Phrase Extraction and Rescoring in Statistical Machine Translation

Ankit Kumar Srivastava

BS, MA

A dissertation submitted in fulfilment of the requirements for the award of

Doctor of Philosophy (Ph.D.)

to the

Dublin City University
School of Computing

Supervisor: Prof. Andy Way

May 2014

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Ph.D. is entirely my own work, that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed:

(Candidate) ID No.:

Date:

Contents

List of Figures

List of Tables

Abstract

The lack of linguistically motivated translation units or phrase pairs in Phrase-based Statistical Machine Translation (PB-SMT) systems is a well-known source of error. One approach to minimise such errors is to supplement the standard PB-SMT models with phrase pairs extracted from parallel treebanks (linguistically annotated and aligned corpora). In this thesis, we extend the treebank-based phrase extraction framework with percolated dependencies – a hitherto unutilised knowledge source – and evaluate its usability through more than a dozen syntax-aware phrase extraction models.

However, the improvement in system performance is neither consistent nor conclusive despite the proven advantages of linguistically motivated phrase pairs. This leads us to hypothesize that the PB-SMT pipeline is flawed as it often fails to access perfectly good phrase-pairs while searching for the highest scoring translation (decoding). A model error occurs when the highest-probability translation (actual output of a PB-SMT system) according to a statistical machine translation model is not the most accurate translation it can produce. In the second part of this thesis, we identify and attempt to trace these model errors across state-of-the-art PB-SMT decoders by locating the position of oracle translations (the translation most similar to a reference translation or expected output of a PB-SMT system) in the *n*-best lists generated by a PB-SMT decoder. We analyse the impact of individual decoding features on the quality of translation output and introduce two rescoring algorithms to minimise the lower ranking of oracles in the *n*-best lists.

Finally, we extend our oracle-based rescoring approach to a reranking framework by rescoring the *n*-best lists with additional reranking features. We observe limited but optimistic success and conclude by speculating on how our oracle-based rescoring of *n*-best lists can help the PB-SMT system (supplemented with multiple treebank-based phrase extractions) get optimal performance out of linguistically motivated phrase pairs.

Acknowledgments

I am thankful to a number of people for their support, guidance, advice, and encouragement throughout the writing and completion of this thesis.

First and foremost, I thank my advisor and guru Prof. Andy Way for his unfailing guidance during each and every step of my research, words of wisdom, and sustained patience from start to end. I am eternally grateful for invaluable lessons learned on moving forward and dealing with experiments generating unexpected results. I could not have asked for a more perfect supervisor with just the right balance of freedom and structure. Sir, you truly made this an enriching experience!

Parts of my research would not have been possible without helpful assistance from several people: Dr. Patrik Lambert (first year postdoctoral mentor), Dr. Yanjun Ma (second year postdoctoral mentor), Dr. Sylwia Ozdowska (manual evaluation of MT output), Dr. John Tinsley (seed treebank data), Dr. Ventsislav Zhechev (help with tree alignment software for dependency trees), Prof. Jinhua Du (system combinaton and rescoring foundations), Dr. Sergio Penkale (sentence-level BLEU evaluation), Dr. Yifan He (discussion on MT evaluation), and Prof. Josef van Genabith (feedback on transfer talk and intriguing exploratory avenues).

This thesis would not have been possible either without the generous support from Science Foundation Ireland (SFI) through Grant 07/CE/I1142, as part of the Centre for Next Generation Localisation (CNGL) at Dublin City University (DCU).

Sincere thanks to Prof. Marcello Federico and Dr. Gareth Jones for their critical and insightful comments during my viva and helping me further improve the overall presentation of this thesis.

I also wish to thank everyone in and around the NCLT/CNGL Lab for spirited discussions during lunch, fun-filled activities during breaks, and general help and friendliness which helped ease the entire process of conducting research and writing this thesis: Anton, Antonio, Debasis, Deirdre, Declan, Eithne, Fiona, Hala, Javed, Jennifer, Jie, Joachim, Johannes, John J, Joseph, Lamia, Maria, Ozlem, Pratyush, Rejwanul, Riona, Sandipan,

Sara, Sarah, Sudip.

Finally, I need to thank Mumma, Dad, and the family (Dadaji, Nanaji, Naniji, Tau, Bhabhi, Bada-papa, Badi-mummy, Mamaji, Mami, Jijaji, Badi, Choti, Bhaiya, MK Bhabhi, Alka, Sagar, Akriti) for believing in me, indulging me in all my moods during writing, be it reticent or chatty, and loving encouragement. Mumma and Dad, I could not have done it without the pep talks and your being a willing audience of two while I babbled on my research. Thank you for keeping me grounded always. I seek your blessings with a *Charan Sparsh*!

