Mach Translat
DOI 10.1007/s10590-010-9082-5

Factored bilingual *n*-gram language models for statistical machine translation

Josep M. Crego · François Yvon

Received: 2 November 2009 / Accepted: 12 June 2010

Abstract In this work, we present an extension of *n*-gram-based translation models based on factored language models (FLMs). Translation units employed in the *n*-gram-based approach to statistical machine translation (SMT) are based on mappings of sequences of raw words, while translation model probabilities are estimated through standard language modeling of such bilingual units. Therefore, similar to other translation model approaches (phrase-based or hierarchical), the sparseness problem of the units being modeled leads to unreliable probability estimates, even under conditions where large bilingual corpora are available. In order to tackle this problem, we extend the *n*-gram-based approach to SMT by tightly integrating more general word representations, such as lemmas and morphological classes, and we use the flexible framework of FLMs to apply a number of different back-off techniques. In this work, we show that FLMs can also be successfully applied to translation modeling, yielding more robust probability estimates that integrate larger bilingual contexts during the translation process.

Keywords Statistical machine translation · Bilingual *n*-gram language models · Factored language models

J. M. Crego (✉)
LIMSI/CNRS, BP 133, 91430 Orsay Cedex, France
e-mail: jmcrego@limsi.fr

F. Yvon
LIMSI/CNRS and Université Paris-Sud, BP 133,
91430 Orsay Cedex, France

Published online: 30 June 2010

1 Introduction

Phrase-based (Och et al. 1999; Zens et al. 2002; Koehn et al. 2003), and hierarchical (Melamed 2004; Chiang 2005) translation models are currently the most widely-used approaches to Statistical Machine Translation (SMT). Each approach is characterized by the elementary *translation units* it models and the way these units are combined to build translations. The units of phrase-based models are *phrase pairs*,[1] which define a probabilistic relationship between variable-length sequences of source and target words; these units are combined through concatenation. Bilingual n-gram language models (Mariño et al. 2006) use and combine *tuples*, which are defined similarly to phrases, but extracted in a different manner. In contrast, *Hierarchical phrases* allow for gaps, and are combined by context-free rewrite operations. The statistical modeling of the translation context is one of the main differences between the above approaches.

Most current approaches in SMT model the mapping of sequences of *word forms*, and are thus plagued by sparseness problems caused by the limited amount of training data. This problem is exacerbated when one (or both) languages under study possesses a rich morphology, which tends to multiply the number of surface forms associated with a given lemma. Furthermore, translation units are extracted by crude heuristics: as a result, the number of extracted units is very large, yielding enormous and noisy translation tables.

In order to mitigate these problems, a common strategy is to integrate richer linguistic representations into the translation model, so as to obtain more reliable statistics. Most attempts to date focus on normalizing the input source stream during a pre-processing step so as to yield more reliable word alignments, that will turn into less sparse counts, and better probability estimates. Normalization can include various reordering operations, word splitting or compounding; normalizing the morphological variants is also typically performed by clustering word forms that have similar translation equivalents (Nießen and Ney 2001, 2004; Yang and Kirchhoff 2006; Talbot and Osborne 2006). A similar approach is used by Habash and Sadat (2006), who show that translation accuracy gains can be obtained by converting the input word forms into a less sparse representation using lemmas, part-of-speech tags, etc., before passing the pre-processed stream to the translation system. Recently, a different approach was proposed in Koehn and Hoang (2007), where the authors introduce a *multi-stream translation model*, which allows to integrate the same type of morphological information; yet, this new framework poses difficult novel computational issues.

Encouraged by the results reported in Koehn and Hoang (2007), we enhance a bilingual n-gram translation model by integrating additional information sources at the word level (e.g. lemmas, roots, part-of-speech tags, etc.) by means of *factored bilingual language models* (FLMs), thus extending the proposal of Bilmes and Kirchhoff (2003). We use the additional information together with the surface form of tuples to build a vector of factors for each translation unit. Factors are taken into account for bilingual n-gram modeling, allowing to back-off to a factored form of a translation unit instead of directly to a lower order model. Accordingly, we aim at producing more

[1] These *phrases* do not necessarily correspond to syntactic constituent in one or the other language.

robust estimations for sparse units while increasing the amount of bilingual context that is taken into account during the translation process.

The rest of this paper is organized as follows. In Sect. 2, we describe the tuple n-gram translation model, which constitutes the core model implemented by an n-gram-based SMT system. Section 3 introduces the basics of factored language models (FLMs). Next, in Sect. 4 we explain how the n-gram translation model has been extended to accommodate this novel formalism. Finally, the experimental framework is presented in Sect. 5, where we also report and discuss our experimental results. We outline further research work and draw conclusions in Sect. 6.

2 Bilingual n-gram language models

The translation model considered in this work is based on bilingual n-grams. In this model, the elementary translation units (*tuples*) are pairs of variable-length sequences (s, t), where s (resp. t) is a sequence of source (resp. target) words. The translation model defines probability over sequences of such units, yielding sentences of a particular bi-language. Assuming a Markovian dependency of order 2 between tuples, the joint probability of a source s_1^J and target t_1^I sentences is computed as:

$$P(s_1^J, t_1^I) = \prod_{l=1}^{L} P((s,t)_l | (s,t)_{l-1}, (s,t)_{l-2}) \tag{1}$$

It is important to notice that, since both languages are linked up via tuples, the contextual information taken into account during translation is bilingual. As for any standard n-gram language model, this translation model is estimated over a training corpus composed of sentences of the language being modeled, in this case, sentences of the *bi-language* previously introduced.

