Confidence-based Rewriting of Machine Translation Output

Benjamin Marie
LIMSI-CNRS, Orsay, France
Lingua et Machina, Le Chesnay, France
benjamin.marie@limsi.fr

Aurélien Max
LIMSI-CNRS, Orsay, France
Univ. Paris Sud, Orsay, France
aurelien.max@limsi.fr

Abstract

Numerous works in Statistical Machine Translation (SMT) have attempted to identify better translation hypotheses obtained by an initial decoding using an improved, but more costly scoring function. In this work, we introduce an approach that takes the hypotheses produced by a state-of-the-art, reranked phrase-based SMT system, and explores new parts of the search space by applying rewriting rules selected on the basis of posterior phrase-level confidence. In the medical domain, we obtain a 1.9 BLEU improvement over a reranked baseline exploiting the same scoring function, corresponding to a 5.4 BLEU improvement over the original `Moses` baseline. We show that if an indication of which phrases require rewriting is provided, our automatic rewriting procedure yields an additional improvement of 1.5 BLEU. Various analyses, including a manual error analysis, further illustrate the good performance and potential for improvement of our approach in spite of its simplicity.

1 Introduction

The standard configuration of modern phrase-based Statistical Machine Translation (SMT) (Koehn et al., 2003) systems can produce very acceptable results on some tasks. However, early integration of better features to guide the search for the best hypothesis can result in significant improvements, an expression of the complexity of modeling translation quality. For instance, improvements have been obtained by integrating features into decoding that better model semantic coherence at the sentence level (Hasan and Ney, 2009) or syntactic well-formedness (Schwartz et al., 2011). However, early use of such complex features typically comes at a high computational cost. Moreover, some informative features require or are better computed when complete translation hypotheses are available. This is addressed in numerous works on reranking of the highest scored sub-space of hypotheses, on so-called n-best lists (Och et al., 2004; Zhang et al., 2006; Carter and Monz, 2011) or output lattices (Schwenk et al., 2006; Blackwood et al., 2010), where many works specifically target the inclusion of better language modelling capabilities, a well-known weakness of current automatic generation approaches (Knight, 2007).

Another way to improve translation *a posteriori* can be done by rewriting initial hypotheses, for instance in a greedy fashion by including new models (Langlais et al., 2007; Hardmeier et al., 2012), or by specifically modeling a task of automatic post-editing targeting a specific system (Simard et al., 2007; Dugast et al., 2007). While such automatic post-editing may seem to be too limited, notably because of the limited initial diversity considered and the fact that it may be in some instances agnostic to the internals of the initial system, it has been shown to potentially improve accuracy of the new translation hypotheses (Parton et al., 2012) and to offer very high oracle performance (Marie and Max, 2013).

However, an important issue for such approaches is their capacity to only rewrite *incorrect* parts of the translation hypotheses and to use appropriate replacement candidates. Many works have tackled the issue of word to *n*-gram confidence estimation in SMT output (Zens and Ney, 2006; Ueffing and Ney, 2007; Bach et al., 2011; de Gispert et al., 2013), and some attempts have been made to exploit confidence estimates for lattice rescoring (Blackwood et al., 2010) or *n*-best reranking (Bach et al., 2011; Luong et al., 2014b).

In this work, we present an approach in which

Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 1261–1272,
October 25-29, 2014, Doha, Qatar.

new complete hypotheses are produced by rewriting existing hypotheses, and are scored using complex models that could not be used during the initial decoding. We will use as competitive baselines systems that rerank the output of an initial decoder using the complete set of available features, and will show that we manage to improve their translation. The difference between our approach and the reranking baseline lies in the manner in which we expand our training data, as well as in our use of high-confidence rewritings to obtain new translation hypotheses. Importantly, this work will only exploit simple confidence estimates corresponding to phrase-based posteriors, which do not require that large sets of human-annotated data be available as in other works (Bach et al., 2011; Luong et al., 2014b).

The remainder of this paper is organized as follows. Section 2 is devoted to the description of our approach, with details on our rewriting approach (2.1), additional features (2.2), rewriting phrase table (2.3), and training examples (2.4). Section 3 presents experiments. We first describe our experimental setup (3.1) and our baseline systems (3.2). We then report results when naive rewriting is performed and then with confidence-based rewriting (3.3). We next devote a significant part of the paper in section 4 to report further results and analyses: an analysis of the performance of our system depending on the quality of initial hypotheses (4.1); a semi-oracle experiment where correct phrases are known (4.2); an oracle experiment where only correct rewriting decisions are made (4.3); a manual error analysis of the main configurations studied in this work (4.4); and, finally, a study of the performance of our approach on a more difficult translation task (4.5). Related work is discussed in section 5 and we conclude and introduce our future work in section 6.

2 Description of the approach

2.1 Rewriting of translation hypotheses

Langlais *et al* (2007) proposed a greedy search procedure to improve translations by reusing the same translation table and scoring function that were used during an initial phrase-based decoding. In our approach, we rewrite hypotheses by using the same greedy search algorithm, adding more complex models and using the most-confident bi-phrases according to the initial decoder's search space. To select the hypothesis to rewrite for each sentence, we produce a n-best list of the initial decoder and rerank this list with a new, better informed scoring function (see section 2.2). The one-best hypothesis obtained after reranking is then rewritten by our system (denoted as `rewriter`). In this way, we ensure that the hypothesis that was rewritten had been so far the best one according to the initial decoding best subspace and the new models used.

At each iteration, new hypotheses are obtained from a current hypothesis by applying one rewriting operation on bi-phrases. The set of all new hypotheses is called the neighborhood of the current hypothesis. Focusing in this work on *local rewriting*, we used the following set of operations (N denotes the number of bi-phrases, T the maximum number of entries per source phrase in a rewriting phrase table (see 2.3), and S the average number of tokens per source phrase)[1]:

1. `replace` ($\mathcal{O}(N.T)$): replaces the translation of a source phrase with another translation from the rewriting phrase table;

2. `split` ($\mathcal{O}(N.S.T^2)$): splits a source phrase into all possible sets of two (contiguous) phrases, and uses `replace` on each of the resulting phrases;

3. `merge` ($\mathcal{O}(T.N)$): merges two contiguous source phrases and uses `replace` on the resulting new phrase.

