From n-gram-based to CRF-based Translation Models

Thomas Lavergne **Josep Maria Crego**
LIMSI/CNRS
BP 133
F-91 403 Orsay Cédex
{lavergne,jmcrego}@limsi.fr

Alexandre Allauzen **François Yvon**
LIMSI/CNRS & Uni. Paris Sud
BP 133
F-91 403 Orsay Cédex
{allauzen,yvon}@limsi.fr

Abstract

A major weakness of extant statistical machine translation (SMT) systems is their lack of a proper training procedure. Phrase extraction and scoring processes rely on a chain of crude heuristics, a situation judged problematic by many. In this paper, we recast the machine translation problem in the familiar terms of a sequence labeling task, thereby enabling the use of enriched feature sets and exact training and inference procedures. The tractability of the whole enterprise is achieved through an efficient implementation of the conditional random fields (CRFs) model using a weighted finite-state transducers library. This approach is experimentally contrasted with several conventional phrase-based systems.

1 Introduction

A weakness of existing phrase-based SMT systems, that has been repeatedly highlighted, is their lack of a proper training procedure. Attempts to design probabilistic models of phrase-to-phrase alignments (e.g. (Marcu and Wong, 2002)) have thus far failed to overcome the related combinatorial problems (DeNero and Klein, 2008) and/or to yield improved training heuristics (DeNero et al., 2006).

Phrase extraction and scoring thus rely on a chain of heuristics see (Koehn et al., 2003), which evolve phrase alignments from "symmetrized" word-to-word alignments obtained with IBM models (Brown et al., 1990) and the like (Liang et al., 2006b; Deng and Byrne, 2006; Ganchev et al., 2008). Phrase scoring is also mostly heuristic and relies on an optimized interpolation of several simple frequency-based scores. Overall, the training procedure of translation models within conventional phrase-based (or hierarchical) systems is generally considered unsatisfactory and the design of better estimation procedures remains an active research area (Wuebker et al., 2010).

To overcome the NP-hard problems that derive from the need to consider all possible permutations of the source sentence, we make here a radical simplification and consider training the translation model given a fixed segmentation and reordering. This idea is not new, and is one of the grounding principle of n-gram-based approaches (Casacuberta and Vidal, 2004; Mariño et al., 2006) in SMT. The novelty here is that we will use this assumption to recast machine translation (MT) in the familiar terms of a sequence labeling task.

This reformulation allows us to make use of the efficient training and inference tools that exists for such tasks, most notably linear CRFs (Lafferty et al., 2001; Sutton and McCallum, 2006). It also enables to easily integrate linguistically informed (describing morphological or morpho-syntactical properties of phrases) and/or contextual features into the translation model. In return, in addition to having a better trained model, we also expect (i) to make estimation less sensible to data sparsity issues and (ii) to improve the ability of our system to make the correct lexical choices based on the neighboring source words. As explained in Section 2, this reformulation borrows much from the general architecture of n-gram MT systems and implies to solve several computational challenges. In our approach, the tractability of the whole enterprise is achieved through an efficient reimplementation of CRFs using a public domain library for weighted finite-state transducers (WFSTs) (see details in Section 3). This approach is experimentally contrasted with more conventional n-gram based and phrase-based approaches on a standard benchmark in Section 4, where we also evaluate the benefits of various feature sets and training regimes. We finally relate our new system with alternative proposals for training discriminatively SMT systems in Section 5, before drawing some lessons and discussing possible extensions of this work.

The main contribution of this work are thus (i) a detailed presentation of the CRF in translation including all necessary implementation details and (ii) an experimental study of various feature functions and of various ways to integrate target side LM information.

2 MT as sequence labeling

In this section, we briefly review the n-gram based approach to SMT, originally introduced in (Casacuberta and Vidal, 2004; Mariño et al., 2006), which constitutes our starting point. We then describe our new proposal, which, in essence, consists in replacing the modeling of compound source-target translation units by a conditional model where the probability of each target side phrase is conditioned on the source sentence.

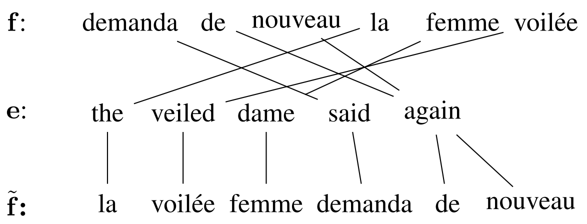
2.1 The n-gram based approach in SMT

The n-gram based approach of (Mariño et al., 2006) is a variation of the standard phrase-based model, characterized by the peculiar form of the translation model. In this approach, the translation model is based on bilingual units called *tuples*. Tuples are the analogous of phrase pairs, as they represent a matching $u = (e, f)$ between a source f and a target e word sequence. The probability of a sequence of tuples is computed using a conventional n-gram model as:

$$p(u_1 \dots u_I) = \prod_{i=1}^{I} p(u_l | u_{i-1} \dots u_{i-n+1}).$$

The probability of a sentence pair $(\mathbf{f}, \mathbf{e})$ is then either recovered by marginalization, or approximated by maximization, over all possible joint segmentations of $\mathbf{f}$ and $\mathbf{e}$ into tuples.

As for any n-gram model, the parameters are estimated using statistics collected in a training corpus made of *sequences of tuples* derived from the parallel sentences in a two step process. First, a word alignment is computed using a standard alignment pipeline[1] based on the IBM models. Source words are then reordered so as to disentangle the alignment links and to synchronize the source and target texts. Special care has to be paid to non-aligned source words, which have to be collapsed with their neighbor words. A byproduct of this process is a *deterministic joint segmentation* of parallel sentences into minimal bilingual units, the tuples, that constitute the basic elements in the model. This process is illustrated on Figure 1, where the unfolding process enables the extraction of tuples such as: (*demanda*, *said*) or (*de nouveau*, *again*).



f:
demanda de nouveau la femme voilée
e:
the veiled dame said again
$\tilde{\mathbf{f}}$:
la voilée femme demanda de nouveau

Figure 1: The tuple extraction process
The original (top) and reordered (bottom) French sentence aligned with its translation.

