of the total number of fragments in the input, regardless of whether the fragment is translated correctly or not. The system may skip fragments for which it can find no solution at all.

In addition to these, the system's output can be compared against the L2 reference translation(s) using established Machine Translation evaluation metrics. We report on BLEU, NIST, METEOR, and word error rate metrics WER and PER. These scores should generally be much better than the typical MT system performances as only local changes are made to otherwise “perfect” L2 sentences.

5 Baselines

A context-insensitive yet informed baseline was constructed to assess the impact of L2 context information in translating L1 fragments. The baseline selects the most probable L1 fragment per L2 fragment according to the phrase-translation table. This baseline, henceforth referred to as the 'most likely fragment' baseline (MLF) is analogous to the 'most frequent sense'-baseline common in evaluating WSD systems.

A second baseline was constructed by weighing the probabilities from the translation table directly with the L2 language model described earlier. It adds a LM component to the MLF baseline. This LM baseline allows the comparison of classification through L1 fragments in an L2 context, with a more traditional L2 context modelling (i.e. target language modelling) which is also cus-
tory in MT decoders. Computing this baseline is done in the same fashion as previously illustrated in Equation 1, where score then represents the normalised \( p(t|s) \) score from the phrase-translation table rather than the class probability from the classifier.

6 Experiments & Results

The data for our experiments were drawn from the Europarl parallel corpus (Koehn, 2005) from which we extracted two sets of 200,000 sentence pairs each for several language pairs. These were used to form the training and test sets. The final test sets are a randomly sampled 5,000 sentence pairs from the 200,000-sentence test split for each language pair.

All input data for the experiments in this section are publicly available\(^2\).

Let us first zoom in to convey a sense of scale on a specific language pair. The actual Europarl training set we generate for English (L1) to Spanish (L2), i.e. English fallback in a Spanish context, consists of 5,608,015 sentence pairs. This number is much larger than the 200,000 we mentioned before because single sentence pairs may be reused multiple times with different marked fragments. From this training set of sentence pairs over 100,000 classifier experts are derived. The eleven largest classifiers are shown in Table 1, along with the number of training instances per classifier. The full table would reveal a Zipfian distribution.

<table>
<thead>
<tr>
<th>Fragment</th>
<th>Training instances</th>
<th>Translations</th>
</tr>
</thead>
<tbody>
<tr>
<td>the</td>
<td>256,772</td>
<td>la, el, los, las</td>
</tr>
<tr>
<td>of</td>
<td>139,273</td>
<td>de, del</td>
</tr>
<tr>
<td>and</td>
<td>128,074</td>
<td>y, de, e</td>
</tr>
<tr>
<td>to</td>
<td>66,565</td>
<td>a, para, que, de</td>
</tr>
<tr>
<td>a</td>
<td>54,306</td>
<td>un, una</td>
</tr>
<tr>
<td>is</td>
<td>40,511</td>
<td>es, está, se</td>
</tr>
<tr>
<td>for</td>
<td>34,054</td>
<td>para, de, por</td>
</tr>
<tr>
<td>this</td>
<td>29,691</td>
<td>este, esta, esto</td>
</tr>
<tr>
<td>European</td>
<td>26,543</td>
<td>Europea, Europeo</td>
</tr>
<tr>
<td>on</td>
<td>23,147</td>
<td>sobre, en</td>
</tr>
<tr>
<td>of the</td>
<td>22,361</td>
<td>de la, de los</td>
</tr>
</tbody>
</table>

Table 1: The top eleven classifier experts for English to Spanish. The eleventh entry is included as an example of a common phrasal fragment

Among the classifier experts are only words and phrases that are ambiguous and may thus map to multiple translations. This implies that such words and phrases must have occurred at least twice in the corpus, though this threshold is made configurable and could have been set higher to limit the number of classifiers. The remaining 246,380 unambiguous mappings are stored in a separate mapping table.

For the classifier-based system, we tested various different feature vector configurations. The first experiment, of which the results are shown in Figure 2, sets a fixed and symmetric local context size across all classifiers, and tests three context widths. Here we observe that a context width of one yields the best results. The BLEU scores, not included in the figure but shown in Table 2, show a similar trend. This trend holds for all the MT metrics.

