A

Lattice BLEU Oracles in Machine Translation

ARTEM SOKOLOV, Institut für Computerlinguistik, Universität Heidelberg
and GUILLAUME WISNIEWSKI and FRANÇOIS YVON, Université Paris Sud & LIMSI–CNRS

The search space of Phrase-Based Statistical Machine Translation (PBSMT) systems can be represented as a directed acyclic graph (lattice). By exploring this search space, it is possible to analyze and understand the failures of PBSMT systems. Indeed, useful diagnoses can be obtained by computing the so-called *oracle* hypotheses, which are hypotheses in the search space that have the highest quality score. For standard SMT metrics, this problem is however NP-hard and can only be solved approximately. In this work, we present two new methods for efficiently computing BLEU oracles on lattices: the first one is based on a linear approximation of the corpus BLEU score and is solved using generic shortest distance algorithms; the second one relies on an Integer Linear Programming (ILP) formulation of the oracle decoding that incorporates count clipping constraints. It can either be solved directly using a standard ILP solver or using Lagrangian relaxation techniques. These new decoders are evaluated and compared with several alternatives from the literature for three language pairs, using lattices produced by two PBSMT systems.

Categories and Subject Descriptors: I.7.2 [**Computing Methodologies**]: Natural Language Processing—*Machine Translation*

General Terms: Languages, Experimentation, Performance

Additional Key Words and Phrases: BLEU, Integer Linear Programming, Lattices, Machine Translation, Oracle Decoding

ACM Reference Format:
Artem Sokolov, Guillaume Wisniewski and François Yvon. 2013. Lattice BLEU Oracles in Machine Translation *ACM Trans. Speech Lang. Process.* V, N, Article A (2013), 30 pages.
DOI: http://dx.doi.org/10.1145/0000000.0000000

1. INTRODUCTION

Phrase-Based Statistical Machine Translation (PBSMT) systems translate source sentences by solving a complex search problem. For computational reasons, search usually only considers a well-defined approximation of the complete translation space, as computing exactly the best achievable translation is known to be NP-hard, even under simplifying assumptions regarding the scoring function [Knight 1999]. In many decoders, an approximation of this search space can be represented either as a n*-best list* containing the n top-scoring hypotheses, or as a phrase or word graph (a *lattice*) which compactly encodes those hypotheses that have survived search space pruning. Lattices usually contain orders of magnitude more hypotheses than n-best lists and thus constitute much better approximations of the search space.

This work has been partially funded by OSEO under the Quaero program.
Author's addresses: A. Sokolov, Institut für Computerlinguistik, Universität Heidelberg, 69120 Heidelberg, Germany; G. Wisniewski, Université Paris Sud & LIMSI–CNRS, BP 133, 91403 Orsay CEDEX, France; F. Yvon, Université Paris Sud & LIMSI–CNRS, BP 133, 91403 Orsay CEDEX, France.

DOI: http://dx.doi.org/10.1145/0000000.0000000

Exploring the PBSMT search space or some approximation of it, is one of the few means to perform *diagnostic analysis* and to better understand the behavior of the system [Turchi et al. 2008; Auli et al. 2009]. It is, for instance, often the case that the actual reference (human) translation of the source sentence does not occur in the search space. There can be many reasons for this undesirable state of affairs: the scantiness of the translation model, the insufficient expressiveness of the reordering model, the use of inadequate scoring functions, or an excessive amount of pruning in search. It can also be because the reference is not literal enough. Differentiating between these cases and understanding the exact causes of failure can help improve existing translation systems.

Useful diagnoses can be obtained by looking at *oracle* hypotheses, which are hypotheses that have the highest quality score in the search space. Oracle hypotheses can be used to analyze cases of failure and to better understand the bottlenecks of existing translation systems [Wisniewski and Yvon 2013; Turchi et al. 2012; Wisniewski et al. 2010]. *Oracle decoding* has several other applications: for instance, as in [Liang et al. 2006; Chiang et al. 2008], it can used as a work-around to the problem of non-reachability of the reference in discriminative training of SMT systems, notwithstanding the warnings of [Chiang 2012], which discuss the risks of always updating towards the best BLEU oracle. Lattice reranking [Li and Khudanpur 2009], a promising way to improve MT systems, also relies on oracle decoding to build the training data for a reranking algorithm.

For sentence-level evaluation metrics, finding oracle hypotheses in n-best lists is a simple issue; however, solving this problem on lattices proves more challenging, as the large number of embedded hypotheses prevents the use of brute-force approaches. The problem is even more difficult when quality is measured with BLEU [Papineni et al. 2002], as is most commonly done in MT evaluation. This is because BLEU is a corpus-level metric which does not decompose over sentences. In fact, even for sentence-level approximations, the oracle decoding problem is known to be NP-hard [Leusch et al. 2008].

In this paper, we propose two original methods for efficiently finding approximate oracle hypotheses on lattices, originally introduced in a conference paper [Sokolov et al. 2012]. The first method is based on a linear approximation of the corpus BLEU which was originally designed for efficient Minimum Bayesian Risk decoding on lattices [Tromble et al. 2008]. The second one, based on Integer Linear Programming techniques, is an extension to lattices of a recent work on failure analysis for phrase-based decoders [Wisniewski et al. 2010; Wisniewski and Yvon 2013]. In this framework, two decoding strategies are considered: one relies on a generic ILP solver, and the other is based on dual Lagrangian relaxation techniques, thus dispensing with the use of an ILP solver.

Our contribution is also experimental: we empirically compare the quality of these oracles with several existing approaches, for three language pairs and using the lattice generation capacities of two publicly available, state-of-the-art phrase-based decoders: Moses [Koehn et al. 2007] and N-code [Crego et al. 2011]. Additionally, we investigate in detail the structure of the PBSMT search spaces with respect to the distribution of oracles inside them and show that they contain many oracle hypotheses of good quality. We also used oracle decoding to try to understand some of the current PBSMT limitations by analyzing the discriminative power of standard features of a PBSMT system and by studying the impact of several of its parameters.

The rest of this paper is organized as follows. In Section 2, we formally define the oracle decoding task, before recalling the formalism of finite-state automata on semirings, which is used to describe several algorithms. We then describe in Section 3 two existing approaches for solving this task, before detailing our new proposals in Section 4 and Section 5. We then report, in Section 6, evaluations of the existing and new oracles on different language pairs and explain, in Section 7, how oracle decoding can be applied to identify various causes of error in large-scale PBSMT systems.

2. PRELIMINARIES

2.1. Phrase-based Machine Translation

In a nutshell, Phrase-Based Statistical Machine Translation (PBSMT) systems compute the target translation $\mathbf{e}$ of a source sentence $\mathbf{f}$ by freely recombining, through concatenation, small translation units called 'phrases'.[1] These phrases are the product of a complex set of processes, starting with word-to-word alignment, and culminating with a phrase-scoring procedure, which assigns various numerical confidence scores to the extracted phrases (see e.g. [Koehn 2010] for a complete description).

Inference (decoding) in PBSMT systems relies on the modeling of $p(\mathbf{e}, \mathbf{a}|\mathbf{f})$, the probability of obtaining an alignment $\mathbf{a}$ of the target $\mathbf{e}$ given the source sentence $\mathbf{f}$. This probability is usually parameterized as a linear model, $p(\mathbf{e}, \mathbf{a}|\mathbf{f}) = Z(\mathbf{f})^{-1} \exp(\boldsymbol{\beta} \cdot \mathbf{h}(\mathbf{e}, \mathbf{a}, \mathbf{f}))$, where $\mathbf{h}(\mathbf{e}, \mathbf{a}, \mathbf{f})$ is a numerical vector of *feature functions* representing various properties of $\mathbf{f}$, of $\mathbf{e}$ and of their alignment $\mathbf{a}$, and β is a parameter vector. For such a model, the MAP decision rule selects $\tilde{\mathbf{e}}_{\mathbf{f}}$ as:

$$\tilde{\mathbf{e}}_{\mathbf{f}}(\boldsymbol{\beta}) = \operatorname*{arg\,max}_{\mathbf{a},\mathbf{e} \in E} p(\mathbf{a}, \mathbf{e}|\mathbf{f}) = \operatorname*{arg\,max}_{\mathbf{a},\mathbf{e} \in E} \boldsymbol{\beta} \cdot \mathbf{h}(\mathbf{e}, \mathbf{a}, \mathbf{f}) \qquad (1)$$

where E is the set of reachable translations/alignments. Each component β_i of $\boldsymbol{\beta}$ regulates the influence of feature $h_i(\mathbf{e}, \mathbf{a}, \mathbf{f})$. A crucial property of $\mathbf{h}(\mathbf{e}, \mathbf{a}, \mathbf{f})$ should be that its components can be decomposed (through summation) over the scores of the individual phrases that are used in the alignment of $\mathbf{e}$ and $\mathbf{f}$. This property is required to obtain a compact representation of the decoder search space, which can then be explored efficiently. This is notably the case of translation models scores, which are directly derived from the scores of the individual phrases; this is also the case of the target language model score, as well as of various functions evaluating the plausibility of phrase reorderings in translation.

The optimal value for β is computed so as to optimize the system overall performance on a suitable training set consisting of pairs of source and target sentences, a procedure referred to as Minimum Error Rate Training (MERT) [Och 2003]. We will come back to MERT in Section 7.3; for the rest of the discussion, it suffices to assume that the system has been properly trained and that the optimal parameters β have been found.

2.2. Oracle Decoding

We now assume that a *phrase-based* decoder is able to produce, for each sentence $\mathbf{f}$ in a source language, a *lattice* $L_{\mathbf{f}} = \langle Q, \Xi \rangle$, with $\# \{Q\}$ vertices (states) and $\# \{\Xi\}$ edges. Each edge carries a source phrase f_i, an associated translation in the target language e_i, as well as the (local) feature vector $\mathbf{h}_i$, which encodes local compatibility scores for the phrase pair made of f_i and e_i.

We further assume that $L_{\mathbf{f}}$ is a *word* lattice, meaning that each f_i and e_i comprises at most a single word[2] and that $L_{\mathbf{f}}$ contains a unique initial state q_I and a unique final state q_F. Let $\Pi_{\mathbf{f}}$ denote the set of all paths from q_I to q_F in $L_{\mathbf{f}}$. Each path $\boldsymbol{\pi} \in \Pi_{\mathbf{f}}$ corresponds to a possible translation $\mathbf{e}_{\boldsymbol{\pi}}$ obtained through the alignment defined by the transition labels. For this representation, the job of a (conventional) decoder is to find the best path(s) in $L_{\mathbf{f}}$ according to Equation (1): $\boldsymbol{\pi}^*_{decoder} = \arg\max_{\boldsymbol{\pi} \in \Pi_{\mathbf{f}}} \boldsymbol{\beta} \cdot \sum_{i \in \boldsymbol{\pi}} \mathbf{h}_i$.

[1] Even though they rarely correspond to a syntactically defined notion of a phrase.

[2] Converting a *phrase* lattice to a *word* lattice is a simple matter of redistributing a compound input or output over a linear chain of arcs.

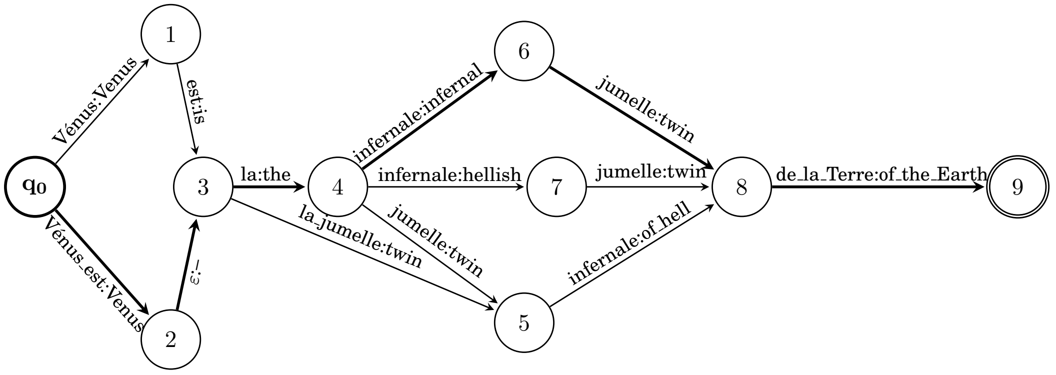


Fig. 1. Toy example of a lattice for the source French sentence '*Vénus est la jumelle infernale de la Terre*'. The path in bold corresponds to the translation hypothesis '*Venus – the infernal twin of the Earth*'. Here the standard notation for the non-consuming, matching-all symbol (ε) is used [Mohri 2009].