At test time, the source text is reordered so as to match the reordering implied by the disentanglement procedure. Various proposals has been made to perform such source side reordering (Collins et al., 2005; Xia and McCord, 2004), or even learning reordering rules based on syntactic or morpho-syntactic information (Crego and Mariño, 2007). The latter approach amounts to accumulate reordering patterns during the training; test source sentences are then non-deterministically reordered *in all possible ways* yielding a word graph. This graph is then monotonously decoded, where the score of a translation hypothesis combines information from the translation models as well as from other information sources (lexicalized reordering model, target

[1]Here, using the MGIZA++ package (Gao and Vogel, 2008).

side language model (LM), word and phrase penalties, etc).

2.2 Translating with CRFs

A discriminative version of the n-gram approach consists in modeling $P(\mathbf{e}|\mathbf{f})$ instead of $P(\mathbf{e},\mathbf{f})$, which can be efficiently performed with CRFs (Lafferty et al., 2001; Sutton and McCallum, 2006). Assuming matched sequences of observations ($\mathbf{x} = x_1^L$) and labels ($\mathbf{y} = y_1^L$), CRFs express the conditional probability of labels as:

$$P(y_1^L|x_1^L) = \frac{1}{Z(x_1^L;\theta)} \exp(\theta^T G(x_1^L, y_1^L)),$$

where θ is a parameter vector and G denotes a vector of *feature functions* testing various properties of $\mathbf{x}$ and $\mathbf{y}$. In the *linear-chain* CRF, each component $G_k(x_1^I, y_1^I)$ of G is decomposed as a sum of local features: $G_k(x_1^I, y_1^I) = \sum_i g_k(x_1^I, y_{i-1}, y_i)$[2]. CRFs are trained by maximizing the (penalized) log-likelihood of a corpus containing observations and their labels.

In principle, the data used to train n-gram translation models provide all the necessary information required to train a CRF[3]. It suffices to consider that the alphabet of possible observations ranges over all possible source side fragments, and that each target side of a tuple is a potential label. The model thus defines the probability of a segmented target $\widetilde{\mathbf{e}} = \widetilde{e}_1^I$ given the segmented and reordered source sentence $\tilde{\mathbf{f}} = \widetilde{f}_1^I$. To complete the model, one just needs to define a distribution over source segmentations $P(\tilde{\mathbf{f}}|\mathbf{f})$. Given the deterministic relationship between $\mathbf{e}$ and $\widetilde{\mathbf{e}}$ expressed by the "unsegmentation" function ϕ which maps $\widetilde{\mathbf{e}}$ with $\mathbf{e} = \phi(\widetilde{\mathbf{e}})$, we then have:

$$\begin{aligned} P(\mathbf{e}|\mathbf{f}) &= \sum_{\tilde{\mathbf{f}},\widetilde{\mathbf{e}}|\phi(\widetilde{\mathbf{e}})=\mathbf{e}} P(\widetilde{\mathbf{e}},\tilde{\mathbf{f}}|\mathbf{f}) \\ &= \sum_{\tilde{\mathbf{f}},\widetilde{\mathbf{e}}|\phi(\widetilde{\mathbf{e}})=\mathbf{e}} P(\widetilde{\mathbf{e}},|\tilde{\mathbf{f}},\mathbf{f})P(\tilde{\mathbf{f}}|\mathbf{f}) \\ &= \sum_{\tilde{\mathbf{f}},\widetilde{\mathbf{e}}|\phi(\widetilde{\mathbf{e}})=\mathbf{e}} P(\widetilde{\mathbf{e}},|\tilde{\mathbf{f}})P(\tilde{\mathbf{f}}|\mathbf{f}) \end{aligned}$$

[2]Assuming first order dependencies.

[3]This is a significant difference with (Blunsom et al., 2008), as we do not need to introduce latent variables during training.

In practice, we will only consider a restricted number of possible segmentation/reorderings of the source, denoted $\mathcal{L}(\mathbf{f})$, and compute the best translation $\mathbf{e}^*$ as $\phi(\widetilde{\mathbf{e}}^*)$, where:

$$\begin{aligned} \widetilde{\mathbf{e}}^* &= \arg\max_{\widetilde{\mathbf{e}}} P(\widetilde{\mathbf{e}}|\mathbf{f}) \\ &\approx \arg\max_{\tilde{\mathbf{f}}\in\mathcal{L}(\mathbf{f}),\widetilde{\mathbf{e}}} P(\widetilde{\mathbf{e}},|\tilde{\mathbf{f}},\mathbf{f})P(\tilde{\mathbf{f}}|\mathbf{f}) \end{aligned} \quad (1)$$

Even with these simplifying assumptions, this approach raises several challenging computational problems. First, training a CRF is quadratic in the number of labels, of which we will have plenty (typically hundreds of thousands). A second issue is decoding: as we need to consider at test time a combinatorial number of possible source reorderings and segmentations, we can no longer dispense with the computation of the normalizer $Z(\tilde{\mathbf{f}};\theta)$ which is required to compute $P(\widetilde{\mathbf{e}},\tilde{\mathbf{f}}|\mathbf{f})$ as $P(\tilde{\mathbf{f}}|\mathbf{f})P(\widetilde{\mathbf{e}}|\tilde{\mathbf{f}})$ and to compare hypotheses associated with different values of $\tilde{\mathbf{f}}$. We discuss our solutions to these problems in the next section.