This rewriting algorithm is described in pseudo-code in Algorithm 1.

Algorithm 1 `rewriter` Algorithm

Require: *source* a sentence to translate

```
nbestList ← TRANSLATE(source)
oneBest ← RERANK(nbestList)
sCurrent ← GET_SCORE(oneBest)
loop
   hypothesesSet ← NEIGHBORHOOD(oneBest)
   newOneBest ← RANK(hypothesesSet)
   s ← GET_SCORE(newOneBest)
   if s ≤ sCurrent then
      return oneBest
   else
      oneBest ← newOneBest
      sCurrent ← s
   end if
end loop
```

[1] Complexity is expressed in terms of the maximum number of hypotheses that will be considered given some hypothesis to rewrite.

The produced hypotheses are then ranked according to a new, better informed scoring function (see 2.2). At the next iteration, the hypothesis now ranked at the top of the list is rewritten, and search terminates when no better hypothesis is found.

Such a greedy search has several obvious limitations, in particular it can only perform a limited exploration of the search space, a situation that can be improved by using a beam (see Section 3.3). However, associated with a small and precise rewriting phrase table, this approach only visits small numbers of more-confident hypotheses, which is a critical property given the cost of computing the new scoring function used.

2.2 Reranking and features

The rerankings of the hypotheses sets describe in this work are all performed with `kb-mira` (Cherry and Foster, 2012) using the initial features set of the decoder in conjunction with the following additional features:[2]

- **`SOUL` models**: `SOUL` models are structured output layer neural network language models (LMs) which have been shown to be useful in reranking tasks, for instance for WMT evaluations (Allauzen et al., 2013; Pécheux et al., 2014). `SOUL` scoring being too costly to be integrated during decoding, it fits perfectly the `reranker` scenario, which furthermore enables to use larger contexts for *n*-grams. We used both monolingual (Le et al., 2011) and bilingual (Le et al., 2012) `SOUL` 10-gram models, which were trained on the WMT'12 data.

- **POS language model**: part-of-speech (POS) LMs have been shown to yield improvements in *n*-best list reranking (Carter and Monz, 2011). In this work, we trained a 6-gram POS LM using Witten-Bell smoothing.

- **IBM1** : the IBM1 scores ($p(e|f)$ and $p(f|e)$) of the complete hypothesis (Och et al., 2004).

- **phrase-based confidence score** : bi-phrases are associated to a posterior probability, inspired from *n*-gram posterior probability estimation as defined in (de Gispert et al., 2013). Let E be the set of all hypotheses in the space of translation hypotheses defined by the n-best list used for source sentence f, and E_α be the subset of E such that word alignments in sentence pairs (e', f), $\forall e' \in E_\alpha$, allow us to extract bi-phrase α. Let also $H(e, f)$ be the score assigned by a baseline decoder (denoted as `1-pass Moses` henceforth) to sentence pair (e, f). We use the following posterior probability for α:

$$P(\alpha|F) = \frac{\sum_{e' \in E_\alpha} exp(H(e', f))}{\sum_{e'' \in E} exp(H(e'', f))} \quad (1)$$

Then, the logarithms of each phrase's confidence score are summed to use as a confidence score for the complete hypothesis.

[2]Note that we did not try to explore the independant contribution of each feature in this work.

2.3 Rewriting phrase table

Taking the whole translation table of the decoder as a rewriting phrase table to perform the greedy search produces very large neighborhoods that `rewriter` cannot handle due to the cost of the models that have to be computed. We tried two different approaches to extract a rewriting phrase table from the translation table of the system.

We first tried a naive approach where the rewriting phrase table of `rewriter` for the test set uses the phrase table of `1-pass Moses`, filtered to keep the k best entries according to the direct translation model. We denote such a configuration `rpt`k`pef`.

Our second approach consists in extracting the rewriting phrase table containing bi-phrases that were the most probable according to the set of all models used in `1-pass Moses`. Selection of bi-phrases for each sentence is done in a binary fashion, depending on their presence in k-best lists of `1-pass Moses` for a given value of k. This configuration will be denoted `conf`k.

2.4 Training examples

We tried several sets of examples to train the ranker of `rewriter`. We used the 1,000-best list of the development set produced by `1-pass Moses` during its tuning. In other configurations we mixed *a*) the neighborhood of the `reranker` n-best hypotheses computed by our system on the development set using a rewriting phrase table containing the bi-phrases found in the k-best list produced by `1-pass Moses`; and *b*) the neighborhood of the one-best hypotheses of `reranker` using a rewriting phrase table containing the 10-best translations from the `1-pass Moses` translation table according to the direct translation

model. Both neighborhoods are produced by a single iteration of rewriter. We denote respectively these sets of hypotheses n-bestNeigh and 10PefNeigh. Our intuition behind the constitution of these training sets is that the ranker of rewriter needs, in order to perform well, training examples that will be similar to hypotheses that it actually generates.

3 Experiments

3.1 Experimental setup

We used two datasets from two different domains: the data provided for the WMT'14 medical translation task[3] (Medical) and a smaller task using the TED talks[4] (TED Talks) data of the IWSLT evaluation campaigns. For the Medical task we used only the English to French translation direction, and both translation directions, English to French and French to English, for the TED Talks task. In this work, the main part of our experiments uses Medical, and TED Talks will be used at a later stage to study a lower-quality situation (cf. 4.5). For the Medical task, initial decodings were produced using a LM trained on all WMT'14 monolingual and bilingual medical data, while for the TED Talks task we used a much larger LM trained on all the data provided for WMT'13[5]. Both are 4-gram LMs estimated with Kneser-Ney smoothing (Chen and Goodman, 1998). For the 6-gram POS LMs used (see 2.2), we used the same data as used for the token-based LM for Medical, and the concatenation of the News Commentaries and Europarl sub-parts of the WMT'13 data for TED Talks. Table 1 provides relevant statistics about the data used.