Chapter 1

Introduction

"Poetry is what is lost in translation."

Robert Frost

"A convincing demonstration of correctness being impossible as long as the mechanism is regarded as a black box."

Edsger W. Dijkstra

Multilingual online chatting, automatic email translation, multilingual video games, relief and aid workers communicating at a disaster-struck foreign country, cross-lingual search on the web, multilingual customer support, machine-aided human translation: each of the afore-mentioned scenarios currently uses or has the potential to use machine translation (MT) technology in some fashion.

Machine translation is the design and implementation of software systems that can automatically translate text occurring in one natural language to another. MT, one of the earliest non-numeric applications of computers (Hutchins, 2000), has gained sustained resurgence since the meteoric rise of multilingual user-generated content on the web in both academia[1] and industry[2] in an attempt to bridge the language barriers in global com-

[1] **Examples in Academia:** Availability of open-source software such as Moses (Koehn et al., 2007) statistical machine translation system [`http://www.statmt.org/moses`] and Apertium (Forcada et al., 2009) rule-based machine translation platform [`http://www.apertium.org`], and freely available parallel corpora such as the Open Parallel Corpus (Opus) (Tiedemann, 2009) [`http://opus.lingfil.uu.se/`] and the European Parliament Proceedings Parallel Corpus (Europarl) (Koehn, 2005) [`http://www.statmt.org/europarl/`].

[2] **Examples in Industry:** Emergence of language technology solutions and services such as SDL Au-

munication in an increasingly globalised economy and information society.

Since its recommendation as a viable enterprise in Warren Weaver's historical memorandum (Weaver, 1949), a number of approaches have been implemented with varying degrees of success. Some of these are Rule-based Machine Translation (RBMT) (Probst et al., 2002; Sanchez-Martinez and Forcada, 2007), data-driven models like Example-based Machine Translation (EBMT) (Nagao, 1984; Carl and Way, 2003) and Statistical Machine Translation (SMT) (Brown et al., 1990; Koehn, 2010).

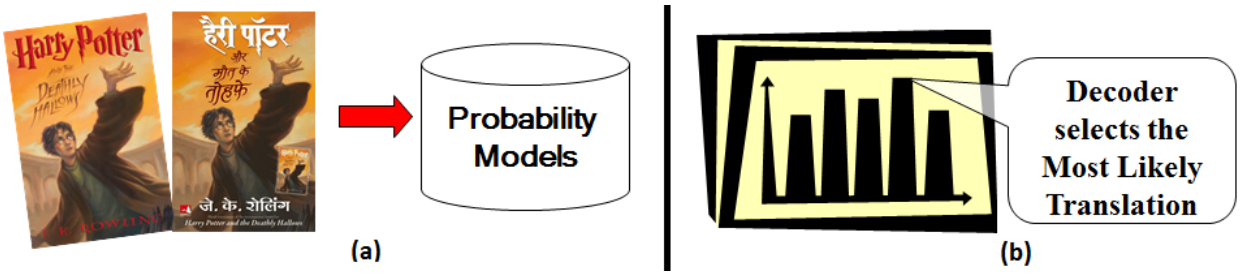


Figure 1.1: (a) Knowledge representation in SMT, (b) Decoding in SMT

These approaches primarily differ with respect to how translation knowledge is stored (knowledge representation) and how it is used to translate unseen text (decoding). The research described in this thesis touches both these areas with respect to SMT. Figure 1.1(a) illustrates the formulation of a text and its translation (parallel corpora) into statistical models, and Figure 1.1(b) depicts a SMT decoder selecting, from amongst a number of possible translations, the candidate with the highest probability.

Currently SMT is the most widely researched paradigm (Figure 1.2), and arguably the most effective as it continues to be the top-performing engine or a core component of the highest ranking multi-engine system at shared tasks and large-scale evaluations like IWSLT[3] (International Workshop on Spoken Language Translation), NIST[4] (National Institute of Standards and Technology Open MT Evaluation), and WMT[5] (Workshop on Statistical Machine Translation).

There exist many different flavours of SMT depending on the representational format

tomated Translation [http://www.sdl.com/products/automated-translation/], Lingo24 [http://www.lingo24.com/], and Asia Online [http://www.asiaonline.net/].