Table 2 shows the results for English to Spanish in more detail and adds a comparison with the two baseline systems. The various \( lXrY \) configurations use the same feature vector setup for all classifier experts. Here \( X \) indicates the left context size and \( Y \) the right context size. The auto configuration does not uniformly apply the same feature vector setup to all classifier experts but instead seeks to find the optimal setup per classifier expert. This shall be further discussed in Section 6.1.

As expected, the LM baseline substantially outperforms the context-insensitive MLF baseline. Second, our classifier approach attains a substantially higher accuracy than the LM baseline. Third, we observe that adding the language model to our classifier leads to another significant gain

\(^2\)Download and unpack http://lst.science.ru.nl/~proycon/colibrita-acl2014-data.zip
Table 2: Europarl results for English to Spanish (i.e. English fallback in Spanish context). Recall = 0.9422

<table>
<thead>
<tr>
<th>Configuration</th>
<th>Accuracy</th>
<th>Word Accuracy</th>
<th>BLEU</th>
<th>METEOR</th>
<th>NIST</th>
<th>WER</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>MLF baseline</td>
<td>0.6164</td>
<td>0.6662</td>
<td>0.972</td>
<td>0.9705</td>
<td>17.0784</td>
<td>1.4465</td>
<td>1.4209</td>
</tr>
<tr>
<td>LM baseline</td>
<td>0.7158</td>
<td>0.7434</td>
<td>0.9785</td>
<td>0.9739</td>
<td>17.1573</td>
<td>1.1735</td>
<td>1.1574</td>
</tr>
<tr>
<td>1l1r1</td>
<td>0.7588</td>
<td>0.7824</td>
<td>0.9801</td>
<td>0.9747</td>
<td>17.1550</td>
<td>1.1625</td>
<td>1.1444</td>
</tr>
<tr>
<td>1l2r2</td>
<td>0.7574</td>
<td>0.7801</td>
<td>0.9800</td>
<td>0.9746</td>
<td>17.1550</td>
<td>1.1750</td>
<td>1.1569</td>
</tr>
<tr>
<td>1l3r3</td>
<td>0.7514</td>
<td>0.7742</td>
<td>0.9796</td>
<td>0.9744</td>
<td>17.1445</td>
<td>1.1946</td>
<td>1.1780</td>
</tr>
<tr>
<td>1l1r1+LM</td>
<td>0.7810</td>
<td>0.7973</td>
<td>0.9816</td>
<td>0.9754</td>
<td>17.1685</td>
<td>1.0946</td>
<td>1.077</td>
</tr>
<tr>
<td>auto</td>
<td>0.7626</td>
<td>0.7850</td>
<td>0.9803</td>
<td>0.9748</td>
<td>17.1544</td>
<td>1.1594</td>
<td>1.1424</td>
</tr>
<tr>
<td>auto+LM</td>
<td>0.7796</td>
<td>0.7966</td>
<td>0.9815</td>
<td>0.9754</td>
<td>17.1664</td>
<td>1.1021</td>
<td>1.0845</td>
</tr>
<tr>
<td>1l1r0</td>
<td>0.6924</td>
<td>0.7223</td>
<td>0.9757</td>
<td>0.9723</td>
<td>17.1087</td>
<td>1.3415</td>
<td>1.3249</td>
</tr>
<tr>
<td>1l2r0</td>
<td>0.6960</td>
<td>0.7245</td>
<td>0.9759</td>
<td>0.9724</td>
<td>17.1091</td>
<td>1.3364</td>
<td>1.3193</td>
</tr>
<tr>
<td>1l2r1</td>
<td>0.7624</td>
<td>0.7849</td>
<td>0.9803</td>
<td>0.9748</td>
<td>17.1558</td>
<td>1.1554</td>
<td>1.1378</td>
</tr>
</tbody>
</table>

Statistical significance on the BLEU scores was tested using pairwise bootstrap sampling (Koehn, 2004). All significance tests were performed with 5,000 iterations. We compared the outcomes of several key configurations. We first tested 1l1r1 against both baselines; both differences are significant at $p < 0.01$ for both. The same significance level was found when comparing 1l1r1+LM against 1l1r1, auto+LM against auto, as well as the LM baseline against the MLF baseline. Automatic feature selection auto was found to perform statistically better than 1l1r1, but only at $p < 0.05$. Conclusions with regard to context width may have to be tempered somewhat, as the performance of the 1l1r1 configuration was found to not be significantly better than that of the 1l2r2 configuration. However, 1l1r1 performs significantly better than 1l3r3 at $p < 0.01$, and 1l2r2 performs significantly better than 1l3r3 at $p < 0.01$.