Tasks	Corpus	Sentences	Tokens (en-fr)
Medical	train	4.9M	78M - 91M
	dev	500	10k - 12k
	test	1,000	21k - 26k
	LM		- 146M
TED Talks	train	107 758	2M - 2.2M
	dev	934	20k - 20k
	test	1,664	31k - 34k
	LM		6B - 2.5B

Table 1: Corpora used in this work.

[3] http://www.statmt.org/wmt14/medical-task/

[4] https://wit3.fbk.eu/mt.php?release=2013-01

[5] http://www.statmt.org/wmt13

We first built a state-of-the-art phrase-based SMT system using Moses (Koehn et al., 2003) with standard settings. We tuned its parameters towards BLEU (Papineni et al., 2002) on the tuning dataset using the kb-mira implementation available in Moses with default parameters.

Our results will be compared using BLEU and TER (Snover et al., 2006) to *a*) the initial best translation produced by the Moses decoder (1-pass Moses) and *b*) the best translation obtained by reranking the 1,000-best list of 1-pass Moses (reranker). Since reranker implements a well-documented approach and uses types of features commonly used in reranking tasks we will consider it as our main baseline. It was trained using kb-mira on the 1,000-best of the development data decoded by 1-pass Moses.

In our experiments, rewriter rewrites the one-best hypothesis[6] produced by reranker using the operators Replace, Split and Merge as described in section 2.1.

3.2 Baseline results

Table 2 gives the results of the 1-pass Moses decoding for the Medical task and the reranking results of reranker applied to the 1-pass Moses 1,000-best list.

1-pass Moses obtains a score of 38.2 BLEU on the test set, which can be considered as a good baseline system.[7] reranker outperforms 1-pass Moses by 3.5 BLEU, indicating a strong performance of the features used on this task. In particular, SOUL is known to be a useful feature for reranking n-best lists on highly-inflected languages such as French. Note also that the SOUL models we used were trained on the WMT'12 monolingual and bilingual data and so were better informed than the models used during the 1-pass Moses decoding.[8] Moreover, as can be seen on Figure 1, the 1,000-best oracle reveals a large potential for improvement over the one-best (+12.4 BLEU). We further observe that the reranked list of reranker shows a much faster potential for translation improvement.

[6] Note that we will also provide results where a beam of *k*-best hypotheses are rewritten.

[7] Distribution of error types on a sub-part of the test set will be provided in section 4.4.

[8] However, SOUL considers only a small sample of the training data for training. For instance, the training of the French monolingual model used roughly only 1% (895K sentences) of all the WMT'12 data.

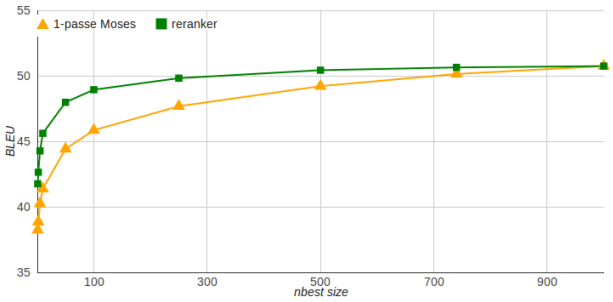


Figure 1: n-best list oracle for `1-pass Moses` and `reranker`

3.3 `rewriter` results

Results for the different rewriting phrase tables and training examples are given in Table 2. First, concerning the rewriting phrase table, for the k=5 (`rpt5pef`) and k=10 (`rpt10pef`) configurations[9] a decrease of 0.7-0.8 BLEU over `reranker` is obtained. This illustrates that naive rewritings applied on the test set cannot be used with our training regime to improve translation quality.

In the next experiments, we used a `conf`k rewriting table. Table 2[10] shows the results of `rewriter` when rewriting the one-best hypothesis from `reranker` for various values of k to define the k-best list from which the rewriting table is built. Various training sets are also considered in the table.

The `1-pass Moses` 1,000-best configuration reused the same set of hypotheses used to train `reranker`. For this configuration, `rewriter` loses 2.6 BLEU over `reranker` on the test set with `conf10k`. Of course, this training data set is of a quite different nature compared to the hypotheses built by `rewriter`.

In the 10pefNeigh training, the ranker is trained with the neighborhoods produced by the first iteration of `rewriter` on the development set with a rewriting phrase table containing only the k-best translations for each source phrase according to the direct translation model. This configuration improves over the previous one by 1.7 BLEU, but is still 0.9 BLEU below `reranker`. Adding the neighborhoods of the `reranker` n-best hypotheses produced with a `conf10k` rewriting phrase table to the training data does not improve over the previous situation for $n = 10$, but increasing n to 30 and then 50 produces strong improvements on the test set (resp. +1.4 and +1.6 BLEU). Considering a larger neighborhood obtained by rewriting the best $n = 90$ hypotheses does not yield further gains. We denote from now on `opti` our best configuration thus far, considering the performance on the development set and having the largest confidence-based rewriting phrase table considered.

Letting `rewriter` perform a beam search on the 10-best hypotheses of the test set, further gains are obtained, corresponding now to an improvement of +1.9 BLEU over our `reranker` baseline, or +5.4 BLEU over `1-pass Moses`.[11] Furthermore, although taking the bi-phrases from the 10,000-best is our best configuration, it is interesting to note that taking bi-phrases from the 10-best only already yields a moderate improvement of +0.6 BLEU over `reranker`. Figure 2a shows that up to $k = 10,000$ higher value of k to extract the rewriting phrase table increase the BLEU score on the test set. [12] We did not experiment with higher values of k, but plan to use the output lattice produced by `1-pass Moses` to compute efficiently posteriors for larger sets of bi-phrases (de Gispert et al., 2013).

