Translation Spotting for Translation Memories

Michel Simard
Laboratoire de recherche appliquée en linguistique informatique (RALI)
Département d'informatique et de recherche opérationnelle
Université de Montréal
C.P. 6128, succursale Centre-ville, Local 2241
Montréal (Québec), Canada H3C 3J7
simardm@iro.umontreal.ca

Abstract

The term *translation spotting* (TS) refers to the task of identifying the target-language (TL) words that correspond to a given set of source-language (SL) words in a pair of text segments known to be mutual translations. This article examines this task within the context of a sub-sentential translation-memory system, i.e. a translation support tool capable of proposing translations for portions of a SL sentence, extracted from an archive of existing translations. Different methods are proposed, based on a statistical translation model. These methods take advantage of certain characteristics of the application, to produce TL segments submitted to constraints of *contiguity* and *compositionality*. Experiments show that imposing these constraints allows important gains in accuracy, with regard to the most probable alignments predicted by the model.

1 Introduction

Translation spotting is the term coined by Véronis and Langlais (2000) for the task of identifying the word-tokens in a target-language (TL) translation that correspond to some given word-tokens in a source-language (SL) text. Translation spotting (TS) takes as input a *couple*, i.e. a pair of SL and TL text segments, which are known to be translations of one another, and a *SL query*, i.e. a subset of the tokens of the SL segment, on which the TS will focus its attention. The result of the TS process consists of two sets of tokens, i.e. one for each language. We call these sets the SL and TL *answers* to the query.

In more formal terms:

- The input to the TS process is a pair of SL and TL text segments $\langle S, T \rangle$, and a contiguous, non-empty sequence of word-tokens in S, $q = s_{i_1} ... s_{i_2}$ (the *query*).
- The output is a pair of sets of tokens $\langle r_q(S), r_q(T) \rangle$, the *SL answer* and *TL answer* respectively.

Figure 1 shows some examples of TS, where the words in italics represent the SL query, and the words in bold are the SL and TL answers.

As can be seen in these examples, the tokens in the query q and answers $r_q(S)$ and $r_q(T)$ may or may not be contiguous (examples 2 and 3), and the TL answer may possibly be empty (example 4) when there is no satisfying way of linking TL tokens to the query.

Translation spotting finds different applications, for example in bilingual concordancers, such as the *TransSearch* system (Macklovitch et al., 2000), and example-based machine translation (Brown, 1996). In this article, we focus on a different application: a *sub-sentential translation memory*. We describe this application context in section 2, and discuss how TS fits in to this type of system. We then propose in section 3 a series of TS methods, specifically adapted to this application context. In section 4, we present an empirical evaluation of the proposed methods.

2 Sub-sentential Translation Memory Systems

A *translation memory system* is a type of translation support tool whose purpose is to avoid the re-translation of segments of text for which a translation has previously been produced. Typically, these systems are integrated to a word-processing environment. Every sentence that the user translates within this environment is stored in a database (the *translation memory* – or TM). Whenever the system encounters some new text that matches a sentence in the TM, its translation is retrieved and proposed to the translator for reuse.

		Sentence Pair	
	Query	SL (English)	TL (French)
1.	*and a growing gap*	Is this our model of the future, regional disparity ***and a growing gap*** between rich and poor?	Est ce là le modèle que nous visons, soit la disparité régionale **et un fossé de plus en plus large** entre les riches et les pauvres?
2.	*the government's commitment*	***The government's commitment*** was laid out in the 1994 white paper.	**Le gouvernement** a exposé **ses engagements** dans le livre blanc de 1994.
3.	*close to* [...] *years*	I have been fortunate to have been travelling for ***close to*** 40 ***years***.	J'ai eu la chance de voyager pendant **près de** 40 **ans** .
4.	*to the extent that*	***To the extent that*** the Canadian government could be open, it has been so.	Le gouvernement canadien a été aussi ouvert qu'il le pouvait.

Figure 1: Translation spotting examples

As suggested in the above paragraph, existing systems essentially operate at the level of sentences: the TM is typically made up of pairs of sentences, and the system's proposals consist in translations of complete sentences. Because the repetition of complete sentences is an extremely rare phenomenon in general language, this level of resolution limits the usability of TM's to very specific application domains – most notably the translation of revised or intrinsically repetitive documents. In light of these limitations, some proposals have recently been made regarding the possibility of building TM systems that operate "below" the sentence level, or *sub-sentential translation memories* (SSTM) – see for example (Langé et al., 1997; McTait et al., 1999).