The translation model implemented as (1) relies on two requirements: (i) that the source and target side of a tuple words are synchronized, i.e. that they occur in the same order in their respective languages and (ii) a joint segmentation of s_1^J and t_1^J in L tuples is available, which uncovers the tuple boundaries. It is a well-known linguistic fact that both requirements are not usually met in natural parallel sentences. This explains why training our model requires a substantial pre-processing of the training corpus so as to derive parallel sentences that actually meet these requirements. This pre-processing is a two-step process: the first step is similar to what is typically done for training conventional phrase-based systems, and consists in deriving word alignments for each sentence in the parallel corpus. Based on this information, the second step produces a joint segmentation of the source and target sentences in tuples.

Segmentation in tuples is made (almost) deterministic by enforcing the following constraints:

- no word inside a tuple can be aligned with a word outside the tuple;
- segmentation must respect the order of words as they occur on the target side. Reordering is permitted in the source side so as to synchronize the source and target sides of a sentence;

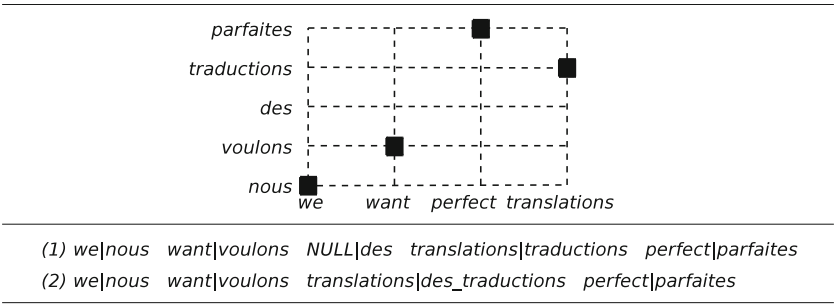

Fig. 1 Tuple/phrase extraction from an aligned sentence pair

– no smaller tuple can be extracted without violating the previous constraints.

Figure 1 presents a simple example illustrating the unique tuple segmentation for a given word-aligned pair of sentences (top).

Note that, in this example, the English source words *perfect* and *translations* have been reordered in the final tuple segmentation, while the French target words are kept in their original order. The resulting sequence of tuples (1) is further refined (2) to avoid *NULL* words in source side of the tuples. Conventionally, non-aligned words on the target side are linked to *NULL* nodes, and yield tuples whose source side is empty. This is, for instance, the case of the third target word in the example presented on Fig. 1, which yields the tuple *NULL*|*des*. Modeling explicitly the apparition of *NULL* words in the input sentence would bog down decoding, as *NULL* words may appear anywhere in the source stream and linked to a huge number of translation options. Hence, a *tuple refinement* must take place, which basically amounts to combining unaligned target words with their left or right neighbour(s) in a deterministic way. In our implementation, the attachment orientation is chosen after evaluating the translation likelihood of both choices using word lexicon probabilities. See Crego and Mariño (2007b) for further details on the tuple extraction process. Note that tuples with *NULL* target side are allowed in our model as none of the previous difficulties apply.

The application of the model defined in Eq. 1 expects synchronized tuple streams in training; likewise, it produces synchronized tuple streams in inference. This means that the input source stream has to be reordered *prior to translation*, so as to reproduce the word order changes introduced during the tuple extraction process. In our system, several possible reorderings of the source are considered in parallel. To this end, the sentence to be translated is first turned into a word lattice containing the most promising reordering hypotheses; it is then straightforward to search this lattice in a monotonous fashion for the best translation hypotheses as defined by Eq. 1. Further details regarding source reordering are given in Crego and Mariño (2007b).

Assuming the source has been properly reordered, the model is used to predict the most likely target side of a tuple, whose source side is known, in the context of the past $n-1$ complete tuples. Larger contexts does not imply here longer units, as is the case in conventional phrase-based models, but *larger n-gram histories*, which allow to make better predictions.

3 Factored language models

In this section, we briefly describe FLMs and the generalized back-off they implement. FLMs (Bilmes and Kirchhoff 2003; Kirchhoff et al. 2008) can be seen as a generalization (or sophistication) of conventional n-gram language models, from which they differ in several important aspects.

Firstly, FLMs use a richer set of features to describe the units of language than n-gram LMs, in order to achieve a more accurate description of the regularities of a particular language. In a FLM, a word, w_t, is seen as a bundle of K factors:

$$w_t \equiv [f_t^1, f_t^2, \ldots, f_t^K] = f_t^{1:K} \tag{2}$$

Factors of a word can be anything, including morphological classes, stems, roots, and any other linguistic features that might correspond to or decompose a word. Assuming that part-of-speech (POS) tags are available, a FLM could be used to define a joint distribution over pairs $w \equiv (f^1 = \text{word}, f^2 = \text{tag})$. Using this additional information has the effect of increasing the number of units, at the risk of worsening sparsity issues. However, it turns out that these linguistic features can also be used to generalize surface forms, thus allowing better smoothing strategies, and more robust probability estimates.

A FLM is then simply a statistical language model over these compounded units. The probability of a sentence containing T words is the probability of the corresponding sequence of factors, as in:

$$P(w_1, w_2, \ldots, w_T) = P(f_1^{1:K}, f_2^{1:K}, \ldots, f_T^{1:K}) \tag{3}$$

As in n-gram LMs, FLMs make the assumption that the word occurring at position t, w_t, only depends on the $n-1$ preceding words, allowing to rewrite the previous equation as:

$$P(w_1, w_2, \ldots, w_T) = \prod_t P(f_t^{1:K} | f_{t-n-1}^{1:K}, \ldots, f_{t-1}^{1:K}) \tag{4}$$

In this way, the occurrence of each word form is not only dependent on a single stream of preceding word forms, but also on additional parallel streams of factors.