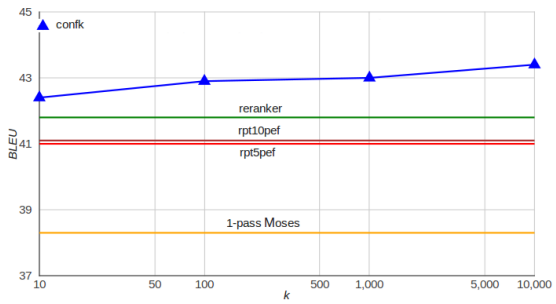
As illustrated on Figure 2b, `rewriter` mostly improves the BLEU score during the three first iterations and then converges at the ninth iteration. However, it is important to note that not all sentences are actually improved by our system. As illustrated on Figure 3a, `opti` improves 40.8% of the sentences of the test set but degrades 29.2% of them according to sentence-BLEU (Lin and Och, 2004). It is certainly the case that more informative confidence features may help idenfity more precisely which fragments of the translations should really undergo rewriting. We will investigate the exploitation of an oracle phrase-based confidence measure in Section 4.2.

[9] We did not experiment with higher values of k because of the computationnal cost of the features used by `reranker`. Indeed, adding more phrase translations increases the size of the neighborhoods corresponding to many additional n-grams to score by `SOUL`, the most expensive model.

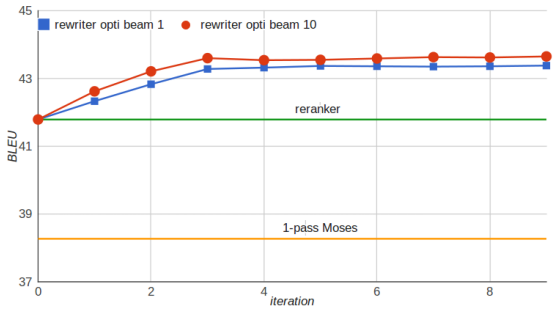
[10] In Table 2 the number of unique bi-phrases for the `rpt` rewriting phrase tables is computed by considering only source phrases appearing in the test set, for the n-best Neighborhood configurations we merged the phrase tables of each sentence into one and count just as one unique entry bi-phrases appearing several times.

[11] Using a beam becomes quickly prohibitive: using 12 threads, 25 mn vs. 3h were needed for the test set for the configurations of size 1 and 10, respectively.

[12] Note that even for $k = 10,000$ the computed neighborhoods are still quite small with an average of 116 hypotheses for each hypothesis to rewrite per iteration, against an average of 788 hypotheses for the `rpt10pef` configuration.



(a) Results of `rewriter` with `rpt5pef`, `rpt10pef` and different values of k for `confk`



(b) Iterations of `rewriter` on test with `opti` and two beam sizes : 1 and 10.

Figure 2: Performance of `rewriter` depending on the type of the rewriting phrase table and the number of iterations and beam sizes.

baseline				dev	test		
				BLEU	BLEU	TER	GOS BLEU
1-pass Moses				40.9	38.3	44.6	
reranker				44.1	41.8	41.6	
training data	rewriting phrase table	unique bi-phrases	beam size				
1-pass Moses 1 000-best	conf10k	38 455	1	44.1	39.2$_{(-2.6)}$	43.8$_{(+2.2)}$	58.7
10pefNeigh	conf10k	38 455	1	43.9	40.9$_{(-0.9)}$	41.2$_{(-0.4)}$	58.7
10-bestNeigh + 10pefNeigh	conf10k	38 455	1	43.8	40.9$_{(-0.9)}$	41.2$_{(-0.4)}$	58.7
30-bestNeigh + 10pefNeigh	conf10k	38 455	1	44.2	43.2$_{(+1.4)}$	40.6$_{(-1.0)}$	58.7
50-bestNeigh + 10pefNeigh	rpt5pef	85 530	1	44.5	41.0$_{(-0.8)}$	42.0$_{(+0.4)}$	50.6
=	rpt10pef	149 887	1	44.5	41.1$_{(-0.7)}$	42.1$_{(+0.5)}$	54.5
=	conf10	21 398	1	44.5	42.4$_{(+0.6)}$	41.0$_{(-0.6)}$	45.9
=	conf100	28 730	1	44.5	42.9$_{(+1.1)}$	40.8$_{(-0.8)}$	50.2
=	conf1k	33 929	1	44.5	43.0$_{(+1.2)}$	40.6$_{(-1.0)}$	53.3
= (opti)	conf10k	38 455	1	44.5	**43.4**$_{(+1.6)}$	**40.4**$_{(-1.2)}$	58.7
=	conf10k	38 455	10	44.5	**43.7**$_{(+1.9)}$	**40.1**$_{(-1.5)}$	59.6
90-bestNeigh + 10pefNeigh	conf10k	38 455	1	44.4	43.4$_{(+1.6)}$	40.4$_{(-1.2)}$	58.7

Table 2: Results on `Medical` for different training configurations, rewriting phrase tables and beam sizes. `opti` denotes our optimal configuration for `rewriter`.