3 Implementation issues

3.1 Training

Basic training The main difficulties in training are caused by the unusually large number of labels, each of which corresponds to a (small) sequence of target words. Hopefully, each observation (source side tuple) occurs with a very small number of different labels. A first simplification is thus to consider that the set of possible "labels" $\widetilde{e}$ for a source sequence $\widetilde{f}$ is limited to those that are seen in training: all the other associations $(\widetilde{f},\widetilde{e})$ are deemed impossible, which amounts to setting the corresponding parameter value to $-\infty$.

A second speed-up is to enforce sparsity in the model, through the use of a ℓ_1 regularization term (Tibshirani, 1996): on the one hand, this greatly reduces the memory usage; furthermore, sparse models are also prone to various optimization of the forward-backward computations (Lavergne et al., 2010). As discussed in (Ng, 2004; Turian et al., 2007), this feature selection strategy is well suited to the task at hand, where the number of possible features is extremely large. Optimization is performed using the Rprop algorithm[4] (Riedmiller and Braun, 1993), which provides the memory efficiency needed to cope with the very large feature sets considered here.

Training with a target language model One of the main strength of the phrase-based "log-linear" models is their ability to make use of powerful target side language models trained on very large amounts of monolingual texts. This ability is crucial to achieve good performance and has to be preserved no matter the difficulties that occur when one moves away from conventional phrase-based systems (Chiang, 2005; Huang and Chiang, 2007; Blunsom and Osborne, 2008; Kääriäinen, 2009). It thus seems appropriate to include a LM feature function in our model or alternatively to define:

$$P(\widetilde{\mathbf{e}}|\tilde{\mathbf{f}}) = \frac{1}{Z(\tilde{\mathbf{f}};\theta)} P_{LM}(\widetilde{\mathbf{e}}) \exp(\theta^T G(\tilde{\mathbf{f}}, \widetilde{\mathbf{e}})),$$

where P_{LM} is the target language model and $Z(\tilde{\mathbf{f}};\theta) = \sum_{\widetilde{\mathbf{e}}} P_{LM}(\widetilde{\mathbf{e}}) \exp(\theta^T G(\tilde{\mathbf{f}}, \widetilde{\mathbf{e}}))$. Implementing this approach implies to deal with the lack of synchronization between the units of the translation models, which are variable-length (possibly empty) tuples, and the units of the language models, which are plain words.

In practice, this extension is implemented by performing training and inference over a graph whose nodes are not only indexed by their position and the left target context, but also by the required n-gram (target) history. In most cases, for small values of n such as considered in this study, the n-gram history can be deduced from the left target tuple. The most problematic case is when the left target tuple is NULL, which require to copy the history from the previous states. As a consequence, for the values of n considered here, the impact of this extension on the total training time is limited.

Reference reachability A recurring problem for discriminative training approaches is *reference unreachability* (Liang et al., 2006a): this happens when the model cannot predict the reference translation, which means in our case that the probability of the reference cannot be computed. In our implementation, this only happens when the reference involves a tuple $(\widetilde{f},\widetilde{e})$ that is too rare to be included in the model. As a practical workaround, when this happens for a given training sentence, we make sure to "locally" augment the tuple dictionary with the missing part of the reference, which is then removed for processing the rest of the training corpus.

[4]Adapted to handle a locally non-differentiable objective.

3.2 Inference

Our decoder is implemented as a cascade of weighted finite-state transducers (WFSTs) using the functionalities of the OpenFst library (Allauzen et al., 2007). This library provides many basic operation for WFSTs, notably the left (π_1) and right (π_2) projections as well as the composition operation ($\circ$). The related notions and algorithms are presented in detail in (Mohri, 2009), to which we refer the reader.

In essence, our decoder is implemented of a finite-state cascade involoving the following steps: (i) source reordering and segmentation (ii) application of the translation model and (optionally) (iii) composition with a target side language model, an architecture that is closely related to the proposal of (Kumar et al., 2006). A more precise account of these various steps is given below, where we describe the main finite-state transducers involved in our decoder:

- S, the acceptor for the source sentence $\mathbf{f}$;
- R, which implements segmentation and reordering rules;
- T, the tuple dictionary, associating source side sequences with possible translations based on the inventory of tuples;
- F, the feature matcher, mapping each feature with the corresponding parameter value;

Source reordering The computation of R mainly follows the approach of (Crego and Mariño, 2007) and uses a part-of-speech tagged version of the reordered training data. Each reordering pattern seen in training is generalized as a non-deterministic reordering rule which expresses a possible rearrangement of some subpart of the source sentence. Each rule is implemented as an elementary finite-state transducer, and the set of possible word reorderings is computed as the composition of these transducers. R is finally obtained by composing the result with a transducer computing all the possible segmentations of its input into sequences of source side tuples[5].

The output of $S \circ R$ are sequences of source side tuples $\tilde{\mathbf{f}}$; each path in this transducer is additionally weighted with a simplistic n-tuple segmentation model, estimated using the source side of the parallel training corpus. Note that these scores are normalized, so that the weight of each path labelled $\tilde{\mathbf{f}}$ in $S \circ R$ is $\log P(\tilde{\mathbf{f}}|f)$.

The feature matcher F The feature matcher is also implemented as a series of elementary weighted transducers, each transducer being responsible for a given *class of feature functions*. The simplest transducer in this family deals with the class of *unigram feature functions*, ie. feature functions that only test the current observation and label. It is represented on the left part of Figure 3.2, where for the sake of readability we only display one example for each test pattern (here: an unconditional feature that always returns true for a given label, a test on the source word, and a test on the source POS label). As long as dependencies between source and/or target symbols remain local, they can be captured by finite-state transducers such as the ones on the mid and right part of Figure 3.2, which respectively compute *bigram* target features, and joint bigram source and target features.