In Table 3 we present some illustrative examples from the English—Spanish Europarl data. We show the difference between the most-likely-fragment baseline and our system.

Likewise, Table 4 exemplifies small fragments from the 1l1r1 configuration compared to the same configuration enriched with a language model. We observe in this data that the language model often has the added power to choose a correct translation that is not the first prediction of the classifier, but one of the weaker alternatives that nevertheless fits better. Though the classifier generally works best in the 1l1r1 configuration, i.e. with context size one, the trigram-based language model allows further left-context information to be incorporated that influences the weights of the classifier output, successfully forcing the system to select alternatives. This combination of a classifier with context size one and trigram-based language model proves to be most effective and reaches the best results so far. We have not conducted experiments with language models of other orders.

### 6.1 Context optimisation

It has been argued that classifier experts in a word sense disambiguation ensemble should be individually optimised (Decadt et al., 2004; van Gompel and van den Bosch, 2013). The latter study on cross-lingual WSD finds a positive impact when conducting feature selection per classifier. This intuitively makes sense; a context of one may seem to be better than any other when uniformly applied to all classifier experts, but it may well be that certain classifiers benefit from different feature selections. We therefore proceed with this line of investigation as well.

Automatic configuration selection was done by performing leave-one-out testing (for small number of instances) or 10-fold-cross validation (for larger number of instances, $n \geq 20$) on the training data per classifier expert. Various configurations were tested. Per classifier expert, the best scoring configuration was selected, referred to as the auto configuration in Table 2. The auto configuration improves results over the uniformly
Mientras no haya prueba en contrario, la financiaci´on de partidos pol´ıticos s´olo se justifica, incluso despu´es del tratado de Niza, desde el momento en que concurra a la expresi´on del sufragio universal, que es la ´unica definici´on aceptable de un partido pol´ıtico.

Esta Directiva es nuestra oportunidad to marcar una verdadera diferencia, reduciendo la tr´agica p´erdida de vidas en nuestras carreteras.

Es la pasado vez que me dirijo a esta C´amara.

Pero el enfoque actual de la Comisi´on no puede conducir a una buena pol´ıtica ya que es tributario del funcionamiento del mercado y de las normas establecidas por la OMC, el FMI y el Banco Mundial, normas que siguen siendo desfavorables para los developing countries.

Table 3: Some illustrative examples of MLF-baseline output versus system output, in which system output matches the correct human reference output. The actual fragments concerned are highlighted in bold. The first example shows our system correcting for number agreement, the second a correction in selecting the right preposition, and the third shows that the English word last can be translated in different ways, only one of which is correct in this context. The last example shows a phrasal translation, in which the determiner was duplicated in the baseline output.

Table 5: Frequency of automatically selected configurations on English to Spanish Europarl dataset this is done, so we do not expect it to make a positive impact. The second reason for omitting this is more practical in nature; to do this in combination with feature selection would add substantial search complexity, making experiments far more time consuming, even prohibitively so.

The bottom lines in Table 2 represent results when all right-context is omitted, emulating a real-time prediction when no right context is available yet. This has a substantial negative impact on re-

Table 3: Some illustrative examples of MLF-baseline output versus system output, in which system output matches the correct human reference output. The actual fragments concerned are highlighted in bold. The first example shows our system correcting for number agreement, the second a correction in selecting the right preposition, and the third shows that the English word last can be translated in different ways, only one of which is correct in this context. The last example shows a phrasal translation, in which the determiner was duplicated in the baseline output.

Table 5: Frequency of automatically selected configurations on English to Spanish Europarl dataset this is done, so we do not expect it to make a positive impact. The second reason for omitting this is more practical in nature; to do this in combination with feature selection would add substantial search complexity, making experiments far more time consuming, even prohibitively so.
Sin ese tipo de protección la gente no aprovecharía la oportunidad para vivir, viajar y trabajar donde les parezca en la Unión Europea.