In oracle decoding, the decoder's job is quite different: the goal is to find the best hypothesis (according to some MT metric) that the system could have generated, independently of the actual model score: this is typically performed by searching $L_{\mathbf{f}}$ for candidates that are close to some ideal translation, obtained from human experts. In MT, the quality of a hypothesis is evaluated by the similarity between a reference and the hypothesis, computed, for instance, using the BLEU score [Papineni et al. 2002]. BLEU is formally defined for two parallel corpora, $\mathcal{E} = \{\mathbf{e}_j\}_{j=1}^{J}$ and $\mathcal{R} = \{\mathbf{r}_j\}_{j=1}^{J}$, each containing J sentences as:

$$\text{BLEU}(\mathcal{E}, \mathcal{R}) = BP \cdot \left(\prod_{m=1}^{n} p_m \right)^{1/n} \tag{2}$$

$$BP = \min(1, e^{1 - c_1(\mathcal{R})/c_1(\mathcal{E})}) \tag{3}$$

where $c_m(\mathcal{E})$ is the total number of word m-grams in $\mathcal{E}$, $c_m(\mathcal{E}, \mathcal{R})$ accumulates over sentences the number of m-grams in $\mathbf{e}_j$ that also belong to $\mathbf{r}_j$ and $p_m = c_m(\mathcal{E}, \mathcal{R})/c_m(\mathcal{E})$ denotes *clipped* (or *modified*) m-gram precision. The counts $c_m(\mathcal{E}, \mathcal{R})$ need to be clipped at the sentence level, so as to ensure that an m-gram occurring k times in a translation hypothesis in $\mathcal{E}$ and l times in the corresponding reference in $\mathcal{R}$, with $k > l$, will only be counted l times. The BLEU score performs a trade-off between precision, directly taken into account in Equation (2), and recall, indirectly introduced by the brevity penalty BP, which penalizes candidates that would be too short. Equation (2) is usually computed with $n = 4$ and we use BLEU as a synonym for 4-BLEU.

BLEU is only well defined for a pair of corpora because of the non-decomposability of the score defined in Equation (2): BLEU is computed as a geometric mean of aggregated corpus-level statistics that cannot be expressed as a combination of individual sentence-level scores. As a result, oracle decoding, which computes oracle hypotheses at the sentence-level, needs to rely on approximations to BLEU that can evaluate the similarity between a *single* hypothesis and its reference. This approximation introduces a discrepancy in the model, since gathering sentences with the highest (local) approximation may not result in the highest possible (corpus-level) BLEU score [Watanabe 2012]. In the following, we will use S-BLEU to denote a sentence-level approximation to BLEU.

With the notations introduced in this section, lattice oracle decoding can be defined as the task of finding, for a given source sentence $\mathbf{f}$, an optimal path $\boldsymbol{\pi}^*(\mathbf{f})$ among all paths in $\Pi_{\mathbf{f}}$ and amounts to solving the following optimization problem:

$$\boldsymbol{\pi}^*(\mathbf{f}) = \operatorname*{arg\,max}_{\boldsymbol{\pi} \in \Pi_{\mathbf{f}}} \text{S-BLEU}(\mathbf{e}_{\boldsymbol{\pi}}, \mathbf{r}_{\mathbf{f}}). \tag{4}$$

2.3. The Trade-Offs of Oracle Decoding

Table I. Comparison of oracle decoders; † denotes novel oracles introduced in this work

oracle	objective	surrogate objective	search	clipping	brevity
LM-2g	2-BLEU	$P_2(\mathbf{e};\mathbf{r})$ (Eq. 7)	exact	no	no
LM-4g	4-BLEU	$P_4(\mathbf{e};\mathbf{r})$ (Eq. 7)	exact	no	no
PB	4-BLEU	partial $\log$ BLEU (Eq. 9)	appr.	no	no
PBℓ	4-BLEU	partial $\log$ BLEU (Eq. 9)	appr.	no	yes
LB-2g†	2-BLEU	linear approx. (Eq. 12)	exact	no	yes
LB-4g†	4-BLEU	linear approx. (Eq. 12)	exact	no	yes
SP†	2-BLEU	1/2-gram counts (Eq. 22)	exact	no	yes
ILP†	2-BLEU	1/2-gram counts (Eq. 22)	exact	yes	yes
RLX†	2-BLEU	1/2-gram counts (Eq. 23)	exact	yes	yes

As proven in [Leusch et al. 2008], even with brevity penalty dropped, the problem of deciding whether a confusion network[3] contains a hypothesis with clipped unigram and bigram precisions all equal to 1.0 is NP-complete [Karp 1972]. Therefore, the associated optimization problem of oracle decoding for any sentence-level approximation to 2-BLEU has the same complexity. The case of more general word and phrase lattices and of the S-BLEU score is consequently also NP-complete. This complexity result stems from the chaining up of local bigram decisions that, due to the clipping constraints, have non-local effects on the precision scores. It is consequently necessary to keep a possibly exponential number of non-recombinable hypotheses (characterized by counts for each n-gram in the reference) during the search.

These NP-hardness results imply that any oracle decoder has to waive either the form of the objective function, replacing S-BLEU with better-behaved scoring functions, or the exactness of the solution, relying on approximate heuristic search algorithms. In Table I, we summarize different trade-offs that the existing (Section 3), as well as our novel (Sections 4 and 5) oracle decoders, have to make. The 'objective' column specifies the targeted score.[4] None of the decoders is actually able to optimize this objective and rather considers an approximation of it, given in the 'surrogate objective' column. Column 'search' details the accuracy of the target replacement optimization. Finally, columns 'clipping' and 'brevity' indicate whether the corresponding properties of the BLEU score are considered in the target surrogate and in the search algorithm.

2.4. Finite-State Acceptors

The implementations of the oracles described in the first part of this work (Sections 3 and 4) use the formalism of finite-state acceptors (FSA) over various semirings. This section quickly reviews the notation necessary to reformulate oracle decoding using this formalism. This greatly simplifies the description of different approaches to oracle computation and, finally, allows us to solve Equation (4) with a generic algorithm.

Recall that a $(\oplus, \otimes)$-semiring $\mathbb{K}$ over a set K is a system $\langle K, \oplus, \otimes, \bar{0}, \bar{1} \rangle$, where $\langle K, \oplus, \bar{0} \rangle$ is an algebraic structure with commutative and associative operation $\oplus$ and identity

[3] A confusion network is a simple form of word graph, introduced in [Mangu et al. 2000].

[4] [Song et al. 2013] studies the quality of several sentence-level BLEU approximations, including the ones used in this work, with respect to both their correlation with human scores and their impact on training.

element $\bar{0}$, meaning that $a \oplus (b \oplus c) = (a \oplus b) \oplus c$, $a \oplus b = b \oplus a$ and $a \oplus \bar{0} = \bar{0} \oplus a = a$. Additionally, the structure $\langle K, \otimes, \bar{1} \rangle$ has commutative and associative operation $\otimes$ and identity element $\bar{1}$. Furthermore, $\otimes$ distributes over $\oplus$ so that $a \otimes (b \oplus c) = (a \otimes b) \oplus (a \otimes c)$ and $(b \oplus c) \otimes a = (b \otimes a) \oplus (c \otimes a)$ and element $\bar{0}$ annihilates $\mathbb{K}$ ($a \otimes \bar{0} = \bar{0} \otimes a = \bar{0}$).

For a finite set of states Q, let $A = (\Sigma, Q, I, F, E)$ be a weighted finite-state acceptor over an alphabet Σ with weights in $\mathbb{K}$, meaning that transitions in A carry both a symbol in Σ and a weight in $\mathbb{K}$. Formally, the transition weighting function E is a mapping from $(Q \times \Sigma \times Q)$ into $\mathbb{K}$; likewise, initial I and final weight F functions are mappings from Q into $\mathbb{K}$.

We borrow here the notations of [Mohri 2009]: an edge ξ from origin state $p(\xi) = q$ to destination state $n(\xi) = q'$ carrying label $w(\xi) = \sigma$ and weight $k = E(q, \sigma, q')$ will be denoted $\xi = (q, \sigma/k, q')$. These notations extend to paths: if $\boldsymbol{\pi}$ is a path in A, $p(\boldsymbol{\pi})$ (resp. $n(\boldsymbol{\pi})$) is its initial (resp. ending) state, $w(\boldsymbol{\pi})$ is the sequence of labels accumulated along the path and $E(\boldsymbol{\pi})$ is the total weight.

A finite-state transducer (FST) is an FSA with an additional output alphabet Σ', so that each transition carries a pair of input/output symbols from, respectively, Σ and Σ'. As for FSA arcs, we use notation $(q, \sigma\!:\!\sigma'/k, q')$ for an arc connecting states q and q', mapping symbol $\sigma \in \Sigma$ to symbol $\sigma' \in \Sigma'$ and carrying weight k.

In our setting, an FSA $A_{\mathbf{f}}$ is derived from a word lattice $L_{\mathbf{f}}$ as follows. We already assumed that $L_{\mathbf{f}}$ has a single start and end states, denoted respectively q_I and q_F and each arc in $L_{\mathbf{f}}$ is labeled with a target word $\mathbf{e}$. Weights are specific to each oracle and will be defined in their respective description. The total weight of a path $\boldsymbol{\pi} = \xi_1 \ldots \xi_l$ in $A_{\mathbf{f}}$ is computed as:

$$E(\boldsymbol{\pi}) = I(p(\xi_1)) \otimes \Big[\bigotimes_{i=1}^{l} E(\xi_i) \Big] \otimes F(n(\xi_l)). \tag{5}$$

The total weight of a path corresponds to the complete translation hypothesis read along $\boldsymbol{\pi}$ as $w(\boldsymbol{\pi}) = w(\xi_1)...w(\xi_l)$. The total weight of all the translation hypotheses in lattice $L_{\mathbf{f}}$ is thus

$$E(A_{\mathbf{f}}) = \bigoplus_{\boldsymbol{\pi} \in \Pi(A_{\mathbf{f}})} E(\boldsymbol{\pi}). \tag{6}$$

As discussed in Sections 3 and 4, several oracle decoding algorithms can be expressed as shortest distance problems, provided a suitable definition of the underlying acceptor and associated semiring. In particular, quantities such as (6) can be efficiently found by generic shortest distance algorithms over acyclic graphs [Mohri 2002]. In the following, for our FSA-based implementations of oracle decoders, we reduce the optimization problem (4) to Equation (6), with the operation $\oplus$ replaced with $\max$ or $\min$ (depending on implementation) and the oracle-specific details incorporated into the definition of the operation $\otimes$.

3. EXISTING ALGORITHMS

Several approximate search algorithms have been proposed in the literature so as to work around the complexity of the exact oracle BLEU decoding problem defined in Equation (2). As mentioned above, these decoders rely on inexact search algorithms and have to disregard many hypotheses in order to avoid a potential exponential explosion, and/or to consider an approximation of the objective function. They will be used as baselines in our experiments.

It should be noted that, in addition to their approximate nature, these decoders do not take into account the fact that the n-gram precisions have to be clipped (as opposed to the novel methods that will be introduced in Section 5).

3.1. Language Model Oracle (LM)

The simplest approach considered in this paper builds on suggestions of [Li and Khudanpur 2009] and [Crego et al. 2010], that reduce oracle decoding to the problem of finding the most likely hypothesis under a trivial n-gram language model trained with just one sentence: the reference translation.

An n-gram language model factorizes the probability $P(\mathbf{e})$ of a sentence as a product of terms $P(e_i|e_{i-1}...e_1)$, where word e_i is further assumed to depend only on the $n-1$ previous words: $P(e_i|e_{i-1}...e_1) = P(e_i|e_{i-1}...e_{i-n+1})$. The probability of a complete sentence $\mathbf{e}$ of length $|\mathbf{e}|$ is given by:

$$P_n(\mathbf{e}) = \prod_{i=0}^{|\mathbf{e}|-n+1} P\left(e_{i+n-1}|e_{i+n-2}...e_i\right). \quad (7)$$

An n-gram language model can conveniently be represented as an FSA denoted A_{LM}, with each arc carrying a negative log-probability weight and with additional ρ-type failure transitions for back-off arcs [Allauzen et al. 2004].

