Unit Completion for a Computer-aided Translation Typing System

Philippe Langlais, George Foster and Guy Lapalme
RALI / DIRO
Université de Montréal
C.P. 6128, succursale Centre-ville
Montral (Qubec), Canada, H3C 3J7
{felipe,foster,lapalme}@iro.umontreal.ca

Abstract

This work is in the context of TRANSTYPE, a system that observes its user as he or she types a translation and repeatedly suggests *completions* for the text already entered. The user may either accept, modify, or ignore these suggestions. We describe the design, implementation, and performance of a prototype which suggests completions of units of texts that are longer than one word.

1 Introduction

TRANSTYPE is part of a project set up to explore an appealing solution to *Interactive Machine Translation* (IMT). In constrast to classical IMT systems, where the user's role consists mainly of assisting the computer to analyse the source text (by answering questions about word sense, ellipses, phrasal attachments, etc), in TRANSTYPE the interaction is directly concerned with establishing the target text.

Our interactive translation system works as follows: a translator selects a sentence and begins typing its translation. After each character typed by the translator, the system displays a proposed completion, which may either be accepted using a special key or rejected by continuing to type. Thus the translator remains in control of the translation process and the machine must continually adapt its suggestions in response to his or her input. We are currently undertaking a study to measure the extent to which our word-completion prototype can improve translator productivity. The conclusions of this study will be presented elsewhere.

The first version of TRANSTYPE (Foster et al., 1997) only proposed completions for the current word. This paper deals with predictions which extend to the next several words in the text. The potential gain from multiple-word predictions can be appreciated in the one-sentence translation task reported in table 1, where a hypothetical user saves over 60% of the keystrokes needed to produce a translation in a word completion scenario, and about 85% in a "unit" completion scenario.

In all the figures that follow, we use different fonts to differentiate the various input and output: *italics* are used for the source text, sans-serif for characters typed by the user and `typewriter-like` for characters completed by the system.

The first few lines of the table 1 give an idea of how TransType functions. Let us assume the unit scenario (see column 2 of the table) and suppose that the user wants to produce the sentence "Ce projet de loi est examiné à la chambre des communes" as a translation for the source sentence "***This bill is examined in the house of commons***". The first hypothesis that the system produces before the user enters a character is `loi` (*law*). As this is not a good guess from TRANSTYPE the user types the first character (c) of the words he or she wants as a translation. Taking this new input into account, TRANSTYPE then modifies its proposal so that it is compatible whith what the translator has typed. It suggests the desired sequence ce projet de loi, which the user can simply validate by typing a dedicated key. Continuing in this way, the user and TRANSTYPE alternately contribute to the final translation. A screen copy of this prototype is provided in figure 1.

2 The Core Engine

The core of TRANSTYPE is a completion engine which comprises two main parts: an *evaluator* which assigns probabilistic scores to completion hypotheses and a *generator* which uses the evaluation function to select the best candidate for completion.

2.1 The Evaluator

The evaluator is a function $p(t|t', s)$ which assigns to each target-text unit t an estimate of its probability given a source text s and the tokens t' which precede t in the current translation of s.[1] Our approach to modeling this distribution is based to a large extent on that of the IBM group (Brown et al., 1993), but it differs in one significant aspect: whereas the IBM model involves a "noisy channel" decomposition, we use a linear combination of separate predictions from a language model $p(t|t')$ and a translation model $p(t|s)$. Although the noisy channel technique

[1] We assume the existence of a deterministic procedure for tokenizing the target text.