The feature matcher F is computed as the composition of these elementary transducers, where we only include source and target labels that can occur given the current input sentence. Weights in F are interpreted in the tropical semiring. $\exp(F)$ is obtained by replacing weights w in F with $\exp(w)$ in the real semiring.

Decoding a word graph If the input segmentation and reordering were deterministically set, meaning that the automaton $I = \pi_1(S \circ R \circ T)$ would only contain one path, decoding would amount to finding the best path in $S \circ R \circ T \circ F$. However, we need to compute:

$$\begin{aligned}\arg\max_{\tilde{\mathbf{e}}} P(\tilde{\mathbf{e}}|\mathbf{f}) &= \arg\max_{\tilde{\mathbf{e}}} \sum_{\tilde{\mathbf{f}}} P(\tilde{\mathbf{e}}, \tilde{\mathbf{f}}|\mathbf{f}) \\ &= \arg\max_{\tilde{\mathbf{e}}} \sum_{\tilde{\mathbf{f}}} P(\tilde{\mathbf{e}}|\tilde{\mathbf{f}}) P(\tilde{\mathbf{f}}|\mathbf{f}).\end{aligned}$$

This requires to compare model scores for multiple source segmentations and reorderings $\tilde{\mathbf{f}}$, hence to compute $P(\tilde{\mathbf{f}}|\mathbf{f})$ and $P(\tilde{\mathbf{e}}|\tilde{\mathbf{f}})$, rather than just the non-normalized value that is usually used in CRFs.

Computing the normalizer $Z(\tilde{\mathbf{f}}; \theta)$ for all sequences in $S \circ R$ is performed efficiently using standard finite-state operations as :

$$D = \det(\pi_1(\pi_2(S \circ R) \circ T \circ \exp(F))).$$

In fact, determinization (in the real semiring) has the effect of accumulating for each $\tilde{\mathbf{f}}$ the corresponding normalizer $Z(\tilde{\mathbf{f}}; \theta)$. Replacing each weight w in D by $-\log(w)$ and using the $\log$ semiring enables to compute $-\log(Z(\tilde{\mathbf{f}}; \theta))$. The best translation is then obtained as: $\mathrm{bestpath}(\pi_2(S \circ R) \circ -\log(D) \circ T \circ F)$ in the tropical semiring.

Decoding and Rescoring with a target language model An alternative manner of using a (large) target side language model is to use it for rescoring purposes. The consistent use of finite-state machines and operations makes it fairly easy to include one during decoding : it suffices to perform the search in $\pi_2(S \circ R) \circ -\log(D) \circ T \circ F \circ L$, where L represents a n-gram language model. When combining several models, notably a source segmentation model and/or a target language model for rescoring, we have made sure to rescale the (log)probabilities so as to balance the language model scores with the CRF scores, and to use a fixed word bonus to make hypotheses of different length more comparable. All these parameters are tuned as part of the decoder development process. It is finally noteworthy that, in our architecture, alternative decoding strategies, such as MBR (Kumar and Byrne, 2004) are also readily implemented.

4 Experiments

4.1 Corpora and metrics

For these experiments, we have used a medium size training corpus, extracted from the datasets made available for WMT 2011[6] evaluation campaign, and have focused on one translation direction, from French to English[7].

Translation model training uses the entire *News-Commentary* subpart of the WMT'2011 training

[5]When none is found, we also consider a maximal segmentation into isolated words.

[6]`statmt.org/wmt11`

[7]Results in the other direction suggest similar conclusions.

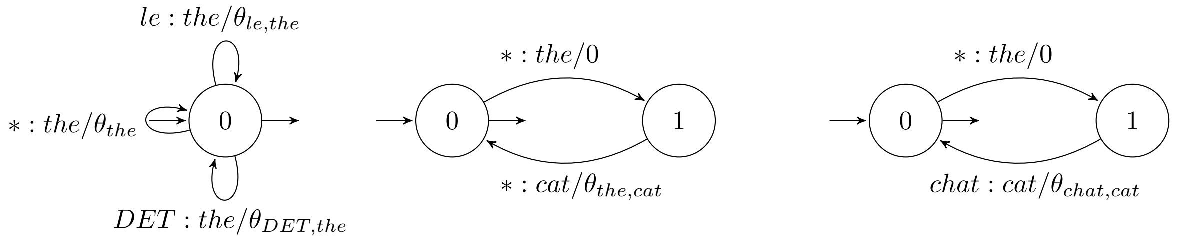

$le : the/\theta_{le,the}$
$* : the/\theta_{the}$
0
$DET : the/\theta_{DET,the}$
$* : the/0$
0
1
$* : cat/\theta_{the,cat}$
$* : the/0$
0
1
$chat : cat/\theta_{chat,cat}$

Figure 2: Feature matchers. The star symbol (*) matches any possible observation.

		French		English	
	sent˙	token	types	token	types
train	115 K	3 339 K	60 K	2 816 K	58 K
test 2008	2.0 K	55 K	9 K	49 K	8 K
test 2009	2.5 K	72 K	11 K	65 K	10 K
test 2010	2.5 K	69 K	10 K	61 K	9 K

Table 1: Corpora used for the experiments

data; for language models, we have considered two approaches (i) a "large" bigram model highly optimized using all the available monolingual data and (ii) a "small" trigram language model trained on just the English side of the NewsCommentary corpus. The regularization parameters used in training are tuned using the WMT 2009 test set; the various parameters implied in the decoding are tuned (for BLEU) on WMT 2008 test set; the internal tests reported below are performed on the 2010 test lines (see Table 1) using the best parameters found during tuning. Various statistics regarding these corpora are reproduced on Table 1.