A specific language model $A_{LM}(\mathbf{r_f})$ is then estimated for each source sentence $\mathbf{f}$ using the reference $\mathbf{r_f}$ as sole training material. Oracle decoding amounts to finding a shortest (most probable) path in the weighted FSA resulting from the composition $L_\mathbf{f} \circ A_{LM}(\mathbf{r_f})$ over the $(\min, +)$-semiring:

$$\boldsymbol{\pi}^*_{LM}(\mathbf{f}) = \texttt{ShortestPath}_{(\min,+)}(L_\mathbf{f} \circ A_{LM}(\mathbf{r_f})). \quad (8)$$

This approach replaces the optimization of n-BLEU with a search for the most likely path under a simplistic language model. One may expect the most probable path to match long k-grams from the reference, thus delivering high n-BLEU solutions.

Note that this simple-minded approach ignores the brevity penalty, contrarily to the proposal of [Li and Khudanpur 2009], which includes an approximation BP during the search. The impact of this shortcut will be assessed in Section 6.2.

3.2. Partial BLEU Oracle (PB)

Another approach is introduced in [Dreyer et al. 2007]: oracle translations are shortest paths in a lattice $L_\mathbf{f}$, in which the weight of each path $\boldsymbol{\pi}$ is an approximation to the sentence-level $\log \text{S-BLEU}(\boldsymbol{\pi})$ score of the corresponding (complete or partial) hypothesis constructed so far:

$$\log \text{S-BLEU}(\boldsymbol{\pi}) = \frac{1}{4}\sum_{m=1}^{4} \log p_m. \quad (9)$$

Here, the brevity penalty is ignored and m-gram precisions are offset for all sizes of m-grams to avoid null counts [Lin and Och 2004]:

$$p_m = \frac{c_m(\mathbf{e}_{\boldsymbol{\pi}}, \mathbf{r}) + 0.1}{c_m(\mathbf{e}_{\boldsymbol{\pi}}) + 0.1} \quad (10)$$

This approach can be readily re-implemented using the formalism of FSA by defining a suitable semiring. Let each weight x of the semiring keep a set of tuples accumulated up to the lattice state in the lattice. Each tuple contains: the partial hypothesis, h, the three most recent words of the hypothesis, w, the hypothesis length ℓ, the vector of n-gram counts $\bar{c}$, and b, the partial sentence-level $\log \text{S-BLEU}(\mathbf{h})$ score defined in Equation (9).

Each arc in $L_\mathbf{f}$ is initialized with a singleton set containing one tuple with a single word as the partial hypothesis. For the semiring operations, we define one common

$\otimes$-operation and two versions of the $\oplus$-operation, that take two argument weights x_1 and x_2:

- $x_1 \otimes_{PB} x_2$ – appends the unique[5] word in the single tuple of x_2 to each tuple in x_1, accordingly updating their recent history w and hypothesis h, n-gram count vector $\bar{c}$, length ℓ, and partial score b;
- $x_1 \oplus_{PB} x_2$ – takes the union of the sets in x_1 and x_2: whenever two tuples have the same recent history ($\mathbf{w}_1 = \mathbf{w}_2$), only the one having the highest partial score is kept.
- $x_1 \oplus_{PB\ell} x_2$ – takes the union of the sets in x_1 and x_2: whenever two tuples have the same recent history ($\mathbf{w}_1 = \mathbf{w}_2$) *and* the same hypothesis length ($\ell_1 = \ell_2$), only the one having the highest partial score is kept. This variant is introduced so as to make the competition between hypotheses more fair, as shortest hypotheses tend to have a better precision.

The optimal path is then found by running a generic shortest distance algorithm over either of the semirings defined above:

$$\boldsymbol{\pi}^*_{PB/PB\ell}(\mathbf{f}) = \texttt{ShortestPath}_{(\oplus_{\{PB/PB\ell\}},\otimes_{PB})}(L). \tag{11}$$

The $(\oplus_{PB\ell}, \otimes_{PB})$-semiring, in which the equal length requirement also implies equal brevity penalties, is more conservative in recombining competing hypotheses and, seemingly, should achieve final a S-BLEU that is least as good as that obtained with the $(\oplus_{PB}, \otimes_{PB})$-semiring. However, in practice, it does not necessarily yield better scores, because of the n-gram clipping in the definition of BLEU, that is ignored by this oracle. This observation was confirmed experimentally, and we found that the seemingly more accurate version of $\oplus$ eventually produced poorer hypotheses (see experiments in Section 6).

4. LINEAR BLEU ORACLE (LB)

In this section, we propose a new oracle based on a linear approximation of the sentence-level contributions to the corpus BLEU. This approximation was originally introduced in [Tromble et al. 2008] for the purpose of Minimum Bayes Risk decoding in lattices [Tromble et al. 2008; Allauzen et al. 2010; Blackwood et al. 2010]. We show here that it can be modified to approximately compute a BLEU-optimal oracle translation.

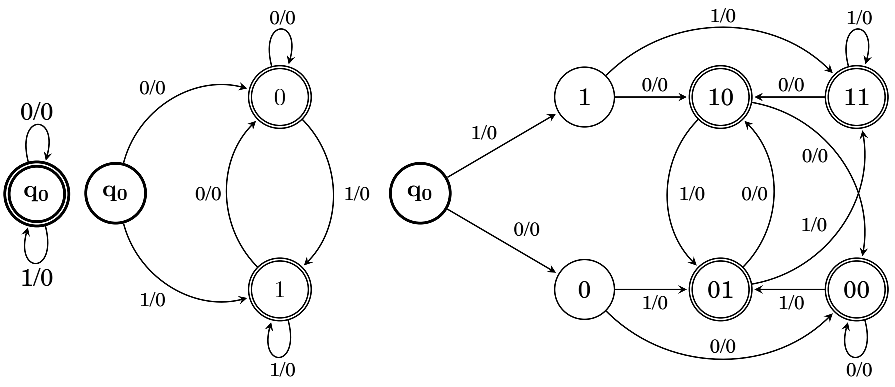
The authors of [Tromble et al. 2008] propose to approximate the contribution of an isolated translated sentence to the corpus-level BLEU score by computing a first order expansion of the change of the $\log$ BLEU score incurred by removing the sentence from the corpus. They show that the increment in $\log$ BLEU implied by a single sentence can be written as a linear function of the (modified) n-gram precisions:

$$\lin \mathrm{BLEU}(\boldsymbol{\pi}) = \theta_0 \, |\mathbf{e}_{\boldsymbol{\pi}}| + \sum_{n=1}^{4} \theta_n \sum_{u \in \Sigma^n} c_u(\mathbf{e}_{\boldsymbol{\pi}}) \delta_u(\mathbf{r}), \tag{12}$$

where $\theta_0 \ldots \theta_4$ are parameters of the method, $c_u(\mathbf{e})$ is the number of times the n-gram u appears in $\mathbf{e}$, and $\delta_u(\mathbf{r})$ is an indicator variable testing the presence of u in $\mathbf{r}$: it is equal to 1 if the n-gram u appears in $\mathbf{r}$, and to 0 otherwise. This approximation basically expresses the contribution of each n-gram match to the final score as a fixed reward θ_n.

To exploit this approximation for oracle decoding, we use the $(\min, +)$ semiring and construct four weighted automata Δ_n containing a (final) state for each possible $(n -$

[5] The right-hand argument x_2 of $\otimes$ always contains a single word because of the possibility to use the topological order during shortest path search, which implies that x_2 always corresponds to the weight found on a single edge.



(a) Δ_1 (b) Δ_2 (c) Δ_3

Fig. 2. Examples of the Δ_n automata for $\Sigma = \{0, 1\}$ and $n = 1 \ldots 3$. Initial and final states are marked, respectively, with bold and with double borders. Arcs connecting final states will carry a null weight only when the corresponding n-gram does not appear in the reference and a unit weight otherwise. Transitions to and from auxiliary states carry a null weight.

1)-gram, and all weighted transitions of the kind:

$$(\sigma_1^{n-1}, \sigma_n/\theta_n \delta_{\sigma_1^n}(\mathbf{r}), \sigma_2^n), \tag{13}$$

where σ are labels in Σ, and the input σ_1^{n-1} and output σ_2^n states correspond to label sequences that are respectively the maximal proper prefix and suffix of the n-gram σ_1^n.

We also add auxiliary states corresponding to k-grams ($k < n - 1$), the purpose of which is to help reach one of the main $(n - 1)$-gram states; all these transitions carry a null weight. There are $\frac{|\Sigma|^{n-1}-1}{|\Sigma|-1}$ such supplementary states and their transitions are formally defined as:

$$(\sigma_1^k, \sigma_{k+1}/0, \sigma_1^{k+1}), k = 1 \ldots n - 2.$$

The single initial state is connected to the auxiliary states with transitions

$$(q_0, \sigma_k/0, \sigma_1^k), k = 1 \ldots n - 2.$$

Apart from these auxiliary states, the initial state and the omitted output n-grams σ_1^n, the rest of the graph (i.e. all final states) reproduces the structure of the well-known de Bruijn graph $B(\Sigma, n)$ (see Figure 2).

To actually compute the best oracle hypothesis, we first weight all arcs in the input FSA $A_{\mathfrak{f}}$ with θ_0 to produce the weighted FSA Δ_0. Weighting makes each word's weight equal in a hypothesis path and the total weight of any path in Δ_0 is proportional to the number of words it contains. Then, by sequentially composing[6] Δ_0 with Δ_i, $i = 1 \ldots n$, we will increase the reward for paths which match longer n-grams in the reference.

Finally, with all operations performed in the $(\min, +)$-semiring, the oracle translation is readily computed as:

$$\boldsymbol{\pi}^*_{LB} = \texttt{ShortestPath}_{(\min,+)}(\Delta_0 \circ \Delta_1 \circ \Delta_2 \circ \Delta_3 \circ \Delta_4). \tag{14}$$

[6] We implicitly identify here FSAs with FSTs having identical input and output symbols; composition is then licit since all these transducers share the same alphabet.

Parameters θ_n are set as in [Tromble et al. 2008]:

$$\theta_0 = 1 \tag{15}$$

$$\theta_n = -(4p \cdot r^{n-1})^{-1} \tag{16}$$

where p and r are average values of, respectively, the corpus unigram precision and common n-gram and $(n-1)$-gram ratio. The values of θ_n are increasing rewards for matching n-grams; the constant value of θ_0 roughly accounts for the brevity penalty (each word in a hypothesis contributes equally to the final path score).

5. ORACLES WITH N-GRAM CLIPPING

In this section, we describe two novel oracle decoders that are able to take n-gram clipping into account. These oracles leverage the well-known fact that shortest distance problems, which lie at the heart of all the oracles described so far, can be straightforwardly reduced to Integer Linear Programming (ILP) problems [Wolsey 1998]. Once oracle decoding is formulated as an ILP problem, it is relatively easy to introduce additional constraints, for instance to enforce n-gram clipping.

We will first introduce the optimization problem describing oracle decoding, then present several ways to solve it in an efficient manner.

5.1. Problem Description

Recall that Q denotes set of states in the lattice, and $\Xi = (\xi_i)_{i=1}^{\#\{\Xi\}}$ denotes its set of edges. Abusing the notations, we will also think of an edge ξ_i as a binary variable describing whether the edge is ‘selected’ or not. The set $\{0,1\}^{\#\{\Xi\}}$ of all possible edge assignments will be denoted $\mathcal{P}$. Note that Π, the set of all paths in the lattice is a subset of $\mathcal{P}$: each assignment satisfying a set of *path constraints* corresponds to a path in the lattice.

As described in Section 2, we assume that each edge ξ_i generates a single word $w(\xi_i)$ and focus first on the simple following problem: given a lattice L_{f}, find the optimal hypothesis with respect to a new sentence-level approximation to the 1-BLEU score defined in the following paragraph.

The S1-BLEU score, with brevity penalty omitted, amounts to the unigram precision and can be made arc-decomposable by defining, for every edge ξ_i, an associated reward θ_i that measures the edge local contribution to the hypothesis score. For instance, for the sentence-level approximation to the 1-BLEU score, rewards are defined as:

$$\theta_i = \begin{cases} \Theta_1 & \text{if } w(\xi_i) \text{ is in the reference,} \\ -\Theta_2 & \text{otherwise,} \end{cases} \tag{17}$$

where Θ_1 and Θ_2 are positive constants chosen to maximize the corpus BLEU score.[7] Θ_1 (resp. Θ_2) is a reward (resp. a penalty) for generating a word in the reference (resp. *not* in the reference). The score of an assignment $\boldsymbol{\xi} \in \mathcal{P}$ is then defined as:

$$\text{score}(\boldsymbol{\xi}) = \sum_{i=1}^{\#\{\Xi\}} \xi_i \cdot \theta_i. \tag{18}$$

This score defines a new approximation to BLEU at the sentence level that can be understood as a trade-off between the number of common words in the hypothesis and the reference (accounting for recall) and the number of words of the hypothesis that do not appear in the reference (accounting for precision).

