Reordering Space Design in Statistical Machine Translation

Nicolas Pécheux, Alexandre Allauzen, Jan Niehues, François Yvon

Reordering Space Design in Statistical Machine Translation

Nicolas Pécheux[1,*] Alexandre Allauzen[1] Jan Niehues[2]
François Yvon[1]

[1]Université Paris-Saclay, CNRS, LIMSI
Campus Universitaire d'Orsay
F-91403 Orsay Cedex
{pecheux,allauzen,yvon}@limsi.fr

[2]Karlsruhe Institute of Technology
Institute for Anthropomatics and Robotics
Adenauerring 2 - 76131 Karlsruhe
jan.niehues@kit.edu

Preprint of a paper published in *Language Resource and Evaluation (2016)*
https://doi.org/10.1007/s10579-016-9353-8

Abstract

In Statistical Machine Translation (SMT), the constraints on word reorderings have a great impact on the set of potential translations that is explored during search. Notwithstanding computational issues, the reordering space of a SMT system needs to be designed with great care: if a larger search space is likely to yield better translations, it may also lead to more decoding errors, because of the added ambiguity and the interaction with the pruning strategy. In this paper, we study the reordering search space, using a state-of-the art translation system, where all reorderings are represented in a permutation lattice prior to decoding. This allows us to directly explore and compare different reordering schemes and oracle settings. We also study in detail a rule-based preordering system, varying the length and number of rules, the tagset used, as well as contrasting with purely combinatorial subsets of permutations. We carry out experiments on three language pairs in both directions: English-French, a close language pair; English-German and English-Czech, two much more challenging pairs. We show that even though it might be desirable to design better reordering spaces, model and search errors seem to be the most important issues. Therefore, improvements of the reordering space should come along with improvements of the associated models to be really effective.

1 Introduction

Reordering is known to be a critical issue for statistical machine translation and the reordering complexity for a language pair can be considered as a relevant indicator of the difficulty to automatically translate from one into the other (Birch et al, 2008).

When translating a source sentence, most machine translation systems, either explicitly or implicitly, have to choose the order in which they will process the source sentence to compute its translation, thereby inducing a reordering of the source words

*Corresponding author

which reflects the target word order. In order to correctly generate the word order, two main problems have to be solved: the identification of a restricted number of possible reorderings and the numerical evaluation of their appropriateness. The first step is necessary due to the intractability of exploring the combinatorial set of all possible permutations. Even for short sentences, this set contains too much ambiguity and an overwhelming number of linguistically meaningless reorderings. It is therefore necessary to rely on methods that filter this space so as to meet the two following conflicting goals: (a) the search space should be large enough to contain good translation hypotheses; (b) yet small enough to be rapidly explored. This first problem thus amounts to identify appropriate *reordering constraints*, which will help to shape the set of permutations of the source that will actually be considered. The second is to design *reordering models* that can assign numerical scores to candidate permutations, so that the most correct word order(s) will receive high scores. Those include distance-based models, lexicalized reordering models (Tillmann, 2004) or hierarchical lexicalized models (Galley and Manning, 2008) among many others.

In this work we mainly focus on analyzing the first issue. While improvements on the reordering models are likely to benefit the overall translation performance, it is less obvious to what extent the reordering constraints are currently impacting the translation process. Indeed, in addition to computational issues, there is a tradeoff when building the reordering space of a machine translation system. On the one hand, a larger space is more likely to contain a permutation that can yield a relevant translation. On the other hand, it may also cause more decoding errors, because of both the ambiguity of natural languages and the necessary pruning of the search space. It is then of great help to understand the current limits of an SMT regarding the reordering space. Thus, the main questions we address in this work are: how good are the current reordering search spaces? how to design them? is it important that they contain the exact needed reorderings or good approximations would be enough? are the models able to make use of the best reorderings from the search space? to what extent would the overall system benefit from much better reordering spaces? Therefore, we investigate several ways to generate the reordering space, in order to evaluate how the SMT system can benefit from a larger/better reordering space. In addition, by studying monotonic as well as various oracle-like reordering spaces, we compute lower and upper bounds on the possible reordering space design, also giving insights on the complex influence of search and model errors on the translation quality.

Various constraints on admissible permutations have been proposed in the past including IBM (Berger et al, 1996), MJ (Kumar and Byrne, 2005) or ITG (Wu, 1997). Those constraints have been compared in terms of performance (Zens and Ney, 2003; Zens et al, 2004) or in oracle settings (Dreyer et al, 2007; Wisniewski and Yvon, 2013). Other approaches include linguistically motivated rules that are automatically learned (Crego and Mariño, 2006; Niehues and Kolss, 2009; Herrmann et al, 2013a). To the best of our knowledge, these two families of approaches, purely combinatorial on the one hand and empirically learned on the other, have never been systematically compared. In this work, we use a rule-based reordering system in which reordering rules are extracted during the training phase (Section 3.2), considering word factors instead of surface word in an attempt to mitigate sparsity issues. We study in detail the effectiveness of the rule-based approach in defining an accurate search space, and show that linguistically motivated constraints define can be used to define a compact search space, and yet, improve the translation quality.

In the phrase-based approach, word reorderings can be divided in two tightly intertwined types: local reorderings that take place within phrases; and longer reorderings

of those phrases. The additional use of pre-ordering methods is introduced eg. in (Xia and McCord, 2004; Collins et al, 2005; Tromble and Eisner, 2009; Genzel, 2010): in this approach, source sentences are reordered in a preprocessing step to match the target word order and then fed into the standard phrase-based pipeline. This further complexifies the analysis of the reorderings that are actually considered in translation. Finally, because of pruning, only a restricted part of the search space is effectively explored. In this paper, we use a state-of-the-art *n*-gram SMT system (Crego et al, 2011), described in Section 2, that splits reordering and decoding into two separate steps. Reorderings of the source sentence are compactly encoded in a permutation lattice, the *reordering space*, that is then translated in a monotonic fashion. This two-step approach allows us to study the reordering space that is explored and then to assess its impact on the whole translation process. This controlled framework also enables to directly compare the size and the coverage of the different reordering spaces. Therefore, even though we only consider one specific phrase-based architecture, we believe that most conclusions would carry over for other phrase-based systems, that mostly use the same reordering mechanisms, and even for hierarchical phrase-based systems Auli et al (2009).

Evaluation is carried out for three language pairs (French-English, German-English and Czech-English in both directions) that differ by the range of the involved reorderings. We measure the impact on the system performance as well as oracle decoding to better understand the potentials of the different reordering spaces as well as the influence of search and model errors on translation quality. We find that while there is ample room for improving the reordering space, this problem might not be the main issue, since search and model errors would prevent the SMT system to fully benefit from a more accurate search space.

The rest of this study is organized as follows. In section 2, we present the *n*-gram-based approach and its peculiarities. Among them, the rule-based method for source reordering is described in Section 3, while Section 4 explains how to build the reordering space explored by the SMT system and how to derive oracle-like reorderings. In Section 5, multiple experimental comparisons are carried out to assess the impact of the reordering space on translation performance.

2 The *n*-gram-Based Approach in SMT

All our experiments use NCODE, an open source SMT toolkit[1], which achieved state-of-the-art performance in recent evaluation campaigns (Callison-Burch et al, 2012; Bojar et al, 2013, 2014). NCODE implements the bilingual *n*-gram approach to SMT (Casacuberta and Vidal, 2004; Mariño et al, 2006; Crego and Mariño, 2006) that is closely related to the standard phrase-based approach (Zens et al, 2002). In this approach, the translation of a source sentence **f** into a target sentence **e** is decomposed into two steps: a source reordering step and a monotonic translation step. Since the translation step is monotonic, the peculiarity of this approach is to rely on the *n*-gram assumption to factor the joint probability of a sentence pair into a product of conditional probabilities involving *bilingual* atomic units called *tuples*: in other words, the translation model is a conventional *n*-gram model of synchronized segments.

NCODE uses a set of feature functions embedded in a log-linear model (Och and Ney, 2002) that is similar to standard phrase-based systems (see Crego et al (2011) for

[1] `http://ncode.limsi.fr`

details). The best translation is selected by solving the following program:

$$\arg\max_{\mathbf{e},\mathbf{a}} p(\mathbf{e},\mathbf{a}|\mathbf{f}) = \arg\max_{\mathbf{e},\mathbf{a}} \frac{1}{\mathscr{Z}_{\mathbf{f}}} \exp\Big(\sum_{k=1}^{K} \lambda_k f_k(\mathbf{f},\mathbf{e},\mathbf{a})\Big) \quad (1)$$

where K feature functions $\{f_k, k = 1 \dots K\}$ are weighted by a set of coefficients $\{\lambda_k\}$, $\mathscr{Z}_{\mathbf{f}}$ is a normalizing factor and $\mathbf{a}$ denotes the set of hidden variables corresponding to the reordering and segmentation of the source sentence. Along with the *n-gram translation model* and the *target n-gram language model*, 13 conventional features are combined: 4 *lexicon models* similar to the ones used in standard phrase-based systems; 6 *lexicalized reordering models* (Tillmann, 2004; Crego et al, 2011) aimed at predicting the orientation of the next translation unit; a "weak" distance-based *distortion model*; and finally a *word-bonus model* and a *tuple-bonus model* which compensate for the system preference for short translations. Features are estimated during the training phase and the corresponding weights (λ_k) are estimated during a tuning phase on held-out development data. The models that have a direct impact on the selected reordering are the monolingual and bilingual n-gram models, the lexicalized reordering models and the distortion model.

During training, source sentences are first reordered so as to match the target word order by *unfolding* the word alignments. In a nutshell, unfolding aims to reorder the source words so as to remove all crossing alignment links; additional heuristic rules handle the movements of non-aligned words on the source side and make the procedure deterministic (see details in (Crego et al, 2005)). Unfolding is performed as follows: the target sentence is first segmented in K segments of *consecutive* words $\mathbf{e} = \mathbf{e}_1 \dots \mathbf{e}_k \dots \mathbf{e}_K$ such that for each segment $\mathbf{e}_k$, if a word f aligns with one word in $\mathbf{e}_k$, it is only aligned with words in $\mathbf{e}_k$, i.e. if $\mathbf{f}_k = \{f \in \mathbf{f} | \exists e \in \mathbf{e}_k, (f,e) \in \mathbf{a}\}$ is the set of source words aligned with $\mathbf{e}_k$, then $\forall f \in \mathbf{f}_k, \forall e \in \mathbf{e}, (f,e) \in a \Rightarrow e \in \mathbf{e}_k$. This is the same as for standard phrase extraction, except that the source words need not be consecutive. One can then output the reordered source words $\tilde{\mathbf{f}} = \mathbf{f}_1 \dots \mathbf{f}_k \dots \mathbf{f}_K$ (using monotonic order withing each $\mathbf{f}_k$)[2] and the tuple sequence $\{(\mathbf{f}_k, \mathbf{e}_k)\}_k$. Figure 1 displays a simple example, where the word `politicians` is moved to the start of the sentence. Unaligned words on the target side, such as `l'` in Figure 1, cause problems as the search does needs input words to generate units; they are consequently attached to the neighbor tuple which maximizes IBM model 1 lexical probabilities (de Gispert and Mariño, 2006). Tuples are then extracted in such a way that a unique segmentation of the bilingual corpus is achieved. A n-gram translation model and optional word factor models are then estimated over the training corpus composed of tuple sequences, using modified Kneser-Ney smoothing (Chen and Goodman, 1998).

During decoding, the source sentence is first reordered so as to reproduce the word order modifications introduced during the tuple extraction process, i.e. to best match the target word order. This generates a word lattice containing the most promising source permutations. This lattice represents the reordering space that is then searched for the best candidate translation. As exhaustive search is intractable, NCODE uses a beam search strategy based on stacks. As future cost estimation is problematic for multiple n-gram models, NCODE uses one stack per hypothesis translating the *same input words*, in contrast to the *same number of words* as in standard phrase-based systems. Thus the memory footprint of the decoding algorithm directly depends on the number of nodes in the reordering lattice.

[2] One has also to decide when to output unaligned source words, for which we use a special tuple with a source NULL token (see Figure 1). In our case, it is output just before the next aligned token.