This bill is examined in the house of commons				
	word-completion task		unit-completion task	
	pref.	completions	pref.	completions
ce	ce+	`/loi · c/'`	c+	`/loi · c/e projet de loi`
projet	p+	`/est · p/rojet`	-	
de	d+	`/très · d/e`	-	
loi	l+	`/très · l/oi`	-	
est	e+	`/de · e/st`	e+	`/de · e/st`
examiné	e+	`/en · e/xaminé`	ex+	`/à la chambre des communes · e/n · ex/aminé`
à la	à+	`/par · à/ la`	+	`/à la chambre des communes`
chambre	+	`/chambre`	-	
des	de+	`/communes · d/e · de/s`	-	
communes	+	`/communes`	-	
54 char.	10 + 10 accept. → **20 keystrokes**		4 + 4 accept. → **8 keystrokes**	

Table 1: A one-sentence session illustrating the word- and unit-completion tasks. The first column indicates the target words the user is expected to produce. The next two columns indicate respectively the prefixes typed by the user and the completions proposed by the system in a word-completion task. The last two columns provide the same information for the unit-completion task. The total number of keystrokes for both tasks is reported in the last line. + indicates the acceptance key typed by the user. A completion is denoted by α/β where α is the typed prefix and β the completed part. Completions for different prefixes are separated by $\cdot$.

is powerful, it has the disadvantage that $p(s|t',t)$ is more expensive to compute than $p(t|s)$ when using IBM-style translation models. Since speed is crucial for our application, we chose to forego the noisy channel approach in the work described here. Our linear combination model is described as follows:

$$p(t|t',s) = \underbrace{p(t|t')\;\alpha(t',s)}_{\text{language}} + \underbrace{p(t|s)\;[1-\alpha(t',s)]}_{\text{translation}} \quad (1)$$

where $\alpha(t',s) \in [0,1]$ are context-dependent interpolation coefficients. For example, the translation model could have a higher weight at the start of a sentence but the contribution of the language model might become more important in the middle or the end of the sentence. A study of the weightings for these two models is described elsewhere. In the work described here we did not use the contribution of the language model (that is, $\alpha(t',s) = 0,\ \forall\, t', s$).

Techniques for weakening the independence assumptions made by the IBM models 1 and 2 have been proposed in recent work (Brown et al., 1993; Berger et al., 1996; Och and Weber, 98; Wang and Waibel, 98; Wu and Wong, 98). These studies report improvements on some specific tasks (task-oriented limited vocabulary) which by nature are very different from the task TRANSTYPE is devoted to. Furthermore, the underlying decoding strategies are too time consuming for our application. We therefore use a translation model based on the simple linear interpolation given in equation 2 which combines predictions of two translation models — M_s and M_u — both based on IBM-like model 2(Brown et al., 1993). M_s was trained on single words and M_u, described in section 3, was trained on both words and units.

$$p(t|s) = \underbrace{\beta p_s(t|s)}_{\text{word}} + \underbrace{(1-\beta)p_u(t|\mathcal{G}(s))}_{\text{unit}} \quad (2)$$

where p_s and p_u stand for the probabilities given respectively by M_s and M_u. $\mathcal{G}(s)$ represents the new sequence of tokens obtained after grouping the tokens of s into units. The grouping operator $\mathcal{G}$ is illustrated in table 2 and is described in section 3.

2.2 The Generator

The task of the generator is to identify units that match the current prefix typed by the user, and pick the best candidate according to the evaluator. Due to time considerations, the generator introduces a division of the target vocabulary into two parts: a small *active* component whose contents are always searched for a match to the current prefix, and a much larger *passive* part over (380,000 word forms) which comes into play only when no candidates are found in the active vocabulary. The active part is computed dynamically when a new sentence is selected by the translator. It is composed of a few entities (tokens and units) that are likely to appear in the translation. It is a union of the best candidates provided by each model M_s and M_u over the set of all possible target tokens (resp. units) that have a non-null translation probability of being translated by any of the current source tokens (resp. units). Table 2 shows the 10 most likely tokens and units in the active vocabulary for an example source sentence.