Putting together this type of system raises the problem of automatically establishing correspondences between arbitrary sequences of words in the TM, or, in other words, of "spotting translations". This process (translation spotting) can be viewed as a by-product of *word-alignment*, i.e. the problem of establishing correspondences between the words of a text and those of its translation: obviously, given a complete alignment between the words of the SL and TL texts, we can extract only that part of the alignment that concerns the TS query; conversely, TS may be seen as a sub-task of the word-alignment problem: a complete word-alignment can be obtained by combining the results of a series of TS operations, covering the entirety of the SL text.

From the point of view of an SSTM application, the TS mechanism should find the TL segments that are the most likely to be useful to the translator in producing the translation of a given SL sentence. In the end, the final criterion by which a SSTM will be judged is *profitability*: to what extent do the system's proposals enable the user to save time and/or effort in producing a new translation.

From that perspective, the two most important characteristics of the TL answers are *relevance*, i.e. whether or not the system's TL proposals constitute valid translations for some part of the source sentence; and *coherence*, i.e. whether the proposed segments are well-formed, at least from a syntactic point of view. As suggested by McTait et al. (1999), "linguistically motivated" sub-sentential entities are more likely than arbitrary sequences of words to lead to useful proposals for the user.

Planas (2000) proposes a fairly simple approach for an SSTM: his system would operate on sequences of syntactic chunks, as defined by Abney (1991). Both the contents of the TM and the new text under consideration would be segmented into chunks; sequences of chunks from the new text would then be looked up *verbatim* in the TM; the translation of the matched sequences would be proposed to the user as partial translations of the current input. Planas's case for using sequences of chunks as the unit of translation for SSTM's is supported by the *coherence* criterion above: chunks constitute "natural" textual units, which users should find easier to grasp and reuse than arbitrary sequences.

The coherence criterion also supports the case for *contiguous TL proposals*, i.e. proposals that take the form of contiguous sequences of tokens from the TM, as opposed to discontiguous sets such as those of examples 2 and 3, in figure 1. This also makes intuitive sense from the more general point of view of profitability: manually "filling holes" within a discontiguous proposal is likely to be time-consuming and counter-productive. On the other hand, filling those holes automatically, as proposed for example by Langé et al. and McTait et al., raises numerous problems with regard to syntactic and semantic well-formedness of the TL proposals. In theory, contiguous sequences of token from the TM should not suffer from such ills.

Finally, and perhaps more importantly, in a SSTM application such as that proposed by Planas, there appears to be statistical argument in favor of contiguous TL proposals: the more frequent a contiguous SL sequences, the more likely it is that its TL equivalent is also contiguous. In other words, there appears to be a natural tendency for frequently-occurring phrases and formulations to correspond to like-structured sequences in other languages. This will be discussed further in section 4. But clearly,

a TS mechanism intended for such a SSTM should take advantage of this tendency.

3 TS Methods

In this section, we propose various TS methods, specifically adapted to a SSTM application such as that proposed by Planas (2000), i.e. one which takes as translation unit contiguous sequences of syntactic chunks.

3.1 Viterbi TS

As mentioned earlier, TS can be seen as a bi-product of word-level alignments. Such alignments have been the focus of much attention in recent years, especially in the field of statistical translation modeling, where they play an important role in the learning process.

For the purpose of statistical translation modeling, Brown et al. (1993) define an alignment as a vector $a = a_1...a_m$ that connects each word of a source-language text $S = s_1...s_m$ to a target-language word in its translation $T = t_1...t_n$, with the interpretation that word t_{a_j} is the translation of word s_j in S ($a_j = 0$ is used to denote words of s that do not produce anything in T).