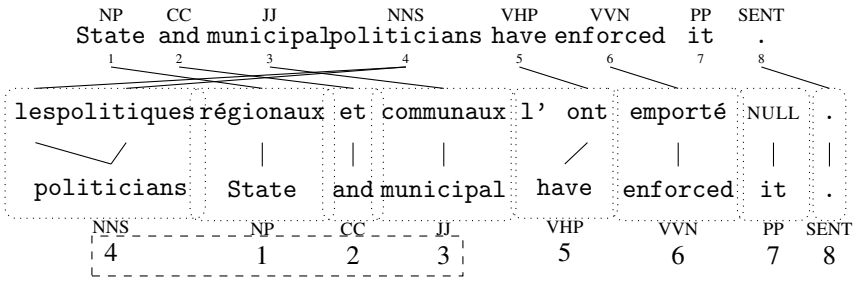


Figure 1: *Unfolding process and reordering rules extraction from word alignment. The source English sentence is aligned with the target French sentence. The unfolding procedure moves the fourth word 'politicians' to the start of the sentence, yielding the unfolded permutation* $\sigma = 41235678$. *Tuples that would be extracted are in a dotted box. The unaligned source word 'it' is associated to the special token 'NULL' while the unaligned target word 'l" is merged with a neighboring tuple. Only one rule would be extracted here, mapping the POS tag sequence* `NP CC JJ NNS` *to the minimal reordering in a dashed box.*

3 Learning Rule-based Reorderings

3.1 Reorderings

We have thus far used the term *word reordering*, even though the definition of how the words "move" during translation is not trivial, as translation is not word-to-word. In fact, standard phrase-based approaches first segment the sentences in phrases and only consider reorderings of those phrases, while local "word moves" are implicitly included withing the phrases[3]. In this work, we are interested in understanding the reorderings that are considered by the overall system, thus the focus on the permutations at the level of words. Word reorderings can be inferred from word alignments, which indeed originate from word-to-word translation models (Berger et al, 1996). It is however not straightforward to induce a permutation from many-to-many alignments and several heuristics, that differ by many subtle details, have been used for evaluating reordering (Birch, 2011) or preordering techniques (Tromble and Eisner, 2009; Khalilov and Sima'an, 2012; Neubig et al, 2012). In this work, we directly make use of the unfolding procedure to obtain word reorderings. As explained above (see Figure 1), unfolding the alignment links directly results in a permutation, that we call the *unfolded reordering*.

Intuitively, a reordering occurs when some words move away from their initial position. In general, a global permutation can be decomposed in many local reorderings. Let $\mathfrak{S}_n$ be the set of permutations of $\{1,\ldots,n\}$ for some integer n and let $\sigma \in \mathfrak{S}_n$ be a permutation $\sigma = (\sigma_1 \ldots \sigma_n)$ with $\forall i, \sigma_i \in \{1,\ldots,n\}$. We define a *reordering* of σ to be any subsequence $\sigma_{[i:j]} = \sigma_i \ldots \sigma_j$ of σ with $|j-i| > 1$ such that:

$$\forall k, i \leq k \leq j \;\Rightarrow\; i \leq \sigma_k \leq j$$

i.e. $\{\sigma_k\}_{i \leq k \leq j} = \{i,\ldots,j\}$. A reordering is said to be *minimal* if it is minimal for this property, i.e if it doesn't (strictly) contain any reordering. Spans $\sigma_{[i:j]}$ correspond to the

[3]Note that our approach makes local word moves explicit, by performing the reordering step before the segmentation one, and therefore subsumes standard phrase-based approaches with or without preordering.

smallest (non trivial) ones to be reordered in order to recover σ. It is easy to see that any permutation can be segmented in an unique way, where each segment is either a fixed point or a minimal reordering. For example, the unfolded reordering $\sigma = 41235678$ in Figure 1 contains only one minimal reordering ($\sigma = 4123$) and four fixed points. Any reordering π can be mapped to a (unique) permutation $\bar{\pi} \in \mathfrak{S}_{|\pi|}$ by renumbering, i.e. $\forall k, \bar{\pi}_k = \pi_k - \min(\pi)$, where $\min(\pi)$ is the smallest integer in π. Let $\mathscr{R}_n \in \mathfrak{S}_n$ be the set of minimal reorderings of $1,\ldots,n$ for $n \geq 2$. The number of minimal reorderings $r_n = |\mathscr{R}_n|$, can be computed recursively as:

$$r(n) = \begin{cases} 1 & if \quad n = 1 \\ n! - \sum_{i=1}^{n-1} r(i) \cdot (n-i)! & else \end{cases} \tag{2}$$

where $r(1) = 1$ is a mathematical convenience.

3.2 Reordering Rules Extraction

Reordering rules are automatically learned during the unfolding procedure. Let $\mathbf{w} = w_1 w_2 \ldots w_n$ be a source sentence and $\mathbf{t} = t_1 t_2 \ldots t_n$ the associated tag sequence. Let $\mathbf{w}_\sigma = w_{\sigma_1} w_{\sigma_2 \ldots} w_{\sigma_n}$ be the reordered sentence produced by the unfolding procedure where $\sigma = \sigma_1 \ldots \sigma_n \in \mathfrak{S}_n$. A reordering rule is extracted for any *minimal reordering* $\sigma_{[i:j]}$ of σ. Rules then have the following form:

$$\mathbf{t}_{[i:j]} \rightarrow \bar{\sigma}_{[i:j]}$$

where $\bar{\sigma}_{[i:j]}$ is the induced permutation in $\mathfrak{S}_{|j-i+1|}$ obtained by renumbering $\sigma_{[i:j]}$ as described above (see § 3.1). An example is in Figure 1, where only one rule:

NP CC JJ NNS$\rightarrow$ 4 1 2 3

would be extracted.

Note that it would also be conceivable to also extract rules $\mathbf{t}_{[i:j]} \rightarrow \bar{\sigma}_{[i:j]}$ for *non-minimal* reorderings (subject to $|j-i| > 1$) in a way similar to the phrase extraction heuristic in MOSES; preliminary experiments showed a slight drop in performance for this variant, which is not explored further in this paper.

To filter out alignment noise and limit the size of the reordering space, rules may be pruned according to a maximum cost threshold (maxcost). The cost of a rule is defined by:

$$\text{cost}(\mathbf{t} \rightarrow \sigma) = -\log \frac{\text{count}(\mathbf{t} \rightarrow \sigma)}{\sum_{\sigma' \in \mathfrak{S}_{|\mathbf{t}|}} \text{count}(\mathbf{t} \rightarrow \sigma')}, \tag{3}$$

where $\mathbf{t}$ is any tag sequence, $\sigma \in \mathfrak{S}_{|\mathbf{t}|}$ is a permutation and the counts are computed on the training data. Since this cost is the negative logarithm of a conditional ratio, a coarser tagset might be more heavily pruned than a fine-grain one, resulting in a smaller set of extracted rules; in principle, the optimal threshold thus depends on the granularity of the tagset, as well as on the translation direction. In our experiments, we use a default value of 4 for the parameter maxcost.

Rules may be also pruned according to their length (by default 10). Preliminary experiments show that further increasing this limit hardly makes any difference in performance. In fact, long rules are too sparse to possibly generalize beyond the training set. Long range reorderings are thus explicitly excluded from the model. Note that in standard phrase-based systems, the maximal reordering span, i.e. the distortion limit is usually even set to a smaller value than ours.[4]

[4]For example, the default distortion limit in MOSES is 6.

3.3 Alternative Tagsets

In (Crego and Mariño, 2006), rewriting rules are built using Part-of-speech (POS), rather than surface word forms in order to increase their generalization power. However, any word factor may possibly be used. To investigate different levels of generalization and the relevance of syntactic word factors, different tagsets are introduced.

- **One single tag** (one): The tagset consist of one single tag. This means that the reordering rules are extracted and applied independently of any syntactic or contextual information. This results in a system which reduces the possible reorderings to all the ones observed in the training data.
- **Universal POS** (ups): The tagset is reduced to 12 simple language-independent categories, in an attempt to limit the sparsity of the extracted rules. In this work we use the universal POS tagset described in (Petrov et al, 2012). For under-resourced languages, universal POS can be projected by cross-lingual transfer or learned from partial annotations (Li et al, 2012; Täckström et al, 2013; Wisniewski et al, 2014), thereby relaxing the need for a POS tagger.
- **Enhanced POS** (e50pos): The POS tags are lexicalized for the 50 most frequent words, resulting in more specific rules. Enhanced tags are closely related to lexicalized rules (Huang and Pendus, 2013).
- **Brown classes** (classes): Statistical word classes were found to be a good approximation for Part-of-Speech tags when a POS tagger is not available. In (Ramanathan and Visweswariah, 2012), word clusters perform worse than POS, but still do reasonably well in a preordering setting. Durrani et al (2014) report some gains when using word clusters, in addition to POS and morphological tags, in an Operation Sequence N-gram model. In this work, we compute statistical word clusters using the method of Brown et al (1992).
- **Plain words** (words): We use the surface word to build the rules, resulting in high-specific fully lexicalized rules with less generalization power.

4 Reordering spaces

The reordering space explored during decoding can be generated in many different ways. In standard phrase-based SMT, all possible reorderings of source segments that do not result in a word move above a distortion threshold is implicitly used (see Lopez (2009) for a detailed account). In this work, the generation of the reordering space is controlled by a set of rewriting rules that non-deterministically reorder the source words. In this section, we detail the procedure use to generate the reordering lattice used in our rule-based system, as well as variants considered in our experiments.

4.1 Reordering Lattice Generation

A *permutation lattice* (Crego, 2008) is an acyclic weighted Finite State Automaton (FSA) $\mathscr{L} = \langle V, E, \Sigma, w \rangle$, where V is a set of nodes, E a set of edges, the alphabet $\Sigma = \{1, \ldots, n\}$, $w : E \rightarrow \mathbb{R}$ a weight function, which generates (or recognize) a language $L(\mathscr{L}) \subset \mathfrak{S}_n$, i.e. in which any path corresponds to a permutation of $\{1, \ldots, n\}$.

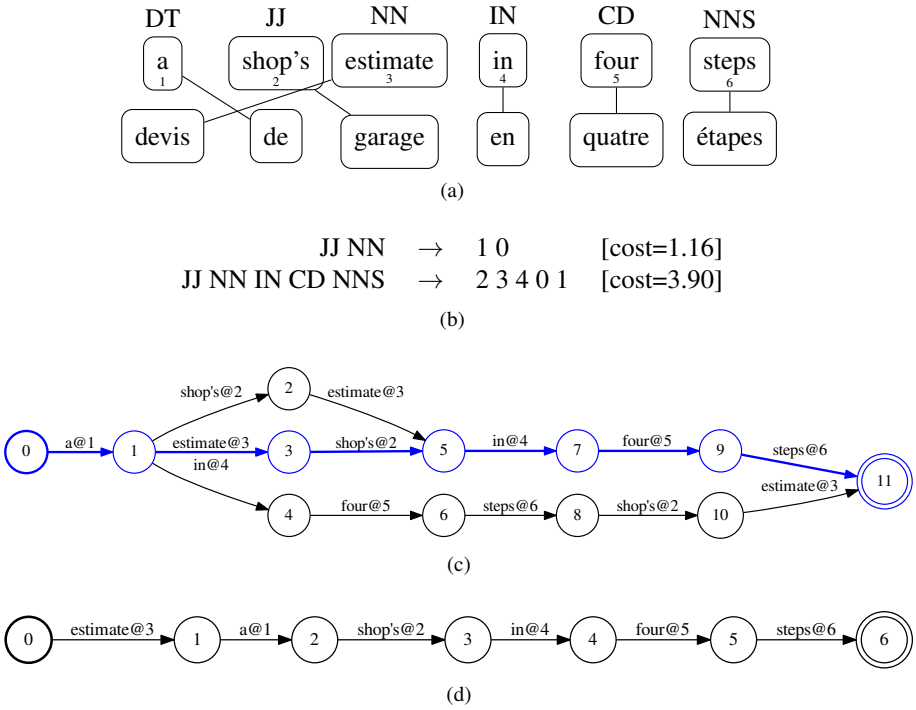


Figure 2: *Example of a tagged source test sentence, the target reference and the forced alignment (a); training rules that apply for this source sentence with costs computed according to* (3) *(b); reordering lattice obtained when applying the rules, with the best reordering as defined in Section 4.3 in bold blue (c); the corresponding unfolded reordering lattice (d).*

For some subsets of permutations, a lattice encodes an exponential number of permutations with a polynomial number of nodes and edges. A strong property of permutation lattices is that all incoming paths that reach a node cover the same word indices.

The permutation lattice is built incrementally for any sentence $\mathbf{w}$ with corresponding tags $\mathbf{t}$ as follows. The monotonic path forms the initial lattice. Then for each segment $[i:j]$ and each rule $\mathbf{t}_{[i:j]} \rightarrow \sigma$, the lattice is expanded by adding the subpath $\sigma([i:j])$. Figure 2 displays an example. This is performed in a parallel fashion so that rewriting rules do not interfere with each other. Applying the reordering rules finally results in a finite-state graph that represents the *reordering space*. This lattice may be weighted, using for example the probability of reordering rules as, in (Herrmann et al, 2013b); this allows us to include the lattice path score as a feature in the log-linear combination of Equation (1). We have not pursued such developments in this study, as previous experiments did not show any improvement of performance.

In principle, one can design any set of permutation constraints and encode them in a lattice. In practice, the number of nodes in the lattice must remain reasonable (polynomial) in the number of words in the sentence.[5] To assess whether constraining the reorderings to those observed in the data is appropriate, the rule-based approach is compared with MaxJump (MJ) constraints (Kumar and Byrne, 2005). In MJ-*i*, a word

[5]This would not be the case for ITG constraints for example. However our framework could be extended to *permutation forests* to take into account more general reordering spaces (Dyer and Resnik, 2010).

move cannot exceed i positions.[6] This is equivalent to using a rule-based system where all possible rules up to size $i+1$ would be considered.