[7] We tried several combinations of Θ_1 and Θ_2 and kept the one that had the highest corpus 4-BLEU score.

As explained in Section 2.4, finding the oracle hypothesis amounts to solving the shortest distance problem of Equation (6), which can be reformulated as the following constrained optimization problem [Wolsey 1998]:

$$\begin{aligned}
&\arg\max_{\boldsymbol{\xi}\in\mathcal{P}} && \sum_{i=1}^{\#\{\Xi\}} \xi_i \cdot \theta_i \\
&\text{subject to} && \sum_{\xi\in\Xi^-(q_F)} \xi = 1, \\
& && \sum_{\xi\in\Xi^+(q_0)} \xi = 1 \\
& && \sum_{\xi\in\Xi^+(q)} \xi - \sum_{\xi\in\Xi^-(q)} \xi = 0,\ \forall q \in Q \setminus \{q_I, q_F\},
\end{aligned} \tag{19}$$

where q_I (resp. q_F) is the initial (resp. final) state in the lattice and $\Xi^-(q)$ (resp. $\Xi^+(q)$) denotes the set of incoming (resp. outgoing) edges of state q. These path constraints ensure that the solution of the problem corresponds to a valid path in the lattice: the assignment of binary indicator variables must have only a single transition from the initial state, a single transition to the final state, and all other non-initial/non-final states must have one incoming and one outgoing transitions.

The optimization problem in Equation (19) can be further extended to take clipping into account. Let us introduce, for each word w, a variable γ_w counting the number of times w appears in the hypothesis, clipped to the number of times $c_w(\mathbf{r})$ it appears in the reference $\mathbf{r}$. Formally, γ_w is defined by:

$$\gamma_w = \min\left\{ c_w(\mathbf{r}), \sum_{\xi\in\Omega(w)} \xi \right\}, \tag{20}$$

where $\Omega\left(w\right)$ is the subset of edges generating w and $\sum_{\xi\in\Omega(w)} \xi$ is the number of occurrences of w in the solution. Using the γ variables, it is possible to define a sentence-level 'clipped' approximation to 1-BLEU:

$$\Theta_1 \cdot \sum_{w\in\Sigma} \gamma_w - \Theta_2 \cdot \left(\sum_{i=1}^{\#\{\Xi\}} \xi_i - \sum_w \gamma_w \right). \tag{21}$$

Indeed, the clipped number of words in the hypothesis that appear in the reference is given by $\sum_w \gamma_w$, and $\sum_{i=1}^{\#\{\Xi\}} \xi_i - \sum_w \gamma_w$ corresponds to the number of words in the hypothesis that do not appear in the reference or that are surplus to the clipped count.

Putting it all together, the lattice oracle with clipped 1-gram precisions is defined by the following optimization problem:

$$\begin{aligned}
\operatorname*{arg\,max}_{\boldsymbol{\xi}\in\mathcal{P},\gamma_w} \quad & (\Theta_1+\Theta_2)\cdot\sum_w \gamma_w - \Theta_2\cdot\sum_{i=1}^{\#\{\Xi\}}\xi_i \\
\text{subject to} \quad & \gamma_w \geq 0, \\
& \gamma_w \leq c_w(\mathbf{r}), \\
& \gamma_w \leq \sum_{\xi\in\Omega(w)}\xi \\
& \sum_{\xi\in\Xi^-(q_F)}\xi = 1 \\
& \sum_{\xi\in\Xi^+(q_0)}\xi = 1 \\
& \sum_{\xi\in\Xi^+(q)}\xi - \sum_{\xi\in\Xi^-(q)}\xi = 0,\ q\in Q\setminus\{q_0,q_F\}
\end{aligned} \tag{22}$$

where the first three sets of constraints derive from the linearization of the definition of γ_w, made possible by the positivity of Θ_1 and Θ_2, and the last three sets of constraints are the path constraints.

It is straightforward to generalize this optimization problem to higher-order n-gram lattices, in which each edge is labeled by the n-gram it generates. Such n-gram FSA can be produced by composing the word lattice with de Bruijn graphs introduced in Section 4.[8] In this case, the reward of an edge will be defined as a combination of the (clipped) number of m-gram matches for $m = 1 \dots n$ and solving the optimization problem yields a n-BLEU optimal hypothesis. The approach can be further generalized to other metrics as long as the reward of an edge can be computed locally, without considering any additional information.

The constrained optimization problem (22) introduced in the previous section can be solved efficiently using off-the-shelf ILP solvers.

5.2. Shortest Path Oracle (SP)

As a trivial special case of the above formulation, we also define a Shortest Path Oracle (SP) that solves the optimization problem in (19). As no clipping constraints apply, it can be solved efficiently using the standard Bellman-Ford algorithm.

5.3. Oracle Decoding through Lagrangian Relaxation (RLX)

In this section, we introduce yet another method to solve problem (22) without having recourse to any external ILP solver. Following [Rush et al. 2010; Chang and Collins 2011], we propose an original method for oracle decoding based on Lagrangian Relaxation. This method takes advantage of the structure of the optimization problem to efficiently compute an optimal solution. It relies on the idea of relaxing — 'forgetting' — some of the clipping constraints in the optimization problem of the ILP oracle decoder: starting from an unconstrained problem, count clipping is enforced by incrementally strengthening the weight of those paths that satisfy the constraints, and accordingly downweighting the weight of paths that do not.

[8] More precisely, a 'transducer' version of these graphs, where arcs read (input) words, and write (output) n-grams σ_1^n.

The oracle decoding problem with clipping constraints amounts to solving:

$$\begin{aligned} &\operatorname*{arg\,min}_{\boldsymbol{\xi} \in \Pi} && -\sum_{i=1}^{\#\{\Xi\}} \xi_i \cdot \theta_i \\ &\text{subject to} && \sum_{\xi \in \Omega(w)} \xi \leq c_w(\mathbf{r}), w \in \mathbf{r} \end{aligned} \tag{23}$$

where, by abusing the notations, $\mathbf{r}$ also denotes the set of words in the reference. For the sake of clarity, path constraints are incorporated into the domain, and the $\arg\min$ runs over Π and no longer over $\mathcal{P}$. To solve this optimization problem, we consider its dual form and use Lagrangian Relaxation to deal with clipping constraints.

Let $\boldsymbol{\lambda} = \{\lambda_w\}_{w \in \mathbf{r}}$ denote the positive Lagrange multipliers, one for each different word type in the reference, then the Lagrangian of problem (23) is expressed as:

$$\mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\xi}) = -\sum_{i=1}^{\#\{\Xi\}} \xi_i \cdot \theta_i + \sum_{w \in \mathbf{r}} \lambda_w \cdot \left(\sum_{\xi \in \Omega(w)} \xi - c_w(\mathbf{r}) \right), \tag{24}$$

The dual objective is then:

$$\mathcal{L}(\boldsymbol{\lambda}) = \min_{\boldsymbol{\xi}} \mathcal{L}(\boldsymbol{\lambda}, \boldsymbol{\xi}) \tag{25}$$

and the dual problem is:

$$\max_{\boldsymbol{\lambda} \succeq 0} \mathcal{L}(\boldsymbol{\lambda}). \tag{26}$$

To solve the dual problem, we first need to work out the dual objective:

$$\boldsymbol{\xi}^* = \operatorname*{arg\,min}_{\boldsymbol{\xi} \in \Pi} \left(-\sum_{i=1}^{\#\{\Xi\}} \xi_i \cdot \theta_i + \sum_{w \in \mathbf{r}} \lambda_w \cdot (\sum_{\xi \in \Omega(w)} \xi - c_w(\mathbf{r})) \right) \tag{27}$$

$$= \operatorname*{arg\,min}_{\boldsymbol{\xi} \in \Pi} \sum_{i=1}^{\#\{\Xi\}} \xi_i \cdot \left(\lambda_{w(\xi_i)} - \theta_i \right), \tag{28}$$

where we assume that $\lambda_{w(\xi_i)}$ is 0 when the word $w(\xi_i)$ is not in the reference. The role of λ_w in these equations will be to reduce the reward of arcs labeled by words whose count exceeds what is found in the reference. In the same way as in Section 5.2, the solution of this problem can be efficiently computed by a shortest path algorithm.

It is possible to solve the dual problem and optimize $\mathcal{L}(\boldsymbol{\lambda})$ by taking advantage of the fact that it is a concave function. In this work, we chose to use a simple gradient descent to solve the dual problem. A subgradient of the dual objective is:

$$\frac{\partial \mathcal{L}(\boldsymbol{\lambda})}{\partial \lambda_w} = \sum_{\xi \in \Omega(w) \cap \xi^*} \xi - c_w(\mathbf{r}). \tag{29}$$

Each component of the gradient corresponds to the difference between the number of times word w appears in the hypothesis and the number of times it appears in the reference. It can finally be shown [Chang and Collins 2011] that, upon convergence, the clipping constraints will be enforced in the optimal solution.

Algorithm 1 summarizes the optimization problem defined by Equation (23). In the algorithm, $\alpha^{(t)}$ corresponds to the step size at the t^{th} iteration. In our experiments, we used $\alpha^{(t)} = \frac{1}{t}$. Compared to the usual gradient descent algorithm, there is an additional projection step of $\boldsymbol{\lambda}$ on the positive orthant, which enforces the constraint $\boldsymbol{\lambda} \succeq 0$.

ALGORITHM 1: Optimization cycle for problem Equation (23)

```
$\forall w, \lambda_w^{(0)} \leftarrow 0$
for $t = 1 \rightarrow T$ do
    $\boldsymbol{\xi}^{*(t)} = \arg\min_{\boldsymbol{\xi}} \sum_i \xi_i \cdot \left(\lambda_{w(\xi_i)} - \theta_i\right)$
    if all clipping constraints are enforced then
        optimal solution found
    else
        for $w \in \mathbf{r}$ do
            $n_w \leftarrow$ n. of occurrences of $w$ in $\boldsymbol{\xi}^{*(t)}$
            $\lambda_w^{(t)} \leftarrow \lambda_w^{(t)} + \alpha^{(t)} \cdot (n_w - c_w(\mathbf{r}))$
            $\lambda_w^{(t)} \leftarrow \max(0, \lambda_w^{(t)})$
```

5.4. Summary

The previous section explained how, by reformulating the shortest path problem as an ILP problem, additional constraints such as clipping constraints can be considered, resulting in a new kind of oracles. In the end, the five oracles described in this work can be classified into two families.

A first family is made of the ILP and RLX oracle and considers oracle decoding as an ILP problem. They both consider, as an approximation to the BLEU score a simple linear combination of n-gram counts and add a penalty for non-matching words.

The LM (Section 3.1), PB (Section 3.2), LB (Section 4) and SP (Section 5.2) oracles are part of a second family: they all rely on shortest distance algorithms and differ only in the way they approximate the BLEU score at the sentence-level. The SP oracle relies on the same approximation as the ILP and RLX oracle allowing us to compare the different approximation; the LB oracle is using a similar approximation although it parameterizes it differently. The PB oracle considers the well-known sentence-level approximation to BLEU introduced by [Lin and Och 2004], usually used in PBSMT training. This approximation is adding pseudocounts to the computation of the n-gram precision in order to avoid null counts. Eventually, the LM oracle approximates the BLEU score at the sentence-level by the probability of a language model.

All the oracles of this family are very close in terms of both their search methods and their objective function. We will see in the next section how close their actual empirical performances are.

6. EVALUATING ORACLE DECODERS

In this section, we describe the experiments that have been conducted to evaluate the performance of the different oracle decoders presented above. We first describe our experimental setup in Section 6.1, then report the performance of the oracle decoding both at the corpus level (Section 6.2) and at the sentence level (Section 6.3) before, finally, evaluating globally the quality of the other hypotheses in the word lattice (Section 6.4).

6.1. Experimental Setup

Experiments were run in parallel on a server with 64G of RAM and 2 Xeon CPUs with 4 cores at 2.3 GHz. Lattices were generated by two state-of-the-art decoders, Moses [Koehn et al. 2007] and N-code [Crego et al. 2011]. Both systems implement a phrase-based approach to SMT, but differ in two important aspects: (a) the translation model of Moses is a unigram model of phrase-pairs of arbitrary length (up to a fixed limit); N-code uses a bilingual n-gram model of shorter phrase-pairs; (b) the reordering model of Moses explores all possible permutations in a fixed-size window, whereas

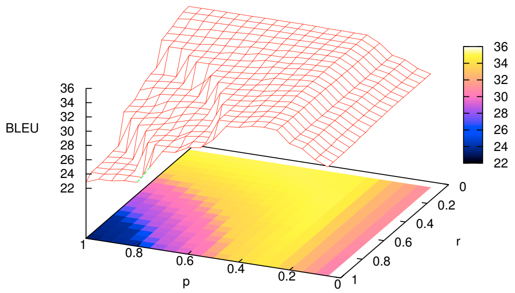


Fig. 3. Performance of the LB-4g oracle for different combinations of parameters p and r for the German to English task.