Training large n-gram models is plagued with sparsity issues: n-grams with small occurrence counts in the training data will lead to poor probability estimates; in the extreme case of n-grams that do not occur at all, maximum likelihood estimates are exactly zero. A widely used solution is resort to *back-off strategies* (Katz 1987): for all unseen k-grams $w_1 \ldots w_k$ ($1 \leq k \leq n$), the corresponding conditional probability estimate $P(w_k|w_1 \ldots w_{k-1})$ is taken to be a fraction of the estimate obtained with a reduced history $P(w_k|w_2 \ldots w_{k-1})$. In other words, in standard n-gram language models, back-off strategies amount to dropping one of the variables on the right of the conditioning bar at a time, *eg* going from a trigram $P(w_t|w_{t-1}, w_{t-2})$ down to a bigram $P(w_t|w_{t-1})$, then possibly down to a unigram, etc. This is illustrated by the *linear back-off path* over words displayed on Fig. 2 (left).

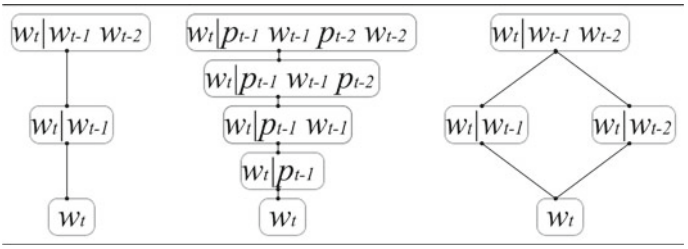

Fig. 2 Different back-off strategies represented in the form of back-off graphs

The availability of linguistically enriched units in FLMs permits to consider novel and more flexible back-off strategies. Firstly, additional factors may be introduced in the history: Fig. 2 (middle) displays a *l*inear back-off path over words and POS-tags (p_t). Continuing on our example, assume we wish to train a bi-gram model over (*word*, *pos*) pairs. For lack of seeing the bigram "(the,DT) (paper,NN)" in the training data, a conventional back-off model would take P((paper,NN)|(the,DT)) to be a fraction of the unigram probability P((paper,NN)). In FLMs, it is possible to derive the estimate of the same conditional probability based on P((paper,NN)|(NN)), where the conditioning history has been generalized at the POS level.

Second, back-off paths more sophisticated than the simple linear chain can be considered. These paths correspond to various ways to recursively generalize the history. In particular, the drop-oldest-first path is not necessarily the unique choice under FLMs. Figure 2 (right) displays several back-off paths on a *back-off graph* over words.

Paths in this graph can be chosen in advance based on linguistic information, or at run-time, based on statistical criteria such as counts in the training set. Hence, a factored LM can fork upon encountering an unseen factor sequence, and estimate the missing conditional probability by combining the estimates obtained through several back-off paths, a procedure called *generalized parallel back-off* (Bilmes and Kirchhoff 2003). Further details on FLMs can be found in Kirchhoff et al. (2008).

4 Factored bilingual *n*-gram language models

As explained above, the purpose of statistical language models is to help accurately predict the next word w_t based on its current history $h_t = [w_0, \ldots, w_{t-1}]$. Over the past years, considerable efforts have been reported to find the factors in h_t that best predict w_t, including the use of syntactic and semantic information, see for instance (Niesler 1997; Rosenfeld 1996; Bod 2000; Chelba and Jelinek 2000; Charniak 2001). Much of the previous work has been carried out to model natural languages, such as French or English, with varied characteristics and singularities. However, in this work we deal with a special language built from *bilingual* units, which further increases the difficulty of making accurate predictions.

Roughly speaking, we are facing a language with much sparser units than natural languages and with a very peculiar structure, as it loosely incorporates the grammar of two different languages. The *bilingual* language inherits the modeling difficulties

Table 1 Tuple factors

Factors	Sequence of (factored) tuples			
$(s,t)^w$	*we*|*nous*	*want*|*voulons*	*translations*|*des_traductions*	*perfect*|*parfaites*
s^w	*we*|	*want*|	*translations*|	*perfect*|
t^w	|*nous*	|*voulons*	|*des_traductions*	|*parfaites*
$(s,t)^l$	*we*|*nous*	*want*|*vouloir*	*translation*|*du_traduction*	*perfect*|*parfait*
s^l	*we*|	*want*|	*translation*|	*perfect*|
t^l	|*nous*	|*vouloir*	|*du_traduction*	|*parfait*
$(s,t)^p$	*pron.*|*pron.*	*verb*|*verb*	*noun*|*prep._noun*	*adj.*|*adj.*
s^p	*pron.*|	*verb*|	*noun*|	*adj.*|
t^p	|*pron.*	|*verb*	|*prep._noun*	|*adj.*

of both the monolingual languages, in particular those that derive from the need to take morphological processes such as inflection, derivation, compounding, agglutination etc., into account; aggravated by the structural characteristics of the bi-language, whose units link up words of two different languages.