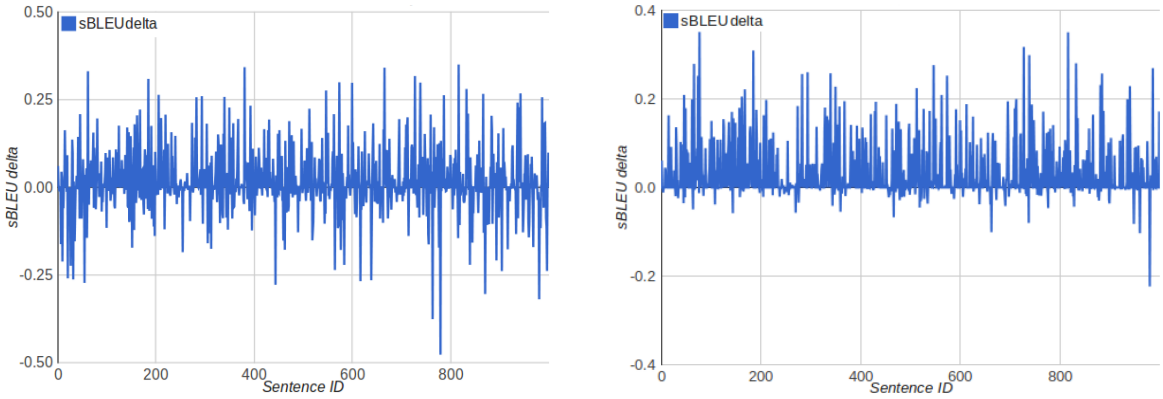
4 Analysis of confidence-based rewriting

4.1 Performance of `rewriter` depending on the quality of initial hypotheses

The first question we address in our analysis of `rewriter` is whether its performance depends on the difficulty of each individual sentence. As a proxy of sentence difficulty we used sentence-BLEU of `1-pass Moses`, and used it to divide the sentences of the test set into quartiles. Figure 4 shows that `reranker` improves more over `1-pass Moses` and that at the same time `rewriter` improves more over `reranker` as the sentences are more difficult. In particular, `rewriter` obtains a 8.6 BLEU improvement over `1-pass Moses` on the more difficult quartile, but only a 1.3 BLEU improvement on the least difficult quartile. We hypothesize that better performance may be achieved if adapting the training and rewriting of `rewriter` to sentences of varying quality, which may, for instance, be estimated with off-the-shelf estimators (Specia et al., 2013).

4.2 Semi-oracle experiments: rewriting only incorrect fragments

We observed in section 3.3 that our `opti` configuration, which obtains strong improvements in translation quality (as given by corpus-BLEU), in fact degrades (as given by sentence-BLEU) a significant proportion of sentences. To further analyze these results, we simulate a situation where oracle confidence information is available at the phrase-level: in particular, `rewriter` is prevented from rewriting bi-phrases whose target phrase appears exactly in the reference transla-



(a) automatic (b) semi-oracle

Figure 3: sBLEU delta, for each sentence, between the `reranker` one-best to rewrite and its automatic (3a) or semi-oracle (3b) rewriting computed by `rewriter` with the `opti` configuration.

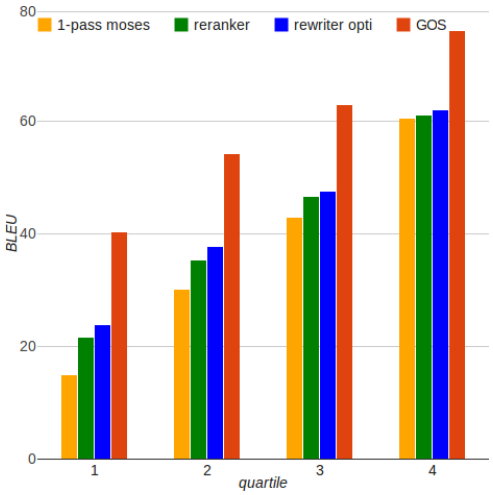


Figure 4: Source sentences were divided into quartiles according to sBLEU of the `1-pass Moses` system. For each quartile we reported the performance of `1-pass Moses`, `reranker`, `rewriter`, `GOS`.

tion.[13] Furthermore, this "freezing" of bi-phrases can be repeated after each iteration of `rewriter`.

Thus, we now have an oracle situation for choosing which source phrases may be rewritten, but the rest of the rewriting procedure is still fully automatic. Moreover, we purposefully did not adapt the training procedure to this new configuration, and reused `opti` as is. Results, reported in Table 3, indicate that an additional 1.5 BLEU is obtained from `opti`, or 3.1 BLEU from `reranker` and 6.6 BLEU from `1-pass Moses`. The use of a larger beam of size 10 did not improve those results any further. At the first iteration, `rewriter` "froze" approximatively 65.6% of the bi-phrases, and 70.5% at the last iteration, demonstrating the ability of `rewriter` to find good rewritings that match the reference translation. Looking at Figure 3b, we now find that, as expected, only a limited number of sentences are now degraded by `rewriter`. The large improvements obtained clearly underlines the important role that better confidence estimates could play in our framework.

System	test	
	BLEU	TER
`reranker`	41.8	41.6
`opti`	43.4	40.4
`semi-oracle`, beam 1	$44.9_{(+1.5)}$	$39.2_{(-1.2)}$
`semi-oracle`, beam 10	$44.9_{(+1.5)}$	$39.0_{(-1.4)}$

Table 3: Results for the semi-oracle using `opti`.

4.3 Oracle experiments: making only the correct decisions

We now turn to the situation where only rewritings that actually improve translation performance would be made. In practice, we use a simple solution: we resort to greedy oracle search (`GOS`) (Marie and Max, 2013), where sentence-BLEU is maximized using rewritings from the `opti` phrase table. At each iteration the rewriting in the neighborhood that maximizes sentence-BLEU is selected until convergence.

Results for this greedy search oracle appear in the last column of Table 2 and allow us to put in perspective the individual potential of the var-

[13] This is obviously not an optimal solution.

ious tested configurations. We can first notice that the rpt5pef phrase table allows the oracle to reach 50.6 BLEU, 8.1 BLEU below the oracle value obtained with conf10k, although rpt5pef contains twice as many bi-phrases. The same conclusion can be made about rpt10pef, which is 3.9 BLEU higher than rpt5pef but contains nearly twice as many bi-phrases. Finally, although conf10k contains approximatively four times fewer bi-phrases than rpt10pef, its oracle value is 4.2 BLEU higher. This points out the fact that conf10k is a lot more precise rewriting phrase table for the translations to rewrite, as well as the fact that rpt5pef and rpt10pef are much noisier and consequently difficult to use efficiently by our automatic rewriting procedure.