Brown et al. also define the *Viterbi alignment* between source and target sentences S and T as the alignment $\hat{a}$ whose probability is maximal under some translation model:

$$\hat{a} = \text{argmax}_{a \in \mathcal{A}} \Pr_{\mathcal{M}}(a|S,T)$$

where $\mathcal{A}$ is the set of all possible alignments between S and T, and $\Pr_{\mathcal{M}}(a|S,T)$ is the estimate of a's probability under model $\mathcal{M}$, which we denote $\Pr(a|S,T)$ from hereon. In general, the size of $\mathcal{A}$ grows exponentially with the sizes of S and T, and so there is no efficient way of computing $\hat{a}$ efficiently. However, under Model 2, the probability of an alignment a is given by:

$$\Pr(a|S,T) = \prod_{i=1}^{m} \Pr(a_i|i,m,n) \quad (1)$$

where

$$\Pr(j|i,m,n) = \frac{\gamma(j,i,m,n)}{\sum_{J=0}^{n} \gamma(J,i,m,n)}, \quad (2)$$

and

$$\gamma(j,i,m,n) = t(s_i|t_j)a(j,i,m,n)$$

In this last equation, $t(s_i|t_j)$ is the model's estimate of the "lexical" distribution $p(s_i|t_j)$, while $a(j,i,m,n)$ estimates the "alignment" distribution $p(j|i,m,n)$. Therefore, with this model, the Viterbi alignment can be obtained by simply picking for each position i in S, the alignment that maximizes $t(s_i|t_j)a(j,i,m,n)$. This procedure can trivially be carried out in $\mathcal{O}(mn)$ operations. Because of this convenient property, we base the rest of this work on this model.

Adapting this procedure to the TS task is straightforward: given the TS query q, produce as TL answer the corresponding set of TL tokens in the Viterbi alignment: $r_q(T) = \{t_{\hat{a}_{i_1}}, ..., t_{\hat{a}_{i_2}}\}$ (the SL answer is simply q itself). We call this method *Viterbi TS*: it corresponds to the most likely alignment between the query q and TL text T, given the probability estimates of the translation model. If q contains I tokens, the Model 2 Viterbi TS can be computed in $\mathcal{O}(In)$ operations. Figure 2 shows an example of the result of this process.

query : *the government 's commitment*

couple:

S = Let us see where *the government's commitment* is really at in terms of the farm community.	T = Voyons quel est le véritable engagement du gouvernement envers la communauté agricole.

Viterbi alignment on query tokens:

the	→	le
government	→	gouvernement
's	→	du
commitment	→	engagement

TL answer:

T = Voyons quel est **le** véritable **engagement du gouvernement** envers la communauté agricole.

Figure 2: Viterbi TS example

3.2 Post-processings

The tokens of the TL answer produced by Viterbi TS are not necessarily contiguous in T which, as remarked earlier, is problematic in a TM application. Various *a posteriori* processings on $r_q(T)$ are possible to fix this; we list here only the most obvious:

expansion : Take the minimum and maximum values in $\{\hat{a}_{i_1}, ..., \hat{a}_{i_2}\}$, and produce the sequence $t_{\min a_i}...t_{\max a_i}$; in other words, produce as TL answer the smallest contiguous sequence in T that contains all the tokens of $r_q(T)$.

longest-sequence : Produce the subset of $r_q(T)$ that constitutes the longest contiguous sequence in T.

zero-tolerance : If the tokens in $r_q(T)$ cannot be arranged in a contiguous sequence of T, then simply discard the whole TL answer.

Figure 3 illustrates how these three strategies affect the Viterbi TS of figure 2.

3.3 Contiguous TS

The various independence assumptions underpinning IBM Model 2 often have negative effects on the resulting Viterbi alignments. In particular, this model assumes

$r_q(T) = \{$le, engagement, du, gouvernement$\}$		
post-processing:		
expansion :	$X(r_q(T)) =$	le véritable engagement du gouvernement
longest-sequence :	$L(r_q(T)) =$	engagement du gouvernement
zero-tolerance :	$Z(r_q(T)) =$	$\emptyset$

Figure 3: Post-processings on Viterbi TS

that all connections within an alignment are independent of each other, which leads to numerous aberrations in the alignments. Typically, each SL token gets connected to the TL token with which it has the most "lexical affinities", regardless of other existing connections in the alignment and, more importantly, of the relationships this token holds with other SL tokens in its vicinity. Conversely, some TL tokens end up being connected to several SL tokens, while other TL tokens are left unconnected.

As mentioned in section 2, in a sub-sentential TM application, contiguous sequences of tokens in the SL tend to translate into contiguous sequences in the TL. This suggests that it might be a good idea to integrate a "contiguity constraint" right into the alignment search procedure.