In this context, we propose to address sparsity issues by using translation units that simultaneously integrate several representations of the source and target words they comprise. The probability of the translation model based on this unit are estimated using back-off schemes that are made available in the formalism of FLMs. A generalized parallel back-off procedure is thus used to allow the most specific representation of a unit (built from surface word forms) when enough available information exists or to back off to a more general representation form. Hence, translation models can draw on more reliable statistics to overcome data sparseness estimation problems. Additionally, many aspects of translation can be better addressed if linguistic information is directly available in the translation model. Since word *lemmas* and *part-of-speech* tags are linguistically motivated and have much smaller cardinality than the word-form vocabulary, we introduce translation unit factors corresponding to these information sources.

Table 1 displays factored translation units for the example used in Fig. 1 (Sect. 2). Each translation unit is represented as a bundle of 9 factors, which are formed from surface word forms (w), lemmas (l) and POS-tags (p), while considering the joint (s, t), source-side (s) and target-side (t) tuple information. Source and target tuple information is considered independently so as to allow more flexible back-off schemes, where we generalize to only the source (or target) side. Note that the set of factored tuples extracted from the training corpus is the same as that of regular tuples detailed in Sect. 2.

Bilingual language models were introduced in Eq. 1. Expending the factors that occur in the history of a given tuple, the probability of a joint source and target pair in the bilingual n-gram model can be expressed as a product of terms such as :

$$P\left((s,t)_0^w | s_1^w, t_1^w, s_1^l, t_1^l, s_1^p, t_1^p, s_2^w, t_2^w, s_2^l, t_2^l, s_2^p, t_2^p\right) \tag{5}$$

Table 2 Factored back-off strategies

$P((s,t)_0^w \mid (s,t)_1^w, (s,t)_2^w)$	*want*|*voulons*	*translations*|*des_traductions*	*perfect*|*parfaites*
$P((s,t)_0^w \mid (s,t)_1^w)$	–	*translations*|*des_traductions*	*perfect*|*parfaites*
$P((s,t)_0^w \mid (s,t)_1^w, (s,t)_2^p)$	**verb**|**verb**	*translations*|*des_traductions*	*perfect*|*parfaites*
$P((s,t)_0^w \mid (s,t)_1^w, t_2^w)$	|**voulons**	*translations*|*des_traductions*	*perfect*|*parfaites*
$P((s,t)_0^w \mid (s,t)_1^p, (s,t)_2^w)$	*want*|*voulons*	**noun**|**prep._noun**	*perfect*|*parfaites*

where the exponents w, l and p denote respectively that the corresponding tuple source- and target-sides correspond respectively to surface forms, lemmas and POS-tag information. In this formulation, indices 0, 1 and 2 denote respectively the current, the preceding and the second preceding token. The rationale behind using a factored n-gram LM instead of a standard n-gram LM is that we may need to translate sequences of tuples that do not occur in training data in their surface word factor form, but that may exist with one or several units in a different factor form. During decoding, this fact can be exploited by generalized back-off algorithms, which allow to take into account larger histories of bilingual contexts for estimating the conditional probability associated with these unseen sequences of tuples. This is illustrated in Table 2 with several examples of back-off strategies.

Following with our example, assume we need to compute the trigram probability of [*perfect*|*parfaites*] which occurs in the context of [*want*|*voulons translations*|*des_traductions*] (first row) and that the corresponding trigram has never been seen during training.

The use of a conventional language model would imply falling back to the bigram estimate (second row); in the factored model, we can back-off to estimates for the probability of [*perfect*|*parfaites*] in different trigram contexts:

- Generalizing the most distant tuple using the factor built from POS-tags (third row).
- Generalizing the most distant tuple using the factor built from target-side information (fourth row).
- Allowing multiple back-off paths instead of the drop-oldest-first typically used (last row). Note that in this case a different back-off path is used in combination with the generalization of the unit using the factor built from POS-tags.

Proceeding this way, we are in a position to introduce larger *factored* contexts in the translation process. Notice that, while Table 2 shows only four back-off strategies, multiple combinations of factors and back-off strategies can be used to estimate a statistical n-gram LM. Furthermore, features such as the n-gram order and discounting method also need to be considered. The previous example shows that FLMs provide a flexible framework for incorporating additional information sources and back-off strategies into language modeling, which may help in the overall objective of finding a model with the best compromise between predictability (lowest perplexity) and estimation error (overfitting).

Table 3 Statistics for the training and test data sets

Set	Language	Sentences	Words	Vocabulary	OOV	Lmean
Train	French	1.41 M	43.5 M	136 k	–	30.7
	English	1.41 M	39.0 M	118 k	–	27.6
Tune	French	2000	64.3 k	7, 507	110	32.17
	English	2000	59.1 k	6, 428	73	29.58
Test	French	2000	65.6 k	7, 648	99	32.82
	English	2000	60.1 k	6, 497	93	30.09

M and k stand respectively for millions and thousands

5 Experimental framework

This section describes the translation tasks we have considered in our experiments. Section 5.2 details the preprocessing, training and optimization steps. Results are presented in Sect. 5.3 and discussed in Sect. 5.4.

5.1 Corpora

We have used the *EPPS* corpus (v4) made available for the fourth Workshop on SMT.[2] The corpus is composed of proceedings of the European Parliament, which exist for 11 European languages. In the case of the experiments presented here, we have only used the French and English versions. Our development and test sets correspond to the *test2007* and *test2008* file sets, hereinafter referred to as *tune* and *test* respectively. Monolingual training data corresponds to the *news-train08* file sets. The above file sets correspond to those made available for the 2008 workshop.