N-code only considers a much smaller set of precomputed permutations. A consequence is that the search space of N-code tends to be much smaller than that of Moses.

Experiments were run for 3 language pairs (French to English, German to English and English to German). Systems were trained on the data provided for the WMT'11 Evaluation task [Callison-Burch et al. 2011], tuned on the WMT'09 test data and evaluated on the WMT'10 official test. A detailed description of the system training procedures is given in [Allauzen et al. 2011]. Because of their size, Moses lattices could not be directly handled by some of our oracle decoders and were therefore generated with a non-default value of the pruning parameter: the beam threshold pruning parameter,[9] was set to 0.5 to generate the lattices used in our oracle finding experiments, which, for the French to English task, amounts to dropping out 95% of the edges of the lattices considered during decoding. Because of the way they are built, N-code lattices are much smaller and all results were obtained with no additional pruning of the lattices. In these experimental conditions, the number of edges is similar in Moses and N-code lattices. Note, that decoding and n-best list generation for baseline results were still run with the default value of the beam threshold parameter. Impact of lattice pruning is evaluated in Section 7.2.

For computational reasons, we have only optimized 2-BLEU in our experiments with ILP and RLX oracles. To make a fair comparison, we have included 2-BLEU versions of the LB and LM oracles, identified below with the '-2g' suffix. The two versions of the PB oracle are respectively denoted as PB and PBℓ, depending on the type of the $\oplus$-operation they consider (see Section 3.2).

Our implementation of the ILP oracle decoder uses Gurobi [Gurobi Optimization 2010], a commercial ILP solver which offers free academic licenses; we consistently used the OpenFST toolkit [Allauzen et al. 2007] to implement all other oracles, except for the RLX oracle, which does not use any third-party software.

6.1.1. Hyper-parameter Selection. All oracle decoders are optimizing sentence-level approximations to the BLEU score that depend on various hyper-parameters. The quality of the results proved to be very sensitive to the choice of these hyper-parameters. For instance, Figure 3 shows the brute force evaluation of LB oracle for each value of p and r (see Equation (16)) for the German to English task: for the considered range of values for p and r, BLEU scores vary from about 22 to almost 36.

[9] In Moses, this threshold is controlled by the `-b` parameter

In all our experiments, these hyper-parameters were chosen by grid-search so as to maximize the corpus level BLEU score for N-code lattices. For the LM oracle decoder, we optimized the value of the smoothing parameter μ used to correct the n-gram counts (an identical value was used for all n-gram orders, and for both types of lattices. The ILP and RLX oracles use the same hyper-parameters Θ_1 and Θ_2, optimized by cross-validation with the ILP oracle. The PB and PBℓ oracles do not have hyper-parameters and do not require any specific tuning.

Table II summarizes the values of the hyper-parameters used in the experiments. Surprisingly enough, for the SP oracle, best performances were systematically achieved when the reward for generating a unigram was 0 and the penalty for generating words not in the reference is always pretty low, suggesting that the BLEU score is more sensitive to precision than to recall. Another unexpected observation is that the values of the hyper-parameters for distinct language pairs are quite different.

Table II. Values of the hyper-parameters used in our experiments

Oracle	French → English	English → German	German → English
LB-4g	$p = 0.250, r = 0.150$	$p = 0.175, r = 0.575$	$p = 0.350, r = 0.425$
LB-2g	$p = 0.300, r = 0.150$	$p = 0.300, r = 0.175$	$p = 0.575, r = 0.100$
LM-4g	$\mu = 0.005$	$\mu = 0.006$	$\mu = 0.004$
LM-2g	$\mu = 0.001$	$\mu = 0.001$	$\mu = 0.001$
SP	$\Theta_0 = -1, \Theta_1 = 0, \Theta_2 = 10$	$\Theta_0 = -1, \Theta_1 = 0, \Theta_2 = 6$	$\Theta_0 = -2, \Theta_1 = 0, \Theta_2 = 6$
ILP	$\Theta_0 = -2, \Theta_1 = 6, \Theta_2 = 10$	$\Theta_0 = -1, \Theta_1 = 2, \Theta_2 = 4$	$\Theta_0 = -2, \Theta_1 = 2, \Theta_2 = 10$

6.2. Corpus-Level Evaluation

Performance, in terms of translation quality, of the oracles introduced in Sections 3 through 5 are evaluated by the BLEU score. Table III presents the BLEU scores achieved by the different oracles for two decoders and the three translation directions we are considering. For the sake of comparison, the scores achieved by the two decoders, as well as the n-best oracle scores are included in the table.

The results in Table III clearly show that the search space of state-of-the-art PBSMT decoders contains very good hypotheses: whatever strategy is used, the BLEU score achieved by oracle decoder is as least twice as large as the score of current systems. This observation suggests that the scoring and feature functions used by decoders to select their translation hypothesis in the search space is a major weakness of existing PBSMT systems. This conclusion is in line with the findings of several other works [Auli et al. 2009; Wisniewski et al. 2010; Turchi et al. 2012]. Section 7 looks deeper into these issues and will discuss reasons for this failure. It also appears that the oracles found in lattices are much better than the oracles found in the n-best list. Such a conclusion is quite expected, given the number of hypotheses represented in a lattice, which is orders of magnitude larger than that of a n-best list. A more detailed comparison between lattice and n-best oracles is presented in Section 6.4.2. It also appears that, while Moses and N-code achieve similar performance, their oracle scores are quite different. As shown in Section 7.2 this performance difference stems from the fact that Moses's lattices are more heavily pruned.

All oracles, except for LM-oracles, achieve comparable performance: their results are always within one or two BLEU points. These results have many practical implications. First, it means that optimizing the 2-BLEU score will achieve the same results as optimizing 4-BLEU at a much smaller computational cost as the lattices considered are smaller. For instance, for the French to English lattices generated by N-code, LB-4g takes 6.01 seconds to decode a single sentence, while LB-2g only takes 0.06

Table III. BLEU score achieved by the different oracles presented in this work.

	N-code			Moses		
	fr → en	de → en	en → de	fr → en	de → en	en → de
RLX	47.84	35.19	24.75	43.82	36.43	28.68
ILP	48.09	35.22	25.15	44.10	37.07	28.64
LB-4g	**48.22**	**35.49**	**25.34**	**44.38**	**37.73**	**29.94**
LB-2g	47.71	35.09	24.85	43.82	36.52	28.94
PB	46.76	34.85	24.78	43.42	36.75	28.76
PBℓ	46.47	34.76	24.73	43.20	36.62	28.65
SP	47.68	35.00	24.83	43.82	36.38	28.94
LM-4g	40.69	30.51	22.02	38.16	30.92	23.28
LM-2g	40.40	30.75	21.99	37.93	30.98	23.16
decoder	27.88	22.05	15.83	27.68	21.85	15.89
100-best oracle	36.36	29.22	21.18	35.25	29.13	22.03

seconds.[10] This result confirms the intuition that hypotheses sharing many 2-grams, would likely have many common 3- and 4-grams as well. Second, taking clipping into account (which, as a direct consequence, makes oracle-decoding NP-hard) produces little to no improvement: in all conditions, the ILP oracle only slightly outperforms the SP oracle, except for the English to German condition in which performance are degraded when clipping is considered. Again, ignoring clipping constraints results in faster and conceptually simpler oracles.

Our experiments show consistently inferior performance of the LM-oracle. A more detailed analysis of the BLEU scores suggests that the LM oracle is outperformed by other oracles for two reasons. First, they do not take the brevity penalty into account and generate shorter sentences than other oracles. For instance, for the German to English direction, the reference corpus contains $61,343$ words; the LM-2g oracle hypotheses only contain $57,194$ words, resulting in a brevity penalty value of 0.93. In comparison, the LB-2g hypotheses contain $61,341$ words and incur no brevity penalty. As can be easily verified, the value of BP alone is not enough to compensate the gap between the performance of LM and other oracles. Indeed the LM's n-gram precisions are also worse than the ones of other oracles: for the German to English direction, the 1-gram precision is 60.4 for LM-2g oracle and 67.6 for the LB-2g oracle.

The PB and PBℓ oracles often performed comparably to our new oracles at the cost, in the length-sensitive case, of a much larger decoding time. Nevertheless, BLEU scores of both PB oracles are only marginally different, so the PBℓ's conservative policy of pruning and, consequently, much heavier memory consumption makes it an unwanted choice. The ILP oracle decoder always managed to find an exact solution in a few seconds, even if solving ILP problems is known to be NP-hard. Likewise, the RLX decoder found a solution in which all clipping constraints are enforced in more than 95% of the cases. All in all, the two ILP-based optimization methods introduced to solve the oracle decoding problem with clipping constraints achieve similar performance, even if the exact solutions found by the ILP oracle decoder are slightly better.

6.3. Sentence-Level Evaluation

As shown in the previous section, oracle hypotheses can achieve high corpus BLEU scores. This global evaluation only tells us part of the story, and could, in principle,

[10] As various oracle decoders have been implemented in different programming languages and their implementations have not always been carefully optimized, a direct comparison of running times is not relevant. Some information regarding running times is however given in [Sokolov et al. 2012].

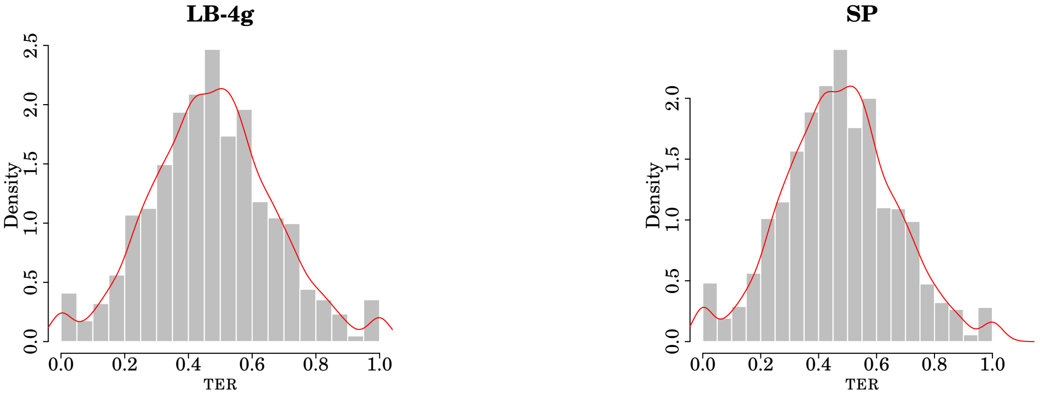


Fig. 4. Density estimation of the TER scores' of oracle hypotheses found by the SP and LB-4g decoders in the German to English lattices generated by N-code

correspond to several situations, depending on the distribution of the n-gram matches that make up the corpus BLEU scores. Two extreme cases would, for instance, be that: i) for all the test sentences, it is possible to find oracles of good quality or ii) some sentences imply very good oracles, while for others only poor quality oracles are found. In trying to distinguish between these two alternatives, we have conducted a sentence-level evaluation using the TER metric [Snover et al. 2006]. In brief, the TER score is a length-normalized extension of the standard edit distance, which gives a low cost to block movements; TER scores are thus in the interval $[0 \dots 1]$; a TER of 0 means that the oracle and the reference translation are identical. Figure 4 plots the distribution of the TER scores of the oracle hypotheses obtained using the LB-4g and the SP oracle decoders for the German to English translation direction using N-code lattices. Similar distributions were observed for other language pairs and other oracle decoders.

As can be seen in Figure 4, only very few oracle hypotheses closely match their references: the overwhelming majority of hypotheses still require a moderate number of edit operations to be transformed into their reference and there is a tail of hypotheses that are very different from the reference and have a TER score of 1.[11] Table IV shows example of such oracle hypotheses. It appears that these differences mainly result from non-literal and erroneous translations: as pointed out by [Wisniewski et al. 2010; Wisniewski and Yvon 2013], not all references can be produced by translating the source sentence.

6.4. Oracle Diversity

Experiments reported above show that the word lattices contain at least one high quality hypothesis. However, they do not give any information about the quality of the other hypotheses contained in the lattice. In this subsection, we describe two experiments that aim at evaluating the 'general' quality of a lattice by first looking at the similarity between the hypotheses found by the different oracle decoders and then by analyzing the n-best lists found by the standard and the oracle decoders.