For example, we can formulate a variant of the Viterbi TS method above, which looks for the alignment that maximizes $\Pr(a|S,T)$, under the constraint that the TL tokens aligned with the SL query must be contiguous. Consider a procedure that seeks the (possibly null) sequence $t_{j_1}...t_{j_2}$ of T, that maximizes:

$$\Pr(a_q|s_{i_1}^{i_2}, t_{j_1}^{j_2})\Pr(a_{\bar{q}}|s_1^{i_1-1}s_{i_2+1}^{m}, t_1^{j_1-1}t_{j_2+1}^{n})$$

Such a procedure actually produces two distinct alignments over S and T: an alignment a_q, which connects the query tokens (the sequence $s_{i_1}^{i_2}$) with a sequence of contiguous tokens in T (the sequence $t_{j_1}^{j_2}$), and an alignment $a_{\bar{q}}$, which connects the rest of sentence S (i.e. all the tokens outside the query) with the rest of T. Together, these two alignments constitute the alignment $a = a_q \cup a_{\bar{q}}$, whose probability is maximal, under a double constraint:

1. the query tokens $s_{i_1}^{i_2}$ can only be connected to tokens within a contiguous region of T (the sequences $t_{j_1}^{j_2}$);
2. the tokens outside the query (in either one of the two sequences $s_1^{i_1-1}$ and $s_{i_2+1}^{m}$) can only get connected to tokens outside $t_{j_1}^{j_2}$.

With such an alignment procedure, we can trivially devise a TS method, which will return the optimal $t_{j_1}^{j_2}$ as TL answer. We call this method *Contiguous TS*. Alignments satisfying the above constraints can be obtained directly, by computing Viterbi alignments a_q and $a_{\bar{q}}$ for each pair of target positions $\langle j_1, j_2 \rangle$. The TS procedure then retains the pair of TL language positions that maximizes the joint probability of alignments a_q and $a_{\bar{q}}$. This operation requires the computation of two Viterbi alignments for each pair $\langle j_1, j_2 \rangle$, i.e. $n(n-1)$ Viterbi alignments, plus a "null" alignment, corresponding to the situation where $t_{j_1}^{j_2} = \emptyset$. Overall, using IBM Model 2, the operation requires $\mathcal{O}(mn^3)$ operations. Figure 4 illustrates a contiguous TS obtained on the example of figure 2.

Alignment:	Let us see	→	Voyons
	where	→	quel
$a_q =$	the	→	engagement
	government	→	gouvernement
	's	→	du
	commitment	→	engagement
	is	→	est
	really	→	véritable
	at	→	la
	in terms of	→	envers
	the	→	la
	farm	→	agricole
	community	→	communauté
	.	→	.

TL answer:
$T =$ Voyons quel est le véritable **engagement du gouvernement** envers la communauté agricole.

Figure 4: Contiguous TS Example

3.4 Compositional TS

As pointed out in section 3.3, In IBM-style alignments, a single TL token can be connected to several SL tokens, which sometimes leads to aberrations. This contrasts with alternative alignment models such as those of Melamed (1998) and Wu (1997), which impose a "one-to-one" constraint on alignments. Such a constraint evokes the notion of *compositionality* in translation: it suggests that each SL token operates independently in the SL sentence to produce a single TL token in the TL sentence, which then depends on no other SL token. This view is, of course, extreme, and real-life translations are full of examples (idiomatic expressions, terminology, paraphrasing, etc.) that show how this compositionality principle breaks down as we approach the level of word correspondences.

However, in a TM application, TS usually needs not go down to the level of individual words. Therefore, compositionality can often be assumed to apply, at least to the level of the TS query. The contiguous TS method pro-

posed in the previous section implicitly made such an assumption. Here, we push it a little further.

Consider a procedure that splits each the source and target sentences S and T into two independent parts, in such a way as to maximise the probability of the two resulting Viterbi alignments:

$$\text{argmax}_{\langle i,j,d\rangle} \begin{cases} d=1 : & \Pr(a_1|s_1^i, t_1^j) \\ & \times\Pr(a_2|s_{i+1}^m, t_{j+1}^n) \\ d=-1 : & \Pr(a_1|s_1^i, t_{j+1}^n) \\ & \times\Pr(a_2|s_{i+1}^m, t_1^j) \end{cases}$$

In the triple $\langle i, j, d\rangle$ above, i represents a "split point" in the SL sentence S, j is the analog for TL sentence T, and d is the "direction of correspondence": $d = 1$ denotes a "parallel correspondence", i.e. $s_1...s_i$ corresponds to $t_1...t_j$ and $s_{i+1}...s_m$ corresponds to $t_{j+1}...t_n$; $d = -1$ denotes a "crossing correspondence", i.e. $s_1...s_i$ corresponds to $t_{j+1}...t_n$ and $s_{i+1}...s_m$ corresponds to $t_1...t_j$.