The training data is preprocessed by using standard tools for tokenization and filtering. French and English lemmas and Part-of-speech tags are computed by means of the *TreeTagger* toolkit.[3] The French set of POS tags contains 34 tags, while the English set contains 45 tags. French and English lemma vocabularies contain respectively 27,721 and 32,105 entries.

Table 3 presents statistics for the training and test datasets for each considered language, once fully pre-processed. More specifically, the statistics reported are the number of sentences, the number of running words, the vocabulary size, the number of out-of-vocabulary words and the average sentence length (in number of words). A single reference translation is available for all test sets.

Table 4 shows the vocabulary sizes of the tuple factors used in this work. As can be seen, units built from word *lemmas* and *part-of-speech* tags have much smaller cardinality than the word-form vocabulary.

[2] http://www.statmt.org/wmt09/.

[3] http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger.

Table 4 Vocabularies of factored units

Factors	French ⇝ English	English ⇝ French
$(s,t)^w$	2,084,527	1,893,078
s^w	1,023,665	870,215
t^w	781,777	944,437
$(s,t)^l$	1,826,096	1,704,677
s^l	818,836	764,593
t^l	699,005	794,790
$(s,t)^p$	425,343	418,310
s^p	56,852	55,096
t^p	63,219	55,875

5.2 System description

After preprocessing, word-to-word alignments are performed in both translation directions using the *GIZA++* toolkit.[4] Then, the *grow-diag-final-and* (Koehn et al. 2005) heuristic is used to compute the final alignments from which tuples are extracted. The overall search process is performed by our in-house *n-code* decoder. It implements a beam-search strategy on top of a dynamic programming algorithm (Crego and Mariño 2007a). Reordering hypotheses are computed in a pre-processing step making use of reordering rules built from the word reorderings introduced in the tuple extraction process. The resulting reordering hypotheses are passed to the decoder in the form of word lattices (Crego and Mariño 2007b).

In addition to the tuple n-gram translation model, our system implements seven additional features functions which are linearly combined following a discriminative modeling framework (Och and Ney 2002):

- a *target-language model* which provides information about the target language structure and fluency;
- two *lexicon models*, which constitute complementary translation models computed for each given tuple;
- two *lexicalized reordering models* aiming at predicting the orientation of the next translation unit. Three orientation choices are considered: continuous/forward/backward. The monotonic and swap orientations typically used in phrase-based systems (Koehn et al. 2005) are lumped together in our continuous orientation, while forward and backward orientations imply skipping one or more words on either side respectively. The orientation of each unit with respect to its left and right neighbors are considered, yielding two separate models;
- a 'weak' distance-based *distortion model*;
- and finally a *word-bonus model* and a *tuple-bonus model* which are used in order to compensate for the system preference for short translations.

[4] http://www.fjoch.com/GIZA++.html.

Monolingual language models are estimated using the *SRI language modeling* toolkit.[5] Kneser–Ney smoothing (Kneser and Ney 1995) and interpolation options are used as they typically achieve the best performance. Model weights are optimized by means of the widely used *MERT* toolkit,[6] slightly modified to perform optimizations embedding our decoder. The *BLEU* (Papineni et al. 2002) score is used as objective function for MERT and to evaluate test performance.

We have experimented with several configurations of factored models to estimate the bilingual n-gram model:

3g The first configuration is our baseline system, a conventional trigram LM following a standard drop-oldest-first back-off path. In this case, the model considers only units built from surface forms:

$$P((s,t)_0^w|(s,t)_1^w,(s,t)_2^w) \tag{6}$$

4g A 4-gram LM is also implemented, following the same drop-oldest-first back-off path:

$$P((s,t)_0^w|(s,t)_1^w,(s,t)_2^w,(s,t)_3^w) \tag{7}$$

factor 3g This configuration considers generalized units. The sequence of factors used to identify the configuration refers to the linear back-off path used to compute probability estimates. This is, the last term, $(s,t)_2^w$ is the first factor dropped when the entire history is not found. The next dropped factor is $(s,t)_2^l$, then $(s,t)_2^p$, etc. to finally fall back to the unigram $(s,t)_0^w$:

$$P((s,t)_0^w|(s,t)_1^p,(s,t)_1^l,(s,t)_1^w,(s,t)_2^p,(s,t)_2^l,(s,t)_2^w) \tag{8}$$

factor 4g Equivalently to the previous configuration, three new factors are considered. This model also follows a linear back-off strategy:

$$\begin{aligned} P((s,t)_0^w|(s,t)_1^p,(s,t)_1^l,(s,t)_1^w,\\ (s,t)_2^p,(s,t)_2^l,(s,t)_2^w,\\ (s,t)_3^p,(s,t)_3^l,(s,t)_3^w) \end{aligned} \tag{9}$$

factor disjoint 3g This model configuration considers independently source and target histories. As for the previous configurations, it also implements