All the training corpora were aligned using MGIZA++ with standard parameters[8], and processed in the standard tuple extraction pipeline. The development and test corpora were also processed analogously. For the sake of comparison, we also trained a standard n-gram-based and a Moses system (Koehn et al., 2007) with default parameters and a 3-gram target LM trained using only the target side of our parallel corpus. The development set (test 2009) was used to tune these two systems. All performance are measured using BLEU (Papineni et al., 2002).

[8]As part of a much larger batch of texts.

4.2 Features

The baseline system is composed only of *translation features* [**trs**] and *target bigram features* [**t2g**]. The former correspond to functions of the form $\mathrm{gu}_{s,t}(\tilde{\mathbf{f}}, \tilde{\mathbf{e}}, i) = \mathbb{I}(\tilde{f}_i = s \wedge \tilde{e}_i = t)$, where s and t respectively denote source and target phrases and $\mathbb{I}()$ is the indicator function. These are also generalized to part-of-speech and also to any possible source phrase, giving rise to features such as $\mathrm{gu}_{*,t} = (\tilde{\mathbf{f}}, \tilde{\mathbf{e}}, i) = \mathbb{I}(\tilde{e}_i = t)$. Target bigram features correspond to functions of the form $\mathrm{gb}_{t,t'}(\tilde{\mathbf{f}}, \tilde{\mathbf{e}}, i) = \mathbb{I}(\tilde{e}_{i-1} = t \wedge \tilde{e}_i = t')$. The last baseline feature is the *copy* feature, which fires whenever the source and target segments are identical.

Supplementary groups of features are considered in further stages:

- suffix/prefix features [**ix**]. These features allow to generalize baseline features on the source side to fixed length prefixes and suffixes, thus smoothing the parameters.
- context features [**ctx**]. These features are similar to unigram features, but also test the left *source* tuple and the corresponding part-of-speech.
- segmentation features [**seg**]. These features are meant to express a preference for longer tuples and to regulate the number of target words per source word. We consider the following feature functions ($|e|$ denotes the length of e):
  - target length features :
    $\mathrm{gl}_{*,l}(\tilde{\mathbf{f}}, \tilde{\mathbf{e}}, i) = \mathbb{I}(|\tilde{e}_i| = l)$
  - source-target length features :
    $\mathrm{gl}_{l,l'}(\tilde{\mathbf{f}}, \tilde{\mathbf{e}}, i) = \mathbb{I}(|\tilde{f}_i| = l \wedge |\tilde{e}_i| = l')$
  - source-target length ratio :
    $\mathrm{gl}_{l}(\tilde{\mathbf{f}}, \tilde{\mathbf{e}}, i) = \mathbb{I}(\mathrm{round}(\frac{|\tilde{f}_i|}{|\tilde{e}_i|}) = l)$

Note that all these features are further conditioned on the *target* label.

- reordering features [**ord**]. These features are meant to model preferences for specific local reordering patterns and take into account neighbor source fragments in $\widetilde{\mathbf{e}}$ together with the current label. Each source side segment $\widetilde{f_i}$ is made of some source words that, prior to source reordering, were located at indices $i_1 \dots i_l$, so that $\widetilde{f_i} = f_{i_1} \dots f_{i_l}$. The highest (resp. lowest) index in this sequence is $\lceil \widetilde{f_i} \rceil$ (resp. $\lfloor \widetilde{f_i} \rfloor$). The leftmost (resp. rightmost) index is $[\widetilde{f_i}[$ (resp. $]\widetilde{f_i}]$).

  Using these notations, our model includes the following patterns:

  – distortion features, measuring the gaps between consecutive source fragments : $\mathrm{go}_{l,t}(\widetilde{\mathbf{f}}, \widetilde{\mathbf{e}}, i) = \mathbb{I}(\Delta(\widetilde{f_i}, \widetilde{e_i}) = l \wedge \widetilde{e_i} = t)$, where $\Delta(\widetilde{f_i}, \widetilde{e_i}) =$
  $$\begin{cases} \lfloor \widetilde{f_i} \rfloor - \lceil \widetilde{f_{i-1}} \rceil \text{ if } (\lceil \widetilde{f_{i-1}} \rceil \leq \lfloor \widetilde{f_i} \rfloor) \\ \lceil \widetilde{f_i} \rceil - \lfloor \widetilde{f_{i-1}} \rfloor \text{ otherwise .} \end{cases}$$

  – lexicalized reordering, identifying monotone, swap and discontinuous configurations (Tillman, 2004). The monotonous test is defined as: $\mathrm{go}_m(\widetilde{\mathbf{f}}, \widetilde{\mathbf{e}}, i) = \mathbb{I}(]e_{i-1}] = [e_i[)$; the swap and discontinuous configurations are defined analogously.

  – ”gappiness” test : this feature is activated whenever the source indices $i_1 ... i_l$ contain one or several gaps.

4.3 Experiments and lessons learned

Training time The first lesson learned is that training can be performed efficiently. Our baseline system, which only contains **trs** and **trg** contains approximately 87 million features, out of which a little bit more than 600K are selected. Adding up all supplementary features raises the number of parameters to about 130M features, out of which 1.5M are found useful. All these systems require between 3 and 5 hours to train[9]. These numbers are obtained with a ℓ^1 penalty term ≈ 1, which offers a good balance between accuracy and sparsity.

[9]All experiments run on a server with 64G of memory and two Xeon processors with 4 cores at 2.27 Ghz.