4.2 Oracle Unfolded Reordering

At training time, source sentences are deterministically reordered to enable the tuple extraction and the estimation of the models. During decoding, one would ideally like to process source sentences in their correct target order, i.e. the unfolded reordering defined in Section 2. For unseen data, this oracle can be derived from forced alignments between a source sentence and the corresponding reference (see Figure 2 (d) for an example). In that sense, the best reordering constraints should be the ones that generate lattices containing the unfolded reordering as the only option. We refer to this oracle-like reordering as the *unfolded reordering*.[7]

4.3 What is the Best Reordering in a Lattice ?

As explained above, the unfolded reordering can be considered as the best possible reordering. However for unseen data, this oracle reordering usually requires long range moves and/or permutations that were not observed in the training data. In our experiments (Section 5.7), only about 20-60% of the test unfolded reorderings are actually reachable by our rule based system, depending on the translation direction and the setting used. Therefore it is also of interest to study the properties of the *best reordering* the system can explore. This best reordering can be defined as the path in the reordering lattice leading to the best translation. Such definition would however make the best reordering depend on the whole SMT system, including the pruning strategy and the translation models. Instead, we follow Herrmann et al (2013b) by defining the best reordering as the one closest to the unfolded reordering. This approximation assumes that the best order is the one that most closely matches the target reference order, which is reasonable as most of the automatic metrics also rely on a similar assumption.

Finding the closest permutation requires to define a metric over permutations. Among many choices (Deza and Huang, 1998), two metrics have been shown to be useful when assessing reordering accuracy: the Kendall's τ (Isozaki et al, 2010a; Birch et al, 2010; Talbot et al, 2011; Neubig et al, 2012) and the fragmentation chunk (or fuzzy reordering) (Banerjee and Lavie, 2005; Talbot et al, 2011; Neubig et al, 2012). In this work, we use the Kendall's τ which proved to correlate strongly with human fluency judgment (Birch et al, 2010). The Kendall's τ metric (Kendall, 1962) counts the number of *pairwise* disagreements between two permutations $\sigma, \pi \in \mathfrak{S}_n$

$$\tau(\sigma,\pi) = \sum_{i=1}^{n}\sum_{j=1}^{n} \mathbb{1}_{\{\sigma_i<\sigma_j\}}\mathbb{1}_{\{\pi_i>\pi_j\}} \tag{4}$$

where $\mathbb{1}_{cond}$ is the indicator function with value 1 if *cond* is true and 0 otherwise. It is also the minimum number of swaps between two adjacent symbols needed to transform one permutation into the other, so the distance is also sometimes called the *bubble-sort* distance. The Kendall's τ is usually normalized, so a value of 1 indicates a maximum disagreement

$$\tau^{\text{norm}} = \sqrt{\frac{2\tau(\sigma,\pi)}{n(n-1)}}. \tag{5}$$

[6] Note that defining MJ constraints over *words* amounts to using a fixed distortion limit.

[7] We avoid the term "oracle reordering" to prevent later confusions with oracle decoding.

In the following, we explain how to efficiently search a lattice and find the closest Kendall's τ permutation to the unfolded reordering. First observe that

$$\tau(\sigma,\pi) = \tau(\pi^{-1}\circ\sigma, id), \tag{6}$$

where *id* is the identity permutation. Up to relabeling, the problem is then to find the permutation in a lattice $\mathscr{L}$ with the minimal number of inversions:

$$\underset{\sigma\in\mathscr{L}}{\arg\min}\ inv(\sigma) = \sum_{i=1}^{n}\sum_{j=i+1}^{n}\mathbb{1}_{\{\sigma_i>\sigma_j\}} = \sum_{j=1}^{n}\sum_{i=1}^{j-1}\mathbb{1}_{\{\sigma_i>\sigma_j\}} = \sum_{j=1}^{n} w(\sigma_j, \{\sigma_i\}_{i=1}^{j}), \tag{7}$$

where $w(k,S) = \sum_{s\in S}\mathbb{1}_{\{s>k\}}$ counts how many times an integer k is lower than any element from an integer set $S \in 2^n$. The number of inversions thus decompose as a sum of local functions that only depend on the *set* of already permuted integers. As observed above, in a permutation lattice $\mathscr{L} = (V,E,\Sigma,w)$, each node $v \in V$ corresponds to a set of integers S_v. Each edge $e \in E$ leaving node v with integer label k will have a contribution $w(k,S_v)$ to the total number of inversions of any path in the lattice leading to e. Therefore, $\mathscr{L}$ can be weighted with $w(k,S_v)$ and the conventional shortest path algorithm can infer the best reordering.

Figure 2 displays an example of a reordering lattice containing three paths, with respective Kendall's τ to the unfolded reordering equal to 1, 2 and 8. The best path has just one arc of non-null weight between states 1 and 2, where the first word ($k = 1$) of the unfolded reordering leaves state 1 covering the second word ($S_1 = \{2\}$); likewise the second best bath has one non-null edge of weight 2 between nodes 2 and 5, since this edge corresponds again to $k = 1$, with a coverage equal to $S_2 = \{2,3\}$.

As many paths in the lattice may have the smallest distance to the unfolded reordering (i.e. the arg min in Equation (7) may not be unique), the shortest path is in fact a sub-lattice of the original one. In our experiments, we found however the best reordering lattices to have about only 1.1 path on average.

4.4 Metrics

Given our assumptions, the reordering space should be the smallest one containing the unfolded reordering. Therefore, as a quality measure on reordering constraints, we define the *coverage* on some test set as the number of time the reordering space contains the reference reordering. On the other hand, we compute the *size* of the reordering space as the number of paths[8] as well as the number of edges in the reordering lattice, as this last number closely relates to the decoding complexity.

This study primarily focuses on the overall translation performance, in relationship to the order in which the source has been translated. Our main translation quality metric is therefore BLEU (Papineni et al, 2002). As BLEU however provides little insight from the perspective of the reordering quality, we also report Kendall's τ metric to separately evaluate reorderings as in (Birch and Osborne, 2010). Appendix A also presents results with two additional metrics that were designed to better take into account reordering issues : BEER (Stanojević and Sima'an, 2014) and RIBES (Isozaki et al, 2010a).

[8] Computed efficiently using the counting semiring.

5 Experimental Results

5.1 Experimental setup

Our experimental setup is based on the WMT[9] evaluation campaign shared task. We consider the following language pairs (on both directions): English-French, English-German and English-Czech. For training, the NEWSCOMMENTARY corpus provided by the organizers of WMT'12 (Callison-Burch et al, 2012) is used; *newstest2009* and *newstest2010* are used for tuning and testing,[10] respectively. Table 1 contains various basic statistics regarding these corpora.

In-house text processing tools are used for the tokenization (Déchelotte et al, 2008) in a "true-case" scheme. As German is morphologically rich, the German source side is normalized using a specific preprocessing scheme (Allauzen et al, 2010; Durgar El-Kahlout and Yvon, 2010) which aims at reducing the lexical redundancy by normalizing the orthography, neutralizing most inflections and splitting complex compounds. The English side of the parallel corpora is POS-tagged with Wapiti (Lavergne et al, 2010), while for French and German we use the TreeTagger (Schmid, 1994) and for Czech the open-source tool MORPHODITA[11] (Straková et al, 2014). In the last case, only the first two characters of the fifteen-letter Prague Dependency Treebank tags are used, resulting in 67 possible POS tags. For all languages, we also use the mappings from Petrov et al (2012) to project to the Universal Tagset.

Word alignments and the 50 word classes[12] are computed using MGIZA++[13] and MKCLS[14] with default settings, using, for English-French and English-German all the parallel data described in (Allauzen et al, 2013), and, for English-Czech, the EUROPARL and COMMONCRAWL parallel WMT'12 corpora.

For each task, a 4-gram language model is estimated using the target side of the training data. We use NCODE with the default setting and an additional bilingual factor model based on POS tags.[15] The beam size is set to 25 for KB-MIRA tuning Cherry and Foster (2012) and to 50 when decoding, a parameter setting that worked well in previous experiments. All results are averaged over 3 runs to control for optimizer instability (Clark et al, 2011). Approximate randomization tests for multiple optimizer samples to assess statistical significance are carried out using MULTEVAL.[16]

Oracles are computed using the linear approximation to the BLEU score introduced by Tromble et al (2008): using a first order Taylor-series approximation to the corpus log(BLEU) gain leads to the following sentence level gain function:

$$G(\mathbf{e},\mathbf{e}') = \theta_0|\mathbf{e}'| + \sum_{n=1}^{4} \sum_{g \in n\text{-gram}(\mathbf{e})} \theta_n \cdot \#_g(\mathbf{e}') \tag{8}$$

for a reference $\mathbf{e}$ and an hypothesis $\mathbf{e}'$, where n-gram($\mathbf{e}$) is the set of n-grams in $\mathbf{e}$ and $\#_g(\mathbf{e}')$ is the number of times a n-gram g appears in $\mathbf{e}'$. As this sentence-level approximation decomposes into a sum of local functions, we can efficiently find the

[9] Workshop on Machine Translation – see `http://www.statmt.org/wmt`.

[10] Note that the English side of *newstest2009* and *newstest2010* is the same for all translation directions, which enables a fair comparison between languages.

[11] With models from Straka and Straková (2013), see `http://ufal.mff.cuni.cz/morphodita`.

[12] Out-of-vocabulary words in decoding are mapped to class 1.

[13] `http://www.kyloo.net/software/doku.php`.

[14] `http://code.google.com/p/giza-pp/`.

[15] Note that this is independent of the choice of tags used in the reordering rules.

[16] `https://github.com/jhclark/multeval`

	NEWSCOMMENTARY					*newstest2010*					# pos
	sent.		% mono	reord.		sent.		% mono	reord.		
	#	len		#	len	#	len		#	len	
$en \rightarrow fr$	137k	25	17	1.8	3.8	2k	25	20	1.6	4.3	44
$fr \rightarrow en$		29	14	2.0	4.7		28	17	1.7	4.9	34
$en \rightarrow de$	158k	24	19	1.6	6.4	2k	25	17	1.5	7.2	44
$de \rightarrow en$		26	16	1.7	6.8		26	16	1.6	7.3	116
$en \rightarrow cs$	139k	23	31	1.0	6.5	2k	25	27	1.1	6.9	44
$cs \rightarrow en$		21	29	1.0	6.3		21	29	1.0	6.6	63

Table 1: *Basic statistics regarding the experimental data: number (#) and average length (len) of source sentences (sent.); percentage of monotone sentence pairs (mono); average number per sentence (#) and average length (len) of the reorderings (reord.); as well as the size of the Part-of-Speech tagset.*

maximum gain path in the lattice. The parameters of Equation (8) are chosen using $\theta_n = (4p \times r^{n-1})^{-1}$ for $n \in \{1,\ldots,4\}$ (Tromble et al, 2008), where the unigram precision p, the precision ratio r and the length bonus θ_0 are chosen so as to maximize corpus-level BLEU. We found $p = 0.4$, $r = 0.8$ and $\theta_0 = -1$ to yield good performance.

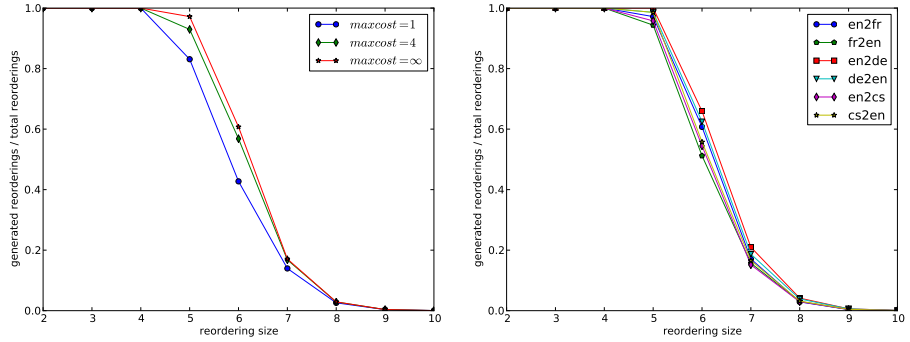
5.2 Coverage, Generalization and Complexity of the Rule-Based Approach

A first question is the coverage and the generalization power of the rule-based approach. Figure 3 displays, for each reordering size n, the ratio between the number of reorderings[17] observed in the data and the total number of possible reorderings of that size (i.e. r_n as defined in Section 3.1). We also vary the values of the cost-based filtering threshold: when $maxcost = \infty$, all the reorderings observed in the training data are considered. In this case, almost all the possible reorderings appear in the data for the rules up to length 5. The ratio then quickly decreases to zero as the size exceeds 8. Moreover, the comparison between $maxcost = 1$ and $maxcost = 4$ shows that the exact value of the threshold has a small impact on the coverage and this trend is observed for all translation directions.