s	*that · is · what · the · prime · minister · said · , · and · i · have · outlined · what · has · happened · since · then · .*
t	c' · est · ce · que · le · premier · ministre · a · dit · , · et · j' · ai · résumé · ce · qui · s' · est · produit · depuis · .
$\mathcal{G}(s)$	*that is what · the prime minister said · , and i · have · outlined · what has happened · since then · .*
A_s	, · . · est · ce · ministre · que · et · a · premier · le
A_u	ce qui s' est produit · et je · c' est ce que · voilà ce que · qu' est · c' est · , et · le premier ministre disait

Table 2: Role of the generator for a sample pair of sentences (t is the translation of s in our corpus). $\mathcal{G}(s)$ is the sequence of source tokens recasted by the grouping operator $\mathcal{G}$. A_s indicates the 10 best tokens according to the word model, A_u the 10 best units according to the unit model.

3 Modeling Unit Associations

Automatically identifying which source words or groups of words will give rise to which target words or groups of words is a fundamental problem which remains open. In this work, we decided to proceed in two steps: a) monolingually identifying groups of words that would be better handled as units in a given context, and b) mapping the resulting source and target units. To train our unit models, we used a segment of the Hansard corpus consisting of 15,377 pairs of sentences, totaling 278,127 english tokens (13,543 forms) and 292,865 french tokens (16,399 forms).

3.1 Finding Monolingual Units

Finding relevant units in a text has been explored in many areas of natural language processing. Our approach relies on distributional and frequency statistics computed on each sequence of words found in a training corpus. For sake of efficiency, we used the suffix array technique to get a compact representation of our training corpus. This method allows the efficient retrieval of arbitrary length n-grams (Nagao and Mori, 94; Haruno et al., 96; Ikehara et al., 96; Shimohata et al., 1997; Russell, 1998).

The literature abounds in measures that can help to decide whether words that co-occur are linguistically significant or not. In this work, the strength of association of a sequence of words $w_1^n = w_1, \ldots, w_n$ is computed by two measures: a likelihood-based one $\rho(w_1^n)$ (where ℓ is the likelihood ratio given in (Dunning, 93)) and an entropy-based one $e(w_1^n)$ (Shimohata et al., 1997). Letting T stand for the training text and m a token:

$$\begin{aligned} \rho(w_1^n) &= \underset{i\in]1,n[}{\operatorname{argmin}} \ell(w_1^i, w_{i+1}^n) & (3) \\ e(w_1^n) &= 0.5 \times \left(\begin{array}{c} \sum_{m|mw_1^n \in T} h\left(\frac{freq(mw_1^n)}{freq(w_1^n)}\right) \\ + \\ \sum_{m|w_1^n m \in T} h\left(\frac{freq(w_1^n m)}{freq(w_1^n)}\right) \end{array} \right) \end{aligned}$$

Intuitively, the first measurement accounts for the fact that parts of a sequence of words that should be considered as a whole should not appear often by themselves. The second one reflects the fact that a salient unit should appear in various contexts (i.e. should have a high entropy score).

We implemented a cascade filtering strategy based on the likelihood score ρ, the frequency f, the length l and the entropy value e of the sequences. A first filter ($\mathcal{F}_1(l_{min}, f_{min}, \rho_{min}, e_{min})$) removes any sequence s for which $l(s) < l_{min}$ or $\rho(s) < \rho_{min}$ or $e(s) < e_{min}$ or $f(s) < f_{min}$. A second filter ($\mathcal{F}_2$) removes sequences that are included in preferred ones. In terms of sequence reduction, applying $\mathcal{F}_1(2, 2, 5.0, 0.2)$ on the 81,974 English sequences of at least two tokens seen at least twice in our training corpus, less than 50% of them (39,093) were filtered: 17,063 (21%) were removed because of their low entropy value, 25,818 (31%) because of their low likelihood value.

