

# Oracle-based Training for Phrase-based Statistical Machine Translation

**Ankit K. Srivastava**  
CNGL, School of Computing  
Dublin City University, Ireland  
asrivastava@computing.dcu.ie

**Yanjun Ma**  
Baidu Inc.  
Beijing, China  
yma@baidu.com

**Andy Way**  
CNGL, School of Computing  
Dublin City University, Ireland  
away@computing.dcu.ie

## Abstract

A Statistical Machine Translation (SMT) system generates an  $n$ -best list of candidate translations for each sentence. A model error occurs if the most probable translation (1-best) generated by the SMT decoder is not the most accurate as measured by its similarity to the human reference translation(s) (an oracle). In this paper we investigate the parametric differences between the 1-best and the oracle translation and attempt to try and close this gap by proposing two rescoring strategies to push the oracle up the  $n$ -best list. We observe modest improvements in METEOR scores over the baseline SMT system trained on French–English Europarl corpora. We present a detailed analysis of the oracle rankings to determine the source of model errors, which in turn has the potential to improve overall system performance.

## 1 Introduction

Phrase-based Statistical Machine Translation (PB-SMT) systems typically learn translation, reordering, and target-language features from a large number of parallel sentences. Such features are then combined in a log-linear model (Och and Ney, 2002), the coefficients of which are optimized on an objective function measuring translation quality such as the BLEU metric (Papineni et al., 2002), using Minimum Error Rate Training (MERT) as described in Och (2003).

An SMT decoder non-exhaustively explores the exponential search space of translations for each source sentence, scoring each hypothesis using the

formula (Och and Ney, 2002) in (1).

$$P(e|f) = \exp\left(\sum_{i=1}^M \lambda_i h_i(e, f)\right) \quad (1)$$

The variable  $h$  denotes each of the  $M$  features (probabilities learned from language models, translation models, etc.) and  $\lambda$  denotes the associated feature weight (coefficient).

The candidate translation (in the  $n$ -best list) having the highest decoder score is deemed to be the best translation (1-best) according to the model. Automatic evaluation metrics measuring similarity to human reference translations can be modified to generate a score on the sentence level instead of at system level. These scores can, in turn, be used to determine the quality or goodness of a translation. The candidate having the highest sentence-level evaluation score is deemed to be the most accurate translation (oracle).

In practice, it has been found that the  $n$ -best list rankings can be fairly poor (i.e. low proportion of oracles in rank 1), and the oracle translations (the candidates closest to a reference translation as measured by automatic evaluation metrics) occur much lower in the list. Model errors (Germann et al., 2004) occur when the optimum translation (1-best) is not equivalent to the most accurate translation (oracle). The aim of this paper is to investigate these model errors by quantifying the differences between the 1-best and the oracle translations, and evaluate impact of the features used in decoding (tuned using MERT) on the positioning of oracles in the  $n$ -best list.

After a brief overview of related approaches in section 2, we describe in section 3 a method to identify the oracles in the  $n$ -best lists, and our analytical approach to determine whether the basic features (used in decoding) help or hurt the oracle rankings. Section 4 lists our experiments on modifying the feature weights to help push the oracles

up the  $n$ -best list, followed by discussion in section 5. We conclude with our remarks on how to obtain the best of the available  $n$  translations from the MT system together with avenues for further research on incorporating our methods in mainstream reranking paradigms.

## 2 Related Work

One manner to minimize the problem of low ranking of higher quality translation candidates in the  $n$ -best lists has been to extract additional features from the  $n$ -best lists and rescore them discriminatively. These reranking approaches differ mainly in the type of features used for reranking and the training algorithm used to determine the weights needed to combine these features.

Och et al. (2004) employed nearly 450 syntactic features to rerank 1000-best translation candidates using MERT optimized on BLEU. These same features were then trained in a discriminative reranking model by replacing MERT with a perceptron-like splitting algorithm and ordinal regression with an uneven margin algorithm (Shen et al., 2004). Unlike the aforementioned approaches, Yamada and Muslea (2009) trained a perceptron-based classifier on millions of features extracted from shorter  $n$ -best lists of size 200 of the entire training set for reranking, and computed BLEU on a sentence level rather than corpus level as we do here.