[3] http://workshop2013.iwslt.org/

[4] http://www.nist.gov/itl/iad/mig/openmt12.cfm

[5] http://www.statmt.org/wmt14/

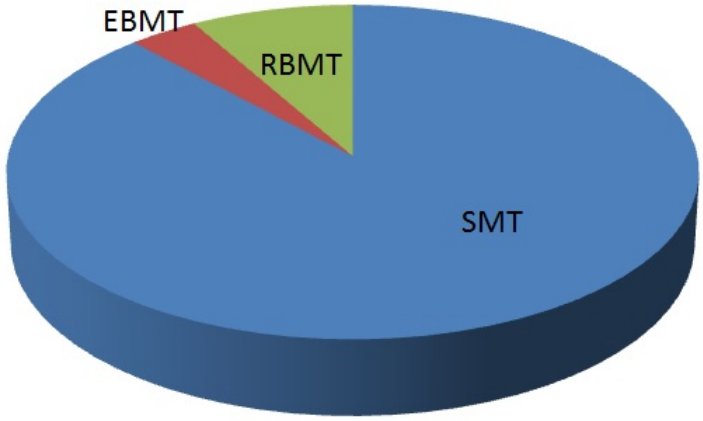


Figure 1.2: Pie chart to show number of research papers published since 2010 within each MT paradigm; Rule-based: RBMT (8%), Example-based: EBMT (4%), and Statistical: SMT (88%). *Source: MT Archive (December 2013) at* `http://www.mt-archive.info`

of bilingual text (also known as translation units) used for statistical modeling: word-based (Brown et al., 1990, 1993), phrase-based (Marcu and Wong, 2002; Koehn et al., 2003), or tree-based (Chiang, 2007; Wang et al., 2010). Figure 1.3 illustrates the three types of representational formats characterised by the manner in which the source (Hindi sentence) and the target (English sentence) is aligned. The first type operates at the word level as its translation unit, while phrase-based models align flat sequence of words or chunks, and tree-based models align recursive or hierarchical chunks (which may be labelled at each node) as translation units.

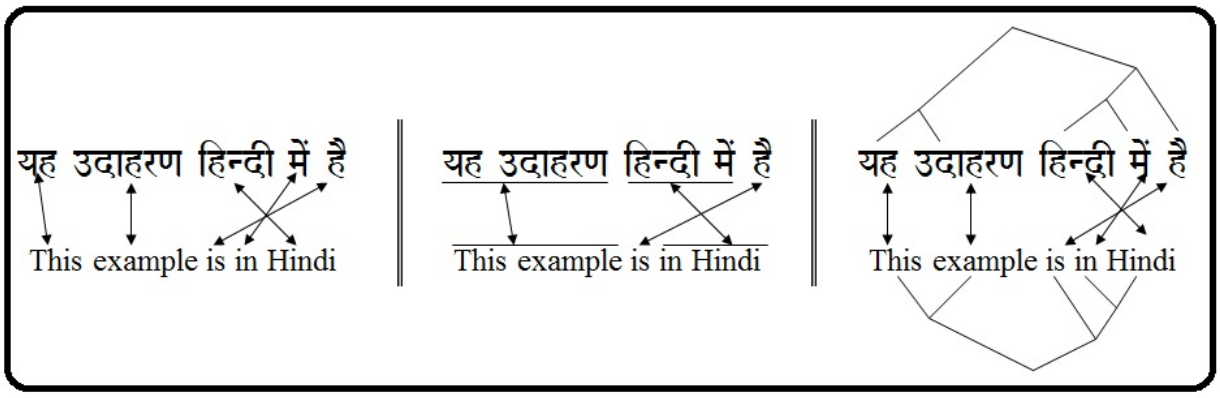


Figure 1.3: Three types of SMT: word-based, phrase-based, and tree-based.

Both phrase-based and tree-based (also known as syntax-based) SMT models are at the forefront of research and experimentation in the field, with improvements to the established methodologies being constantly proposed and implemented. Thus the state-of-the-art in MT is a dynamic target. For the sake of consistency and homogeneity in this thesis, we conduct all our experiments, modifications, analyses on the Phrase-based Statistical

Machine Translation (PB-SMT) model.[6]

While there are several alternatives to designing a PB-SMT system, we describe the state-of-the-art log-linear phrase-based model (Och and Ney, 2002) with standard configurations, as implemented in the open-source statistical machine translation system Moses[7] (Koehn et al., 2007), and used throughout in all our experiments. The schematics are demonstrated in Figure 1.4 and referred to throughout this thesis.