The triple $\langle I, J, D\rangle$ produced by this procedure refers to the most probable alignment between S and T, under the hypothesis that both sentences are made up of two independent parts ($s_1...s_I$ and $s_{I+1}...s_m$ on the one hand, $t_1...t_J$ and $t_{J+1}...t_n$ on the other), that correspond to each other two-by-two, following direction D. Such an alignment suggests that translation T was obtained by "composing" the translation of $s_1...s_I$ with that of $s_{I+1}...s_m$.

This "splitting" process can be repeated recursively on each pair of matching segments, down to the point where each SL segment contains a single token. (TL segments can always be split, even when empty, because IBM-style alignments make it possible to connect SL tokens to the "null" TL token, which is always available.) This gives rise to a word-alignment procedure that we call *Compositional word alignment*.

This procedure actually produces two different outputs: first, a parallel partition of S and T into m pairs of segments $\langle s_i, t_j^k\rangle$, where each t_j^k is a (possibly null) contiguous sub-sequence of T; second, an IBM-style alignment, such that each SL and TL token is linked to at most one token in the other language: this alignment is actually the concatenation of individual Viterbi alignments on the $\langle s_i, t_j^k\rangle$ pairs, which connects each s_i to (at most) one of the tokens in the corresponding t_j^k.

Of course, such alignments face even worst problems than ordinary IBM-style alignments when confronted with non-compositional translations. However, when adapting this procedure to the TS task, we can hypothesize that compositionality applies, at least to the level of the SL query. This adaptation proceeds along the following modifications to the alignment procedure described above:

1. forbid splittings within the SL query: $i_1 \leq i \leq i_2$;
2. at each level of recursion, only consider that pair of segments which contains the SL query;
3. stop the procedure as soon as it is no longer possible to split the SL segment, i.e. it consists of $s_{i_1}...s_{i_2}$.

The TL segment matched with $s_{i_1}...s_{i_2}$ when the procedure terminates is the TL answer. We call this procedure *Compositional TS*. It can be shown that it can be carried out in $\mathcal{O}(m^3n^2)$ operations in the worst case, and $\mathcal{O}(m^2n^2 \log m)$ on average. Furthermore, by limiting the search to split points yielding matching segments of comparable sizes, the number of required operations can be cut by one order of magnitude (Simard, 2003).

Figure 5 shows how this procedure splits the example pair of figure 2 (the query is shown in italics).

4 Evaluation

We describe here a series of experiments that were carried out to evaluate the performance of the TS methods described in section 3. We essentially identified a number of SL queries, looked up these segments in a TM to extract matching pairs of SL-TL sentences, and manually identified the TL tokens corresponding to the SL queries in each of these pairs, hence producing manual TS's. We then submitted the same sentence-pairs and SL queries to each of the proposed TS methods, and measured how the TL answers produced automatically compared with those produced manually. We describe this process and the results we obtained in more details below.

4.1 Test Material

The test material for our experiments was gathered from a translation memory, made up of approximately 14 years of Hansard (English-French transcripts of the Canadian parliamentary debates), i.e. all debates published between April 1986 and January 2002, totalling over 100 million words in each language. These documents were mostly collected over the Internet, had the HTML markup removed, were then segmented into paragraphs and sentences, aligned at the sentence level using an implementation of the method described in (Simard et al., 1992), and finally dumped into a document-retrieval system (MG (Witten et al., 1999)). We call this the *Hansard TM*.

To identify SL queries, a distinct document from the Hansard was used, the transcript from a session held in March 2002. The English version of this document was segmented into syntactic chunks, using an implementation of Osborne's chunker (Osborne, 2000). All sequences of chunks from this text that contained three or more word tokens were then looked up in the Hansard TM. Among the sequences that did match sentences in the TM, 100 were selected at random. These made up the *test SL queries*.