Hasan et al. (2007) observed that even after the reference translations were included in the  $n$ -best list, less than 25% of the references were actually ranked as the best hypotheses in their reranked system. They concluded that better reranking models were required to discriminate more accurately amongst the  $n$ -best lists. In this paper we take a step in that direction by trying to observe the impact of existing features (used in MERT and decoding) on the positioning of oracle-best hypotheses in the  $n$ -best lists to motivate new features for a reranking model.

Our work is most related to Duh and Kirchhoff (2008) in that they too devise an algorithm to recompute the feature weights tuned in MERT. However, they focus on iteratively training the weights of additional reranking features to move towards a non-linear model, using a relatively small dataset. While most papers cited above deal with feature-based reranking (and as such are not directly related to our proposed approach), they constitute

a firm foundation and serve as motivation for our oracle-based study. We focus on the features used in decoding itself and recompute their weights to determine the role of these features in moving oracles up (and down) the  $n$ -best list.

## 3 Methodology

The central thrust of our oracle-based training is the study of the position of oracle translations in the  $n$ -best lists and an analysis of sentences where the most likely translation (1-best) does not match with the best-quality translation (oracle). In this section, we describe the selection procedure for our oracles followed by an overview of the baseline system settings used in all our experiments, the rescoring strategies, and a filtering strategy to increase oracle confidence.

### 3.1 $N$ -best Lists and Oracles

The oracle sentence is selected by picking the candidate translation from amongst an  $n$ -best list closest to a given reference translation, as measured by an automatic evaluation metric. We chose BLEU for our experiments, as despite shortcomings such as those pointed out by (Callison-Burch et al., 2006), it remains the most popular metric, and is most often used in MERT for optimizing the feature weights. Our rescoring experiments focus heavily on these weights. Note that BLEU as defined in (Papineni et al., 2002) is a geometric mean of precision  $n$ -grams (usually 4), and was not designed to work at the sentence-level, as is our requirement for the oracle selection. Several sentence-level implementations known as smoothed BLEU have been proposed (Lin and Och, 2004; Liang et al., 2006). We use the one proposed in the latter, as shown in (2).

$$S_{BLEU} = \sum_{i=1}^4 \frac{BLEU_i(cand, ref)}{2^{4-i+1}} \quad (2)$$

Figure 1 shows a sample of 10 candidate English translations from an  $n$ -best list for a French sentence. The first column gives the respective decoder cost (log-linear score) used to rank an  $n$ -best list and the third column displays the  $sBLEU$  (sentence-level BLEU score) for each candidate translation. The candidate in the first position in the figure is the **1-best** according to the decoder. The 7th-ranked sentence is most similar to the reference translation and hence awarded the highest

Decoder	Sentence	sBLEU
-5.32	is there not here two weights , two measures ?	0.0188
-5.50	is there not here double standards ?	0.147
-5.66	are there not here two weights , two measures ?	0.0125
-6.06	is there not double here ?	0.025
-6.15	is there not here double ?	0.025
-6.17	is it not here two sets of standards ?	0.0677
<b>-6.28</b>	<b>is there not a case of double standards here ?</b>	<b>0.563</b>
-6.37	is there not here two weights and two yardsticks ?	0.0188
-6.38	is there no double here ?	0.0190
-6.82	is there not here a case of double standards ?	0.563

sBLEU score. This sentence is the **oracle translation** for the given French sentence. Note that there may be ties where the oracle is concerned (the 7th and the 10th ranking sentence have the same sBLEU score). Such issues are discussed and dealt with in section 3.4. Oracle-best hypotheses are a good indicator of what could be achieved if our MT models were perfect, i.e. discriminated properly between good and bad hypotheses.