3.2 Mapping

Mapping the identified units (tokens or sequences) to their equivalents in the other language was achieved by training a new translation model (IBM 2) using the EM algorithm as described in (Brown et al., 1993). This required grouping the tokens in our training corpus into sequences, on the basis of the unit lexicons identified in the previous step (we will refer to the results of this grouping as the ***sequence-based corpus***). To deal with overlapping possibilities, we used a dynamic programming scheme which optimized a criterion $\mathcal{C}$ given by equation 4 over a set $\mathcal{S}$ of all units collected for a given language plus all single words. $\mathcal{G}(w_1^n)$ is obtained by returning the path that maximized $B(n)$. We investigated several $\mathcal{C}$-criteria and we found $\mathcal{C}_l$—a length-based measure—to be the most satisfactory. Table 2 shows an output of the grouping function.

$$B(i) = \begin{cases} 0 \ if \ \ i = 0 \\ \underset{I\in[1,i[\ , w_{i-I}^i \in \mathcal{S}}{\operatorname{argmax}} \left(\begin{array}{c} \mathcal{C}(w_{i-I}^i) \\ + \\ B(i-I-1) \end{array} \right) \end{cases} \quad (4)$$

$$\text{with: } \mathcal{C}_l(w_i^j) = \begin{cases} 0 & if \ \ j <= i \\ j - i + 1 & else \end{cases}$$

source unit (s)	$f(s)$	target units ($[\alpha, p]$)
we have	1748	[nous,0.49] [avons,0.41] [, nous avons,0.07]
we must	720	[nous devons,0.61] [il faut,0.19] [nous,0.14]
this bill	640	[ce projet de loi,0.35] [projet de loi .,0.21] [projet de loi,0.18]
people of canada	282	[les canadiens,0.26] [des canadiens,0.21] [la population,0.07]
mr. speaker :	269	[m. le président :,0.80] [a,0.07] [à la,0.06]
what is happening	190	[ce qui se passe,0.21] [ce qui se,0.16] [et,0.15]
of course ,	178	[évidemment ,,0.26] [naturellement,0.08] [bien sûr,0.08]
is it the pleasure of the house to adopt the	14	[plaît-il à la chambre d' adopter,0.49] [la motion ?,0.42] [motion ?,0.04]
the world	201	[le monde,0.46] [du monde,0.33] [le monde entier,0.19]
child care	86	[les garderies,0.59] [la garde d' enfants,0.23] [des services de garde d' enfants,0.13]
the free trade agreement	75	[l' accord de libre-échange,0.96] [la décision du gatt,0.04]
post-secondary education	66	[l' enseignement postsecondaire,0.75] [l' éducation postsecondaire,0.15] [des fonds,0.06]
the first time	62	[la première fois,1.00]
the canadian aviation safety board	36	[le bureau canadien de la sécurité aérienne,0.55] [du bureau canadien de la sécurité aérienne,0.31] [l' un,0.14]
the next five years	26	[au cours des cinq prochaines années,0.53] [cinq prochaines années,0.27] [25 milliards de d ollars,0.10]
the people of china	17	[le peuple chinois,0.38] [la population chinoise,0.25] [les chinois,0.13]

Table 3: Bilingual associations. The first column indicates a source unit, the second one its frequency in the training corpus. The third column reports its 3-best ranked target associations (α being a token or a unit, p being the translation probability). The second half of the table reports NP-associations obtained after the filter described in the text.

We investigated three ways of estimating the parameters of the unit model. In the first one, $\mathcal{E}_1$, the translation parameters are estimated by applying the EM algorithm in a straightforward fashion over all entities (tokens and units) present at least twice in the sequence-based corpus [2]. The two next methods filter the probabilities obtained with the $\mathcal{E}_1$ method. In $\mathcal{E}_2$, all probabilities $p(t|s)$ are set to 0 whenever s is a token (not a unit), thus forcing the model to contain only associations between source units and target entities (tokens or units). In $\mathcal{E}_3$ any parameter of the model that involves a token is removed (that is, $p(t|s) = 0$ if t or s is a token). The resulting model will thus contain only unit associations. In both cases, the final probabilities are renormalized. Table 3 shows a few entries from a unit model (M_u) obtained after 15 iterations of the EM-algorithm on a sequence corpus resulting from the application of the length-grouping criterion ($\mathcal{C}_l$) over a lexicon of units whose likelihood score is above 5.0. The probabilities have been obtained by application of the method $\mathcal{E}_2$.