Recursion Level	*SL segment*		*TL segment*	*direction (d)*
1	[Let us see] [where *the government 's commitment* is really at in terms of the farm community]	$\longleftrightarrow$	[Voyons] [quel est le véritable engagement du gouvernement envers la communauté agricole]	$d = 1$
2	[where *the government 's commitment* is really at] [in terms of the farm community]	$\longleftrightarrow$	[quel est le véritable engagement du gouvernement] [envers la communauté agricole]	$d = 1$
3	[where] [*the government 's commitment* is really at]	$\longleftrightarrow$	[quel] [est le véritable engagement du gouvernement]	$d = 1$
4	[*the government 's commitment*] [is really at]	$\longleftrightarrow$	[est le véritable] [engagement du gouvernement]	$d = -1$
Answers:	$r_q(S)$ =*the government 's commitment*	$\longleftrightarrow$	$r_q(T)$ =engagement du gouvernement	

Figure 5: Compositional TS Example

While some SL queries yielded only a handful of matches in the TM, others turned out to be very productive, producing hundreds (and sometimes thousands) of couples. For each test segment, we retained only the 100 first matching pair of sentences from the TM. This process yielded 4100 pairs of sentences from the TM, an average of 41 per SL query; we call this our *test corpus*. Within each sentence pair, we spotted translations manually, i.e. we identified by hand the TL word-tokens corresponding to the SL query for which the pair had been extracted. These annotations were done following the TS guidelines proposed by Véronis (1998); we call this the *reference TS*.

4.2 Evaluation Metrics

The results of our TS methods on the test corpus were compared to the reference TS, and performance was measured under different metrics. Given each pair $\langle S, T \rangle$ from the test corpus, and the corresponding reference and evaluated TL answers r^* and r, represented as sets of tokens, we computed:

exactness : equal to 1 if $r^* = r$, 0 otherwise;

recall : $|r^* \cap r|/|r^*|$

precision : $|r^* \cap r|/|r|$

F-measure : $2\ \frac{|r \cap r^*|}{|r|+|r^*|}$

In all the above computations, we considered that "empty" TL answers ($r = \emptyset$) actually contained a single "null" word. These metrics were then averaged over all pairs of the test corpus (and not over SL queries, which means that more "productive" queries weight more heavily in the reported results).

4.3 Experiments

We tested all three methods presented in section 3, as well as the three "post-processings" on Viterbi TS proposed in section 3.2. All of these methods are based on IBM Model 2. The same model parameters were used for all the experiments reported here, which were computed with the GIZA program of the Egypt toolkit (Al-Onaizan et al., 1999). Training was performed on a subset of about 20% of the Hansard TM. The results of our experiments are presented in table 1.

	Metric			
method	exact	precision	recall	F
Viterbi	0.17	0.60	0.57	0.57
+ Expansion	0.26	0.51	0.71	0.55
+ Longest-sequence	0.03	0.63	0.20	0.29
+ Zero-tolerance	0.20	0.28	0.28	0.28
Contiguous	0.36	0.75	0.66	0.68
Compositional	0.40	0.72	0.70	0.69

Table 1: Results of experiments

The *Zero-tolerance* post-processing produces empty TL answers whenever the TL tokens are not contiguous. On our test corpus, over 70% of all Viterbi alignments turned out to be non-contiguous. These empty TL answers were counted in the statistics above (*Viterbi + Zero-tolerance* row), which explains the low performance obtained with this method. In practice, the intention of *Zero-tolerance* post-processing is to filter out non-contiguous answers, under the hypotheses that they probably would not be usable in a TM application. Table 2 presents the performance of this method, taking into account only non-empty answers.

	Metric			
method	exact	precision	recall	F
Viterbi + Zero-tolerance	0.56	0.83	0.82	0.81

Table 2: Performance of *zero-tolerance* filter on non-empty TL answers

4.4 Discussion

Globally, in terms of exactness, compositional TS produces the best TL answers, with 40% correct answers, an

improvement of 135% over plain Viterbi TS. This gain is impressive, particularily considering the fact that all methods use exactly the same data. In more realistic terms, the gain in F-measure is over 20%, which is still considerable.

The best results in terms of precision are obtained with contiguous TS, which in fact is not far behind compositional TS in terms of recall either. This clearly demonstrates the impact of a simple contiguity constraint in this type of TS application. Overall, the best recall figures are obtained with the simple *Extension* post-processing on Viterbi TS, but at the cost of a sharp decrease in precision. Considering that precision is possibly more important than recall in a TM application, the contiguous TS would probably be a good choice.