### 3.2 Baseline System

The set of parallel sentences for all our experiments is extracted from the WMT 2009<sup>1</sup> Europarl (Koehn, 2005) dataset for the language pair French–English after filtering out sentences longer than 40 words (1,050,398 sentences for training and 2,000 sentences each for development (test2006 dataset) and testing (test2008 dataset)). We train a 5-gram language model using SRILM<sup>2</sup> with Kneser-Ney smoothing (Kneser and Ney, 1995). We train the translation model using GIZA++<sup>3</sup> for word alignment in both directions followed by phrase-pair extraction using grow-diag-final heuristic described in Koehn et al., (2003). The reordering model is configured with a distance-based reordering and monotone-swap-discontinuous orientation conditioned on both the source and target languages with respect to previous and next phrases.

We use the Moses (Koehn et al., 2007) phrase-based beam-search decoder, setting the stack size to 500 and the distortion limit to 6, and switching on the  $n$ -best-list option. Thus, this baseline model uses 15 features, namely 7 distortion features ( $d1$  through  $d7$ ), 1 language model feature ( $lm$ ), 5 translation model features ( $tm1$  through  $tm5$ ), 1 word penalty ( $w$ ), and 1 unknown word penalty feature. Note that the unknown word fea-

<sup>1</sup><http://www.statmt.org/wmt09/>

<sup>2</sup><http://www-speech.sri.com/projects/srilm/>

<sup>3</sup><http://code.google.com/p/giza-pp/>

Figure 1: Sample from an  $n$ -best list of translation candidates for the input sentence *N’y a-t-il pas ici deux poids, deux mesures?*, whose reference translation is: *Is this not a case of double standards?*

ture applies uniformly to all the candidate translations of a sentence, and is therefore dropped from consideration in our experiments.

### 3.3 Recalculating Lambdas

In contrast to mainstream reranking approaches in the literature, this work analyzes the 14 remaining **baseline features** optimized with MERT and used by the decoder to generate an initial  $n$ -best list of candidates. No new features are added, the existing feature values are not modified, and we only alter the feature weights used to combine the individual features in a log-linear model. We are interested in observing the influence of each of these baseline features on the position of oracles in the  $n$ -best lists. This is achieved by comparing a specific feature value for a 1-best translation against its oracle. These findings are then used in a novel way to recompute the lambdas using one of the following two formulae.

- $RESC_{sum}$ : For each of the 14 features, the new weight factors in the difference between the mean feature value of oracles and the mean feature value of the 1-bests.

$$\lambda_{new} = \lambda_{old} + (\bar{f}_{oracle} - \bar{f}_{1best}) \quad (3)$$

- $RESC_{prod}$ : For each of the 14 features, the new weight factors in the ratio of the mean feature value of oracles to the mean feature value of the 1-bests.

$$\lambda_{new} = \lambda_{old} * \frac{\bar{f}_{oracle}}{\bar{f}_{1best}} \quad (4)$$

Both formulae aim to close the gap between feature values of oracle translations and those of the baseline 1-best translations. The recalculated weights are then used to rescore the  $n$ -best lists, as described in section 4.

Accordingly, our experiments are essentially focused on recomputing the original set of feature weights rather than the feature values. We reiterate that the huge mismatch between oracles and 1-best translations implies that MERT is sub-optimal (He and Way, 2009) despite being tuned on translation quality measures such as (document-level) BLEU. In recomputing weights using oracle translations, the system tries to learn translation hypotheses which are closest to the reference. These computations and rescorings are learned on the development set (**devset**), and then carried over to rescoring the  $n$ -best lists of the **testset** (blind dataset).

### 3.4 Oracle Filtering

A system composed of all the oracle hypotheses serves as an upper bound on any improvement due to reranking. However, one must carefully evaluate these so-called oracle translations. There is inherent noise due to:

- the existence of a large population of identical surface-level hypotheses (but different phrase segmentations) in the  $n$ -best list;
- the tendency of BLEU and other metrics to award the same score to sentences differing in the order or lexical choice of one or two words only.