We found many partially correct associations (*over the years/au fils des*, *we have/nous*, etc) that illustrate the weakness of decoupling the unit identification from the mapping problem. In most cases however, these associations have a lower probability than the good ones. We also found few erratic associations (*the first time/ c'était*, *some hon. members/!*, etc) due to distributional artifacts. It is also interesting to note that the good associations we found are not necessary compositional in nature (*we must/il faut*, *people of canada/les canadiens*, *of course/évidemment*, etc).

3.3 Filtering

One way to increase the precision of the mapping process is to impose some linguistic constraints on the sequences such as simple noun-phrase contraints (Gaussier, 1995; Kupiec, 1993; hua Chen and Chen, 94; Fung, 1995; Evans and Zhai, 1996). It is also possible to focus on non-compositional compounds, a key point in bilingual applications (Su et al., 1994; Melamed, 1997; Lin, 99). Another interesting approach is to restrict sequences to those that do not cross constituent boundary patterns (Wu, 1995; Furuse and Iida, 96). In this study, we filtered for potential sequences that are likely to be noun phrases, using simple regular expressions over the associated part-of-speech tags. An excerpt of the association probabilities of a unit model trained considering only the NP-sequences is given in table 3. Applying this filter (referred to as $\mathcal{F}_{NP}$ in the following) to the 39,093 english sequences still surviving after previous filters $\mathcal{F}_1$ and $\mathcal{F}_2$ removes 35,939 of them (92%).

[2] The entities seen only once are mapped to a special "unknown" word

	model	*spared*	*ok*	*good*	*nu*	*u*
1	*baseline* – model 1	*48.98*	*0*	*0*	*747*	*0*
2	**baseline** – model 2	**51.83**	**0**	**0**	**747**	**0**
3	$\mathcal{E}_1 + \mathcal{F}_1(2,2,0,0.2)$	50.98	527	1702	5	626
4	$\mathcal{E}_1 + \mathcal{F}_1(2,2,5,0.2)$	51.61	596	2149	5	658
5	$\mathcal{E}_1 + \mathcal{F}_1(2,2,5,0.2) + \mathcal{F}_2$	51.72	633	2265	5	657
6	$\mathcal{E}_2 + \mathcal{F}_1(2,2,0,0.2)$	51.39	514	1551	43	578
7	$\mathcal{E}_2 + \mathcal{F}_1(2,2,5,0.2)$	51.99	470	1889	46	614
8	$\mathcal{E}_2 + \mathcal{F}_1(2,2,5,0.2) + \mathcal{F}_2$	52.12	493	1951	46	606
9	$\mathcal{E}_3 + \mathcal{F}_1(2,2,0,0.2)$	51.07	577	1699	43	588
10	$\mathcal{E}_3 + \mathcal{F}_1(2,2,5,0.2)$	51.47	629	2124	46	618
11	$\mathcal{E}_3 + \mathcal{F}_1(2,2,5,0.2) + \mathcal{F}_2$	51.68	665	2209	46	615
12	$\mathcal{E}_1 + \mathcal{F}_1(2,2,5,0.2) + \mathcal{F}_2 + \mathcal{F}_{NP}$	52.83	416	1302	4	564
13	$\mathcal{E}_3 + \mathcal{F}_1(2,2,5,0.2) + \mathcal{F}_{NP}$	53.12	439	1031	228	425
14	$\mathcal{E}_3 + \mathcal{F}_1(2,2,5,0.2) + \mathcal{F}_2 + \mathcal{F}_{NP}$	53.16	458	1052	199	439
15	$\mathcal{E}_3 + \beta = 0.4 + \mathcal{F}_1(2,2,5,0.2) + \mathcal{F}_{NP}$	53.22	495	1031	228	425