Test conditions In order to better assess the strengths and weaknesses of our approach, we compare several test settings: the most favorable considers only one possible segmentation/reordering $\widetilde{\mathbf{f}}$ for each $\mathbf{f}$, obtained through forced alignment with the reference; we then consider the more challenging case where the reordering is fixed, but several segmentations are considered; then the regular decoding task, where both segmentation and reordering are unknown and where the entire space of all segmentations and reordering is searched. For each condition, we also vary (i) the set of features used and (ii) the target language model used, if any. Wherever applicable, we also report contrasts with n-gram-based systems subject to the same input and comparable resources, varying the order of the tuple language model, as well as with Moses. Results are in Table 2.

	dev	test	# feat.
decoding with optimal segmentation/reordering			
CRF (**trs,trg**)	23.8	25.1	660K
CRF **+ctx**	24.1	25.4	1.5M
CRF **+ix,ord,seg**	24.3	25.6	1.5M
decoding with optimal reordering			
n-gram (2g,3g)	20.6	24.1	755K
n-gram (3g,3g)	21.5	25.2	755K
CRF **trs,trg**	-	22.8	660K
CRF **+ctx**	-	23.1	1.5M
CRF **+ix,ord,seg**	-	23.5	1.5M
regular decoding			
Moses (3g)	21.2	20.5	
n-gram (2g,3g)	20.6	20.2	755K
n-gram (3g,3g)	21.5	21.2	755K
CRF (**trs,trg**)	-	18.3	660K
CRF **+ctx**	-	18.8	1.5M
CRF **+ix,ord,seg**	-	19.1	1.5M
CRF **+ix,ord,seg**+3g	-	19.1	1.5M

Table 2: Translation performance

Extending the feature set As expected, the use of increasingly complex feature sets seems beneficial in all experimented conditions. It is noteworthy that throwing in reordering and contextual features is helping, *even when decoding one single segmentation and reordering*. This is because these features do not help to select the best input reordering, but help choose the best target phrase.

Searching a larger space Going from the simpler to the more difficult conditions yields significant degradations in the model, as our best score drops down from 25.6 to 23.5 (with known reordering) then to 19.1 (regular decoding). This is a clear indication that our current segmentation/reordering model is not delivering very useful scores. A similar loss is incurred by the n-gram system, which loses 4 bleu points between the two conditions.

LM rescoring Our results to date with target side language models have proven inconclusive, which might explain why our best results remain between one and two BLEU points behind the n-gram based system using comparable information. Note also that preliminary experiments with incorporating a large bigram during training have also failed to date to provide us with improvements over the baseline.

Summary In sum, the results accumulated during this first round of experiments tend to show that our CRF model is still underperforming the more established baseline by approximately 1 to 1.5 BLEU point, when provided with comparable resources. Sources of improvements that have been clearly identified is the scoring of reordering and segmentations, and the use of a target language model in training and/or decoding.

5 Related work

Discriminative learning approaches have proven successful for many NLP tasks, notably thanks to their ability to cope with flexible linguistic representations and to accommodate potentially redundant descriptions. This is especially appealing for machine translation, where the mapping between a source word or phrase and its target correlate(s) seems to involve an large array of factors, such as its morphology, its syntactic role, its meaning, its lexical context, etc. (see eg. (Och et al., 2004; Gimpel and Smith, 2008; Chiang et al., 2009), for inspiration regarding potentially useful features in SMT).

Discriminative learning requires (i) a parameterized scoring function and (ii) a training objective. The scoring function is usually assumed to be linear and ranks candidate outputs y for input x according to $\theta^T G(x,y)$, where θ is the parameter vector. θ and G deterministically imply the input/output mapping as $x \rightarrow \arg\max_y \theta^T G(x,y)$. Given a set of training pairs $\{x^i, y^i, i = 1 \dots N\}$, parameters are learned by optimizing some regularized loss function of θ, so as to make the inferred input/output mapping faithfully replicate the observed instances.

Machine translation, like most NLP tasks, does not easily lend itself to that approach, due to the complexity of the input/output objects (word or label strings, parse trees, dependency structures, etc). This complexity makes inference and learning intractable, as both steps imply the resolution of the $\arg\max$ problem over a combinatorially large space of candidates y. Structured learning techniques (Bakir et al., 2007), developed over the last decade, rely on decompositions of these objects into sub-parts as part of a *derivation* process, and use conditional independence assumptions between sub-parts to render the learning and inference problem tractable. For machine translation, this only provides part of the solution, as the training data only contain pairs of word aligned sentences $(\mathbf{f}, \mathbf{e})$, but lack the explicit derivation $\mathbf{h}$ from $\mathbf{f}$ to $\mathbf{e}$ that is required to train the model in a fully supervised way.

The approach of (Liang et al., 2006a) circumvents the issue by assuming that the hidden derivation $\mathbf{h}$ can be approximated through forced decoding. Assuming that $\mathbf{h}$ is in fact observed as the optimal (Viterbi) derivation $\mathbf{h}^*$ from $\mathbf{f}$ to $\mathbf{e}$ given the current parameter value[10], it is straightforward to recast the training of a phrase-based system as a standard structured learning problem, thus amenable to training algorithms such as the averaged perceptron of (Collins, 2002). This approximation is however not genuine, and the choice of the most appropriate derivation seems to raises intriguing issues (Watanabe et al., 2007; Chiang et al., 2008).

The authors of (Blunsom et al., 2008; Blunsom and Osborne, 2008) consider models for which it is computationally possible to marginalize out all possible derivations of a given translation. As demonstrated in these papers, this approach is tractable even when the derivation process is a based on synchronous context-free grammars, rather that finite-state devices. However, the computational cost as-

[10]If one actually exists in the model, thus raising the issue of *reference reachability*, see discussion in Section 3.

sociated with training and inference remains very high, especially when using a target side language model, which seems to preclude the application to large-scale translation tasks[11]. The recent work of (Dyer and Resnik, 2010) proceeds from a similar vein: translation is however modeled as a two step process, where a set of possible source reorderings, represented as a parse forest, are associated with possible target sentences, using, as we do, a finite-state translation model. This translation model is trained discriminatively by marginalizing out the (unobserved) reordering variables; inference can be performed effectively by intersecting the input parse forest with a transducer representing translation options.