4.4 Manual error analysis

In the previous sections, we have shown that our automatic rewriting procedure can improve translation quality over both an initial Moses baseline, and a reranked baseline using the same features as our procedure. We have further shown in section 4.3 that much larger improvements could be obtained by using an oracle procedure.

We now focus on the four following configurations: 1-pass Moses, reranker, rewriter and GOS. Although this four configurations are well separated both in terms of BLEU and TER scores, it is informative to look more precisely into what makes their results different. We performed a small-scale manual error analysis of these four configurations. A French native speaker annotated 70 translation hypotheses using an error typology adapted from (Vilar et al., 2006).

Results of the manual error analysis are reported in Table 4. The most significant results are for the *disamb(iguation)* and *form* error types, the former being more related to translation accuracy, and the later to fluency. In both cases, we first observe a strong reduction of errors between 1-pass Moses and reranker, which demonstrates the positive impact of the features used on these levels. Then, another, similar reduction is obtained between reranker and rewriter, demonstrating that our reranking procedure manages to identify more precise and fluent hypotheses. Finally, a further reduction is found between rewriter and GOS, indicating that our proposed local, greedy rewriting can still be improved, notably by using more informative features and better confidence estimates.

The other types of error categories are less informative. We find no clear differences in error types attributable to style issues, which seem to be irrecoverable even for GOS. reranker and rewriter both improve on order-related errors over 1-pass Moses, but our local rewriting unsurprisingly did not fix any of these errors. Finally, reranker and rewriter decreased slightly the number of extra words from 1-pass Moses, while GOS sometimes artificially introduces extra words.

4.5 Lower-quality SMT experiments

We now turn to the question of how our rewriting system fares on a more difficult task, and used TED Talks, 6 BLEU below Medical for the English to French direction, for this purpose. In the same way as we did for Medical, we first tried to find the best training configuration for the ranker of the rewriting system. For this task, mixing the n-best neighborhood and 10pefNeigh with n=10 seemed to be sufficient to have no more improvement on the development set by increasing n for both language directions, so we used this training configuration. As for the rewriting phrase table used on the test set, we simply selected conf10k as in the Medical task. Results are reported in Table 5 for French to English and English to French.

We first observe that reranker performed similarly for the two translation directions, by improving 1-pass Moses by 0.5 BLEU. The smaller improvements may be partly attributed to the better LM used in 1-pass Moses, implying a better early modeling of grammaticality, but also by the fact that models such as SOUL and POS LMs rely on accurate contexts and are therefore more apt to help in choosing translations among generally better candidates.

Finally, rewriter obtains smaller but consistent improvements over reranker: +0.4 BLEU for translation into English, and +0.9 BLEU for translation into French. The smaller improvement in the former situation may be attributed to the nature of the target language which has a simpler agreement system. Consequently, the form-related errors discussed in Section 4.4 are possibly less subject to improvement here.

	extra	*missing*	*incorrect*				*unknown*	
	word	word	disamb	form	style	order	word	**all**
`1-pass Moses`	11	1	57	91	13	31	10	214
`reranker`	5	3	47	73	11	19	10	168
`rewriter`	4	4	40	55	12	19	10	144
`rewriter` oracle	19	2	26	44	14	22	10	137

Table 4: Results for manual error analysis for the first 70 test sentences.

System	fr-en		en-fr	
	BLEU	TER	BLEU	TER
`1-pass Moses`	32.5	47.7	32.3	49.9
`reranker`	33.0	47.3	32.8	49.4
`rewriter`	$33.4_{(+0.4)}$	$47.4_{(+0.1)}$	$33.7_{(+0.9)}$	$49.3_{(-0.1)}$
`semi-oracle`	$34.1_{(+1.1)}$	$46.6_{(-0.7)}$	$34.2_{(+1.4)}$	$48.6_{(-0.8)}$

Table 5: Results for the baselines, our best configuration and the semi-oracle for the `TED Talks`.

5 Related work

Reranking of translation hypotheses n-best list reranking was extensively studied in (Och et al., 2004), using features not used in the initial decoder such as IBM1 scores (which also proved useful for word-level confidence estimation (Blatz et al., 2004)) and generative syntactic models. While the experiments in (Och et al., 2004) did not show any clear contribution of syntactic information used in this manner, the later work by Carter and Monz (2011) managed to successfully exploit syntactic features using discriminative language modeling for n-best reranking. Gimpel *et al.* (2013) outperformed n-best reranking by generating, with an expensive but simple method, diverse hypotheses used as training data. Recently, Luong *et al.* (2014b) reranked n-best lists using confidence scores at the hypothesis level computed from word-level confidence measures learnt from roughly 10,000 SMT system outputs annotated by humans.

Rewriting of translation hypotheses Langlais *et al.* (2007) described a greedy search decoder, first introduced in (Germann et al., 2001), able to improve translations produced by a dynamic programming decoder using the same scoring function and translation table. However, the more recent work by Arun *et al.* (2010) using a Gibbs sampler for approximating maximum translation decoding showed the adequacy of the approximations made by state-of-the-art decoders for finding the best translation in their search space. Other works were more directly targeted at automatic post-editing of SMT output, and approached the problem as one of second-pass translation between automatic predictions and correct translations (Simard et al., 2007; Dugast et al., 2007). The recent work of Zhu *et al.* (2013) attempts to repair translations by exploiting confidence estimates for examples derived from the similarity between source words in the input text and in training examples. Luong *et al.* (2014a) obtained improvements by computing word confidence estimation, trained on human annotated data, and large sets of lexical, syntactic and semantic features, for the words in the n-best list produced during a first-pass decoding, and performing a second-pass decoding exploiting these new scores.