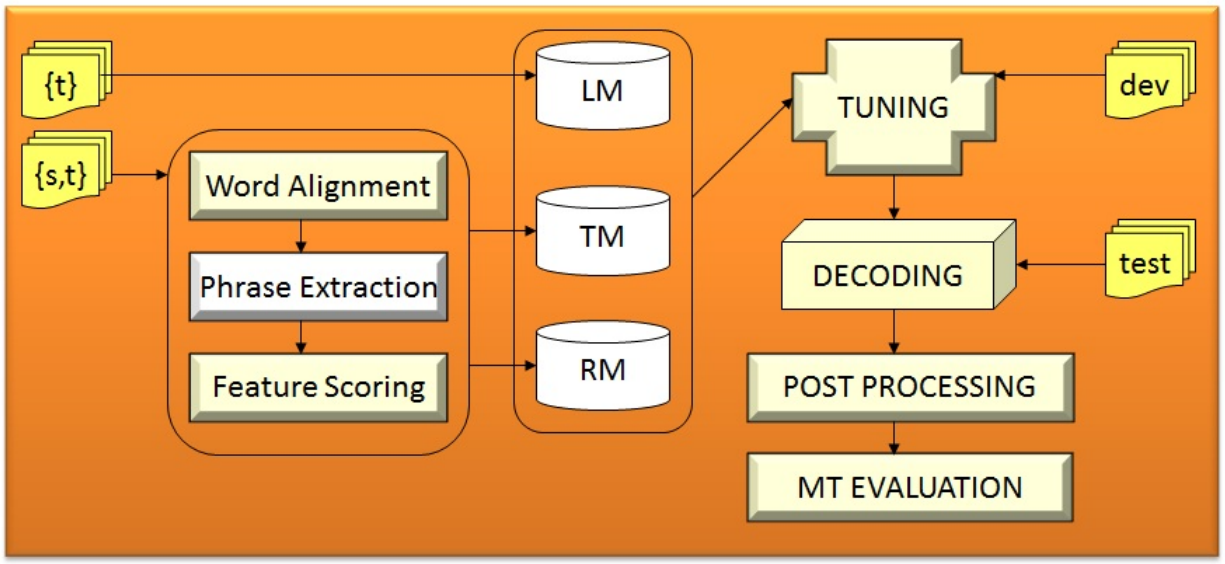


Figure 1.4: Schematic diagram of all the modules in a Phrase-based Statistical Machine Translation System.

A PB-SMT system takes as input a large corpus of sentences in a source (input) language ('s' for short) and their human translations in the target (output) language ('t' for short). Additional target-language data is also often used to build language models. PB-SMT systems extract knowledge in the form of sub-sentential alignments from large amounts of parallel corpora (sentence-aligned bilingual texts, also known as bitexts) to represent them as source–target phrase pair probability models. In PB-SMT, knowledge representation is also known as modeling. This step includes processes like extraction of translation units (phrase pairs) from word-aligned parallel corpora and defining a num-

[6] Note that, in theory, our methods can be adapted to the tree-based SMT models with trivial changes: (1) The treebank-based phrase extraction system can easily be implemented in syntax-based systems by either retaining the node labels or assigning generic labels in order to maintain the structure. (2) The oracle-based rescoring system is independent of the decoding algorithm and can thus be implemented in syntax-based systems as well.

[7] `http://www.statmt.org/moses/`

ber of probabilistic attributes (features) for each of these translation units (feature scoring in Figure 1.4). Some of the features in PB-SMT systems include source–target translation conditional probabilities (translation model or TM), target-language probabilities (language model or LM), and source–target relative order conditional probabilities (reordering model or RM).