La Comisión también está acometiendo medidas en el ámbito social y educativas con vistas a mejorar la situación de los niños.

Table 4: Some examples of l1r1 versus the same configuration enriched with a language model.

In order to draw accurate conclusions, experiments on a single data set and language pair are not sufficient. We therefore conducted a number of experiments with other language pairs, and present the abridged results in Table 6.

There are some noticeable discrepancies for some experiments in Table 6 when compared to our earlier results in Table 2. We see that the language model baseline for English→French shows the same substantial improvement over the baseline as our English→Spanish results. The same holds for the Chinese→English experiment. However, for English→Dutch and English→Chinese we find that the LM baseline actually performs slightly worse than baseline. Nevertheless, in all these cases, the positive effect of including a Language Model to our classifier-based system again shows. Also, we note that in all cases our system performs better than the two baselines.

Another discrepancy is found in the BLEU scores of the English→Chinese experiments, where we measure an unexpected drop in BLEU score under baseline. However, all other scores do show the expected improvement. The error rate metrics show improvement as well. We therefore attach low importance to this deviation in BLEU here.

In all of the aforementioned experiments, the system produced a single solution for each of the fragments, the one it deemed best, or no solution at all if it could not find any. Alternative evaluation metrics could allow the system to output multiple alternatives. Omission of a solution by definition causes a decrease in recall. In all of our experiments recall is high (well above 90%), mostly because train and test data lie in the same domain and have been generated in the same fashion, lower recall is expected with more real-world data.

7 Discussion and conclusion

In this study we have shown the feasibility of a classifier-based translation assistance system in which L1 fragments are translated in an L2 context, in which the classifier experts are built individually per word or phrase. We have shown that such a translation assistance system scores both above a context-insensitive baseline, as well as an L2 language model baseline.

Furthermore, we found that combining this cross-language context-sensitive technique with an L2 language model boosts results further.

The presence of a one-word right-hand side context proves crucial for good results, which has implications for practical translation assistance application that translate as soon as the user finishes an L1 fragment. Revisiting the translation when right context becomes available would be advisable.

We tested various configurations and conclude that small context sizes work better than larger ones. Automated configuration selection had positive results, yet the system with context size one and an L2 language model component often produces the best results. In static configurations, the failure of a wider context window to be more suc-
Table 6: Results on different datasets and language pairs. The iwslt12ted set is the dataset used in the IWSLT 2012 Evaluation Campaign (Federico et al., 2012), and is formed by a collection of transcriptions of TED talks. Here we used of just over 70,000 sentences for training. Recall for each of the four datasets is 0.9498 (en-nl), 0.9494 (en-fr), 0.9386 (en-zh), and 0.9366 (zh-en).

The idea of a comprehensive translation assistance system may extend beyond the translation of L1 fragments in an L2 context. There are more NLP components that might play a role if such a system were to find practical application. Word completion or predictive editing (in combination with error correction) would for instance seem an indispensable part of such a system, and can be implemented alongside the technique proposed in this study. A point of more practically-oriented future research is to see how feasible such combinations are and what techniques can be used.

An application of our idea outside the area of translation assistance is post-correction of the output of some MT systems that, as a last-resort heuristic, copy source words or phrases into their output, producing precisely the kind of input our system is trained on. Our classification-based approach may be able to resolve some of these cases operating as an add-on to a regular MT system – or as a independent post-correction system.

Our system allows L1 fragments to be of arbitrary length. If a fragment was not seen during training stage, and is therefore not covered by a classifier expert, then the system will be unable to translate it. Nevertheless, if a longer L1 fragment can be decomposed into subfragments that are known, then some recombination of the translations of said sub-fragments may be a good translation for the whole. We are currently exploring this line of investigation, in which the gap with MT narrows further.

Finally, an important line of future research is the creation of a more representative test set. Lacking an interactive system that actually does what we emulate, we hypothesise that good approximations would be to use gap exercises, or cloze tests, that test specific aspects difficulties in language learning. Similarly, we may use L2 learner corpora with annotations of code-switching points or errors. Here we then assume that places where L2 errors occur may be indicative of places where L2 learners are in some trouble, and might want to fall back to generating L1. By then manually translating gaps or such problematic fragments into L1 we hope to establish a more realistic test set.

References