Confidence estimation of Machine Translation The Word Posterior Probability (WPP) proposed by Ueffing and Ney (2007), derived from information from the n-best list produced by a decoder, proved to be useful for estimating word-level confidence. Bach *et al.* (2011) worked on the issue of predicting sentence-level and word-level MT errors by using WPP and other features derived from the source context, the source-target alignment, and dependency structures, but relied on a significantly large manually annotated corpus of MT errors. De Gispert *et al.* (2013) calculate k-

gram posterior probabilities from n-best lists or word lattices, and demonstrated that they were reasonably accurate indications of whether specific k-grams would be found or not in human reference translations. Finally, the work of Blackwood *et al.* (2010) proposed to segment translation lattices according to confidence measures over the maximum likelihood translation hypothesis to focus on regions with potential translation errors. Hypothesis space constraints based on monolingual coverage are then applied to the low confidence regions to improve translation fluency.

6 Conclusions and perspectives

In this paper, we have described an approach that improves translations *a posteriori* by applying simple local rewritings. We have shown that the quality of phrase-level confidence estimates has a direct impact of the amplitude of the improvements that can be obtained, as well as the initial quality of the rewritten hypotheses. We have used a very simple definition for confidence estimates under the form of phrase posteriors estimated from n-best lists from an initial decoder, which obtained good empirical performance, in spite of not requiring large human-annotated datasets as in other approaches (Bach et al., 2011; Luong et al., 2014b).

Our work could be extended in several directions. First, we could use a larger set of rewriting operations (Langlais et al., 2007), including the `rewrite` (*sic*) operation introduced in (Marie and Max, 2013) that paraphrases source phrases and then translates them.

We could also possibly consider any phrase segmentation compatible with a specific word alignment rather than rely on specific phrase segmentations. This would allow us to attain faster some rewritings that could otherwise require several rewriting iterations and may never be attained by the greedy procedure.

More features could also be used, for instance to model more fine-grained syntax (Post, 2011) or document-level lexical coherence (Hardmeier et al., 2012). However, anticipating that some features might be very expensive to compute, we could adapt our procedure to work in several passes: initial passes would tend to restrict the search space more and more using an initial set of features, before a more expensive pass would concentrate on a limited number of hypotheses. Figure 1 indeed already showed a much faster oracle improvement between `1-pass Moses` and `reranker` for n-best list of small sizes.

Another avenue for improvement lies in the possibility to perform the training of our `rewriter` by providing it with more reference translations. As these are typically not readily available, we could resort to targeted paraphrasing (Madnani and Dorr, 2013) to rewrite reference translations into acceptable paraphrases that reuse n-grams from the best hypotheses of the system so far. Contrarily to (Madnani and Dorr, 2013), we could bias the paraphrasing table so that it only contains paraphrases that correspond to target phrases of high confidence values, which would add new n-grams likely of being produced by `rewriter`.

It is furthermore worth noticing that our work proposes a potential answer to an original question: contrarily to typical works on sub-sentencial MT confidence estimation, which predict whether a word or phrase is correct or not, our `rewriter` system could be used to determine automatically whether a rewriting system *could* (if asked to) attempt to improve locally a translation, or whether a human post-editor should already tackle working on improving it. As we showed in our manual error analysis in section 4.4, there are in fact many instances of errors that could not be recovered by our approach, be it because of its local rewriting strategy or of the bilingual resources or models used, so that some knowledge would have to be provided as hard constraints by a human translator, as hinted in (Crego et al., 2010). We could then finally have our `rewriter` system work in a turn-based fashion in collaboration with a human translator, fixing errors or making improvements that are being made possible by the last edits from the translator.

Acknowledgments

The authors would like to thank the anonymous reviewers and Guillaume Wisniewski for their useful remarks. Additional thanks go to Hai Son Le for "anticipating" the need for a large and efficient cache in his SOUL implementation, Quoc Khanh Do for his assistance on using SOUL, and Li Gong and Nicolas Pécheux for providing the authors with data used in the experiments. The work of the first author is supported by a CIFRE grant from French ANRT.

References

Alexandre Allauzen, Nicolas Pécheux, Quoc Khanh Do, Marco Dinarelli, Thomas Lavergne, Aurélien Max, Hai-son Le, and François Yvon. 2013. LIMSI @ WMT13. In *Proceedings of WMT*, Sofia, Bulgaria.

Abhishek Arun, Phil Blunsom, Chris Dyer, Adam Lopez, Barry Haddow, and Philipp Koehn. 2010. Monte Carlo inference and maximization for phrase-based translation. In *Proceedings of CoNLL*, Boulder, USA.

Nguyen Bach, Fei Huang, and Yaser Al-Onaizan. 2011. Goodness: A Method for Measuring Machine Translation Confidence. In *Proceedings of ACL*, Portland, USA.

Graeme Blackwood, Adrià de Gispert, and William Byrne. 2010. Fluency Constraints for Minimum Bayes-Risk Decoding of Statistical Machine Translation Lattices. In *Proceedings of COLING*, Beijing, China.

John Blatz, Erin Fitzgerald, George Foster, Simona Gandrabur, Cyril Goutte, Alex Kulesza, Alberto Sanchis, and Nicola Ueffing. 2004. Confidence Estimation for Machine Translation. In *Proceedings of COLING*, Geneva, Switzerland.

Simon Carter and Christof Monz. 2011. Syntactic Discriminative Language Model Rerankers for Statistical Machine Translation. *Machine Translation*, 25(4):317–339.

Stanley F. Chen and Joshua T. Goodman. 1998. An Empirical Study of Smoothing Techniques for Language Modeling. Technical Report TR-10-98, Computer Science Group, Harvard University.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of NAACL*, Montréal, Canada.

Josep M. Crego, Aurélien Max, and François Yvon. 2010. Local lexical adaptation in Machine Translation through triangulation: SMT helping SMT. In *Proceedings of COLING*, Beijing, China.

Adrià de Gispert, Graeme Blackwood, Gonzalo Iglesias, and William Byrne. 2013. N-gram posterior probability confidence measures for statistical machine translation: an empirical study. *Machine Translation*, 27(2):85–114.

Loïc Dugast, Jean Senellart, and Philipp Koehn. 2007. Statistical Post-Editing on SYSTRANs Rule-Based Translation System. In *Proceedings of WMT*, Prague, Czech Republic.