Table 4: Completion results of several translation models. *spared*: theoretical proportion of characters saved; *ok*: number of target units accepted by the user; *good*: number of target units that matched the expected whether they were proposed or not; *nu*: number of sentences for which no target unit was found by the translation model; *u*: number of sentences for which at least one helpful unit has been found by the model, but not necessarily proposed.

More than half of the 3,154 remaining NP-sequences contain only two words.

4 Results

We collected completion results on a test corpus of 747 sentences (13,386 english tokens and 14,506 french ones) taken from the Hansard corpus. These sentences have been selected randomly among sentences that have not been used for the training. Around 18% of the source and target words are not known by the translation model.

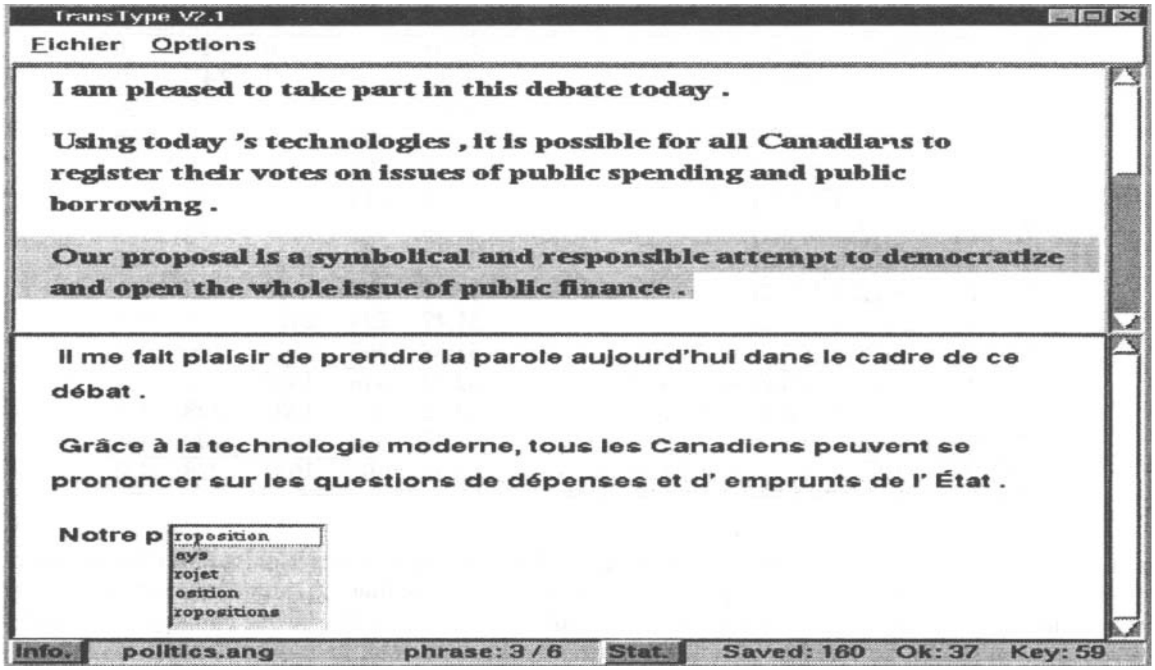
The baseline models (line 1 and 2) are obtained without any unit model (*i.e.* $\beta = 1$ in equation 2). The first one is obtained with an IBM-like model 1 while the second is an IBM-like model 2. We observe that for the pair of languages we considered, model 2 improves the amount of saved keystrokes of almost 3% compared to model 1. Therefore we made use of alignment probabilities for the other models.