Revisiting the  $n$ -best list given in Figure 1, note that both the 7th and the 10th sentence as well as the 1st and 8th sentence were awarded the same sBLEU score. There is no way to distinguish between the two as far as the oracle is concerned. Furthermore, note that this sample was carefully selected to show the variety of the  $n$ -best list. That is, in reality, approximately 20 hypotheses (identical to the 1-best hypothesis at the surface-level) occur between the 1st and the 2nd sentence in the figure.

<b>N-BEST</b>	<b>DIFF</b>	<b>DIVERSE</b>	<b>ACCEPTED</b>
100	62.10%	48.55%	27.10%
500	55.50%	57.75%	30.50%
1000	54.05%	61.40%	32.80%

Table 1: Statistics of % of oracle sentences considered for rescoring experiments

Since the underlying strength of all our experiments relies primarily on the goodness of oracles,

we explore a combination of two filtering strategies to increase the confidence in oracles, namely **DIFFERENCE** and **DIVERSITY**.

The **DIFFERENCE** filter computes the difference in the sentence-level BLEU scores of the hypotheses at rank 1 and rank 2. Note that it is often the case that more than one sentence occupies the same rank. Thus when we compute the difference between rank 1 and rank 2, these are in actuality a cluster of sentences having the same scores. The purpose of this filter is to ensure that oracles (rank 1) are “different enough” compared to the rest of the sentences (rank 2 and beyond).

The **DIVERSITY** filter aims at ensuring that the specific sentence has a wide variety of hypotheses leading to a distinguishing oracle (selected using the previous filter). This is computed from the proportion of  $n$ -best translations represented by the sentences in rank 1 and rank 2 clusters (based on how many sentences are present in rank 1 or 2). The motivation behind this filter is to drop sentences whose  $n$ -best lists contain no more than 2 or 3 clusters. In such cases, all the hypotheses are very similar to each other, when scored by the sBLEU metric. We used both filters in tandem because this ensured that the sentences selected in our final list had an oracle which was significantly different from the rest of the  $n$ -best list, and the  $n$ -best list itself had a good variety of hypotheses to choose from.

Thresholds for both filters were empirically determined to approximate the average of their respective mean and median values. Sentences which possessed a value above both thresholds constituted the set of true oracles used to recalculate the lambdas for our rescoring experiments. Table 1 shows the number of sentences passing the Difference filter (column 2), the Diversity filter (column 3) and both (column 4: the accepted set of true oracles). Experiments were carried out for 3 different sizes of  $n$ -best lists. It is observed that all three sets follow the same trend.

## 4 Experimental Analysis

Our analyses of the differences between the 1-best and the oracle translations follows. We perform all our experiments on 3 different  $n$ -best list sizes—100, 500, and 1000.

RANGE	(a) DEVSET			(b) TESTSET		
	100-BEST	500-BEST	1000-BEST	100-BEST	500-BEST	1000-BEST
Rank 1	725	402	308	725	415	324
Rank 2 to 5	194	87	68	176	95	69
Rank 6 to 10	121	52	37	125	67	53
Rank 11 to N	960	1459	1587	974	1423	1554

Table 2: Number of times an oracle occurs in a particular range of ranks in the  $n$ -best lists of (a)DEVSET and (b)TESTSET

#### 4.1 Distribution of Oracles

Before proceeding with our rescoring experiments, it is important to determine how the oracle translations are distributed across the space of the baseline systems. Table 2 gives a summary of where (at what rank) each oracle candidate is placed in the  $n$ -best list of the development and test sets of 2000 sentences each. It is evident that with increasing  $n$ -best list size, the number of oracles in the top ranks decreases. This is alarming as this increases the complexity of our problem with increasing  $n$ -best list sizes. This is another reason why we filter oracles, as described in the previous section. Oracle filtering clearly shows that not all sentences have a good quality oracle. This balances the tendency of high-ranking translations to be placed lower in the list.