A third strategy is to consider a simpler class of derivation process, which only *partly* describe the mapping between $\mathbf{f}$ and $\mathbf{e}$. This is, for instance, the approach of (Bangalore et al., 2007), where a simple *bag-of-word* representation of the target sentence is computed using a battery of boolean classifiers (one for each target word). In this approach, discriminative training is readily applicable, as the required supervision is overtly present in example source-target pairs $(\mathbf{f}, \mathbf{e})$; however, a complementary reshaping/reordering step is necessary to turn the bag-of-word into a full-fledged translation. This work was recently revisited in (Mauser et al., 2009), where a conditional model predicting the presence of each target phrase provides a supplementary score for the standard "log-linear" model.

This line of research has been continued notably in (Kääriäinen, 2009), which introduces an exponential model of *bag of phrases* (allowing some overlap), that enables to capture localized dependencies between target words, while preserving (to some extend) the efficiency of training and inference. Supervision is here indirectly provided by word alignment and correlated phrase extraction processes implemented in conventional phrase-based systems (Koehn et al., 2003). If this model seems to deliver state-of-the-art performance on large-scale tasks, it does so at a very high computational cost. Moreover, for lack of an internal modeling of reordering processes, this approach, like the bag-of-word approach, seems only appropriate for language pairs with similar or related word ordering.

The approach developed in this paper fills a gap between the hierarchical model of (Blunsom et al., 2008) and the phrase-based model (Kääriäinen, 2009), with whom we share several important assumptions, such as the use of alignment information to provide supervision, and the resort to a an "external", albeit a more powerful, reordering component. Using a finite-state model enables to process reasonably large corpora, and gives some hopes as to the scalability of the whole enterprise; it also makes the integration of a target side language model much easier than in hierarchical models.

6 Discussion and future work

In this paper, we have given detailed description of an original phrase-based system implementing a discriminative version of the n-gram model, where the translation model probabilities are computed with conditional random fields. We have showed how to implement this approach using a memory efficient implementation of the optimization algorithms needed for training: in our approach, training a mid-scale translation system with hundred of thousands sentence pairs and millions of features only takes a couple of hours on a standalone desktop machine. Using ℓ_1 regularization has enabled to assess the usefulness of various families of features.

We have also detailed a complete decoder implemented as a pipeline of finite-state transducers, which allows to efficiently combine several models, to produce n-best lists and word lattices.

The results obtained in a series of preliminary experiments show that our system is already delivering competitive translations, as acknowledged by a comparison with two strong phrase-based baselines. We have already started to implement various optimizations and to experiment with somewhat larger datasets (up to 500K sentence pairs) and larger feature sets, notably incorporating word sense disambiguation features: this work needs to be continued. In addition, we intend to explore a number of extensions of this architecture, such as implementing MBR decoding (Kumar and Byrne, 2004) or adapting the translation model to new domains and conditions, using, for instance, the proposal of

[11]For instance, the experiments reported in (Blunsom and Osborne, 2008) use the English-Chinese BTEC, where the average sentence length is lesser than 10.

(Daume III, 2007)[12].

One positive side effect of experimenting with new translation models is that they help reevaluate the performance of the whole translation system pipeline: in particular, discriminative training seems to be more sensible to alignments errors than the corresponding n-gram system, which suggests to pay more attention to possible errors in the training data; we have also seen that the current reordering model defines a too narrow search space and delivers insufficiently discriminant scores: we will investigate various ways to further improve the computation and scoring of hypothetical source reorderings.

Acknowledgements

The authors wish to thank the reviewers for comments and suggestions. This work was achieved as part of the *Quaero* Programme, funded by OSEO, French State agency for innovation.

References

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A general and efficient weighted finite-state transducer library. In *Proceedings of the Ninth International Conference on Implementation and Application of Automata, (CIAA 2007)*, volume 4783 of *Lecture Notes in Computer Science*, pages 11–23. Springer. http://www.openfst.org.

Gökhan Bakir, Thomas Hofmann, Bernhard Schölkopf, Alexander J.Smola, Ben Taskar, and S.V.N. Vishwanathan. 2007. *Predicting structured output*. MIT Press.

Srinivas Bangalore, Patrick Haffner, and Stephan Kanthak. 2007. Statistical machine translation through global lexical selection and sentence reconstruction. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 152–159, Prague, Czech Republic.

Phil Blunsom and Miles Osborne. 2008. Probabilistic inference for machine translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 215–223, Honolulu, Hawaii.

Phil Blunsom, Trevor Cohn, and Miles Osborne. 2008. A discriminative latent variable model for statistical machine translation. In *Proceedings of ACL-08: HLT*, pages 200–208, Columbus, Ohio.

Peter F. Brown, John Cocke, Stephen Della Pietra, Vincent J. Della Pietra, Frederick Jelinek, John D. Lafferty, Robert L. Mercer, and Paul S. Roossin. 1990. A statistical approach to machine translation. *Computational Linguistics*, 16(2):79–85.

Francesco Casacuberta and Enrique Vidal. 2004. Machine translation with inferred stochastic finite-state transducers. *Computational Linguistics*, 30(3):205–225.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 224–233, Honolulu, Hawaii.

D. Chiang, K. Knight, and W. Wang. 2009. 11,001 new features for statistical machine translation. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 218–226. Association for Computational Linguistics.

David Chiang. 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)*, pages 263–270, Ann Arbor, Michigan.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan.

Michael Collins. 2002. Discriminative training methods for hidden markov models: Theory and experiments with perceptron algorithms. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 1–8. Association for Computational Linguistics, July.