The three next blocks in table 4 show how the parameter estimation method affects performance. Training models under the $\mathcal{E}_1$ method gives the worst results. This results from the fact that the word-to-word probabilities trained on the sequence based corpus (predicted by M_u in equation 2) are less accurate than the ones learned from the token based corpus. The reason is simply that there are less occurrences of each token, especially if many units are identified by the grouping operator.

In methods $\mathcal{E}_2$ and $\mathcal{E}_3$, the unit model of equation 2 only makes predictions $p_u(t|s)$ when s is a source unit, thus lowering the noise compared to method $\mathcal{E}_1$. We also observe in these three blocks the influence of sequence filtering: the more we filter, the better the results. This holds true for all estimation methods tried. In the fifth block of table 4 we observe the positive influence of the NP-filtering, especially when using the third estimation method.

The best combination we found is reported in line 15. It outperforms the baseline by around 1.5%. This model has been obtained by retaining all sequences seen at least two times in the training corpus for which the likelihood test value was above 5 and the entropy score above 0.2 ($\mathcal{F}_1(2,2,5,0.2)$). In terms of the coverage of this unit model, it is interesting to note that among the 747 sentences of the test session, there were 228 for which the model did not propose any units at all. For 425 of the remaining sentences, the model proposed at least one helpful (good or partially good) unit. The active vocabulary for these sentences contained an average of around 2.5 good units per sentence, of which only half (495) were proposed during the session. The fact that this model outperforms others despite its relatively poor coverage (compared to the others) may be explained by the fact that it also removes part of the noise introduced by decoupling the identification of the salient units from the training procedure. Furthermore, as we mentionned earlier, the more we filter, the less the grouping scheeme presented in equation 4 remains necessary, thus reducing a possible source of noise.

The fact that this model outperforms others, despite its relatively poor coverage, is due to the fact



I am pleased to take part in this debate today .

Using today 's technologies , it is possible for all Canadians to register their votes on issues of public spending and public borrowing .

Our proposal is a symbolical and responsible attempt to democratize and open the whole issue of public finance .

Il me fait plaisir de prendre la parole aujourd'hui dans le cadre de ce débat .

Grâce à la technologie moderne, tous les Canadiens peuvent se prononcer sur les questions de dépenses et d' emprunts de l' État .

Notre p

politics.ang phrase: 3 / 6 Saved: 160 Ok: 37 Key: 59

Figure 1: Example of an interaction in TRANSTYPE with the source text in the top half of the screen. The target text is typed in the bottom half with suggestions given by the menu at the insertion point.

that it also removes part of the noise that is introduced by dissociating the identification of the salient units from the training procedure. Furthermore, as we mentioned earlier, the more we filter, the less the grouping scheme presented in equation 4 remains necessary, thus further reducing an other possible source of noise.

5 Conclusion

We have described a prototype system called TRANSTYPE which embodies an innovative approach to interactive machine translation in which the interaction is directly concerned with establishing the target text. We proposed and tested a mechanism to enhance TRANSTYPE by having it predict sequences of words rather than just completions for the current word. The results show a modest improvement in prediction performance which will serve as a baseline for our future investigations. One obvious direction for future research is to revise our current strategy of decoupling the selection of units from their bilingual context.

Acknowlegments

TRANSTYPE is a project funded by the Natural Sciences and Engineering Research Council of Canada. We are undebted to Elliott Macklovitch and Pierre Isabelle for the fruitful orientations they gave to this work.

References

Adam L. Berger, Stephen A. Della Pietra, and Vincent J. Della Pietra. 1996. A maximum entropy approach to natural language processing. *Computational Linguistics*, 22(1):39–71.

Peter F. Brown, Stephen A. Della Pietra, Vincent Della J. Pietra, and Robert L. Mercer. 1993. The mathematics of machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–312, June.

Ted Dunning. 93. Accurate methods for the statistics of surprise and coincidence. *Computational Linguistics*, 19(1):61–74.