6.4.1. Similarity between Oracle Hypotheses. The various oracle decoders studied in this work have shown to achieve comparable performance in term of BLEU scores. A natural question raised by their comparison is the one of the similarity between the different

[11] Recall that TER score are clipped to 1 and that a TER score of 1 means that the number of operations is larger than the number of words in the translation hypothesis.

Table IV. Oracle hypotheses found by the LB-4g decoder in N-code lattices that are the most different from the reference according to their TER score

①	source	Wir liegen etwa um fünf Prozent besser als im Vorjahr.
	reference	We’re up about 5 percent.
	oracle	We are about five percent better than last.
②	source	So bleibt uns nichts Anderes übrig, als uns zu wehren.
	reference	So all we can do is defend ourselves.
	oracle	So there is nothing different other than us to defend ourselves.
③	source	In unserem Angebot ist etwa die Hälfte der Plätze immer noch frei.
	reference	Half of our capacity is still available.
	oracle	Our offer is roughly half of the seats is still free.
④	source	Dieser Raketentyp ermöglicht den Abschuss aus dem U-Boot.
	reference	This type of rocket can be launched from a craft which is moving and even submerged.
	oracle	This Raketentyp allows the rocket from the submarine.

Table V. Average TER scores between the hypotheses found by all possible pairs of oracle decoders.

	LM-2g	LM-4g	ILP	LB-2g	LB-4g	PB	PBℓ	RLX	SP
LM-2g	—	0.01	0.35	0.36	0.36	0.37	0.37	0.35	0.16
LM-4g	0.01	—	0.35	0.36	0.36	0.37	0.37	0.35	0.16
ILP	0.40	0.40	—	0.14	0.16	0.21	0.22	0.12	0.34
LB-2g	0.41	0.41	0.14	—	0.09	0.18	0.19	0.15	0.34
LB-4g	0.41	0.41	0.16	0.09	—	0.14	0.16	0.17	0.34
PB	0.39	0.38	0.19	0.17	0.13	—	0.04	0.19	0.35
PBℓ	0.38	0.38	0.20	0.18	0.14	0.04	—	0.19	0.35
RLX	0.39	0.39	0.11	0.15	0.16	0.20	0.20	—	0.33
SP	0.19	0.19	0.33	0.34	0.34	0.37	0.37	0.34	—

oracle hypotheses, that will also, indirectly, give some information about the number of ‘good’ hypotheses found in a lattice. The similarity between two oracle hypotheses can be again estimated using the TER metric.

Table V thus reports the average TER score between the hypotheses found by all possible pairs of oracle decoders in the German to English lattices generated by N-code. It appears that the solutions of the best oracles are relatively close as their average TER is pretty small. However, when considering under-performing oracles, the hypotheses start to be very different and, even when the (average) TER is small, there are outliers for which the TER score remains very high (i.e. a lot of operations are required to transform the solution of one oracle into the solution of the other). When looking more precisely at the operations used to compute the edit distance (and their frequency) no clear trend is observed: the most frequent operations concern punctuations or stop words and most edit operations are observed only once.

6.4.2. General Quality of the Lattices. The results of previous section suggest that all oracle hypotheses are very similar, which might indicate that the lattices contain only a few good translations and that most of the other hypotheses are of worse quality. To test this hypothesis, we propose to look at the evolution of the BLEU score in n-best lists of both decoder hypotheses and oracle hypotheses to assess the scarcity of oracle translations and to get an idea of the ‘general’ quality of lattices. All the results reported in this section were obtained considering N-code lattices for the French to English language pair and LB-4g oracle decoder.

Decoder n-best Lists. As pointed out in Section 6.2, the oracles found in lattices are much better than oracles found in n-best list. A more detailed evaluation of the quality

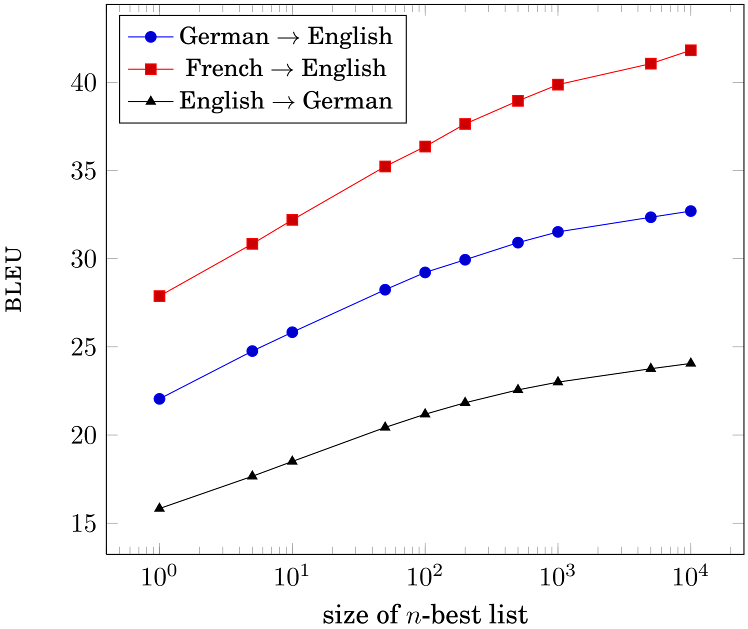


Fig. 5. n-best list oracle performances for N-code's lattices with respect to the size of the n-best list.

of the hypotheses of a n-best list is represented in Figure 5 which plots, for different n-best list sizes, the quality (corpus-BLEU) of the best hypothesis found in this n-best list. It appears that even when considering $10,000$-best lists, the best hypothesis found by the decoder is still outperformed by the oracle hypothesis by several BLEU points. These observations show that the gap in quality between oracle and decoder hypotheses is large and suggest that their feature representations are very different.

Oracle n-best Lists. Recall that all oracle decoding algorithms discussed in this paper, except for the ILP and RLX oracles, can be formalized as shortest distance algorithms for the appropriate semiring. The shortest distance algorithm can be extended to search, at almost no additional computational burden, for a list of n-best shortest paths [Mohri 2002]. This extension allows us to efficiently compute lists of n-best oracle hypotheses.

To characterize the quality of this 'oracle n-best list', we have represented, in Figure 6, the evolution of the (corpus) BLEU score when considering, for each sentence, its i-th best oracle hypothesis. It appears that most n-best oracle hypotheses are pretty good: even when considering the 10,000th best oracle, the quality of the oracle hypotheses, as estimated by the corps BLEU score, is much higher than the quality of the decoder hypotheses. Even if there is a sharp drop in oracle quality at the beginning of the oracle n-best list, translation quality quickly stabilizes itself and the quality of all first oracles 10,000 is within 30% of the quality of the best oracle. For instance, for the French to English direction, the 100th oracle hypotheses is outperformed by the 1st oracle hypotheses by almost 3 BLEU point, but another drop of 1 BLEU point is not achieved before going to the 500th oracle hypotheses.

A more precise evaluation of the quality of oracle n-best lists can be performed by assessing the loss in quality as we move away from the optimal oracle translation. We

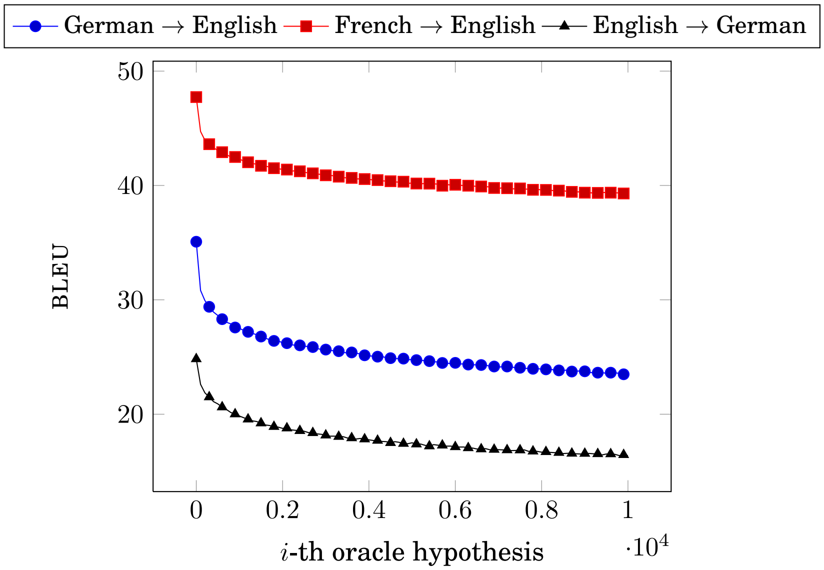


Fig. 6. Corpus BLEU score achieved when considering the i-th best oracle hypothesis. Only lattices generated by N-code have been considered.

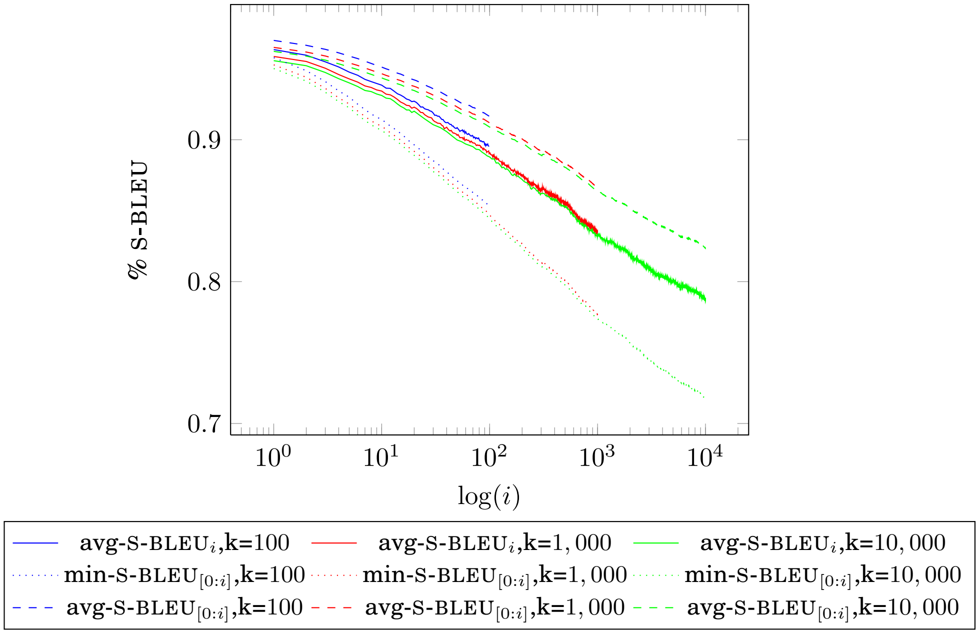


Fig. 7. Evolution of relative scores of oracle hypotheses.

propose three ways to estimate this loss:

$$\text{avg-S-BLEU}_i = \frac{1}{\#\{\mathcal{F}\}} \sum_{\mathbf{f}\in\mathcal{F}} \frac{\text{S-BLEU}(h_i)}{\text{S-BLEU}_{\textbf{oracle}}(\mathbf{f})} \tag{30}$$

$$\text{min-S-BLEU}_{[0:i]} = \min_i \frac{1}{\#\{\mathcal{F}\}} \sum_{\mathbf{f}\in\mathcal{F}} \frac{\text{S-BLEU}(h_i)}{\text{S-BLEU}_{\textbf{oracle}}(\mathbf{f})} \tag{31}$$

$$\text{avg-S-BLEU}_{[0:i]} = \frac{1}{i} \sum_{j=0}^{i} \frac{1}{\#\{\mathcal{F}\}} \sum_{\mathbf{f}\in\mathcal{F}} \frac{\text{S-BLEU}(h_i)}{\text{S-BLEU}_{\textbf{oracle}}(\mathbf{f})} \tag{32}$$

where $\mathcal{F}$ denotes the set of all source sentences, h_i is the hypothesis of the lattice with the i-th highest S-BLEU score, $\text{S-BLEU}_{\textbf{oracle}}(\mathbf{f})$ is the best BLEU score that can be achieved for a sentence **f**. The first value (Equation (30)) directly assesses the degradation of quality as a function of the position i; the second one (Equation (31)) can be seen as the minimum quality of oracles up to position i, while the last one (Equation (32)) describes the average quality of oracles up to position i.

As can be seen in Figure 7, the oracle translations enjoy only a slow degradation of quality within the oracle n-best list. Even the worst oracle translations (lower curves) in a 1,000-best list are within about 80% quality of the best oracle translation. If the dependency continues to be almost linear for larger values of n, then the 50% drop in quality (that approximately corresponds to the decoder's performance) would occur not earlier than for $n = 10^6$. Oracle translations therefore represent a very abundant source of high quality[12] translation examples.