Josep M. Crego and José B. Mariño. 2007. Improving SMT by coupling reordering and decoding. *Machine Translation*, 20(3):199–215.

Hal Daume III. 2007. Frustratingly easy domain adaptation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 256–263, Prague, Czech Republic. Association for Computational Linguistics.

John DeNero and Dan Klein. 2008. The complexity of phrase alignment problems. In *Proceedings of ACL-08: HLT, Short Papers*, pages 25–28, Columbus, Ohio.

John DeNero, Dan Gillick, James Zhang, and Dan Klein. 2006. Why generative phrase models underperform

[12]In a nutshell, this proposal amounts to having three different parameters for each feature; one parameter is trained as usual; the other two parameters are updated conditionally, depending whether the training instance comes from the in-domain or from the out-domain training dataset.

surface heuristics. In *Proceedings of the ACL workshop on Statistical Machine Translation*, pages 31–38, New York City, NY.

Yonggang Deng and William Byrne. 2006. MTTK: An alignment toolkit for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Demonstrations*, pages 265–268, New York City, USA.

Chris Dyer and Philip Resnik. 2010. Context-free reordering, finite-state translation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 858–866, Los Angeles, California. Association for Computational Linguistics.

Kuzman Ganchev, João V. Graça, and Ben Taskar. 2008. Better alignments = better translations ? In *Proceedings of ACL-08: HLT*, pages 986–993, Columbus, Ohio.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *SETQA-NLP '08*.

Kevin Gimpel and Noah A. Smith. 2008. Rich source-side context for statistical machine translation. In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 9–17, Columbus, Ohio, June.

Liang Huang and David Chiang. 2007. Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic.

Matti Kääriäinen. 2009. Sinuhe – statistical machine translation using a globally trained conditional exponential family translation model. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1027–1036, Singapore.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistic*, pages 127–133, Edmondton, Canada.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proc. Annual Meeting of the Association for Computational Linguistics (ACL), demonstration session*, pages 177–180, Prague, Czech Republic.

Shankar Kumar and William Byrne. 2004. Minimum bayes-risk decoding for statistical machine translation. In Daniel Marcu Susan Dumais and Salim Roukos, editors, *HLT-NAACL 2004: Main Proceedings*, pages 169–176, Boston, Massachusetts, USA. Association for Computational Linguistics.

Shankar Kumar, Yonggang Deng, and William Byrne. 2006. A weighted finite state transducer translation template model for statistical machine translation. *Natural Language Engineering*, 12(1):35–75.

John Lafferty, Andrew McCallum, and Fernando Pereira. 2001. Conditional random fields: probabilistic models for segmenting and labeling sequence data. In *Proceedings of the International Conference on Machine Learning*, pages 282–289. Morgan Kaufmann, San Francisco, CA.

Thomas Lavergne, Olivier Capp, and Franois Yvon. 2010. Practical very large scale crfs. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 504–513, Uppsala, Sweden.

Percy Liang, Alexandre Bouchard-Côté, Dan Klein, and Ben Taskar. 2006a. An end-to-end discriminative approach to machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 761–768, Sydney, Australia.

Percy Liang, Ben Taskar, and Dan Klein. 2006b. Alignment by agreement. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 104–111, New York City, USA.

Daniel Marcu and Daniel Wong. 2002. A phrase-based, joint probability model for statistical machine translation. In *Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing*, pages 133–139.

José B. Mariño, Rafael E. Banchs, Josep M. Crego, Adrià de Gispert, Patrick Lambert, José A.R. Fonollosa, and Marta R. Costa-Jussà. 2006. N-gram-based machine translation. *Computational Linguistics*, 32(4):527–549.

Arne Mauser, Saša Hasan, and Hermann Ney. 2009. Extending statistical machine translation with discriminative and trigger-based lexicon models. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 210–218, Singapore.

Mehryar Mohri. 2009. Weighted automata algorithms. In Manfred Droste, Werner Kuich, and Heiko Vogler, editors, *Handbook of Weighted Automata*, chapter 6, pages 213–254. Springer Verlag.

Andrew Y. Ng. 2004. Feature selection, l1 vs. l2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, pages 78–86.

Franz J. Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL 2004: Main Proceedings*, pages 161–168, Boston, Massachusetts, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318.

Martin Riedmiller and Heinrich Braun. 1993. A direct adaptive method for faster backpropagation learning: The RPROP algorithm. In *Proceedings of the IEEE International Conference on Neural Networks*, pages 586–591, San Francisco, USA.

Charles Sutton and Andrew McCallum. 2006. An introduction to conditional random fields for relational learning. In Lise Getoor and Ben Taskar, editors, *Introduction to Statistical Relational Learning*, Cambridge, MA. The MIT Press.

Robert Tibshirani. 1996. Regression shrinkage and selection via the lasso. *J.R.Statist.Soc.B*, 58(1):267–288.

Christoph Tillman. 2004. A unigram orientation model for statistical machine translation. In Susan Dumais, Daniel Marcu, and Salim Roukos, editors, *HLT-NAACL 2004: Short Papers*, pages 101–104, Boston, Massachusetts, USA.

J. Turian, B. Wellington, and I.D. Melamed. 2007. Scalable discriminative learning for natural language parsing and translation. In *Proc. Neural Information Processing Systems (NIPS)*, volume 19, pages 1409–1417.

Taro Watanabe, Jun Suzuki, Hajime Tsukada, and Hideki Isozaki. 2007. Online large-margin training for statistical machine translation. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 764–773, Prague, Czech Republic.

Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 475–484, Uppsala, Sweden.

Fei Xia and Michael McCord. 2004. Improving a statistical mt system with automatically learned rewrite patterns. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING)*, pages 508–514, Geneva, Switzerland.