The *Zero-tolerance* strategy, used as a filter on Viterbi alignments, turns out to be particularily effective. It is interesting to note that this method is equivalent to the one proposed by Marcu (Marcu, 2001) to automatically construct a sub-sentential translation memory. Taking only non-null TS's into consideration, it outclasses all other methods, regardless of the metric. But this is at the cost of eliminating numerous potentially useful TL answers (more than 70%). This is particularily frustrating, considering that over 90% of all TL answers in the reference are indeed contiguous.

To understand how this happens, one must go back to the definition of IBM-style alignments, which specifies that each SL token is linked to at most one TL token. This has a direct consequence on Viterbi TS's: if the SL queries contains K word-tokens, then the TL answer will itself contain at most that number of tokens. As a result, this method has systematic problems when the actual TL answer is longer than the SL query. It turns out that this occurs very frequently, especially when aligning from English to French, as is the case here. For example, consider the English sequence *airport security*, most often translated in French as *sécurité dans les aéroports*. The Viterbi alignment normally produces links *airport* $\rightarrow$ *aéroport* and *security* $\rightarrow$ *sécurité*, and the sequence *dans les* is then left behind (or accidentally picked up by erroneous links from other parts of the SL sentence), thus leaving a non-contiguous TL answer.

The *Expansion* post-processing, which finds the shortest possible sequence that covers all the tokens of the Viterbi TL answer, solves the problem in simple situations such as the one in the above example. But in general, integrating contiguity constraints directly in the search procedure (contiguous and compositional TS) turns out to be much more effective, without solving the problem entirely. This is explained in part by the fact that these techniques are also based on IBM-style alignments. When "surplus" words appear at the boundaries of the TL answer, these words are not counted in the alignment probability, and so there is no particular reason to include them in the TL answer. Consider the following example:

- These companies **indicated their support** for the government 's decision.
- Ces compagnies ont déclaré qu' elles appuyaient la décision du gouvernement .

When looking for the French equivalent to the English *indicated their support*, we will probably end up with an alignment that links *indicated* $\rightarrow$ *déclaré* and *support* $\rightarrow$ *appuyaient*. As a result of contiguity constraints, the TL sequence *qu' elle* will naturally be included in the TL answer, possibly forcing a link *their* $\rightarrow$ *elles* in the process. However, the only SL that could be linked to *ont* is the verb *indicated*, which is already linked to *déclaré*. As a result, *ont* will likely be left behind in the final alignment, and will not be counted when computing the alignment's probability.

5 Conclusion

We have presented different translation spottings methods, specifically adapted to a sub-sentential translation memory system that proposes TL translations for SL sequences of syntactic chunks, as proposed by Planas (2000). These methods are based on IBM statistical translation Model 2 (Brown et al., 1993), but take advantage of certain characteristics of the segments of text that can typically be extracted from translation memories. By imposing contiguity and compositionality constraints on the search procedure, we have shown that it is possible to perform translation spotting more accurately than by simply relying on the most likely word alignment.

Yet, the accuracy of our methods still leave a lot to be desired; on closer examination most of our problems can be attributed to the underlying translation model. Computing word alignments with IBM Model 2 is straightforward and efficient, which made it a good choice for experimenting; however, this model is certainly not the state of the art in statistical translation modeling. Thenagain, the methods proposed here were all based on the idea of finding the most likely word-alignment under various constraints. This approach is not dependent on the underlying translation model, and similar methods could certainly be devised based on more elaborate models, such as IBM Models 3–5, or the HMM-based models proposed by Och et al. (1999) for example.

Alternatively, there are other ways to compensate for Model 2's weaknesses. Each IBM-style alignment between two segments of text denotes one particular explanation of how the TL words emerged from the SL words, but it doesn't tell the whole story. Basing our TS methods on a set of likely alignments rather than on the single most-likely alignment, as is normally done to estimate the

parameters of higher-level models, could possibly lead to more accurate TS results. Similarly, TS applications are not bound to translation directionality as statistical translation systems are; this means that we could also make use of a "reverse" model to obtain a better estimate of the likelihood of two segments of text being mutual translation.

These are all research directions that we are currently pursuing.