[5] http://www.speech.sri.com/projects/srilm/

[6] http://www.statmt.org/moses/

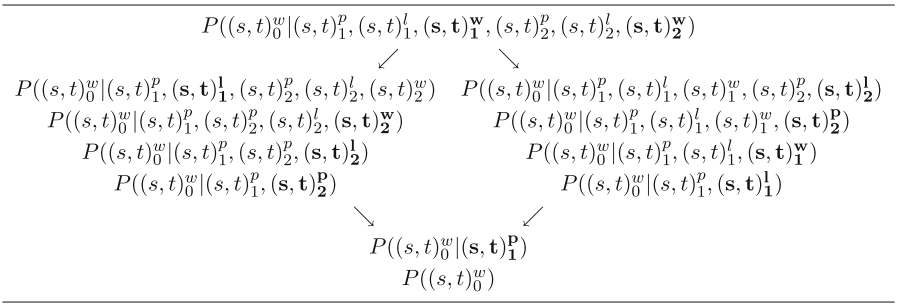

Fig. 3 Generalized back-off strategy

the drop-oldest-first linear back-off strategy. Note that this model estimates the probability of a given tuple conditioned to a history of 12 terms:

$$P((s,t)_0^w|t_1^p,s_1^p,t_1^l,s_1^l,t_1^w,s_1^w,t_2^p,s_2^p,t_2^l,s_2^l,t_2^w,s_2^w) \quad (10)$$

factor mean 3g This model implements a generalized back-off strategy. Rather than using just one back-off path, as in the first four configurations, two back-off paths are combined. When a requested n-gram is missing, both back-off paths are triggered and the arithmetic mean of both lower level model probabilities is returned. Figure 3 illustrates the back-off path used for this configuration.

5.3 Results

In this section we evaluate the performance of the factored models proposed in terms of perplexity (Sect. 5.3.1) and translation accuracy (Sect. 5.3.2). Furthermore, we assess the introduction of additional bilingual context in the translation process (Sect. 5.3.3).

5.3.1 Perplexity

The quality of an n-gram language model can be measured by its perplexity. However, perplexity measurements require samples from the language under study, in this case samples of the bilingual language over tuples. We have therefore re-aligned the entire training corpus, including the tune set, and used the word alignments to parse each pair of parallel sentences in the tune set into the corresponding sequence of tuples. Perplexity measurements are computed based on these 2,000 pairs of sentences. Table 5 displays the perplexity scores obtained by the factored models.

As can be seen, all bilingual n-gram models built with additional factors and different back-off techniques achieve lower perplexity scores than the baseline model. Numbers in boldface are used to outline the best perplexity scores, while italic letters are used to outline the results of our baseline model. Compared to the baseline models,

Table 5 Perplexity scores

FLM	French ⇝ English	English ⇝ French
3g	*150*	*117*
4g	146	114
factor 3g	**135**	**103**
factor 4g	140	109
factor mean 3g	145	115
factor disjoint 3g	148	117

Table 6 Translation accuracy (BLEU) scores

FLM	French ⇝ English					English ⇝ French				
	75 k	150 k	300 k	600 k	All	75 k	150 k	300 k	600 k	All
3g	*28.23*	*28.90*	*29.61*	*30.41*	*30.85*	*27.88*	*28.77*	*29.62*	*29.81*	*30.40*
4g	28.01	28.92	29.36	29.93	30.37	27.77	28.54	29.37	29.61	29.97
factor 3g	28.27	**29.15**	**30.02**	30.43	30.78	28.20	**29.23**	**29.90**	**30.31**	30.33
factor 4g	28.17	28.95	29.56	29.87	30.21	27.88	28.95	29.63	29.78	29.95
factor mean 3g	28.19	**29.09**	29.81	**30.64**	30.56	**28.12**	**29.16**	**30.01**	**30.19**	30.28
factor disjoint 3g	28.19	**29.19**	**30.08**	30.28	30.64	**28.09**	**29.10**	29.79	**30.25**	30.44
Moses	27.71	28.65	29.62	29.80	30.19	27.27	28.21	28.95	28.73	29.37

3g, the highest perplexity reduction is achieved by models *factor 3g*, with near 10% reduction in both translation directions.

5.3.2 Translation accuracy

In Table 6, we report translation accuracy results in terms of *BLEU*, obtained over the test set for both translation tasks and factored bilingual model configurations. In order to measure the impact of the new bilingual models according to different data size conditions, we show results for systems trained over the first $75k$, $150k$, $300k$, $600k$ and $1,41M$ (all) sentences of the training corpus detailed in Table 3. We also report the *BLEU* scores obtained by the *Moses* SMT system Koehn et al. (2007), a state-of-the-art phrase-based decoder. Moses models were built using the same training data, alignment and preprocessing conditions than the n-gram-based SMT systems; Moses systems were tuned with a standard implementation of Minimum Error Rate Training. We have used Moses with default option settings, including lexicalized reordering.

As expected, translation accuracy results do not entirely correlate with perplexity measures, as multiple additional variables must be considered for the accuracy evaluation, such as the additional models used in the systems. Numbers in italics are used to outline the baseline results while numbers in boldface outline the best scores when higher than the baseline.

The best results are obtained by the systems performing translation with the *factor 3g*, *factor mean 3g* and *factor disjoint 3g* bilingual FLMs, all of which outperform

Table 7 n-gram hits (%)

factor 3g	%	%	*3g*
$(s,t)_0^w(s,t)_1^p(s,t)_1^l(s,t)_1^w(s,t)_2^p(s,t)_2^l(s,t)_2^w$	48	48	$(s,t)_0^w(s,t)_1^w(s,t)_2^w$
$(s,t)_0^w(s,t)_1^p(s,t)_1^l(s,t)_1^w(s,t)_2^p(s,t)_2^l$	1	–	–
$(s,t)_0^w(s,t)_1^p(s,t)_1^l(s,t)_1^w(s,t)_2^p$	14	–	–
$(s,t)_0^w(s,t)_1^p(s,t)_1^l(s,t)_1^w$	18	33	$(s,t)_0^w(s,t)_1^w$
$(s,t)_0^w(s,t)_1^p(s,t)_1^l$	1	–	–
$(s,t)_0^w(s,t)_1^p$	9	–	–
$(s,t)_0^w$	9	19	$(s,t)_0^w$