David A. Evans and Chengxiang Zhai. 1996. Noun-phrase analysis in unrestricted text for information retrieval. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 17–24, Santa Cruz, California.

George Foster, Pierre Isabelle, and Pierre Plamondon. 1997. Target-text Mediated Interactive Machine Translation. *Machine Translation*, 12:175–194.

Pascale Fung. 1995. A pattern matching method for finding noun and proper noun translations from noisy parallel corpora. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pages 236–243, Cambridge, Massachusetts.

Osamu Furuse and Hitoshi Iida. 96. Incremental translation utilizing constituent boundray patterns. In *Proceedings of the 16th International Conference On Computational Linguistics*, pages 412–417, Copenhagen, Denmark.

Éric Gaussier. 1995. *Modles statistiques et patrons morphosyntaxiques pour l'extraction de lexiques bilingues*. Ph.D. thesis, Universit de Paris 7, janvier.

Masahiko Haruno, Satoru Ikehara, and Takefumi Yamazaki. 96. Learning bilingual collocations by word-level sorting. In *Proceedings of the 16th International Conference On Computational Linguistics*, pages 525–530, Copenhagen, Denmark.

Kuang hua Chen and Hsin-Hsi Chen. 94. Extracting noun phrases from large-scale texts: A hybrid approach and its automatic evaluation. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 234–241, Las Cruces, New Mexico.

Satoru Ikehara, Satoshi Shirai, and Hajine Uchino. 96. A statistical method for extracting uinterupted and interrupted collocations from very large corpora. In *Proceedings of the 16th International Conference On Computational Linguistics*, pages 574–579, Copenhagen, Denmark.

Julian Kupiec. 1993. An algorithm for finding noun phrase correspondences in bilingual corpora. In *Proceedings of the 31st Annual Meeting of the Association for Computational Linguistics*, pages 17–22, Colombus, Ohio.

Dekang Lin. 99. Automatic identification of noncompositional phrases. In *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, pages 317–324, College Park, Maryland.

I. Dan Melamed. 1997. Automatic discovery of noncompositional coumpounds in parallel data. In *Proceedings of the 2nd Conference on Empirical Methods in Natural Language Processing*, pages 97–108, Providence, RI, August, 1st-2nd.

Makoto Nagao and Shinsuke Mori. 94. A new method of n-gram statistics for large number of n and automatic extraction of words and phrases from large text data of japanese. In *Proceedings of the 16th International Conference On Computational Linguistics*, volume 1, pages 611–615, Copenhagen, Denmark.

Franz Josef Och and Hans Weber. 98. Improving statistical natural language translation with categories and rules. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 985–989, Montréal, Canada.

Graham Russell. 1998. Identification of salient token sequences. Internal report, RALI, University of Montreal, Canada.

Sayori Shimohata, Toshiyuki Sugio, and Junji Nagata. 1997. Retrieving collocations by cooccurrences and word order constraints. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics*, pages 476–481, Madrid Spain.

Keh-Yih Su, Ming-Wen Wu, and Jing-Shin Chang. 1994. A corpus-based approach to automatic compound extraction. In *Proceedings of the 32nd Annual Meeting of the Association for Computational Linguistics*, pages 242–247, Las Cruces, New Mexico.

Ye-Yi Wang and Alex Waibel. 98. Modeling with structures in statistical machine translation. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, volume 2, pages 1357–1363, Montréal, Canada.

Dekai Wu and Hongsing Wong. 98. Machine translation with a stochastic grammatical channel. In *Proceedings of the 36th Annual Meeting of the Association for Computational Linguistics*, pages 1408–1414, Montréal, Canada.

Dekai Wu. 1995. Stochastic inversion transduction grammars, with application to segmentation, bracketing, and alignment of parallel corpora. In *Proceedings of the International Joint Conference on Artificial Intelligence*, volume 2, pages 1328–1335, Montréal, Canada.