Figure 4 characterizes the complexity of the reorderings for three conditions: (a) the reorderings observed in the training data; (b) the reordering in test data; (c) the test reorderings that are not captured[18] by the rule-based approach (*miss*). The complexity of reorderings, as a function of their size, is described with three indicators: the proportion of extracted permutations that are in the ITG family, the average normalized Kendall's τ and the normalized fragmentation chunk distance to the identity permutation. Statistics are computed at the level of rules (rather than sentences): this is because rules decompose sentences into chunks that are reordered independently. For long sentences with many independent local reorderings, the properties of each local reordering are more relevant than considering the sentence as a whole. For instance, a non-ITG

[17] In all this section we always consider *minimal* reorderings, see Section 3.1.

[18] Because reorderings are minimal, it is easy to see that a test reordering for a given sequence of tags is captured by the rule-based approach if and only if a rule with the correct right-hand side exists for that POS sequence.



(a) $en \rightarrow fr$ (b) $maxcost = \infty$

Figure 3: *Ratio of reorderings observed in the training data as a function of the reordering size, when using the POS tagset rules and different filtering thresholds (maxcost) (a) or with maxcost* $= \infty$ *(b) across all language directions. Plots for the five other language directions were almost identical to the* $en \rightarrow fr$ *case. Reorderings of size larger than* 10 *are not reported, as their ratio is indistinguishable from* 0*.*

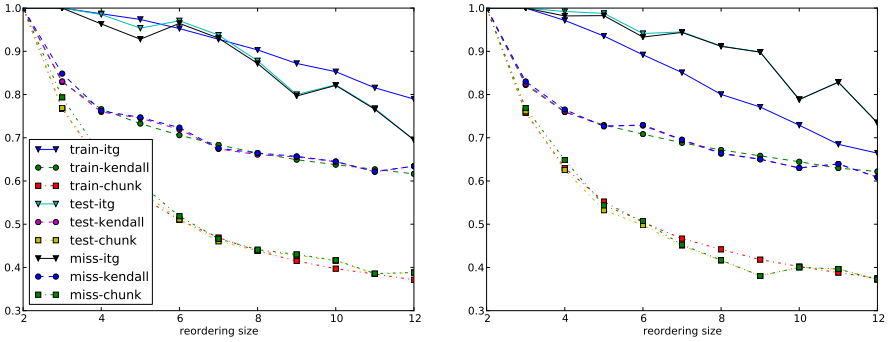
sentence may exhibit several local ITG reorderings, in addition to the non-ITG one(s). As shown in Figure 4, the complexity measures for train, test and missed reorderings are nearly identical. Therefore, reorderings that are not captured by the rule-based approach cannot be characterized by their complexity. For small size reorderings, we observe the missed reordering to have a slightly lower ITG ratio, a phenomena that quickly vanishes, as in fact almost all large test reorderings are missed.

Figure 5 displays the number of missed reorderings on the test data as a function of the size. We also vary the tagsets used to build the reordering rules.[19] For English-French, most reorderings concern moderate size spans (2 to 4) and half of the reorderings spread over only two words. Long range reorderings (i.e. more than ten word) are rare: if some correspond to genuine linguistic patterns, such as the alternation between active and passive voice, most of these permutations are due to alignment errors, mistranlations or complex constructions. In contrast, for English-Czech and even more so for English-German, the rule length is more evenly spread, with many medium size (5-10) reorderings as well a significant number of long range reorderings, which cannot be fully attributed to alignment errors. These results are in line with the numbers in Table 1, which suggest that French-English is the easiest pair (having the shortest average reordering length), and that German-English is the hardest (with longer reorderings and fewer monotonous alignments), Czech-English being in-between with a large number of monotone reorderings, yet a much larger average reordering length than French.

Figure 5 enables to distinguish three types of reorderings, the proportion of which varies depending the language pairs:

- short range reorderings, corresponding to permutations involving 2 to 4 words; these are accurately captured by the rule-based approach and the number of misses is accordingly very small; in this case, it makes sense to use a syntactic context to make sure they fire only in likely positions.

[19] The case where no rule is applied gives the number of reorderings needed on the test, data broken down by reordering size.



(a) *en* → *de* (b) *en* → *cs*

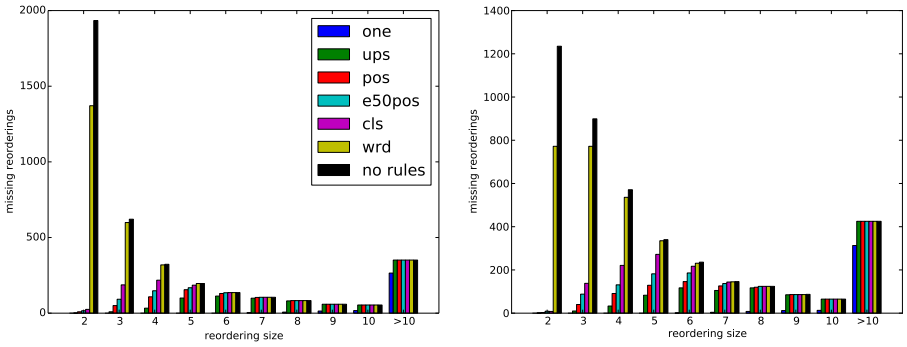
Figure 4: *Complexity of reorderings, as a function of the reordering size, in the training data (train), in the test data (test); (miss) denotes the reorderings in the test data but are not be captured by any rule). Complexity is measured by the percentage of ITG reorderings (itg), and the average normalized Kendall's τ (kendall) and fragmentation chunk (chunk) distance to the identity permutation. The plots for the four other language directions where similar to en → de.*

- medium-size reorderings (permutations of 5-10 words): most of the test situations are observed in training (as acknowledged by the small number of misses when we only use one single POS tag); the rule-based approach is less successful here, and many misses are observed when syntactic constraints are introduced;
- long-range reorderings (involving more than 10 words): most of these are missed, even with the most general tagset. These means that most of these test permutations are not seen in training and suggests that learning such long permutations is useless.

Figure 5 also gives some insights regarding the generalization power of the tagsets introduced in Section 3.3. With fully lexicalized rules, the coverage of test reorderings is rather poor: for instance, more than half of the swaps are missed, for all translation directions. We also note that class-based rules are always worst than rules based on linguistic tags. Additionally, for reordering size greater than 5, whatever the tagset, half of the test reordering are missed. This means that the reordering rules, as used in this work, are only useful for very small range reorderings. Differences between tagsets thus mainly impact such reorderings and enable to vary the trade-off between coverage and ambiguity.

5.3 From Monotone to Rule-Based Reordering: Impact on MT Performance

In this section, we assess the impact of the rule-based approach in terms of MT performance. For this purpose, Table 2 reports BLEU scores on test data for several reordering spaces of varying "quality". Additional figures for BEER and RIBES metrics are provided in Appendix A (see Table 5). Note that results obtained with those metrics are consistent with our observations based on BLEU scores throughout. The first reordering space only considers the original source sentence order (monotone). The



(a) $en \rightarrow fr$ (b) $fr \rightarrow en$

(c) $en \rightarrow de$ (d) $en \rightarrow cs$

Figure 5: *Number of missed reordering on the test data as a function of the reordering size when using rules built over different tagsets (see text for details; no filtering is used here, i.e maxcost* $= \infty$*) or without any rule. Plots for de* $\rightarrow$ *en and cs* $\rightarrow$ *en are similar to the ones for en* $\rightarrow$ *de and en* $\rightarrow$ *cs, respectively.*

	tun.	dec.	$en \rightarrow fr$	$fr \rightarrow en$	$en \rightarrow de$	$de \rightarrow en$	$en \rightarrow cs$	$cs \rightarrow en$
NCODE	*rules*	*mono*	19.1*$_{\pm 0.0}$	19.8*$_{\pm 0.0}$	12.3*$_{\pm 0.0}$	17.0*$_{\pm 0.0}$	9.9$_{\pm 0.0}$	14.7*$_{\pm 0.0}$
	rules	*rules*	22.2$_{\pm 0.0}$	21.9$_{\pm 0.0}$	12.7$_{\pm 0.0}$	18.1$_{\pm 0.0}$	9.9$_{\pm 0.0}$	14.8$_{\pm 0.0}$
	rules	*best*	22.8*$_{\pm 0.0}$	24.2*$_{\pm 0.0}$	13.5*$_{\pm 0.0}$	19.0*$_{\pm 0.0}$	10.2*$_{\pm 0.0}$	15.3*$_{\pm 0.0}$
	rules	*unfo*	24.0*$_{\pm 0.0}$	25.8*$_{\pm 0.0}$	15.6*$_{\pm 0.0}$	22.1*$_{\pm 0.0}$	10.9*$_{\pm 0.1}$	16.4*$_{\pm 0.1}$
	rules	*aug*	22.3$_{\pm 0.0}$	21.9*$_{\pm 0.0}$	12.8$_{\pm 0.0}$	18.6*$_{\pm 0.1}$	9.9$_{\pm 0.0}$	14.8*$_{\pm 0.0}$
	rules	*duel*	23.1*$_{\pm 0.0}$	24.5*$_{\pm 0.0}$	14.1*$_{\pm 0.0}$	20.2*$_{\pm 0.0}$	10.3*$_{\pm 0.0}$	15.5*$_{\pm 0.0}$
Oracle		*mono*	47.0	50.0	37.8	43.0	30.1	36.2
		rules	54.2	57.5	42.8	49.0	33.1	40.2
		best	52.5	56.2	40.7	46.8	31.7	38.0
		unfo	54.8	59.2	45.0	52.4	34.7	40.4
		aug	56.0	59.9	46.1	53.4	35.5	42.0
		duel	54.9	59.3	45.2	52.6	35.0	40.8
NCODE	*aug*	*aug*	22.3$_{\pm 0.0}$	22.0*$_{\pm 0.0}$	13.7*$_{\pm 0.0}$	19.9*$_{\pm 0.1}$	10.1*$_{\pm 0.0}$	14.9*$_{\pm 0.0}$
	duel	*duel*	23.9*$_{\pm 0.0}$	25.6*$_{\pm 0.0}$	15.6*$_{\pm 0.0}$	21.9*$_{\pm 0.0}$	11.2*$_{\pm 0.0}$	16.6*$_{\pm 0.0}$
	aug	*rules*	22.2$_{\pm 0.0}$	22.0$_{\pm 0.0}$	12.1*$_{\pm 0.1}$	17.6*$_{\pm 0.2}$	10.0*$_{\pm 0.0}$	14.8$_{\pm 0.0}$

Table 2: *BLEU scores on test data obtained by* NCODE *systems and oracle decoding, when no reorderings are allowed (monotone (mono)); when using our lattice reordering space (rules); when given only the best lattice reordering (best); when given only the reference (unfolded) reordering (unfo); when adding the unfolded reordering to the lattice (aug) or to the best lattice reordering (duel) during tuning phase on development data (tun.) and/or when decoding the test (dec.). BLEU scores are averages across 3 runs of MIRA; standard deviation across runs are reported in script size. A statistical significance ($p < 0.005$) difference to the baseline (rules; rules) is indicated by the * symbol.*

second uses our rule-based approach to create a reordering lattice (rules). The remaining four are oracle-like reordering spaces that will be detailed in Sections 5.4 and 5.5. An example of sentence translation for these configurations is in Figure 6. Table 2 also gives the best possible BLEU scores (oracle decoding) for the six conditions.[20]

For English-French and English-German, we can observe BLEU improvements from monotone to lattice reordering, as one would expect. For English-French, the increase is as high as 3 BLEU points, which illustrates the importance of taking word moves into account during translation, even for closely related languages. In Figure 6 for example, the monotone translation fails to invert `president's` and `spokesman`, resulting in a mistranslation (meaning "by the president, spokesman, Radim Ochvat"). For English-German, gains are however much lower, especially for *en* $\rightarrow$ *de* (only about a half BLEU point). This suggests that our reordering system does not succeed in predicting the German word order. Finally and perhaps more surprisingly, there is no gain at all for English-Czech in neither direction, which indicates that our rule-base might not be particularly adapted for capturing ordering variation for this particular

[20]The oracle BLEU scores are significantly lower in this article than the ones published in Pécheux et al (2014). For the latter, the tuple tables were extracted from the test data to isolate the contribution of the reordering, while in this work tuple tables are derived from the training corpora, thus measuring the impact on the actual potential of the systems.

```
   src : the meeting was announced by the president's spokesman Radim Ochvat .
   ref : c' est le porte-parole présidentiel Radim Ochvat qui a informé de la réunion .
  mono : la réunion a été annoncée par le président porte-parole Radim Ochvat .
 rules : la réunion a été annoncée par le porte-parole du président Radim Ochvat .
  unfo : le porte-parole du président Radim a été annoncée par la réunion Ochvat .
oracle : de la réunion est le porte-parole présidentiel Radim Ochvat qui
```

Figure 6: *Translations of a source sentence* (src) *from* newstest2010*, along with the reference translation* (ref)*, contrasting monotone* (mono) *lattice-based* (rules) *and reference (unfolded) reordering* (unfo) *constraints, as well as oracle decoding* (oracle) *in the lattice reordering space. For this specific example, the best reordering and the augmented lattice constraints yielded the same translation as the lattice one (rules).*

language pair.[21] Two main explanations may account for this negative result. Either the reordering mechanism is not expressive enough to generate good reordering variants in the search space, or the model is not able to recognize these better paths. We will see in Section 5.7 (Table 3) that for only about 30-40% percent of the sentences, the correct order is encoded in the lattice. However, oracle decoding shows that in all cases, even for the most challenging translation directions, the reordering lattice contains much better reorderings than the monotonic order which could be exploited to achieve better BLEU scores. This suggests that model and/or search errors are largely responsible for the lack of improvement observed when moving from a monotone to an enriched reordering space.