the baseline results in most of the cases. No gains over the baseline are observed for both translation directions under the *all* data condition. FLMs mainly address the sparseness problems of conventional translation models. Hence, lowering its performance under *'enough'* training data conditions. In all cases, all differences fall within the confidence margin, as approximately, ±0.92 points *BLEU* must be added for a 95% confidence level. Note that all system models and search settings are identical in all experiments, with the exception of the bilingual n-gram language model being analyzed. As can be seen, 4-gram models, *4g* and *factor 4g*, do not improve over the corresponding trigram models. The results obtained with *Moses* yield slightly lower accuracy results than those obtained by our n-code system, which can be explained by the fact that our Moses system only uses surface forms, whereas the n-code systems have access to richer linguistic representations.

5.3.3 Back-off strategies

To further analyze the impact of factored models, we report in Table 7 statistics related to the n-grams that are actually used in the system for the *factor 3g* model, and compare them with the baseline back-off path. The numbers reported were computed using those tuples that are actually used to translate the test sets; for each of these, we analyze the translation context (n-grams) where they appeared. Notice that percentages are approximate, the total number of tuples employed to translate the test sets is approximately the same for all configurations. Results are only shown for the French-to-English translation direction.

As expected, the percentage of hits of trigram units (using all the available context) is the same for both models (48%). However, while the standard trigram LM falls back to bigrams in 33% of the cases, the factored model distributes this percentage between bigrams (18%) and generalized trigrams hits (1% at the level of lemmas and 14% at the level of POS-tags). A similar situation is found for unigrams. The baseline model backs off to unigrams for 19% of its predictions, while the factored model distributes this percentage between unigrams (9%) and bigrams (1 and 9%). The results in Table 7 clearly show that a larger amount of bilingual context is actually taken into account (25%) during the translation process when we use factored models.

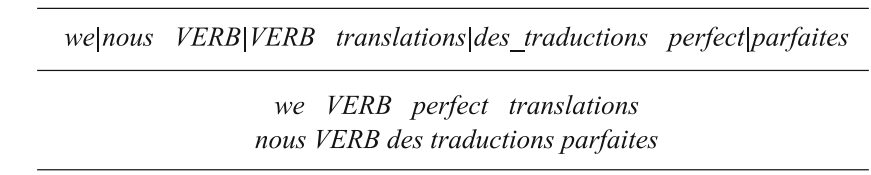

Fig. 4 Translation unit in the form of tuple n-grams with a factored unigram (*top*) and its equivalent phrase (*bottom*)

5.4 Discussion

Tuple n-grams can be viewed as the analog of phrases of the phrase-based approach to SMT. A tuple n-gram with factored units such as the one showed at the top of Fig. 4 is thus analogous, under a phrase-based approach, to the phrase shown at the bottom of the same Figure. As we have seen, the generalized parallel back-off procedure of FLMs allows to back-off from the n-gram of tuples to a generalized form (using lemmas or POS-tags) of any *of its individual tuples*. In this example, the second tuple is represented generalized using its POS-tag.

To obtain a similar behavior under the phrase-based approach, a huge number of probability estimates would need to be computed for the multiple combinations of phrases with *'sub-phrases'* built from alternative factored forms. A computationally extremely expensive work. In addition, it is not clear how generalization steps (back-off) would be applied at decoding time.

The factored phrase-based model presented in Koehn and Hoang (2007) addresses this problem by performing translation in two steps: first a phrase-based model is used for each factor (*surface forms, lemmas, POS-tags etc.*) *in independence*, to produce hypothetic parallel streams of factors; these streams are then recombined in a generation step, where factor vectors are projected back to their corresponding surface form. Computing translation independently for the various components makes this approach unable to manipulate *hybrid* phrases such as the one in Fig. 4 (bottom), where the *'sub-phrase' wants|voulons* appears in a generalized form, and which, we have showed, allow to take larger context into account during the translation process. The tuple n-gram of the example has a higher potential of re-usability as it can be used with multiple instantiations of the tuple $VERB|VERB$. Furthermore, under factored bilingual LMs, the computational complexity of the translation process is not modified: decoding is only performed once (rather than once per factor) and no generation step needs to be introduced.

6 Conclusions and further work

We have presented an extension to a bilingual n-gram language model based on FLMs. We have shown that this approach allows to integrate larger bilingual context during the translation process, yielding small, yet consistent gains over a baseline. It thus seems to provide a principled way to alleviate the impact of sparseness problems encountered in conventional phrase-based approaches. Moreover, we introduced factored bilingual

LMs without increasing the decoding complexity. No additional translation/generation step is needed. It is finally worth mentioning that the integration of this novel kind of translation model has required only a minor programming effort: in fact, our phrase-based formalism, which primarily relies on n-gram language model technologies, can readily benefit from all the improvements of such models.

The framework of factored bilingual n-gram models is a general framework where new information sources can be straightforward incorporated to overcome some of the shortcomings of current SMT systems. A first natural extension of this work will be to consider other language pairs, for instance involving two languages with a rich morphology, and to take more linguistic features into account. We also plan to explore the possibilities of incorporating bilingual syntactic information as a factor form, in order to improve long distance reorderings. We are finally considering the possibility of dynamically extending the search space with new tuples for which no probability estimates are available, as is done for instance in Yang and Kirchhoff (2006). This can be achieved in our approach by generalizing not only the contexts in which tuples occur, but also the source side of tuples: this should allow to estimate (conditional) probabilities for an unseen tuple (s, t) in the (generalized) context h based on the available estimates for $P((s', t)|h)$, when s and s' correspond to the same lemma (or part-of-speech).