References

[Abney1991] Steven Abney. 1991. Parsing by Chunks. In R.C. Berwick, editor, *Principle-Based Parsing: Computation and Psycholinguistics*, pages 257–278. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[Al-Onaizan et al.1999] Yaser Al-Onaizan, Jan Curin, Michael Jahr, Kevin Knight, John Lafferty, Dan Melamed, Franz-Josef Och, David Purdy, Noah H. Smith, and David Yarowsky. 1999. Statistical Machine Translation - Final Report, JHU Workshop 1999. Technical report, Johns Hopkins University.

[Brown et al.1993] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. 1993. The Mathematics of Machine Translation: Parameter Estimation. *Computational Linguistics*, 19(2):263–311.

[Brown1996] Ralf D. Brown. 1996. Example-Based Machine Translation in the Pangloss System. In *Proceedings of the International Conference on Computational Linguistics (COLING) 1996*, pages 169–174, Copenhagen, Denmark, August.

[Langé et al.1997] Jean-Marc Langé, Éric Gaussier, and Béatrice Daille. 1997. Bricks and Skeletons: Some Ideas for the Near Future of MAHT. *Machine Translation*, 12(1–2):39–51.

[Macklovitch et al.2000] Elliott Macklovitch, Michel Simard, and Philippe Langlais. 2000. TransSearch: A Free Translation Memory on the World Wide Web. In *Proceedings of the Second International Conference on Language Resources & Evaluation (LREC)*, Athens, Greece.

[Marcu2001] Daniel Marcu. 2001. Towards a Unified Approach to Memory- and Statistical-Based Machine Translation. In *Proceedings of the 39th Annual Meeting of the Association for Computational Linguistics (ACL)*, Toulouse, France, July.

[McTait et al.1999] Kevin McTait, Maeve Olohan, and Arturo Trujillo. 1999. A Building Blocks Approach to Translation Memory. In *Proceedings of the 21st ASLIB International Conference on Translating and the Computer*, London, UK.

[Melamed1998] I. Dan Melamed. 1998. Word-to-Word Models of Translational Equivalence. Technical Report 98-08, Dept. of Computer and Information Science, University of Pennsylvania, Philadelphia, USA.

[Och et al.1999] Franz Josef Och, Christoph Tillmann, and Hermann Ney. 1999. Improved Alignment Models for Statistical Machine Translation. In *Proceedings of the 4th Conference on Empirical Methods in Natural Language Processing (EMNLP)and 7th ACL Workshop on Very Large Corpora (WVLC)*, pages 20–28, College Park, USA.

[Osborne2000] Miles Osborne. 2000. Shallow Parsing as Part-of-Speech Tagging. In Claire Cardie, Walter Daelemans, Claire Nédellec, and Erik Tjong Kim Sang, editors, *Proceedings of the Fourth Conference on Computational Natural Language Learning*, Lisbon, Portugal, September.

[Planas2000] Emmanuel Planas. 2000. Extending Translation Memories. In *EAMT Machine Translation Workshop*, Ljubljana, Slovenia, May.

[Simard et al.1992] Michel Simard, George Foster, and Pierre Isabelle. 1992. Using Cognates to Align Sentences in Bilingual Corpora. In *Proceedings of the 4th Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, pages 67–82, Montréal, Canada.

[Simard2003] Michel Simard. 2003. *Mémoires de traduction sous-phrastiques*. Ph.D. thesis, Université de Montréal. to appear.

[Véronis and Langlais2000] Jean Véronis and Philippe Langlais. 2000. Evaluation of Parallel Text Alignment Systems – The ARCADE Project. In Jean Véronis, editor, *Parallel Text Processing*, Text, Speech and Language Technology. Kluwer Academic Publishers, Dordrecht, The Netherlands.

[Véronis1998] Jean Véronis. 1998. Tagging guidelines for word alignment. http://www.up.univ-mrs.fr/ veronis/arcade/2nd/word/guide/index.html, April.

[Witten et al.1999] Ian H. Witten, Alistair Moffat, and Timothy C. Bell. 1999. *Managing Gigabytes: Compressing and Indexing Documents and Images*. Morgan Kaufmann Publishing, San Francisco, USA, 2nd edition edition.

[Wu1997] Dekai Wu. 1997. Stochastic Inversion Transduction Grammars and Bilingual Parsing of Parallel Corpora. *Computational Linguistics*, 23(3):377–404, September.