Phrase-based SMT	**Tree-based SMT**
Translation Model avails of string-based chunks	Translation Model avails of recursive string-based chunks (labeled or unlabeled)
Language Model consists of *n*-gram probabilistic database of target language	Language Model same as PB-SMT; alternatively with labeled or unlabeled recursive structure (Shen et al., 2008; Tu et al., 2010)
Reordering Model operates on distance-based model	Reordering Model not required (taken care of by translation model)
Decoding algorithm employed is Stack-based (Beam search)	Decoding algorithm employed is Chart-based
Time & Space Complexity better	Time & Space Complexity worse

Table 1.1: Phrase-based SMT and Tree-based SMT: A contrastive analysis

PB-SMT systems use phrases as their basic translation unit. These phrases are merely strings of consecutive words, having no linguistic motivation whatsoever. Hence, unlike approaches in RBMT and certain EBMT variants, standard PB-SMT systems do not attempt to utilize linguistic syntax, leading to several errors like reordering, i.e. incorrect word order in the translation (Galley and Manning, 2008), and dropping of significant words like verbs (Ma and McKeown, 2009). SMT research is empirically driven and motivated by ideas that seek to reduce errors and improve system performance. Tree structures seek to overcome the shortcomings of PB-SMT arising from the lack of exploiting knowledge from formal or linguistic theories of syntax of any kind (Koehn, 2010).[8] Table 1.1 summarizes the major differences between phrase-based and tree-based SMT in terms of individual modules. Thus the tree-based SMT paradigm attempts to counter the afore-mentioned shortcomings of PB-SMT by exploiting the syntactic (structural) rela-

[8] For a detailed overview of application of these theories in MT, see Chapter 11 of Koehn (2010) pp.331–369.

tionships between chunks (with varying degrees of success), but at a cost to the decoding complexity (Chiang, 2007).

In another line of research, Tinsley (2010) extracted linguistically motivated chunks (i.e. flat sequences of words respecting syntactic boundaries) from parallel treebanks (node-aligned parse trees of parallel corpora or bitext) to be used directly in the PB-SMT framework. We too incorporate syntax in PB-SMT using the parallel treebank extraction framework (cf. Section 2.2) as an alternate method of phrase extraction; syntactic information is in the same format as string-based phrases and incurs no additional decoding cost.

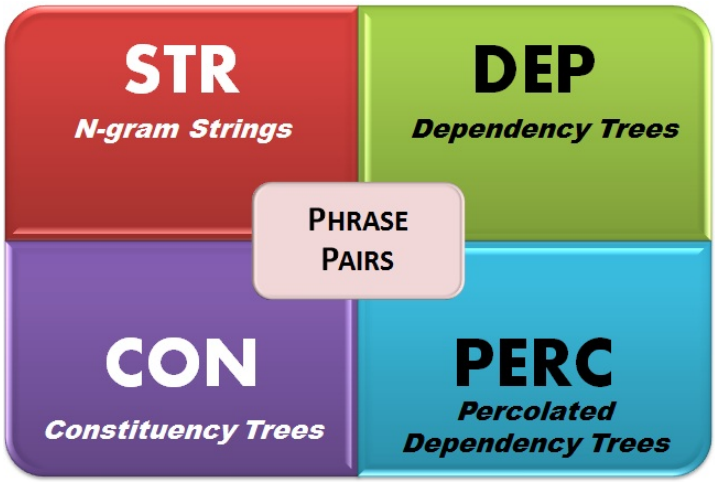


Figure 1.5: Four types of phrase pairs used in PB-SMT.

Parallel treebanks are obtained by parsing (assigning syntactic tree-like structure to text) both the source-language and target-language data and aligning corresponding nodes from these parse trees. There exist different types of treebanks depending on the annotation or syntactic theory used to parse the text: constituency,[9] dependency.[10] Figure 1.5 shows four types of phrase pairs we implement in our PB-SMT system (one standard non-linguistic [STR] and three linguistically motivated extracted from parallel treebanks [CON, DEP, and PERC]). We introduce a hitherto new annotation format called **percolated dependencies** – obtained via a method of deriving dependency trees from

[9] Example of a phrase structure treebank is the Penn Treebank for English (Marcus et al., 1993) [`http://www.cis.upenn.edu/~treebank/`].

[10] Example of a dependency structure treebank is the Prague Dependency Treebank for Czech (Hajič et al., 2000) [`https://ufal.mff.cuni.cz/pdt2.0/`].

consituency trees using Head Percolation (Magerman, 1995) – and investigate their incorporation into the PB-SMT pipeline with several new syntax-aware models. This leads us to our first research question:

(RQ1) *Are phrase pairs extracted from percolated dependency treebanks a useful knowledge source for PB-SMT?*

Through a range of MT evaluation experiments on a combination of one or more of the syntax-aware models in Chapter 2, we show that adding percolated dependency induced phrase pairs to a MT system generally improves the translation performance. Therefore the answer to RQ1 is yes. However we observe that our syntax-aware models do not show consistent performance in automatic evaluation and translation accuracy is lost when all four types of phrase pairs are merged into one MT system. A qualitative analysis then leads us to believe that good phrase pairs fail to be selected by the decoder as the optimal translation. This leads us to the second part of our research.