#### 4.2 System-level Evaluation

We extract the 14 baseline features for sentences from the devset of 2000 sentences using the test2006 dataset selected via oracle filtering mentioned previously. For each of these sentences, we compare the 1-best and oracle-best features and compute the mean value per feature. This is then used to compute two new sets of weights using the  $\text{RESC}_{sum}$  and  $\text{RESC}_{prod}$  rescoring strategies, described in the previous section. We implemented our rescoring strategies on the devset and then applied the 2 new sets of weights computed on the testset of  $n$ -bests. Evaluation is done at a system level for both the development and testsets using BLEU (Papineni et al., 2002) and METEOR (Banerjee and Lavie, 2005). We also evaluate how many sentences contain the oracle candidates in the top position (rank 1). This is shown in Table 3. The last row in each subsection labeled ORACLE gives the upper bound on each system, i.e. performance if our algorithm was perfect and all the oracles were placed at position 1.

We also perform a Top5-BLEU oracle evaluation (shown in Table 4). The difference between the evaluations in Tables 3 and 4 is that the lat-

ter evaluates on a list of top-5 hypotheses for each sentence instead of the usual comparison of a single translation hypothesis with the reference translation. The sentences used in Table 3 are present in the top 1 position of sentences used in Table 4. This means that when BLEU and METEOR scores are evaluated at system-level, for each sentence, the translation (among 5) with the highest sBLEU score is selected as the translation for that sentence. This is similar to the post-editing scenario where human translators are shown  $n$  translations and are asked to either select the best or rank them. Some studies have used as many as 10 translations together (Koehn and Haddow, 2009). We only use 5 in our evaluation.

We observe that overall the  $\text{RESC}_{sum}$  system shows a modest improvement over the baseline in terms of METEOR scores, but not BLEU scores. This trend is consistent across all the 3  $n$ -best list sizes. We speculate that perhaps the reliance of METEOR on both precision and recall as opposed to precision-based BLEU is a factor for this disagreement between metrics. We also observe that the degree of improvement in the BLEU and METEOR scores of each system from top-1 (Table 3) to top-5 (Table 4) is more obvious in the rescored systems  $\text{RESC}_{sum}$  and  $\text{RESC}_{prod}$  compared to the baseline. This gives weight to our observation that the oracles have moved up, just not to the top position.

#### 4.3 Per feature Comparison

Figure 2 analyses which features favour how many oracles over 1-best translations. The figures are in percentages. We only give values for 1000-best lists, because the results are consistent across the various  $n$ -best list sizes.

The oracles seems to be favoured by d2 (monotone orientation) and tm5 (phrase penalty) features. Note that this selection is arbitrary and changes when the dataset changes. This means that if we use a different DEVSET, a different set of features will favour the oracle rankings. Further

SYSTEM	(a) DEVSET			(b) TESTSET		
	BLEU	MET	ORC	BLEU	MET	ORC
<i>rescored on 100-best list</i>						
BASE	32.17	61.34	36.25	32.47	61.80	36.25
RESC <sub>sum</sub>	31.99	<b>61.45</b>	36.55	32.33	61.75	35.65
RESC <sub>prod</sub>	32.13	61.35	36.30	32.46	61.78	35.60
ORACLE	34.90	63.65	100	35.26	64.01	100
<i>rescored on 500-best list</i>						
BASE	32.17	61.34	20.10	32.47	61.80	20.75
RESC <sub>sum</sub>	31.56	<b>61.62</b>	20.15	31.99	<b>62.00</b>	19.65
RESC <sub>prod</sub>	32.08	61.30	20.15	32.43	61.75	20.65
ORACLE	36.45	64.70	100	36.80	65.12	100
<i>rescored on 1000-best list</i>						
BASE	32.17	61.34	15.4	32.47	61.80	16.2
RESC <sub>sum</sub>	31.45	<b>61.48</b>	15.7	31.84	61.87	15.45
RESC <sub>prod</sub>	32.04	61.26	15.6	32.41	61.73	16.2
ORACLE	37.05	65.14	100	37.50	65.65	100

Table 3: Summary of the Fr–En translation results on WMT (a)test2006 (devset) and (b)test2008 (testset) data, using BLEU and METEOR metrics. The column labeled ORC refers to the % of sentences selected as the oracle w.r.t. BLEU metric.