Table 2 also shows that BLEU scores for $en \rightarrow de$, and even more so for $en \rightarrow cs$, are much worse than in the other translation direction. Assuming that the reordering complexity is more or less symmetric, the difference here may be due to the complex morphology of German and Czech, which is difficult to generate when translating from English.

The high oracle BLEU scores in Table 2 finally suggest that larger gains may be achieved by improving the translation models than by increasing the size of the reordering space. Note however that oracle BLEU scores may be overly optimistic and a large part of these gains may be due to over-fitting the BLEU metric. An illustration is in Figure 6, with a mumbo-jumbo oracle translation.

5.4 Oracle Reorderings, an Upper Bound on MT Performance

To better understand the impact of model and search errors, we carry out additional experiments with two informed reordering spaces. Results are again in Table 2. The *best* configuration refers to the situation where the reordering space only contains the best(s) reordering as defined in Section 4.3, *unfo* denotes the forced unfolded reordering. Table 2 shows that for all translation directions, the prior knowledge of the best(s) reordering is actually useful, with particularly large gains for French to English. The gap between *unfo* and *aug* and the corresponding oracle conditions indicates that model and search errors cause the system to often miss good reordering paths in the lattice, and reveals quantitatively the impact of these errors in decoding.

As also noted by Herrmann et al (2013b), the best reordering in the lattice is not necessarily the one leading to the best translation. In fact, alignment errors may yield

[21] Surprisingly, Czech which is usually described as a free-order language, has both the highest number of monotone alignments to English, as well as a significant number of non-local reorderings: this makes the identification of good reorderings especially challenging.

low quality unfolded reorderings, which will then affect the best reordering approximation. Moreover, the resulting word order is somewhat artificial and may not correspond to an optimal matching of phrase pairs.[22] Oracle decoding gives a quantitative illustration as oracle decoding with best(s) reordering is only half way between monotone and lattice-based oracle decoding. This means that, in many cases, an even better reordering would yield a larger improvement. On the other hand, giving hints about a given reference translation (here information about its order) biases the system towards the order of that reference. So the oracle-like reordering results may be slightly more optimistic than the actual performance a real system could ever achieve.

Previous experiments have delivered an upper bound of the performance for a constrained reordering space. It is also interesting to contrast the best *achievable* reordering with best *conceivable* one, as this gives information about the quality of the approximation of the reordering by the generation mechanism (Herrmann et al, 2013b). In addition, this also gives an upper bound for any reordering mechanism that would be used to generate the reordering space. As shown in Table 2, all language directions benefit from knowing the unfolded reordering, sometimes by a wide margin, with however some disparities between language directions. The difference between the best reordering and the unfolded one measures the improvements that could be obtained by relaxing the reordering constraints (assuming no model/search errors), while the difference between the lattice and the unfolded cases measures the improvements that could be obtained by designing a better reordering space *and* a better model score function.

The observed gaps in performance, up to 4 BLEU points when translating into English from French or German shows that there is indeed room for improvement. The improvements for *en* $\rightarrow$ *cs* are however not so clear. In other words, "solving" the reordering problem at decoding time has only a slight effect on performance for this language direction. Therefore, the reordering constraints might currently not be the main limitation of our system for this language pair. Note finally that the reordering length is unbounded in the unfolded reordering case, hence the lack of long range reorderings in our model can not be the main explanation.

5.5 Discrimination of the Unfolded Reordering

In this section, we design two additional experiments where we simulate a "competition" between some of the previous lattices.

In the first one, the rule-based reordering space is augmented with the unfolded reordering (line denoted by (*aug*) in Table 2). In this setting, the reordering space now contains the "expected" reordering, as well as many alternative permuations. This experiment allows us to further understand how well the decoding system is able to find the unfolded reordering in the lattice. Table 2 shows that there is almost no difference with regular lattice decoding, except for *de* $\rightarrow$ *en*. Therefore, it seems that one of the main issue with reordering is not the lack of good reorderings in the search space, but rather the failure to select these permutations, due to models and/or search errors.

It should be noted that the unfolded reordering does not always result in a better translation. For instance, in Figure 6, the lattice translation is valid though different from the reference. However, the translation in the unfolded reordering condition is mistranslated as the initial sentence is in the passive voice (the translated sentence means "President Radim's spokesman has been announced by the Ochvat meeting").

[22] This effect is probably small for NCODE, as the phrase pairs (tuples) are minimal.

This illustrates the situation where a seemingly optimal reordering leads to a poor translation. Oracle decoding for the augmented lattice gives a more quantitative analysis: results in Table 2 for oracle decoding with the augmented lattice are always superior to the ones obtained with just the unfolded reordering. This means that in many cases, the search space contains a better reordering than the unfolded one.

The second experiment is a face-to-face comparison of the unfolded and the *best* reorderings (see line *duel* in Table 2). This experiment aims at understanding whether the decoder could be able to choose the unfolded reordering in the absence the ambiguity introduced by many competing spurious reorderings. For each source sentence, a *duel* permutation lattice is built containing only the unfolded and the best reorderings. Table 2 reveals that for English-French and English-Czech, the scores are only slightly better than the ones with the best reorderings, suggesting that the decoder would only choose the unfolded reordering if forced to. One explanation is that the unfolded permutation may contain long-range reorderings that are severely penalized by the distortion mode; this is nonetheless revealing of the pitfalls of the scoring function. The improvement is minimal for English-Czech, for which the scoring function seems to be the less appropriate. Oracle results also show that in some cases, the best reorderings might be easier to use than the unfolded ones, as the scores slightly increase.

Note that in this paper, we have not attempted to sort model from search errors. In future work, we plan to evaluate the oracle coverage using the lattices that are effectively explored during decoding (i.e. after beam search pruning), to better understand the relative contributions of these types of errors. However, our duel setting already suggests that model errors play an important role, as it is unlikely to observe many search errors when only two alternative reorderings are competing.

5.6 Reordering Space when Tuning

We finally explore the importance of the reordering space during the tuning step. The bottom part of Table 2 shows that tuning, then decoding with the augmented lattice could yield some improvements; these are particularly significant for English-German. This means that the tuning benefits from seeing the unfolded reordering as a possible candidate. The effect is all the more important when tuning and decoding with the duel lattices, with an increase of about one BLEU point for all directions. We must then mitigate the conclusions of previous paragraphs: the decoder is indeed able to select the unfolded reordering, but has to be trained with lattices containing good non-monotone paths.[23]

Unfortunately, tuning with an augmented lattice, while decoding with the rule-based one, as is the case in real-world scenario, actually harms performance, as shown in the last line of Table 2.

5.7 Reordering Space Tradeoff

Table 3 reports reordering space size, coverage, oracle and decoding scores[24] when varying the rule filtering threshold. We observe that while the number of rules is almost twice as large for $en \rightarrow de$ than for $en \rightarrow fr$, the generated reordering spaces are comparable in size, but with a much lower coverage for $en \rightarrow de$. English-Czech has

[23] As hinted to above, the distortion model is largely responsible for this. Indeed, when tuning with the standard lattice the distortion penalty weight is 0.01, while it is -0.05 and -0.19 when tuning with the augmented and duel lattices, respectively, *encouraging* to deviate from monotonic translation.

[24] Results obtained with BEER and RIBES are in Appendix A, Table 6.

the smallest reordering space; yet the rule coverage is higher than for English-German. Interestingly, when compared with English-German, English-Czech has almost twice as many monotonic translations;[25] yet, we also observe much lower BLEU scores both when using NCODE or oracle decoding, indicating again that reordering is only part of the complexity of this language pair.

By relaxing the rules pruning, we see large increases in the size of the reordering space, in coverage and in oracle BLEU. For English-French, in regular test condition, we observe a slight degradation of the BLEU scores, even though the size of the search space drastically increases. This shows the importance of the trade-off regarding the design of the reordering space, as noisy reorderings may introduce spurious, but plausible, alternatives. This is however not the case for the other language pairs, for which the BLEU scores do not change much as the reordering space increases, and with it the unfolded reordering coverage. Again, $en \rightarrow cs$ is the most challenging translation pair, where the best performance are obtained for the monotone condition, reflecting again the inability of our systems to take advantage of a richer search space.

Table 3 also displays the average Kendall's τ distance from NCODE hypotheses to the unfolded permutations as well as the number of times the unfolded reordering is actually used by the decoder (which corresponds to a null value for Kendall's τ). For English-German and English-Czech, we observed that the unfolded coverage of the search space increases when relaxing the filtering strategy; this is however not reflected in the final reordering chosen by the decoder, which remains the same, and even slightly decreases (for maxcost= ∞). Improving the reordering search space does not seem to benefit the decoder in making better reordering decisions for those language pairs.

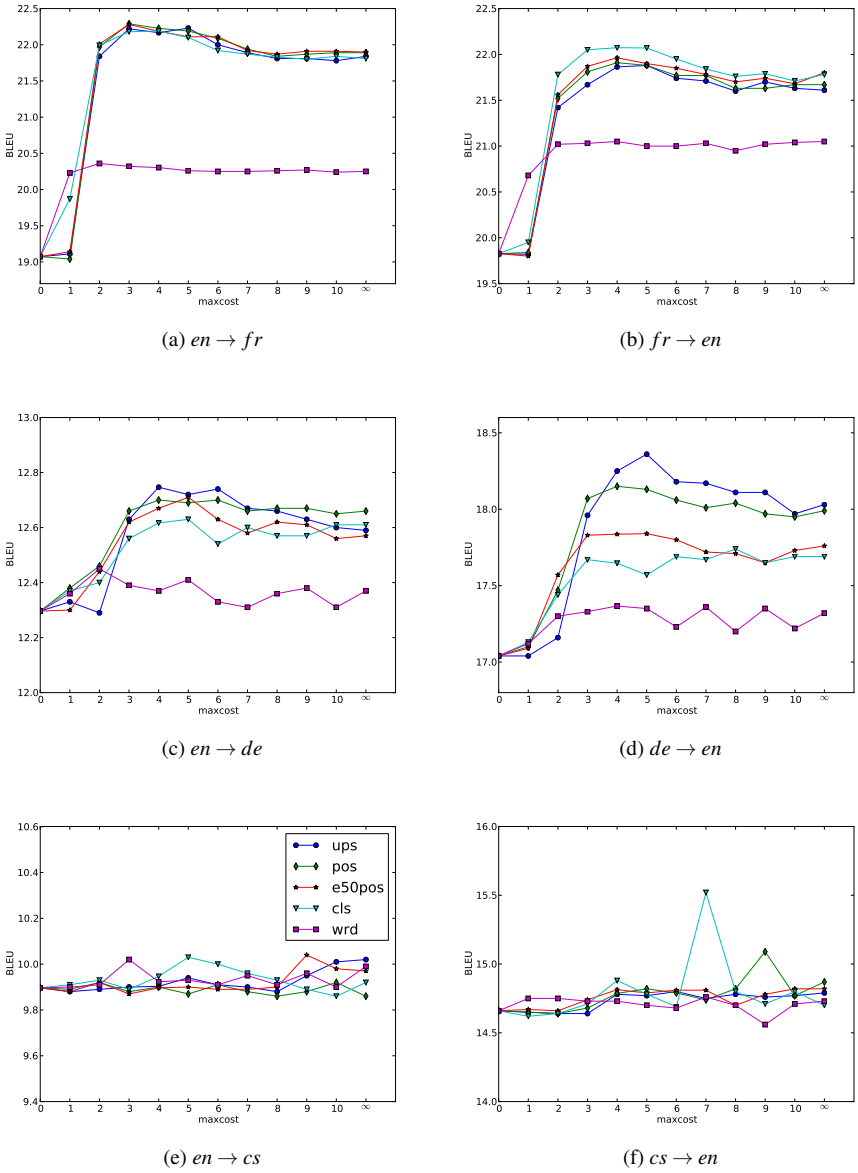
It might be surprising, at a first glance, to see that using low cost rules does have a small effect on the reordering space. For instance, for $en \rightarrow de$, using a threshold of 2 selects 101 K rules, which however generate lattices that have only two paths on average. Similarly, the 53 K $en \rightarrow cs$ rules do not significantly increase the number of paths with respect to the monotone condition. This shows that most of the extracted reordering rules do not generalize to the test data, mainly because the corresponding tag sequences are observed too rarely. However, even useless, they are not harmful as they never fire.