Ulrich Germann, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2001. Fast Decoding and Optimal Decoding for Machine Translation. In *Proceedings of ACL*, Toulouse, France.

Kevin Gimpel, Dhruv Batra, Chris Dyer, Gregory Shakhnarovich, and Virginia Tech. 2013. A Systematic Exploration of Diversity in Machine Translation. In *Proceedings of EMNLP*, Seatlle, USA.

Christian Hardmeier, Joakim Nivre, and Jorg Tiedeman. 2012. Document-Wide Decoding for Phrase-Based Statistical Machine Translation. In *Proceedings of EMNLP*, Jeju Island, Korea.

Saša Hasan and Hermann Ney. 2009. Comparison of Extended Lexicon Models in Search and Rescoring for SMT. In *Proceedings of NAACL, short papers*, Boulder, USA.

Kevin Knight. 2007. Automatic Language Translation Generation Help Needs Badly. In *MT Summit (invited talk)*, Copenhagen, Denmark.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of NAACL*, Edmonton, Canada.

Philippe Langlais, Alexandre Patry, and Fabrizio Gotti. 2007. A Greedy Decoder for Phrase-Based Statistical Machine Translation. In *Proceedings of Conference on Theoretical and Methodological Issues in Machine Translation (TMI)*, Skovde, Sweden.

Hai-Son Le, Ilya Oparin, Alexandre Allauzen, Jean-Luc Gauvain, and François Yvon. 2011. Structured Output Layer Neural Network Language Model. In *Proceedings of ICASSP*, Prague, Czech Republic.

Hai-Son Le, Alexandre Allauzen, and François Yvon. 2012. Continuous Space Translation Models with Neural Networks. In *Proceedings of NAACL*, Montréal, Canada.

Chin Y. Lin and Franz J. Och. 2004. ORANGE: a method for evaluating automatic evaluation metrics for machine translation. In *Proceedings of COLING*, Geneva, Switzerland.

Ngoc-Quang Luong, Laurent Besacier, and Benjamin Lecouteux. 2014a. An Efficient Two-Pass Decoder for SMT Using Word Confidence Estimation. In *Proceedings of EAMT*, Dubrovnik, Croatia.

Ngoc-Quang Luong, Laurent Besacier, and Benjamin Lecouteux. 2014b. Word Confidence Estimation for SMT N -best List Re-ranking. In *Proceedings of the Workshop on Humans and Computer-assisted Translation (HaCaT)*, Gothenburg, Sweden.

Nitin Madnani and Bonnie J. Dorr. 2013. Generating Targeted Paraphrases for Improved Translation. *ACM Transactions on Intelligent Systems and Technology, special issue on Paraphrasing*, 4(3).

Benjamin Marie and Aurélien Max. 2013. A Study in Greedy Oracle Improvement of Translation Hypotheses. In *Proceedings of IWSLT*, Heidelberg, Germany.

Franz Josef Och, Daniel Gildea, Sanjeev Khudanpur, Anoop Sarkar, Kenji Yamada, Alex Fraser, Shankar Kumar, Libin Shen, David Smith, Katherine Eng, Viren Jain, Zhen Jin, and Dragomir Radev. 2004. A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of NAACL*, Boston, USA.

Kishore Papineni, Salim Roukos, Todd Ward, and Weijing Zhu. 2002. BLEU: a Method for Automatic Evaluation of Machine Translation. In *Proceedings of ACL*, Philadelphia, USA.

Kristen Parton, Nizar Habash, Kathleen R. McKeown, Gonzalo Iglesias, and Adrià de Gispert. 2012. Can Automatic Post-editing Make MT more Meaningful? In *Proceedings of EAMT*, Trento, Italy.

Nicolas Pécheux, Li Gong, Quoc Khanh Do, Benjamin Marie, Yulia Ivanishcheva, Alexander Allauzen, Thomas Lavergne, Jan Niehues, Aurélien Max, and François Yvon. 2014. LIMSI @ WMT'14 Medical Translation Task. In *Proceedings of WMT*, Baltimore, USA.

Matt Post. 2011. Judging Grammaticality with Tree Substitution Grammar Derivations. In *Proceedings of ACL, short papers*, Portland, USA.

Lane Schwartz, Chris Callison-Burch, William Schuler, and Stephen Wu. 2011. Incremental Syntactic Language Models for Phrase-based Translation. In *Proceedings of ACL*, Portland, USA.

Holger Schwenk, Daniel Déchelotte, and Jean-Luc Gauvain. 2006. Continuous Space Language Models for Statistical Machine Translation. In *Proceedings of COLING-ACL*, Sydney, Australia.

Michel Simard, Cyril Goutte, and Pierre Isabelle. 2007. Statistical Phrase-based Post-editing. In *Proceedings of NAACL*, Rochester, USA.

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 AMTA*, Cambridge, USA.

Lucia Specia, Kashif Shah, Jose G.C. de Souza, and Trevor Cohn. 2013. QuEst - A Translation Quality Estimation Framework. In *Proceedings of ACL, System Demonstrations*, Sofia, Bulgaria.

Nicola Ueffing and Hermann Ney. 2007. Word-Level Confidence Estimation for Machine Translation. *Computational Linguistics*.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error Analysis of Statistical Machine Translation Output. In *Proceedings of LREC*, Genoa, Italy.

Richard Zens and Hermann Ney. 2006. N -Gram Posterior Probabilities for Statistical Machine Translation. In *Proceedings of WMT*, New York, USA.

Ying Zhang, Almut Silja Hildebrand, and Stephan Vogel. 2006. Distributed Language Modeling for N-best List Re-ranking. In *Proceedings of EMNLP*, Sydney, Australia.

Junguo Zhu, Muyun Yang, Sheng Li, and Tiejun Zhao. 2013. Repairing Incorrect Translation with Examples. In *Proceedings of IJCNLP*, Nagoya, Japan.