SYSTEM	(a) DEVSET			(b) TESTSET		
	BLEU	MET	ORC	BLEU	MET	ORC
<i>rescored on 100-best list</i>						
BASE <sub>5</sub>	32.83	61.95	45.95	33.17	62.34	45.05
RESC <sub>sum5</sub>	32.72	<b>62.04</b>	45.75	33.08	<b>62.40</b>	45.65
RESC <sub>prod5</sub>	32.78	61.92	45.80	33.16	62.34	45.00
ORACLE	34.90	63.65	100	35.26	64.01	100
<i>rescored on 500-best list</i>						
BASE <sub>5</sub>	32.83	61.95	24.45	33.17	62.34	25.50
RESC <sub>sum5</sub>	32.49	<b>62.31</b>	27.20	32.95	<b>62.71</b>	27.90
RESC <sub>prod5</sub>	32.74	61.89	24.75	33.12	62.30	25.80
ORACLE	36.45	64.70	100	36.80	65.12	100
<i>rescored on 1000-best list</i>						
BASE <sub>5</sub>	32.83	61.95	18.80	33.17	62.34	19.65
RESC <sub>sum5</sub>	32.45	<b>62.27</b>	20.90	32.85	<b>62.68</b>	21.85
RESC <sub>prod5</sub>	32.70	61.88	18.60	33.13	62.30	19.85
ORACLE	37.05	65.14	100	37.50	65.65	100

Table 4: Top5 Eval: Summary of the Fr–En translation results on WMT (a)test2006 (devset) and (b)test2008 (testset) data, using BLEU and METEOR metrics on best of top 5 hypotheses. The column labeled ORC refers to the % of sentences selected as the oracle w.r.t. BLEU metric.

experimentation is required to determine whether there is a pattern to this. Nevertheless, this computation provides some clue as to how the baseline feature weights change during rescoring.

#### 4.4 Movement in Rankings

Table 5 shows the number (n) of sentences (out of 2000) which were moved up ( $\uparrow$ ), moved up to a position in the top-5, moved down ( $\downarrow$ ), or moved down from a position in the top-5, and the average number of positions moved (p) for both our rescoring strategies. We observe that RESC<sub>sum</sub> is more effective in promoting oracles than RESC<sub>prod</sub>. Perhaps it is no surprise that the RESC<sub>sum</sub> formula resembles the highly effective perceptron formula (without the iterative loop) of Liang et al., (2006). The similarity between the number of positions

moved up and down explains why our rescoring strategies fail to record a more marked improvement at the system level.

## 5 Discussion and Future Work

### 5.1 Impact of MERT features on oracles

We try to re-estimate the weights of the baseline features and observe the impact of them on oracle reranking. While a substantial amount of oracles are moved to the top-5 ranks (not necessarily to the top-1), it does not automatically imply a better BLEU score. However, there is up to a 0.5% relative improvement in the METEOR scores. Perhaps this implies low quality oracles for at least some of the sentences. Note that although we filter away sentences before recomputing lambdas, we imple-

SYS	(a) DEVSET						(b) TESTSET					
	n↑	p↑	n <sub>5</sub> ↑	n↓	p↓	n <sub>5</sub> ↓	n↑	p↑	n <sub>5</sub> ↑	n↓	p↓	n <sub>5</sub> ↓
	<i>rescored on 100-best list</i>											
R <sub>sum</sub>	637	24	267	776	23	278	627	24	260	794	22	278
R <sub>prod</sub>	590	10	94	534	11	89	559	10	93	587	12	93
	<i>rescored on 500-best list</i>											
R <sub>sum</sub>	840	122	212	875	121	185	869	129	277	850	111	199
R <sub>prod</sub>	856	54	75	722	74	64	831	55	84	739	69	80
	<i>rescored on 1000-best list</i>											
R <sub>sum</sub>	908	237	180	878	248	147	933	247	198	870	215	176
R <sub>prod</sub>	918	114	63	758	163	51	895	117	73	785	148	66

Table 5: Movement of oracles in  $n$ -bests of (a) development set and (b) test set after rescoring the baseline system with weights learned from RESC<sub>sum</sub> and RESC<sub>prod</sub>: how many & how much?