5.8 Using Alternative Tagsets

Figure 7 displays the results when building the reordering rules over different tagsets and when using different filtering thresholds. It is interesting to see different behaviors between languages pairs, irrespective of the translation direction. For both English-French and English-German, using fully lexicalized rules, as predicted in Section 5.2, performs significantly worst, with improvement however over the monotonic case. For English-French, we see little differences between the other tags, and, as previously noted, larger reordering spaces, corresponding to higher filtering thresholds slightly degrade the performance. It is surprising to see that for $fr \rightarrow en$, the rules built on the word classes slightly outperform the other tags, as this was not predictable from Figure 5 (a). Word classes however do not perform so well for English-German for which differences between tagsets are larger. In this case, smaller tagsets seem to perform better, probably because they enable a better generalization.

As previously observed, the case of $en \rightarrow cs$ is peculiar, as we do not observe any quantifiable score change when varying the parameters. Note, however, that this result

[25] Which also explains the high coverage.



(a) $en \rightarrow fr$ (b) $fr \rightarrow en$

(c) $en \rightarrow de$ (d) $de \rightarrow en$

(e) $en \rightarrow cs$ (f) $cs \rightarrow en$

Figure 7: *Comparison of the performance (BLEU score, single MIRA run) of different tagsets on the test data as a function of the filtering threshold (maxcost).*

rules out a possible explanation that the $cs \rightarrow en$ direction would be penalized by a large POS tagset, as when using the smaller Universal tagset no changes is observed. Finally, the results for $cs \rightarrow en$ show a surprising outlier when using word classes and a very specific threshold parameter, with about one BLEU point improvement with respect to other conditions.[26]

[26]The most likely explanation being a "lucky" tuning, as we did not average results across multiple runs.

In general, the competitive results obtained with the coarse-grain tagset and the automatic word classes show that they can be used as a workaround for under-resourced language, as for a new language pair, one would not been able to predict which tagset should be used anyways.

5.9 Comparison with MJ-*i*

Table 4 finally reports the results[27] of a head-to-head comparison between MJ-*i* constraints and the rule-based approach. The MJ reordering spaces are several orders of magnitude larger than their ruled-based counterparts but yield the same or significantly lower results. This warrants the use of linguistically motivated rules, instead of allowing all local permutations, and corroborates the trade-off discussed earlier. Training time is also an issue here: for $en \rightarrow fr$, the tuning step with MJ-3 constraints is twenty times longer than $maxlen = 4$.

6 Related Work

The reordering problem has been addressed in many ways since the advent of machine translation. Researchers tried to solve this problem via new approaches, varying the modeling strategies, or by restricting the possible word reordering operations. In this work, we are interested in how the *reordering space* is defined, either explicitly or implicitly, and in methods that could help to understand the importance and the impact of reordering space design.

Early work on word reordering constraints includes ITG constraints (Wu, 1997) and IBM constraints (Berger et al, 1996), which are compared in Zens and Ney (2003). Goh et al (2011) partition sentences into several clauses and restrict word reordering to occur within clauses. The definition of the reordering space is also closely related to the generative mechanisms used in SMT. In phrase-based SMT (Zens et al, 2002) local reorderings are modeled within phrases, which may then be reordered according to some constraints, e.g. a simple distortion limit on words. Other constraints on phrases include ITG constraints (Zens et al, 2004; Feng et al, 2010; Cherry et al, 2012) and MJ constraints (Kumar and Byrne, 2005).[28] Syntax-based MT systems handle the reordering problem by embedding syntactic analysis in the decoding process (Wu, 1997; Yamada and Knight, 2001; Galley et al, 2004). Finally, the hierarchical approach or Chiang (2005) is mainly motivated by the recursive nature of reorderings. However, Auli et al (2009) showed that the search space explored by phrase-based and hierarchical-based models are very close. All these approaches generally fail to handle long range reorderings, hence the motivation of approaches that rearrange the source sentence in a target-like word order before translating and that handle reorderings at the word level, as ours.

This line of work has been pioneered by Xia and McCord (2004), who automatically learn reordering rules from source and target language dependency trees. Many subsequent approaches have proposed to manually design reordering rules based on syntactic or dependency parse trees (Collins et al, 2005; Xu et al, 2009; Carpuat et al, 2010; Isozaki et al, 2010b), or to automatically learn them (Zhang et al, 2007; Li et al, 2007; Khalilov et al, 2009; Elming and Habash, 2009; Genzel, 2010; Dyer and Resnik, 2010; Khalilov and Sima'an, 2011; Lerner and Petrov, 2013). As source parse trees are

[27] See Table 7 in Appendix A for BEER and RIBES metrics.

[28] Note that in this work we consider MJ constraints on *words* instead of on *phrases*.

not always available, other approaches cast the preordering directly as a permutation modeling problem (Tromble and Eisner, 2009; Visweswariah et al, 2011) or infer the parse trees automatically from parallel text (DeNero and Uszkoreit, 2011; Neubig et al, 2012). Note that these approaches require high-quality manual word alignments. However, Visweswariah et al (2013) propose an approach that jointly improves alignment and reordering in the presence of noisy alignments.

Another widely-used approach is to automatically learn shallow reordering rules based on POS tags or syntactic chunks (Rottmann and Vogel, 2007; Zhang et al, 2007; Crego and Habash, 2008; Niehues and Kolss, 2009). Herrmann et al (2013a) further combine POS based reordering on the morphosyntactic level and syntax tree-based on the constituent level. Alternatively, Costa-jussà and Fonollosa (2006) cast the word reordering problem as a translation task, using word class information to translate the original source sentence into the reordered source sentence.

In some cases, pre-ordering fails to improve translation performance (Howlett and Dras, 2010). These authors investigate in detail several factors to understand when preordering may be useful; one reason being that it enables to better match the inner mechanism of phrase-based SMT (Zwarts and Dras, 2006) .

In the majority of previous works, only one deterministic preordering of the source sentence is computed: this is because preordering is used in a preprocessing step, which is then followed by the whole translation pipeline, inducing further reorderings. In contrast, in our approach, all the possible reorderings are computed once and for all in the reordering step, whose single goal is to *generate* the reordering space; the selection of the best reordering path is then left to the decoder.

Bisazza and Federico (2013b) claim that long-range reorderings issues should not be attributed to the deficiencies of existing reordering *models*, but rather to too coarse definition of the reordering *search space*. They introduce a word after word reordering similar to the preordering model of Visweswariah et al (2011) to *dynamically shape* the search space while decoding with a very high distortion limit (Bisazza and Federico, 2013a). This approach enables to achieve fast decoding and performance improvements for the reordering of verbs in Arabic to English translation.

Oracle experiments are a valuable method for analyzing different aspects of machine translation, e.g. identify translation errors in the phrase-table (Wisniewski et al, 2010), or to perform failure analysis (Wisniewski and Yvon, 2013). Sokolov et al (2012) describe efficient methods to find the best translation hypothesis in a lattice and apply them to compare the lattices explored by MOSES and NCODE. In this work, we compute the oracle on the *full* search lattice. Another line of study is to assess the limitations induced by various reordering constraints. Dreyer et al (2007) compute a lower bounds of the best achievable BLEU score using dynamic programming techniques for IBM and ITG constraints, while Sokolov et al (2012) show a very limited influence of the distortion limit both on the decoder and on the oracle quality. Wisniewski and Yvon (2013) study in details various reordering constraints, including the distortion limit, IBM and MJ-*i* constraints. We share the main conclusions of these studies: the scoring functions (or models) seem to be the main limitation for phrase-based systems, while they are expressive enough to achieve higher translation performance. As for oracle-like reorderings, Khalilov and Sima'an (2012) introduce an upper bound, similar to our unfolded reordering, and show potential improvement for preordering performance, albeit limited when considering tree structure constraints.

Auli et al (2009) explore *induction errors* in the search space of phrase-based and hierarchical phrase-based model (HPBT), and promote the use of reference reachability metric, which corresponds to our notion of coverage. They only consider different

reordering spaces by varying the distortion limit and show that both types of model explore almost similar search spaces, and mostly differ by the way they score derivations. In contrast to previous work, we make use of many complementary approaches to assess the importance of the reordering space, by studying and comparing jointly the actual performance, the oracle best possible performance and the search space size, both for rule-lattice based and for oracle reordering spaces.

The most similar work to ours is certainly the study of Herrmann et al (2013b), which also contains oracle experiments aimed to analyze the potential of the preordering approach and the impact of various restriction of the reordering space. Based on oracle results for the English-German pair that are in line with our own findings, the authors suggest that closing the observed gap between the unfolded and rule-based reorderings could yield significant improvements in performance. Based on our experiments, notably the comparison between lattice and augmented reordering spaces (Section 5.5), we are enclined to mitigate these conclusions: in fact, little gains will be obtained from such endeavours unless decisive progresses are made to reordering models.

Note finally that this study extends our own previous work (Pécheux et al, 2014) in several ways: (a) experiments on English-Czech, a challenging translation language pair; (b) a detailed study of the rule-based approach and its efficiency; (c) additional oracle-like conditions which shed light on the importance of model/search errors.[29]

7 Conclusions

In this work, we have compared the search space generated by different reordering rules as well as local permutation constraints. Linguistically motivated reordering rules lead to a much smaller search space and improve the translation quality, and only moderately depends on the abstraction used to generalize rules beyond purely lexical patterns. However, this simple rule-based approach is only effective for small range reorderings, and other techniques would be needed to generate more accurate reordering spaces, in particular for English-Czech. To assess the potential of a better reordering search space, we use a n-gram SMT tool that decorrelates reordering and decoding; but our results are more general and hold for any system for which the reordering space could be encoded in a lattice prior to decoding. This framework allows us to specifically study the the impact of the reordering space on the overall translation performance. We find that there is a large room of improvement by designing a better reordering space. This improvement is however less substantial for English to Czech, the most difficult translation direction, suggesting that the reordering search space is not the only critical issue in a system design; for this particular language pair, the complexity of Czech morphology also contributes to make to make SMT very challenging. However, there is little hope to generate reordering spaces composed of solely a few good reordering candidates. We showed that because of model/search errors, simply adding a good reordering in the search space would not be enough. Therefore, improving the reordering space should come with improvements on the reordering models if one wants to expect some gains.

It is worth mentioning that, in this work, we aim to understand the importance and the expressiveness of the *reordering decoding search space*, all other things being

[29] Notwithstanding various minor changes such as the use of MIRA instead of MERT and a different BLEU evaluation script, which explains that the figures presented here do not exactly match those of (Pécheux et al, 2014).

equal. Word order differences between languages intervene however at many other levels in a statistical machine translation system. The importance of the reordering search space has to be conditioned on the fact we are using a (particular) phrase-based approach relying on alignment links. What we claim is that currently, at least for the systems and language pairs studied in this work, the main sources of errors, even from the reordering point of view, can not be attributed to the decoding search space design. This does not negate that word ordering issues might still play a critical role, in particular, it presides over tuple extraction, which is at the root of phrase-based approaches like ours. In addition, our system is plagued by alignment error which intervene at various levels, affecting the system performance as well as our analysis. We plan to further study the impact of word alignment noise on the reorderings.

8 Acknowledgments

We would like to thank Thomas Lavergne and Guillaume Wisniewski for their feedback and for providing the oracle and semiring frameworks used in this work. We would also like to thank our anonymous reviewers for their comments and suggestions.