Acknowledgements The authors would like to thank the reviewers for their detailed comments on earlier versions of this article. This work was partially funded by OSEO under the Quaero program.

References

Bilmes JA, Kirchhoff K (2003) Factored language models and generalized parallel backoff. In: NAACL '03: Proceedings of the 2003 conference of the North American chapter of the association for computational linguistics on human language technology, pp 4–6

Bod R (2000) Combining semantic and syntactic structure for language modeling. In: Proceedings of the 8th international conference on spoken language processing, ICSLP'00, vol III. Beijing, China, pp 106–109

Charniak E (2001) Immediate-head parsing for language models. In: Proceedings of the 39th annual meeting on association for computational linguistics. Toulouse, France, pp 124–131

Chelba C, Jelinek F (2000) Structured language modeling. Comput Speech Lang 14(4):283–332

Chiang D (2005) A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd annual meeting of the association for computational linguistics (ACL'05). Ann Arbor, Michigan, pp 263–270

Crego JM, Mariño JB (2007a) Extending MARIE: an N-gram-based SMT decoder. In: Proceedings of the 45rd annual meeting of the association for computational linguistics (ACL'07). Ann Arbor, Michigan

Crego JM, Mariño JB (2007b) Improving SMT by coupling reordering and decoding. Mach Transl 20(3):199–215

Habash N, Sadat F (2006) Arabic preprocessing schemes for statistical machine translation. In: NAACL '06: proceedings of the human language technology conference of the NAACL, Companion Volume: Short Papers on XX, pp 49–52

Katz SM (1987) Estimation of probabilities from sparse data for the language model component of a speech recognizer. IEEE Trans Acoust Speech Signal Process 35(3):400–401

Kirchhoff K, Bilmes J, Duh K (2008) Factored language models tutorial. Technical report. Deptartment of Electrical Engineering, University of Washington

Kneser R, Ney H (1995) Improved backing-off for m-gram language modeling. In: Proceedings of the international conference on acoustics, speech, and signal processing, ICASSP'95. Detroit, MI, pp 181–184

Koehn P, Hoang H (2007) Factored translation models. In: Proceedings of the 2007 joint conference on empirical methods in natural language processing and computational natural language learning (EMNLP-CoNLL), pp 868–876

Koehn P, Och FJ, Marcu D (2003) Statistical phrase-based translation. In: Proceedings of the human language technology conference of the North American Chapter of the association for computational linguistic. Edmondton, Canada, pp 127–133

Koehn P, Axelrod A, Birch A, Callison-Burch C, Osborne M, Talbot D (2005) Edinburgh system description for the 2005 IWSLT speech translation evaluation. In: Proceedings of the international workshop on spoken language translation, IWSLT'05. Pittsburgh, PA

Koehn P, Hoang H, Birch A, Callison-Burch C, Federico M, Bertoldi N, Cowan B, Shen W, Moran C, Zens R, Dyer C, Bojar O, Constantin A, Herbst E (2007) Moses: open source toolkit for statistical machine translation. In: Proceedings of the annual meeting of the association for computational linguistics (ACL), demonstration session. Prague, Czech Republic

Mariño JB, Banchs RE, Crego JM, de Gispert A, Lambert P, Fonollosa JA, Costa-Jussà MR (2006) N-gram-based machine translation. Comput Linguist 32(4):527–549

Melamed ID (2004) Statistical machine translation by parsing. In: ACL '04: Proceedings of the 42nd annual meeting on association for computational linguistics. Morristown, NJ, USA, p 653

Niesler TR (1997) Category-based statistical language models. Ph.D. thesis, University of Cambridge

Nießen S, Ney H (2001) Toward hierarchical models for statistical machine translation of inflected languages. In: Proceedings of the ACL 2001 workshop on data-driven methods in machine translation. Toulouse, France, pp 47–51

Nießen S, Ney H (2004) Statistical machine translation with scarce resources using morpho-syntactic information. Comput Linguist 30(2):181–204

Och FJ, Ney H (2002) Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings of ACL. Philadelphia, PA, pp 295–302

Och F-J, Tillmann C, Ney H (1999) Improved alignment models for statistical machine translation. In: Proceedings of the joint conference of empirical methods in natural language processing and very large corpora. University of Maryland, College Park, MD, pp 20–28

Papineni K, Roukos S, Ward T, Zhu W-J (2002) Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the annual meeting of the association for computation linguistics. Philadelphia, PA, pp 311–318

Rosenfeld R (1996) A maximum entropy approach to adaptative statistical language modeling. Comput Speech Lang 10:187–228

Talbot D, Osborne M (2006) Modelling Lexical Redundancy for Machine Translation. In: Proceedings of the 21st international conference on computational linguistics and 44th annual meeting of the association for computational linguistics. Sydney, Australia, pp 969–976

Yang M, Kirchhoff K (2006) Phrase-based backoff models for machine translation of highly inflected languages. In: Proceedings of the European Chapter of the ACL. Trento, Italy, pp 41–48

Zens R, Och FJ, Ney H (2002) Phrase-based statistical machine translation. In: Jarke M, Koehler J, Lakemeyer G (eds) KI-2002: advances in artificial intelligence, vol 2479 of LNAI. pp 18–32