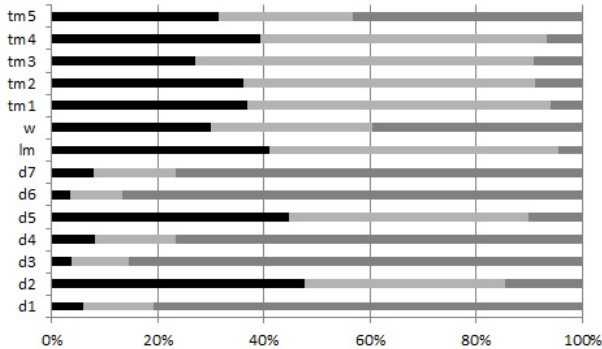


Figure 2: Results for a 1000-best list of filtered oracles: For how many sentences (% given on the X-axis) does a baseline feature (given on the Y-axis) favour the oracle translation (black bar) over the 1-best translation (light grey bar). The dark grey bar (third band in each bar) denotes percentage of sentences having the same value for its oracle and 1-best hypothesis

ment our rescoring strategies on the entire set (i.e. no filtering). Therefore the devset and testset may contain noise which makes it difficult for any improvements to be seen. Overall, there are certain baseline features (see section 4.3), which favour oracles and help in pushing them up the  $n$ -best list.

Duh and Kirchhoff, (2008) conclude that log-linear models often underfit the training data in MT reranking and that is the main reason for the discrepancy between oracle-best hypothesis and reranked hypothesis of a system. We agree with this statement (cf. figure 2). However, we believe that there is scope for improvement on the baseline features (used in decoding) before extracting more complex features for reranking.

## 5.2 Role of oracles in boosting translation accuracy

We believe oracle-based training to be a viable method. In future work, we intend to explore more features (especially those used in the reranking literature such as Och et al., (2004)) to help promote oracles. We believe that our oracle-based method can help select better features for reranking. We also plan to use a host of reranking features (Shen et al., 2004) and couple them with our RESC<sub>sum</sub> rescoring strategy. We will also generate a feature based on our rescoring formula and use it as an additional feature in discriminative reranking frameworks. We have used here sentence-level BLEU as opposed to system-level BLEU as used in MERT for oracle identification. We plan to use metrics better suited for sentence-level like TER (Snover et al., 2006).

## 6 Conclusion

We analyze the relative position of oracle translations in the  $n$ -best list of translation hypotheses to help reranking in a PB-SMT system. We propose two new rescoring strategies. In general, the improvements provided by reranking the  $n$ -best lists is dependent on the size of  $n$  and the type of translations produced in the  $n$ -best list. We see an improvement in METEOR scores. To conclude, oracles have much to contribute to the ranking of better translations and reducing the model errors.

## Acknowledgements

This work is supported by Science Foundation Ireland (grant number: 07/CE/I1142). This work was carried out during the second author’s time at CNGL in DCU. The authors wish to thank the anonymous reviewers for their helpful insight.