References

Allauzen A, Crego JM, Durgar El-Kahlout I, Yvon F (2010) LIMSI's statistical translation systems for WMT'10. In: Proceedings of the Joint Workshop on Statistical Machine Translation and Metrics, Uppsala, Sweden, pp 54–59

Allauzen A, Pécheux N, Do QK, Dinarelli M, Lavergne T, Max A, Le HS, Yvon F (2013) LIMSI @ WMT13. In: Proceedings of the Workshop on Statistical Machine Translation, Sofia, Bulgaria, WMT, pp 62–69

Auli M, Lopez A, Hoang H, Koehn P (2009) A systematic analysis of translation model search spaces. In: Proceedings of the Fourth Workshop on Statistical Machine Translation, Association for Computational Linguistics, StatMT, pp 224–232

Banerjee S, Lavie A (2005) METEOR: An automatic metric for MT evaluation with improved correlation with human judgments. In: Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization, Association for Computational Linguistics, Ann Arbor, Michigan, pp 65–72

Berger AL, Brown PF, Della Pietra SA, Della Pietra VJ, Kehler AS, Mercer RL (1996) Language translation apparatus and method using context-based translation models

Birch A (2011) Reordering metrics for statistical machine translation. PhD thesis, University of Edinburgh

Birch A, Osborne M (2010) Lrscore for evaluating lexical and reordering quality in mt. In: Proceedings of the Joint Workshop on Statistical Machine Translation and Metrics, Association for Computational Linguistics, Uppsala, Sweden, pp 327–332

Birch A, Osborne M, Koehn P (2008) Predicting success in machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Honolulu, USA, EMNLP

Birch A, Osborne M, Blunsom P (2010) Metrics for MT evaluation: Evaluating reordering. Machine Translation 24(1):15–26

Bisazza A, Federico M (2013a) Dynamically shaping the reordering search space of phrase-based statistical machine translation. Transactions of the Association of Computational Linguistics – Volume 1 pp 327–340

Bisazza A, Federico M (2013b) Efficient solutions for word reordering in German-English phrase-based statistical machine translation. In: Proceedings of the Workshop on Statistical Machine Translation, Association for Computational Linguistics, Sofia, Bulgaria, WMT, pp 440–451

Bojar O, Buck C, Callison-Burch C, Federmann C, Haddow B, Koehn P, Monz C, Post M, Soricut R, Specia L (2013) Findings of the 2013 Workshop on Statistical Machine Translation. In: Proceedings of the Workshop on Statistical Machine Translation, Sofia, Bulgaria, WMT, pp 1–44

Bojar O, Buck C, Federmann C, Haddow B, Koehn P, Leveling J, Monz C, Pecina P, Post M, Saint-Amand H, Soricut R, Specia L, Tamchyna A (2014) Findings of the 2014 workshop on statistical machine translation. In: Proceedings of the Workshop on Statistical Machine Translation, Association for Computational Linguistics, Baltimore, Maryland, USA, WMT, pp 12–58

Brown PF, deSouza PV, Mercer RL, Pietra VJD, Lai JC (1992) Class-based n-gram models of natural language. Computational Linguistic 18(4):467–479

Callison-Burch C, Koehn P, Monz C, Post M, Soricut R, Specia L (2012) Findings of the 2012 workshop on statistical machine translation. In: Proceedings of the Workshop on Statistical Machine Translation, Montréal, Canada, WMT, pp 10–51

Carpuat M, Marton Y, Habash N (2010) Improving Arabic-to-English statistical machine translation by reordering post-verbal subjects for alignment. In: Proceedings of the Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, ACL, pp 178–183

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

Chen SF, Goodman JT (1998) An empirical study of smoothing techniques for language modeling. Tech. Rep. TR-10-98, Computer Science Group, Harvard University

Cherry C, Foster G (2012) Batch tuning strategies for statistical machine translation. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, Montréal, Canada, pp 427–436

Cherry C, Moore RC, Quirk C (2012) On hierarchical re-ordering and permutation parsing for phrase-based decoding. In: Proceedings of the Workshop on Statistical Machine Translation, Association for Computational Linguistics, Stroudsburg, PA, USA, WMT, pp 200–209

Chiang D (2005) A hierarchical phrase-based model for statistical machine translation. In: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, ACL, pp 263–270

Clark JH, Dyer C, Lavie A, Smith NA (2011) Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Association for Computational Linguistics, HLT, pp 176–181

Collins M, Koehn P, Kucerova I (2005) Clause restructuring for statistical machine translation. In: Proceedings of the Annual Meeting of the Association for Computational Linguistics, Ann Arbor, Michigan, ACL, pp 531–540

Costa-jussà MR, Fonollosa JAR (2006) Statistical machine reordering. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Sydney, Australia, EMNLP, pp 70–76

Crego JM (2008) Architecture and modeling for n-gram-based statistical machine translation. PhD thesis, Universitat Politècnica de Catalunya

Crego JM, Habash N (2008) Using shallow syntax information to improve word alignment and reordering for SMT. In: Proceedings of the Workshop on Statistical Machine Translation, Association for Computational Linguistics, Columbus, Ohio, WMT, pp 53–61

Crego JM, Mariño JB (2006) Improving statistical MT by coupling reordering and decoding. Machine Translation 20(3):199–215

Crego JM, Costa-jussà MR, Mariño JB, Fonollosa JAR (2005) Ngram-based versus phrasebased statistical machine translation. In: Proceedings of International Workshop on Spoken Language Translation, IWSLT, pp 177–184

Crego JM, Yvon F, Mariño JB (2011) N-code: an open-source bilingual N-gram SMT toolkit. Prague Bulletin of Mathematical Linguistics 96:49–58

Déchelotte D, Adda G, Allauzen A, Galibert O, Gauvain JL, Maynard H, Yvon F (2008) LIMSI's statistical translation systems for WMT'08. In: Proceedings of the NAACL-HTL Statistical Machine Translation Workshop, Columbus, Ohio

DeNero J, Uszkoreit J (2011) Inducing sentence structure from parallel corpora for reordering. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Edinburgh, Scotland, UK., EMNLP, pp 193–203

Deza M, Huang T (1998) Metrics on permutations, a survey. Journal of Combinatorics, Information and System Sciences

Dreyer M, Hall KB, Khudanpur SP (2007) Comparing reordering constraints for SMT using efficient BLEU oracle computation. In: Proceedings of th Workshop on Syntax and Structure in Statistical Translation, Rochester, New York, NAACL-HLT, pp 103–110

Durgar El-Kahlout I, Yvon F (2010) The pay-offs of preprocessing for German-English statistical machine translation. In: Federico M, Lane I, Paul M, Yvon F (eds) Proceedings of International Workshop on Spoken Language Translation, IWSLT, pp 251–258

Durrani N, Koehn P, Schmid H, Fraser A (2014) Investigating the usefulness of generalized word representations in SMT. In: Proceedings of the 25th International Conference on Computational Linguistics, Dublin, Ireland, COLING, pp 421–432

Dyer C, Resnik P (2010) Context-free reordering, finite-state translation. In: Proceedings of the Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, NAACL, pp 858–866

Elming J, Habash N (2009) Syntactic reordering for English-Arabic phrase-based machine translation. Proceedings of the EACL Workshop on Computational Approaches to Semitic Languages p 69

Feng Y, Mi H, Liu Y, Liu Q (2010) An efficient shift-reduce decoding algorithm for phrased-based machine translation. In: Proceedings of the International Conference on Computational Linguistics, Association for Computational Linguistics, Stroudsburg, PA, USA, COLING, pp 285–293

Galley M, Manning CD (2008) A simple and effective hierarchical phrase reordering model. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, EMNLP, pp 848–856

Galley M, Hopkins M, Knight K, Marcu D (2004) What's in a translation rule? In: Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics, HLT-NAACL

Genzel D (2010) Automatically learning source-side reordering rules for large scale machine translation. In: Proceedings of the International Conference on Computational Linguistics, Association for Computational Linguistics, COLING, pp 376–384

de Gispert A, Mariño JB (2006) Linguistic tuple segmentation in n-gram-based statistical machine translation. In: Proceedings of the European Conference on Speech Communication and Technology, ISCA, INTERSPEECH

Goh CL, Onishi T, Sumita E (2011) Rule-based reordering constraints for phrase-based SMT. In: Proceedings of the Conference of the European Association for Machine Translation, pp 113–120

Herrmann T, Niehues J, Waibel A (2013a) Combining word reordering methods on different linguistic abstraction levels for statistical machine translation. In: Proceedings of the Seventh Workshop on Syntax, Semantics and Structure in Statistical Translation, Association for Computational Linguistics, Atlanta, Georgia, pp 39–47

Herrmann T, Weiner J, Niehues J, Waibel A (2013b) Analyzing the potential of source sentence reordering in statistical machine translation. In: Proceedings of International Workshop on Spoken Language Translation, Heidelberg, Germany, IWSLT

Howlett S, Dras M (2010) Dual-path phrase-based statistical machine translation. In: Proceedings of the Australasian Language Technology Association Workshop, pp 32–40

Huang F, Pendus C (2013) Generalized reordering rules for improved SMT. In: Proceedings of the Annual Meeting on Association for Computational Linguistics, The Association for Computer Linguistics, Sofia, Bulgaria, ACL, vol 25380, pp 387–392

Isozaki H, Hirao T, Duh K, Sudoh K, Tsukada H (2010a) Automatic evaluation of translation quality for distant language pairs. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, EMNLP, pp 944–952

Isozaki H, Sudoh K, Tsukada H, Duh K (2010b) Head finalization: A simple reordering rule for sov languages. In: Proceedings of the Joint Workshop on Statistical Machine Translation and Metrics, Association for Computational Linguistics, WMT, pp 244–251

Kendall MG (1962) Rank Correlation Methods. Theory and applications of rank order-statistics, Hafner Pub. Co.

Khalilov M, Sima'an K (2011) Context-sensitive syntactic source-reordering by statistical transduction. In: Proceedings of the International Joint Conference on Natural Language Processing, Asian Federation of Natural Language Processing, CoNLL, pp 38–46

Khalilov M, Sima'an K (2012) Statistical translation after source reordering: Oracles, context-aware models, and empirical analysis. Natural Language Engineering 18:491–519

Khalilov M, Fonollosa JA, Dras M (2009) A new subtree-transfer approach to syntax-based reordering for statistical machine translation. In: Proceedings of the Annual Conference of the European Association for Machine Translation, EAMT, pp 198–204

Kumar S, Byrne W (2005) Local phrase reordering models for statistical machine translation. In: Proceedings of Human Language Technology Conference and Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Vancouver, British Columbia, Canada, EMNLP, pp 161–168

Lavergne T, Cappé O, Yvon F (2010) Practical very large scale CRFs. In: Proceedings the Annual Meeting of the Association for Computational Linguistics, Association for Computational Linguistics, ACL, pp 504–513

Lerner U, Petrov S (2013) Source-side classifier preordering for machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Seattle, Washington, USA, EMNLP, pp 513–523

Li CH, Li M, Zhang D, Li M, Zhou M, Guan Y (2007) A probabilistic approach to syntax-based reordering for statistical machine translation. In: Proceedings of the Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, ACL, pp 720–727

Li S, Graça JaV, Taskar B (2012) Wiki-ly supervised part-of-speech tagging. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, Korea, EMNLP-CoNLL, pp 1389–1398

Lopez A (2009) Translation as weighted deduction. In: Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009), Athens, Greece, pp 532–540

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

Neubig G, Watanabe T, Mori S (2012) Inducing a discriminative parser to optimize machine translation reordering. In: Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning, Association for Computational Linguistics, EMNLP-CoNLL, pp 843–853

Niehues J, Kolss M (2009) A POS-based model for long-range reorderings in SMT. In: Proceedings of the Workshop on Statistical Machine Translation, Association for Computational Linguistics, StatMT, pp 206–214

Och FJ, Ney H (2002) Discriminative training and maximum entropy models for statistical machine translation. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, Philadelphia, Pennsylvania, USA, ACL, pp 295–302

Papineni K, Roukos S, Ward T, Zhu WJ (2002) BLEU: a method for automatic evaluation of machine translation. In: Proceedings of the Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, ACL, pp 311–318

Petrov S, Das D, Mcdonald R (2012) A universal part-of-speech tagset. In: Proceedings of the International Conference on Language Resources and Evaluation, European Language Resources Association, Istanbul, Turkey, LREC

Pécheux N, Alexandre A, François Y (2014) Rule-based reordering spaces in statistical machine translation. In: Proceedings of the International Conference on Language Resources and Evaluationa, European Language Resources Association, Reykjavik, Iceland, LREC

Ramanathan A, Visweswariah K (2012) A study of word-classing for mt reordering. In: Proceedings of the International Conference on Language Resources and Evaluation, Istanbul, Turkey, LREC, pp 3971–3976

Rottmann K, Vogel S (2007) Word reordering in statistical machine translation with a pos-based distortion model. Proceedings of the International Conference on Theoretical and Methodological Issues in Machine Translation pp 171–180

Schmid H (1994) Probabilistic part-of-speech tagging using decision trees. In: Proceedings of International Conference on New Methods in Language Processing, Manchester, UK, pp 44–49

Sokolov A, Wisniewski G, Yvon F (2012) Computing lattice bleu oracle scores for machine translation. In: Proceedings of the Conference of the European Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, Avignon, France, EACL, pp 120–129

Stanojević M, Sima'an K (2014) Beer: Better evaluation as ranking. In: Proceedings of the Workshop on Statistical Machine Translation, Association for Computational Linguistics, Baltimore, Maryland, USA, WMT, pp 414–419

Straka M, Straková J (2013) Czech models (MorfFlex CZ + PDT) for MorphoDiTa

Straková J, Straka M, Hajič J (2014) Open-source tools for morphology, lemmatization, POS tagging and named entity recognition. In: Proceedings of Annual Meeting of the Association for Computational Linguistics: System Demonstrations, Association for Computational Linguistics, Baltimore, Maryland, pp 13–18

Täckström O, Das D, Petrov S, Mcdonald R, Nivre J (2013) Token and type constraints for cross-lingual part-of-speech tagging. Transactions of the Association for Computational Linguistics 1:1–12

Talbot D, Kazawa H, Ichikawa H, Katz-Brown J, Seno M, Och FJ (2011) A lightweight evaluation framework for machine translation reordering. In: Proceedings of the Workshop on Statistical Machine Translation, Association for Computational Linguistics, WMT, pp 12–21

Tillmann C (2004) A unigram orientation model for statistical machine translation. In: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics, NAACL, pp 101–104

Tromble R, Eisner J (2009) Learning linear ordering problems for better translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP, pp 1007–1016

Tromble R, Kumar S, Och F, Macherey W (2008) Lattice Minimum Bayes-Risk decoding for statistical machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Honolulu, Hawaii, EMNLP, pp 620–629