(RQ2) *Can the PB-SMT system obtain optimal performance out of linguistically motivated phrase pairs?*

In order to investigate this, we need to delve deeper and understand what is meant by optimal performance of a PB-SMT system. PB-SMT systems decode a source-language sentence into the target-language by selecting the most likely translation. There is an intermediate step between modeling and decoding known as parameter estimation or tuning which determines the optimal way of combining the features in a log-linear model (Och and Ney, 2002). The TM, LM, and RM features from Figure 1.4 are combined in a log-linear model, 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). Under the standard procedure, this parameter optimization is computed using a small parallel corpus, known as the development ('dev' for short) set.

An SMT decoder non-exhaustively explores the exponential search space of translations for each source sentence (in the test set), scoring each hypothesis using the formula (Och and Ney, 2002) in (1.1) (decoding in Figure 1.4).

$$score(t|s) = \sum_{i=1}^{M} \lambda_i h_i(s,t) \tag{1.1}$$

The variable *h* denotes each of the *M* features (probabilities learned from language models, translation models, etc.) and λ denotes the associated feature weight (coefficient). The candidate translation, amongst all the competing hypotheses, having the highest decoder score is deemed to be the best translation according to the model. The list of hypotheses of candidate translations for a particular sentence ranked according to their decoder score is called the *n*-best list of translations (where *n* refers to the number of candidate translations or hypotheses generated) and the highest-scoring candidate is labelled as the 1-best translation.

The post-processing module from Figure 1.4 is often an optional module and involves processes like recasing, detokenization, and most importantly reranking or rescoring of the *n*-best list of translations.

The last module as per the schemata given in Figure 1.4 concerns the evaluation of the output of a MT system. Typically system performance is assessed by using automatic evaluation metrics like BLEU (Papineni et al., 2002), which measure the similarity of MT output to a human-produced translation (reference translation). SMT systems consist of a number of components engaged in complex interactions and automatic evaluation provides very little insight into where the translation errors occur. Often improvements in the model are not registered by these metrics. Germann et al. (2004) identify several types of translation system errors, i.e. cases when the heuristic search-based decoder fails to output a high-quality optimum translation. Search errors occur when the decoder fails to find the optimum or highest scoring translation according to the model. Model errors occur when a good translation (candidate translation most similar to the reference translation, also known as the oracle translation) is not the highest scoring translation, i.e. it is posi-

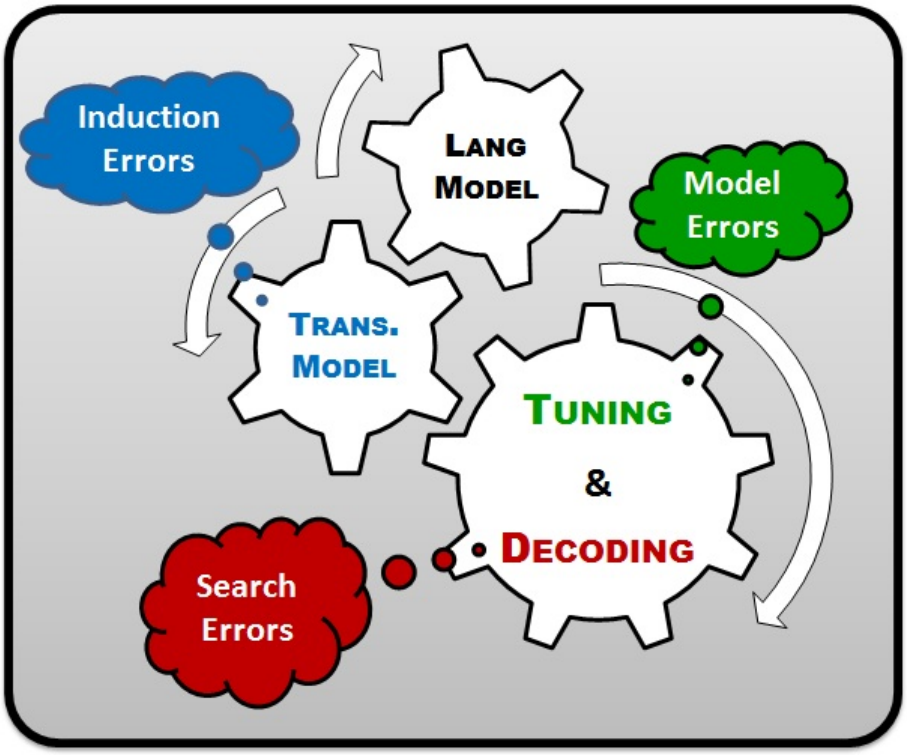


Figure 1.6: Types of system errors in a SMT system.