7. FAILURE ANALYSIS

Oracle decoding can be used to identify and analyze the limits of existing SMT systems by comparing the hypothesis generated by a system with the best hypothesis it could have produced. Oracle decoders presented in this work have a main advantage with respect to the ones introduced in previous work (e.g. in [Wisniewski et al. 2010]): as they directly explore the decoder's approximation of the true search space, they can deliver the features associated to the best achievable hypotheses. This provides us a way to directly compare the internal scores of oracle and decoder hypotheses, which can help understand why the oracle hypothesis is not found by decoders. A detailed comparison of the features of oracle and decoder hypotheses is presented in Section 7.1, the impact of search space pruning is assessed in Section 7.2 and the evolution of oracle score during MERT training is studied in Section 7.3.

All experiments were conducted on WMT'11 datasets, the decoders and language pairs used will be mentioned as needed in each subsection.

7.1. How do Oracle Hypotheses Differ from 1-best?

In this section, we focus on the N-code decoder and the French to English language pair. In order to compare features of oracle and decoder hypotheses, we represent in Figure 8 the scatter plot of the feature values for different hypotheses: for a given feature, each sentence is represented by a point (x, y): the abscissa x is the feature value of the decoder hypothesis and its ordinate y is equal to the feature value of the corresponding oracle hypothesis. Note that all values have been normalized by the target sentence length, to make them comparable across different hypotheses. In order to limit the impact of outliers, only the 90% of hypotheses closest to the origin

[12] At least when quality is measured by BLEU.

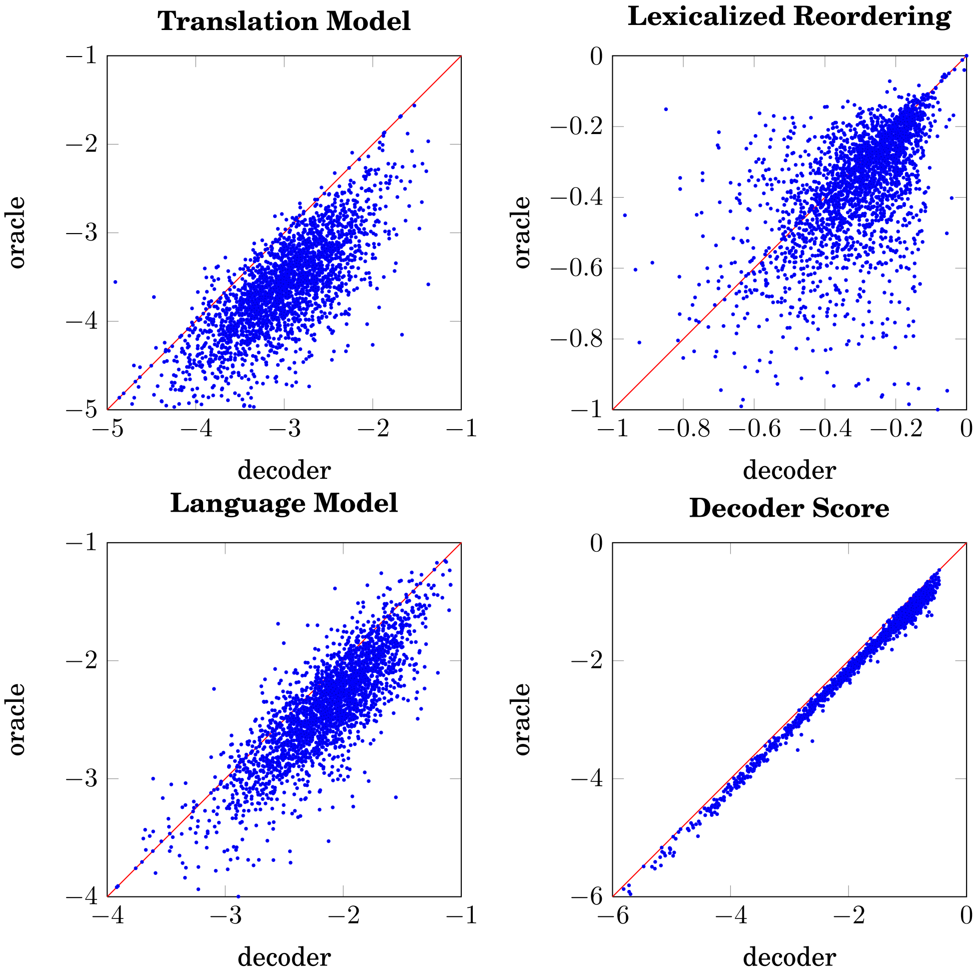


Fig. 8. Comparison between model scores of the decoder and oracle hypotheses.

have been represented. For clarity reasons, only the scatter plots of the three following features are shown: target language model, translation model and a lexicalized reordering model [Crego and Yvon 2010]. Additionally, the scatter plot of the 'global' decoder score (i.e. the weighted average of the feature values used to score hypotheses) is also represented.

The comparison between the scores of oracle and decoder hypotheses shows that the score of a decoder hypothesis is always higher than the score of the corresponding oracle hypothesis. This is graphically reflected by the scatter plot of the global score being located under the main diagonal ($y = x$). This is expected, since no hypothesis in the pruned search space can outscore the 1-best.

Figure 8 shows that the score of the language model is highly correlated in decoder and oracle hypotheses, contrarily to the score of the distortion feature. The translation model scores seems to be an intermediate case. More precisely, the Pearson correlation coefficient between the language model score of the decoder and oracle hypotheses is very high ($r = 0.810$), slightly lower for the translation model ($r = 0.740$) and much

Table VI. Influence of the Moses distortion limit parameter on the oracle quality (French → English)

distortion limit	avg. number of arcs ($\times 10^3$)	BLEU decoder	BLEU oracle
0 (monotone)	85.5	26.26	51.88
1 (swap)	85.5	26.29	51.88
2	68.8	26.93	54.20
3	52.4	27.24	55.39
4	39.9	27.56	55.49
5	31.4	27.58	54.78
6 (default)	25.1	27.68	53.82
7	20.5	27.55	52.83
8	17.0	27.69	51.87
9	14.5	27.41	50.83
10	12.6	27.23	49.93
unlimited	8.1	25.08	42.83

smaller for the reordering model ($r = 0.555$). More importantly, while the score of the language and translation models in an oracle hypothesis are almost always smaller than that in the corresponding decoder hypothesis (in more than 95% of the cases for the language model and 99% for the translation model), the score of the reordering model in an oracle hypothesis can be higher (in almost 40% of the cases) or much lower than its score in the corresponding decoder hypothesis.

All these observations highlight some of the limits of current PBSMT systems. Current reordering models seem to be particularly flawed, as they are not able to distinguish oracle hypotheses from decoder hypotheses, which means that the scores they provide are only poorly related to the actual translation quality. Translation and language models seem to play a primary role in the choice of the best hypotheses as their score is always larger in the decoder hypotheses. Their importance is probably overestimated, and their reliability should also be questioned. This is especially true for the translation model, since we sometimes find good oracle hypotheses having a rather poor translation model score.

7.2. Assessing the Impact of Search Parameters

To reduce the overall complexity of decoding, the search space is typically pruned using simple heuristics. For instance, the state-of-the-art phrase-based decoder Moses [Koehn et al. 2007] considers only a restricted number of translations for each source phrase and enforces a strict distortion limit which controls the span of possible reorderings. Oracle decoding offers a new way to evaluate the impact of this pruning.[13]

Tables VI and VII show, for two translation directions (French to English and English to German), the impact of Moses's distortion limit parameter (`-dl`) on the decoder score, on the oracle score of the PB oracle decoder and on the lattice size.[14] Surprisingly enough, for the two translation directions considered, increasing the distortion limit reduces the lattices size. A possible explanation is that with a higher distortion limit, reordering scores become more discriminant and, consequently, more hypotheses can be pruned. Another rather counter-intuitive observation is that the distortion limit has only a very small impact on both the decoder and oracle quality: even for language pair requiring long reorderings, such as in the German to English task, increasing the distortion improves the decoder score by less than 1 BLEU point at best and monotone decoding is only marginally worse.

[13] [Wisniewski et al. 2010] and [Wisniewski and Yvon 2013] have already evaluated the oracle performance of phrase-based systems when their search space is not pruned.

[14] The distortion limit is changed only during the production of the test lattice and not during training.

Table VII. Influence of the Moses distortion limit parameter on the oracle quality (German → English)

distortion limit	avg. number of arcs ($\times 10^3$)	BLEU decoder	BLEU oracle
0 (monotone)	82.2	20.91	42.28
1 (swap)	82.2	20.91	42.28
2	70.3	21.02	43.96
3	49.5	21.32	45.14
4	36.5	21.57	45.57
5	28.6	21.86	45.35
6 (default)	18.9	21.85	44.69
7	15.9	21.85	43.93
8	13.8	21.80	42.99
9	12.3	21.76	42.09
10	8.8	21.63	41.02
unlimited	8.1	19.87	35.08

Another pruning parameter is the threshold for beam pruning that controls the number of hypotheses kept in each stack during decoding [Koehn 2010]: hypotheses whose score is worse than the best in a stack by this factor are dropped. The impact of this parameter for Moses's lattices in the French to English task is described Table VIII. In this experiment, the oracle hypotheses have been obtained by the LB-2g decoder using the values of p and r described in Table II.

As expected, the threshold for beam pruning has a huge impact on the lattice size and consequently on the oracle BLEU score: reducing the beam threshold to 0.2 (recall that in all experiments presented so far, the beam threshold was set to 0.5) results in a 10 BLEU points increase and in lattices with ten times more edges. However, further reducing the beam hardly changes the quality of oracle hypotheses while the lattice size continues to grow quickly, making oracle decoding slower and slower. Using the default value of the beam threshold (10^{-5}) results in lattices that are so large that oracle decoding becomes impractical. Results presented in Table VIII also show that beam threshold has almost no impact on 1-gram precision and only hurts higher order n-gram precisions, suggesting that the oracle hypotheses found for the different values of the beam threshold differ only in their word order. It therefore appears that the main effect of using a larger beam during decoding is to increase the number of reorderings explored by Moses and not the number of translation hypotheses considered for each source phrase. In comparison, pruning[15] hardly impacts the quality of N-code lattices: in the configuration used in our experiments, the lattices contain $3,612,454$ edges[16] and the oracle BLEU score is 47.71; reducing pruning by a factor of 3 results in lattices containing $23,360,698$ edges and in a oracle BLEU score of 50.48.

7.3. Evolution of Oracle Scores during MERT Training

Until now, only 'final' lattices, produced by fully trained systems, have been considered. The scoring function used by these systems to select the best translation hypothesis results from a computationally intensive training procedure like MERT [Och 2003] or MIRA [Chiang et al. 2008] that requires to iteratively translate the complete training set. In this section, we study the impact of parameter training on oracle decoding: rather than looking only at lattices generated with optimal parameter values, we also consider lattices generated at different steps of the training process.

[15] In practice, N-code uses a histogram pruning strategy, in which only the n best hypotheses in a stack are kept. The beam value is therefore not directly comparable to Moses's which, as previously described, implements a threshold pruning strategy.
[16] The number of edges is accumulated over the whole dataset.

Table VIII. Impact of the Moses's beam threshold value on the LB-2g oracle decoder's performance (French → English); the number of edges is accumulated over the whole corpus

beam threshold	# edges	oracle BLEU	n-gram precisions
0.001	62,555,362	52.77	76.2/60.4/46.3/36.4
0.01	62,394,971	52.78	76.3/60.4/46.3/36.4
0.1	49,127,382	53.18	76.9/60.8/46.7/36.7
0.2	33,347,219	53.14	77.5/60.7/46.5/36.5
0.3	17,673,431	51.32	77.3/59.3/44.9/34.8
0.4	7,586,123	47.92	76.4/56.8/42.0/31.8
0.5	3,154,311	43.82	74.5/52.9/37.9/28.0

Table IX. Evolution of (corpus) 4-BLEU scores during MERT training for the French to English pair

MERT iteration	decoder	ILP oracle	LB-2g oracle
1	24.28	–	43.40
2	24.90	42.01	44.01
3	27.47	43.73	45.90
4	27.63	43.08	45.67
5	27.63	43.06	45.68

Table X. Evolution of (corpus) 4-BLEU scores during MERT training for the German to English pair

MERT iteration	decoder	ILP oracle	LB-2g oracle
1	17.14	27.91	27.84
2	19.20	28.59	30.50
3	20.23	31.15	31.03
4	20.46	30.35	31.28
5	20.54	30.06	31.12
6	20.72	30.40	31.21
7	20.72	30.39	31.22
8	20.72	30.39	31.22
9	20.73	30.40	31.22

Tables IX and X report the 4-BLEU scores achieved by N-code and two oracle decoders for the French to English and German to English tasks across iterations of the training process (calculated on the WMT'09 dev data). Similarly to most state-of-the-art systems, N-code relies on the MERT procedure: starting with well-chosen initial weights, the decoder is used to generate n-best lists for each training sentences; weights are then updated so as to find the best (corpus) BLEU score achievable for this set of n-best lists; these two steps are repeated until convergence.