Visweswariah K, Rajkumar R, Gandhe A, Ramanathan A, Navratil J (2011) A word reordering model for improved machine translation. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Stroudsburg, PA, USA, EMNLP, pp 486–496

Visweswariah K, Khapra MM, Ramanathan A (2013) Cut the noise: Mutually reinforcing reordering and alignments for improved machine translation. In: Proceedings of the Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, Sofia, Bulgaria, ACL, pp 1275–1284

Wisniewski G, Yvon F (2013) Oracle decoding as a new way to analyze phrase-based machine translation. Machine Translation 28(2):1–24

Wisniewski G, Allauzen A, Yvon F (2010) Assessing phrase-based translation models with oracle decoding. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, EMNLP, pp 933–943

Wisniewski G, Pécheux N, Gahbiche-Braham S, Yvon F (2014) Cross-lingual part-of-speech tagging through ambiguous learning. In: Proceedings of the Conference on Empirical Methods in Natural Language Processing, Association for Computational Linguistics, Doha, Qatar, EMNLP, pp 1779–1785

Wu D (1997) Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. Computational Linguistics 23(3):377–403

Xia F, McCord M (2004) Improving a statistical mt system with automatically learned rewrite patterns. In: Proceedings of the International Conference on Computational Linguistics, Geneva, Switzerland, COLING, pp 508–514

Xu P, Kang J, Ringgaard M, Och F (2009) Using a dependency parser to improve SMT for subject-object-verb languages. In: Proceedings of Annual Conference of the North American Chapter of the Association for Computational Linguistics, Association for Computational Linguistics, NAACL, pp 245–253

Yamada K, Knight K (2001) A syntax-based statistical translation model. In: Proceedings of the Annual Meeting on Association for Computational Linguistics, Association for Computational Linguistics, ACL, pp 523–530

Zens R, Ney H (2003) A comparative study on reordering constraints in statistical machine translation. In: Proceedings of Annual Meeting of the Association for Computational Linguistics, ACL, pp 144–151

Zens R, Och FJ, Ney H (2002) Phrase-based statistical machine translation. In: Jarke M, Koehler J, Lakemeyer G (eds) Lecture Notes in Artificial Intelligence, Springer Verlag, LNAI, vol 2479, pp 18–32

Zens R, Ney H, Watanabe T, Sumita E (2004) Reordering constraints for phrase-based statistical machine translation. In: Proceedings of the International Conference on Computational Linguistics, Geneva, Switzerland, COLING, pp 205–211

Zhang Y, Zens R, Ney H (2007) Chunk-level reordering of source language sentences with automatically learned rules for statistical machine translation. In: Proceedings of the Workshop on Syntax and Structure in Statistical Translation, Association for Computational Linguistics, Rochester, New York, SSST-NAACL, pp 1–8

Zwarts S, Dras M (2006) This phrase-based SMT system is out of order: Generalised word reordering in machine translation. In: Proceedings of the Australasian Language Technology Workshop, pp 149–156

A Appendix

	maxcost	BLEU		#rules	size	cov. (%)	τ	reach. (%)
		ncode	oracle					
$en \rightarrow fr$	0	19.1*	47.0	0k	27 / 1	20	0.16	20
	2	22.0*	52.4	23k	34 / 45	41	0.15	25
	4	22.2	54.2	33k	52 / 10^5	51	0.15	25
	∞	21.9*	57.5	42k	10^2 / 10^{22}	62	0.15	23
$fr \rightarrow en$	0	19.8*	50.0	0k	30 / 1	17	0.17	17
	2	21.5*	53.6	20k	36 / 18	27	0.16	20
	4	21.9	57.5	32k	73 / 10^7	43	0.16	20
	∞	21.7*	56.6	50k	10^3 / 10^{27}	59	0.16	19
$en \rightarrow de$	0	12.3*	37.8	0k	27 / 1	17	0.22	17
	2	12.5*	38.6	64k	31 / 2	18	0.23	16
	4	12.7	42.8	87k	65 / 10^5	26	0.23	14
	∞	12.7	45.9	102k	10^2 / 10^{22}	33	0.23	15
$de \rightarrow en$	0	17.0*	43.0	0k	28 / 1	16	0.23	16
	2	17.5*	44.8	71k	34 / 3	19	0.23	16
	4	18.1	49.0	92k	68 / 10^5	26	0.22	16
	∞	18.0	50.0	105k	10^2 / 10^{26}	33	0.23	15
$en \rightarrow cs$	0	9.9	30.1	0k	27 / 1	27	0.17	27
	2	9.9	30.5	33k	29 / 1	27	0.17	27
	4	9.9	33.1	46k	55 / 10^3	34	0.17	26
	∞	9.9	34.8	57k	10^2 / 10^{21}	47	0.18	26
$cs \rightarrow en$	0	14.7	36.2	0k	23 / 1	29	0.18	29
	2	14.6	36.5	30k	25 / 1	29	0.18	28
	4	14.8	40.2	41k	52 / 10^4	39	0.18	28
	∞	14.9	44.0	51k	10^2 / 10^{21}	51	0.18	27

Table 3: *Impact of rule filtering strategy (maxcost), using POS tagset on the test set: BLEU scores obtained by* NCODE *system and oracle decoding; on the number of reordering rules (#rules); on the size of the lattice reordering space (averaged number of arcs / average number of paths); on the coverage (see Section 4.4) (cov.); on the average Kendall's τ distance from* NCODE *hypotheses to the reference unfolded permutations (τ); and on the percentage of times the reference (unfolded) reordering is reached by the decoder (reach.). BLEU scores are averages across 3 runs of MIRA. A statistical significance ($p < 0.005$) difference from the baseline (maxcost=4) is indicated by a *.*

	$en \rightarrow fr$		$fr \rightarrow en$		$en \rightarrow de$		$de \rightarrow en$		$en \rightarrow cs$		$cs \rightarrow en$	
	BLEU	size	BLEU	size	BLEU	size	BLEU	size	BLEU	size	BLEU	size
maxlen=2	21.8	10^2	21.4	10^2	12.5	77	17.2	7	9.9	7	14.7	67
MJ-1	21.5*	10^{14}	21.3*	10^{17}	12.5	10^{14}	17.2	10^{19}	9.9	10^{14}	14.7	10^{14}
maxlen=3	22.1	10^4	21.7	10^5	12.6	10^3	17.5	10^2	9.9	10^2	15.0	10^3
MJ-2	21.7*	10^{24}	21.5*	10^{28}	12.5*	10^{24}	17.3*	10^{31}	9.9	10^{24}	15.0	10^{24}
maxlen=4	22.3	10^4	21.9	10^6	12.6	10^4	17.7	10^4	9.9	10^3	14.8	10^4
MJ-3	21.7*	10^{30}	21.6*	10^{36}	12.5	10^{30}	17.5*	10^{40}	9.9	10^{30}	14.8	10^{30}

Table 4: *Comparison between rule-based reordering with a rule length limit (maxlen) and purely combinatorial MaxJump constraints (MJ-i). Reported BLEU scores are averages across 3 runs of MIRA. A statistical significance ($p < 0.005$) difference between $maxlen = i$ and MJ-$(i-1)$ is indicated by a *.*

	tun.	dec.	$en \rightarrow fr$	$fr \rightarrow en$	$en \rightarrow de$	$de \rightarrow en$	$en \rightarrow cs$	$cs \rightarrow en$
NCODE	*rules*	*mono*	12.5	11.1	7.6	9.1	14.9	7.5
	rules	*rules*	14.2	12.4	8.5	10.0	15.0	7.5
	rules	*best*	14.6	13.2	8.5	10.3	15.0	7.6
	rules	*unfo*	15.1	14.0	9.4	11.5	15.3	8.1
	rules	*aug*	14.2	12.4	8.6	10.3	14.9	7.5
	rules	*duel*	14.7	13.4	9.0	10.9	15.0	7.7
Oracle		*mono*	25.3	21.4	10.7	17.0	27.9	15.9
		rules	33.1	29.5	14.5	22.1	28.9	17.6
		best	31.7	28.3	13.4	20.5	28.4	16.9
		unfo	34.2	31.9	17.1	25.9	29.7	18.2
		aug	35.1	32.4	17.5	26.5	30.0	18.7
		duel	34.3	32.0	17.1	26.0	29.8	18.3
NCODE	*aug*	*aug*	14.3	12.5	9.2	10.8	15.0	7.5
	duel	*duel*	15.1	14.1	9.7	11.7	15.5	8.1
	aug	*rules*	14.2	12.4	8.8	10.0	15.0	7.5

(a) BEER evaluation metric

	tun.	dec.	$en \rightarrow fr$	$fr \rightarrow en$	$en \rightarrow de$	$de \rightarrow en$	$en \rightarrow cs$	$cs \rightarrow en$
NCODE	*rules*	*mono*	76.1	77.7	73.0	75.6	70.8	74.7
	rules	*rules*	77.6	78.7	73.2	76.2	70.8	74.8
	rules	*best*	78.1	79.9	73.8	77.3	71.1	75.7
	rules	*unfo*	80.2	82.2	76.6	80.7	72.5	77.7
	rules	*aug*	77.6	78.7	73.1	76.6	70.8	74.6
	rules	*duel*	78.2	80.1	74.2	78.3	71.2	75.9
Oracle		*mono*	88.1	88.0	85.2	85.7	81.4	82.9
		rules	89.9	89.4	86.2	86.7	82.2	83.9
		best	89.9	89.7	86.2	87.2	82.2	84.0
		unfo	92.2	92.2	89.2	90.5	85.0	86.6
		aug	92.0	92.0	88.8	90.0	84.8	86.3
		duel	92.1	92.2	89.0	90.3	85.0	86.5
NCODE	*aug*	*aug*	77.7	78.8	74.3	78.4	70.9	74.6
	duel	*duel*	80.1	82.0	76.6	80.6	73.0	77.7
	aug	*rules*	77.6	78.7	72.2	76.2	70.8	74.7

(b) RIBES evaluation metric

Table 5: *Metrics scores on test data obtained by* NCODE *systems and oracle decoding, when no reorderings are allowed (monotone (mono)); when using our lattice reordering space (rules); when given only the best lattice reordering (best); when given only the reference (unfolded) reordering (unfo); when adding the unfolded reordering to the lattice (aug) or to the best lattice reordering (duel) during tuning phase on development data (tun.) and/or when decoding the test (dec.).*

	maxcost	BEER		RIBES	
		ncode	oracle	ncode	oracle
$en \rightarrow fr$	0	12.5	25.3	76.1	88.1
	2	14.1	31.3	77.3	89.5
	4	14.2	33.1	77.6	89.9
	∞	14.1	34.7	77.4	89.3
$fr \rightarrow en$	0	11.1	21.4	77.7	88.0
	2	12.1	25.3	78.5	88.8
	4	12.4	29.5	78.7	89.4
	∞	12.3	32.2	78.5	88.7
$en \rightarrow de$	0	7.6	10.7	73.0	85.2
	2	8.2	11.2	73.0	85.4
	4	8.5	14.5	73.2	86.2
	∞	8.4	15.8	73.0	86.1
$de \rightarrow en$	0	9.1	17.0	75.6	85.7
	2	9.6	18.5	75.9	86.2
	4	10.0	22.1	76.2	86.7
	∞	9.8	24.1	76.1	86.1
$en \rightarrow cs$	0	14.9	27.9	70.8	81.4
	2	14.9	27.9	70.7	81.5
	4	15.0	28.9	70.8	82.2
	∞	15.0	29.1	70.7	82.2
$cs \rightarrow en$	0	7.5	15.9	74.7	82.9
	2	7.4	16.0	74.5	83.0
	4	7.5	17.6	74.8	83.9
	∞	7.5	18.9	74.7	84.3

Table 6: *BEER and RIBES scores obtained by* NCODE *system and oracle decoding when varying the rule filtering strategy (maxcost), using POS tagset on the test set.*

	$en \rightarrow fr$		$fr \rightarrow en$		$en \rightarrow de$		$de \rightarrow en$		$en \rightarrow cs$		$cs \rightarrow en$	
	BEER	RIBES	BEER	RIBES	BEER	RIBES	BEER	RIBES	BEER	RIBES	BEER	RIBES
maxlen=2	14.0	77.3	12.0	78.3	8.1	73.0	9.4	75.6	15.0	70.8	7.5	74.6
MJ-1	13.9	77.2	12.0	78.3	8.1	73.0	9.3	75.7	15.0	70.8	7.5	74.6
maxlen=3	14.2	77.5	12.2	78.5	8.3	72.9	9.6	75.7	14.9	70.7	7.5	74.9
MJ-2	14.0	77.3	12.1	78.3	8.3	73.1	9.5	75.6	14.9	70.8	7.5	74.8
maxlen=4	14.2	77.6	12.3	78.6	8.5	73.1	9.7	75.9	15.0	70.8	7.5	74.7
MJ-3	14.0	77.3	12.1	78.3	8.3	72.9	9.6	75.9	14.9	70.7	7.5	74.6

Table 7: *Comparison between rule-based reordering with a rule length limit (maxlen) and purely combinatorial MaxJump constraints (MJ-i) for BEER and RIBES metrics.*