tioned lower in the *n*-best list of translations. Recently, Auli et al. (2009) defined a third error type called induction errors which refers to cases when the optimum translation is absent from the search space owing to various pruning strategies, etc. Figure 1.6 visualises these different type of errors. There are still other methods of error analysis which focus on manually or semi-automatically classifying errors in the system outputs (Vilar et al., 2006; Toral et al., 2012). These error classifications mainly deal with a surface-level linguistic check of the MT output in terms of lexical elements and grammatical constructions and are outside the scope of the research in this thesis which seeks to identify the cause for the surface-level errors in terms of model errors.

SMT research is empirically driven and motivated by ideas that reduce errors and improve system performance. Our research focusses on identifying and minimizing model errors which indicate that the 1-best candidate translations are not always the best translations (most accurate or closest to a reference translation) produced by the system. The term *oracle* is used to denote the candidate translation in the *n*-best list of translations which is most similar to the reference translation. This leads to our third research ques-

tion.

(RQ3) *Does pushing oracles up the* n*-best list minimise model errors and improve performance of a PB-SMT system?*

The decoding task for the PB-SMT models considered in this research has been shown to be NP-hard (Knight, 1999). This exponential complexity of the search space implies that the decoder performs a non-exhaustive search (using heuristic search methods) to find the best possible translation for a given input leading to a number of system errors mentioned previously. Our research aims to conduct a deep diagnostic analysis of this complex system by using **oracle reranking** to identify the origin of model errors. We then propose to implement modifications in terms of features and their weights to minimize the model errors.

We also aim to show that model errors can be reduced by reranking oracles and improving the optimization algorithm. This leads to our fourth research question.

(RQ4) *Can additional features in a reranking framework help minimise model errors?*

In Chapter 4, we test out our oracle-based methods on extra features (such as more sophisticated language models like part-of-speech LMs) not used in the decoding stage. This helps us in answering the fourth research question. Finally, we determine the answer to our final research question that ties our oracle-based rescoring work with our initial parallel treebank-based phrase extraction experiments.

(RQ5) *Can the oracle-rescored system help the PB-SMT system to better exploit linguistically motivated chunks?*

In order to answer this question, we modify our oracle-based rescoring framework by considering a translation hypothesis from a syntax-aware system (a MT system composed of syntax-aware models induced from parallel treebanks introduced in Chapter 2) as the oracle in contrast to a standard non-linguistic PB-SMT system hypothesis.

In Machine Learning (learning statistical models from large amounts of data), overfitting and underfitting are two important concepts.[11] Underfitting implies a condition when the model is too sparse to be effective. On the other hand, overfitting refers to the condition when there are too many features in the model resulting in an overgeneralization. Broadly speaking, augmenting the phrase pair set with treebank-aware constraints[12] and discarding any linguistic labels (Chapter 2) can be likened to an underfitting scenario while reranking MT outputs with numerous coarse-grained and fine-grained features (Chapter 4) can be seen approaching the overfitting scenario. Thus a systematic diagnosis based on an oracle-based study (Chapter 3) can be said to be a mid-point, bridging the deficiencies in underfitting and shortcomings of overfitting in MT models.

Having set the scene, and noted the specific RQs to be explored in this thesis, an overall summary of the research goals is an exploration of what is *lost in translation*, i.e. knowledge in the models that fails to be retrieved at the termination of the MT process. At the end of Chapter 2 we will conclude that, in spite of proving more accurate, syntax-aware phrases fail to be scored by the decoder as the most likely (highest probability) translation. This *knowledge* is lost in the pipeline. Consequently, we seek to trace and rescore translation hypotheses to minimize this loss in the two remaining core research chapters (3 and 4). This sort of error analysis in turn leads to shedding light on some of the *black box* processes[13] in the MT pipeline and demonstrate why the MT system decides upon a particular translation as the output.

Thus far, we have identified two avenues for research in the dominant framework of PB-SMT: linguistically motivated knowledge representation, and using system error diagnosis and analysis of the PB-SMT modeling-optimization-decoding pipeline to exploit this linguistic knowledge. We present the treebank-induced PB-SMT chunks as a case-

[11] Okita (2012) also deals with the issue overfitting in SMT. However it focusses on the word alignment stage of PB-SMT.