## References

- Banerjee, Satyanjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with Improved Correlation with Human Judgments. *ACL 2005, Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization at the 43rd Annual Meeting of the Association for Computational Linguistics*, Ann Arbor, Michigan. 65–72.
- Callison-Burch, Chris, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of BLEU in Machine Translation Research. *EACL 2006, Proceedings of the 11th Conference of the European Chapter of the Association for Computational Linguistics*, Trento, Italy. 249–256.
- Duh, Kevin and Katrin Kirchhoff. 2008. Beyond Log-Linear Models: Boosted Minimum Error Rate Training for N-best Re-ranking. *ACL 2008, Proceedings of the 48rd Annual Meeting of the Association for Computational Linguistics Short Papers*, Columbus, Ohio. 37–40.
- Germann, Ulrich, Michael Jahr, Kevin Knight, Daniel Marcu, and Kenji Yamada. 2004. Fast and Optimal Decoding for Machine Translation. *Artificial Intelligence*, 154. 127–143.
- Hasan, Saša, Richard Zens, and Hermann Ney. 2007. Are Very Large N-best List Useful for SMT?. In *Proceedings of NAACL-HLT '07*, Rochester, New York. 57–60.
- He, Yifan and Andy Way. 2009. Improving the Objective Function in Minimum Error Rate Training. In *Proceedings of Machine Translation Summit XII*, Ottawa, Canada. 238–245.
- Kneser, R. and Hermann Ney. 1995. Improved Backing-off for n-gram Language Modeling. In *Proceedings IEEE International Conference on Acoustics, Speech, and Signal Processing*, Vol. 1, Detroit, Michigan. 181–184.
- Koehn, Philipp, Franz Och, and Daniel Marcu. 2003. Statistical Phrase-Based Translation. In *Proceedings of NAACL '03*, Edmonton, Canada. 48–54.
- Koehn, Philipp. 2005. Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of Machine Translation Summit X*, Phuket, Thailand. 79–86.
- Koehn, Philipp, 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. *ACL 2007, 45th Annual Meeting of the Association for Computational Linguistics, demonstration session*, Prague, Czech Republic. 177–180.
- Koehn, Philipp and Barry Haddow. 2009. Interactive Assistance to Human Translators using Statistical Machine Translation Methods. In *Proceedings of MT Summit XII*, Ottawa, Canada. 73–80.
- Liang, Percy, Alexandre Bouchard-Cote, Dan Klein, and Ben Taskar. 2006. An end-to-end Discriminative Approach to Machine Translation. *COLING-ACL 2006, 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, Sydney, Australia. 761–768.
- Lin, Chin-Yew and Franz J Och. 2004. ORANGE: A Method for Evaluating Automatic Evaluation Metrics for Machine Translation. *COLING 2004, Proceedings of the 20th International Conference on Computational Linguistics*, Geneva, Switzerland. 501–507.
- Och, Franz J and Hermann Ney. 2002. Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. *ACL 2002, 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA. 295–302.
- Och, Franz J. 2003. Minimum Error Rate Training in Statistical Machine Translation. *ACL 2003, 41st Annual Meeting of the Association for Computational Linguistics*, Sapporo, Japan. 160–167.
- Och, Franz J., Dan 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. *HLT-NAACL 2004, the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics*, Boston, MA. 161–168.
- Papineni, Kishore, Salim Roukos, Todd Ward, and Wei-Jung Zhu. 2002. BLEU: A Method for Automatic Evaluation of Machine Translation. *ACL 2002, 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, PA. 311–318.
- Shen, Libin, Anoop Sarkar, and Franz J. Och. 2004. Discriminative Reranking for Machine Translation. *HLT-NAACL 2004, the Human Language Technology Conference and the North American Chapter of the Association for Computational Linguistics*, Boston, MA. 177–184.
- Snover, Matthew, Bonnie Dorr, Richard Schwartz, Linnea Micciula, and John Makhoul. 2006. A Study of Translation Edit Rate with targeted Human Annotation. *AMTA 2006, 7th Conference of the Association for Machine Translation in the Americas*, Cambridge, MA. 223–231.
- Yamada, Kenji and Ion Muslea. 2009. Reranking for Large-Scale Statistical Machine Translation. In Cyril Goutte, Nicola Cancedda, Marc Dymetman, and George Foster (eds.), *Learning Machine Translation*. MIT Press, Cambridge, MA. 151–168.