It appears that from the very beginning, very good oracle translations are present in the search space of the decoder and that the quality of the search space, as evaluated by the quality of the best hypothesis it contains, improves only slightly in the course of training. This observation can be of practical importance for machine learning algorithms, as it suggests that there might be no need to regenerate lattices between parameter updates, dispensing with repetitive and expensive calls to the decoder. Similar arguments have already been made to justify the use of MERT on lattices [Macherey et al. 2008].

8. CONCLUSION

In this paper, we have performed a systematic comparison of various ways to compute oracle hypotheses in word lattices for phrase-based MT systems. In particular, we have proposed two original methods for finding such oracle translations, based, respectively, on finite-state automata and on Integer Linear Programming techniques. We have also proposed a variant of the latter approach based on Lagrangian relaxation, that dispenses with using a third-party ILP solver. These new oracle decoders rely on better approximations of BLEU than was previously done, taking the corpus-based nature of BLEU or clipping constraints into account. Experiments considering several decoders and translation directions show that the proposed oracles improve over the existing ones in terms of translation quality.[17]

Our experiments also demonstrate how oracle decoding can be used for understanding the limits of current PBSMT systems. For instance, we have presented evidence that current reordering models are at pain to discriminate good translation hypotheses from bad ones and that they only have a small impact on translation quality. Translation and language models, on the contrary, seem to have a crucial influence but fail as well to correctly score hypotheses according to their quality. Overall, our results highlight the scoring problems that plague existing decoders: for most input sentences, very good hypotheses are present in the search space, but are poorly evaluated by a linear combination of standard feature functions. It is important to realize that these observations do not necessarily imply that using high-BLEU oracle hypotheses instead of references will actually help training: as discussed in [Chiang 2012], these so-called pseudo-references also need to have good model scores to ensure a smooth convergence of learning algorithms. Finally, oracle hypotheses can only be helpful if their features can be shown to decompose *consistently* throughout the corpus, so that a set of weights that would uniformly give a high score to all pseudo-references can be found. Further analyses and experiments are thus required to get a better understanding of the potential impact of using oracle hypotheses as pseudo-references in training. Our observation that the search space often contains many good hypotheses to choose from suggests that learners actually have additional degrees of freedom during training; this provides us with some hope that better pseudo-references can finally turn into better learning.

The oracles introduced in this work can be extended in several ways. A possible extension would be to take several references into account. This can be done, for instance, for the LB-oracles by replacing the definition of the indicator function $\delta_u(\mathbf{r})$ (in Equation (12)) to fire whenever the n-gram u is found in at least one of the references in case there are several of them. Our experiments have shown that using a sentence-level 2-BLEU approximations allows to find very good oracle translations, provided the right choice of hyper-parameters. This suggests that the proposed methods can be extended straightforwardly to oracle decoding for hyper-graphs [Li and Khudanpur 2009]. Another extension would be to consider other metrics such as Meteor [Denkowski and Lavie 2011] or TERplus [Snover et al. 2009] that rely on fuzzy unigram matchings between the hypothesis and a reference. Rewarding fuzzy matchings is easily done in our framework; however, taking into account global factors such as the fragmentation in Meteor, raises new computational challenges and demands new approximations.

Our work also opens new perspectives for discriminative training of large scale PBSMT systems: weight updates in discriminative methods are made towards the best achievable translation that, for computational reasons, has, until now, always been computed on n-best lists [Liang et al. 2006; Arun and Koehn 2007]. This work shows that this approximation can be avoided and the 'true' oracle translation be used to compute the update, such as advocated by [Chiang 2012].

[17] At least when quality is loosely measured with automatic metrics such as 4-BLEU.

REFERENCES

Alexandre Allauzen, Hélène Bonneau-Maynard, Hai-Son Le, Aurélien Max, Guillaume Wisniewski, François Yvon, Gilles Adda, Josep Maria Crego, Adrien Lardilleux, Thomas Lavergne, and Artem Sokolov. 2011. LIMSI@WMT11. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, Scotland, 309–315.

Cyril Allauzen, Shankar Kumar, Wolfgang Macherey, Mehryar Mohri, and Michael Riley. 2010. Expected Sequence Similarity Maximization. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Los Angeles, California, 957–965.

Cyril Allauzen, Mehryar Mohri, Michael Riley, and Brian Roark. 2004. A generalized construction of integrated speech recognition transducers. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Vol. 1. IEEE, Sapporo, Japan, 761–764.

Cyril Allauzen, Michael Riley, Johan Schalkwyk, Wojciech Skut, and Mehryar Mohri. 2007. OpenFst: A General and Efficient Weighted Finite-State Transducer Library. In *Proceedings of the International Conference on Implementation and Application of Automata*. Springer, Prague, Czech Republic, 11–23.

Abhishek Arun and Philipp Koehn. 2007. Online Learning Methods For Discriminative Training of Phrase Based Statistical Machine Translation. In *MT Summit XI*. EAMT, Copenhagen, Denmark, 15–20.

Michael Auli, Adam Lopez, Hieu Hoang, and Philipp Koehn. 2009. A systematic analysis of translation model search spaces. In *Proceedings of the Fourth Workshop on Statistical Machine Translation (StatMT '09)*. Association for Computational Linguistics, Athens, Greece, 224–232.

Graeme Blackwood, Adrià de Gispert, and William Byrne. 2010. Efficient path counting transducers for minimum bayes-risk decoding of statistical machine translation lattices. In *Proceedings of the ACL 2010 Conference Short Papers (ACL-Short '10)*. Association for Computational Linguistics, Uppsala, Sweden, 27–32.

Chris Callison-Burch, Philipp Koehn, Christof Monz, and Omar Zaidan. 2011. Findings of the 2011 Workshop on Statistical Machine Translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, Scotland, 22–64.

Yin-Wen Chang and Michael Collins. 2011. Exact Decoding of Phrase-Based Translation Models through Lagrangian Relaxation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Edinburgh, Scotland, UK., 26–37.

David Chiang. 2012. Hope and Fear for Discriminative Training of Statistical Translation Models. *J. Mach. Learn. Res.* 98888 (June 2012), 1159–1187.

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online Large-Margin Training of Syntactic and Structural Translation Features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Honolulu, Hawaii, 224–233.

Josep Maria Crego, Aurélien Max, and François Yvon. 2010. Local lexical adaptation in machine translation through triangulation: SMT helping SMT. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING '10)*. Association for Computational Linguistics, Stroudsburg, PA, USA, 232–240.

Josep Maria Crego and François Yvon. 2010. Improving Reordering with Linguistically Informed Bilingual n-grams. In *Coling 2010: Posters*. Coling 2010 Organizing Committee, Beijing, China, 197–205.

Josep Maria Crego, François Yvon, and José B. Mariño. 2011. Ncode: an Open Source Bilingual N-gram SMT Toolkit. *Prague Bull. Math. Linguistics* 96 (2011), 49–58.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic Metric for Reliable Optimization and Evaluation of Machine Translation Systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Edinburgh, Scotland, 85–91.

Markus Dreyer, Keith Hall, and Sanjeev Khudanpur. 2007. Comparing reordering constraints for SMT using efficient Bleu oracle computation. In *Proceedings of the NAACL-HLT 2007/AMTA Workshop on Syntax and Structure in Statistical Translation (SSST '07)*. Association for Computational Linguistics, Rochester, NY, USA, 103–110.

Gurobi Optimization. 2010. Gurobi Optimizer. (April 2010). Version 3.0.

Richard Karp. 1972. Reducibility among combinatorial problems. In *Complexity of Computer Computations*, R. Miller and J. Thatcher (Eds.). Plenum Press, New York, NY, USA; London, UK, 85–103.

Kevin Knight. 1999. Decoding complexity in word-replacement translation models. *Comput. Linguist.* 25, 4 (Dec. 1999), 607–615.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press, Cambridge, UK.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*. Association for Computational Linguistics, Prague, Czech Republic, 177–180.

Gregor Leusch, Evgeny Matusov, and Hermann Ney. 2008. Complexity of Finding the BLEU-optimal Hypothesis in a Confusion Network. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Honolulu, Hawaii, 839–847.

Zhifei Li and Sanjeev Khudanpur. 2009. Efficient extraction of oracle-best translations from hypergraphs. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers (NAACL-Short '09)*. Association for Computational Linguistics, Boulder, CO, USA, 9–12.

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

Chin-Yew Lin and Franz Josef Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of the 20th international conference on Computational Linguistics (COLING '04)*. Association for Computational Linguistics, Geneva, Switzerland, Article 501, 7 pages.

Wolfgang Macherey, Franz Och, Ignacio Thayer, and Jakob Uszkoreit. 2008. Lattice-based Minimum Error Rate Training for Statistical Machine Translation. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, Honolulu, Hawaii, 725–734.

Lidia Mangu, Eric Brill, and Andreas Stolcke. 2000. Finding consensus in speech recognition: word error minimization and other applications of confusion networks. *Computer Speech and Language* 14, 4 (2000), 373–400.

Mehryar Mohri. 2002. Semiring frameworks and algorithms for shortest-distance problems. *J. Autom. Lang. Comb.* 7 (2002), 321–350. Issue 3.

Mehryar Mohri. 2009. Weighted automata algorithms. In *Handbook of Weighted Automata*, Manfred Droste, Werner Kuich, and Heiko Vogler (Eds.). Springer, Berlin Heidelberg, Chapter 6, 213–254.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*. Association for Computational Linguistics, Morristown, NJ, USA, 160–167.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics, Philadelphia, PA, USA, 311–318.

Alexander M. Rush, David Sontag, Michael Collins, and Tommi Jaakkola. 2010. On dual decomposition and linear programming relaxations for natural language processing. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*. Association for Computational Linguistics, Cambridge, MA, USA, 1–11.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A Study of Translation Edit Rate with Targeted Human Annotation. In *Proceedings of the Conference of the Association for Machine Translation in the America (AMTA)*. Springer, Cambridge, MA, USA, 223–231.

Matthew Snover, Nitin Madnani, Bonnie Dorr, and Richard Schwartz. 2009. Fluency, Adequacy, or HTER? Exploring Different Human Judgments with a Tunable MT Metric. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Athens, Greece, 259–268.

Artem Sokolov, Guillaume Wisniewski, and Francois Yvon. 2012. Computing Lattice BLEU Oracle Scores for Machine Translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*. Association for Computational Linguistics, Avignon, France, 120–129.

Xingyi Song, Trevor Cohn, and Lucia Specia. 2013. BLEU deconstructed: Designing a Better MT Evaluation Metric. In *Proceedings of the 14th International Conference on Intelligent Text Processing and Computational Linguistics*. Springer, Samos, Greece.

Roy W. Tromble, Shankar Kumar, Franz Och, and Wolfgang Macherey. 2008. Lattice Minimum Bayes-Risk decoding for statistical machine translation. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP '08)*. Association for Computational Linguistics, Honolulu, HA, USA, 620–629.

Marco Turchi, Tijl De Bie, and Nello Cristianini. 2008. Learning Performance of a Machine Translation System: a Statistical and Computational Analysis. In *Proceedings of the Third Workshop on Statistical Machine Translation*. Association for Computational Linguistics, Columbus, OH, USA, 35–43.

Marco Turchi, Tiji De Bie, Cyril Goutte, and Nello Cristianini. 2012. Learning to Translate: a statistical and computational analysis. *Advances in Artificial Intelligence* 2012, Article 1 (2012), 15 pages.

Taro Watanabe. 2012. Optimized Online Rank Learning for 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, 253–262.

Guillaume Wisniewski, Alexandre Allauzen, and François Yvon. 2010. Assessing phrase-based translation models with oracle decoding. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing (EMNLP '10)*. Association for Computational Linguistics, Cambridge, MA, USA, 933–943.

Guillaume Wisniewski and François Yvon. 2013. Oracle decoding as a new way to analyze phrase-based machine translation. *Machine Translation* 27, 2 (2013), 115–138.

L. Wolsey. 1998. *Integer Programming*. John Wiley & Sons, Inc., New York, NY, USA.