[12] Treebank-aware constraints refer to limiting the non-linguistic phrase pairs by filtering out those phrase pairs which do not adhere to linguistic boundaries introduced by the parallel treebanks.

[13] Although the software system used in our experiments (Moses) is open-source and none of the processes can be labeled *black box* in the technical sense, a majority of PB-SMT research, unlike the focus in this thesis, tends to avoid modifying certain processes like phrase extraction and tuning and treat them as black boxes.

study for identifying sub-optimal performance of PB-SMT modules. In this way we hope to trace the origin of deficiencies in the PB-SMT pipeline and minimize model errors through optimization and reranking. To summarize, this thesis presents our research on analysing errors in the PB-SMT system.

The remaining chapters of this thesis will seek to address the research questions through the inclusion of background information, overviews of past approaches and a series of experiments.

Chapter 2 In this chapter, we investigate the uniqueness and utility of linguistically motivated phrase pairs induced from percolated dependency trees in a standard PB-SMT system. After replicating the results of Hearne et al. (2008), we introduce the percolated dependency-induced translation model and scale up the training data used in our experiments by a factor of 13. Furthermore, we experiment with concatenating all possible combinations of the four types of phrase pairs (STR, CON, DEP and PERC) giving rise to 15 translation models with varying degrees of syntax-awareness. We also report our analyses of the degree of overlap and contribution of each phrase pair type in decoding. After presenting our results on employing several phrase pair combination and selection strategies including confusion network-based system combination and all-option candidate selection criteria, we conclude the chapter with the insight that the PB-SMT modeling-optimisation-decoding pipeline does not always select the most accurate candidate translation as the most likely (highest scoring) translation. This addresses RQ1 and RQ2.

Chapter 3 In this chapter, we explore the realm of model errors in PB-SMT by identifying the rank of oracle translations in the *n*-best lists generated by the decoder. We investigate rescoring the *n*-best lists to push the oracles up the ranks by reestimating the weights of the features used by the PB-SMT decoder. We introduce two novel rescoring methods. We experiment along several dimensions (two MT evaluation metrics used to identify the oracle, four language directions, and seven *n*-best list sizes) giving rise to

140 different MT systems. After conducting a range of contrastive analyses we conclude with our recommendations for minimising model errors in specific language pairs. This addresses RQ3.

Chapter 4 In this chapter, we extend the oracle-based training of the previous chapter and rescore the *n*-best lists by introducing additional features in a reranking (post-decoding) framework. Note that we distinguish between rescoring and reranking in that rescoring implies using the same set of features as used in decoding, while reranking implies introducing additional features. This addresses RQ4 and RQ5.

Chapter 5 In this chapter we summarise the contributions of this thesis and complement it with a number of potential research directions in the near future.

1.1 Publications

The research presented in this dissertation is more of an analytical and diagnostic study into the inner workings of a PB-SMT system. The novel concepts introduced in this research (percolated dependencies in PB-SMT, sub-optimal performance in syntax-aware PB-SMT, and oracle-based rescoring in PB-SMT) were published in several peer-reviewed conference proceedings.

- Srivastava and Way (2009) introduces the concept of phrase pairs induced from percolated dependencies as a unique and useful knowledge source for syntax-aware PB-SMT (Machine Translation Summit 2009)

- Srivastava et al. (2009) extends the previous work and is a multi-author collaboration demonstrating combination strategies for systems with multiple phrase extraction models and concluding that PB-SMT systems give sub-optimal performance (Example-Based Machine Translation Workshop 2009)

- Srivastava et al. (2011) introduces our oracle-based rescoring strategies for training in PB-SMT (European Association for Machine Translation Conference 2011).

Chapter 2

Treebank-based Phrase Extraction

In this chapter, we will address the first two research questions **(RQ1)** and **(RQ2)** posed in Chapter 1 by describing the extraction of phrase pairs from parallel treebanks annotated with percolated dependencies and evaluating the incorporation of these linguistically motivated chunks into the PB-SMT system. We extend this parallel treebank-based framework by introducing a novel annotation format called **percolated dependencies** and investigate their incorporation into the PB-SMT pipeline with several new translation models (cf. 2.5). As shown in Figure 2.1, this chapter focuses on alternative phrase extraction methodologies in a PB-SMT system.

The research strands covered in this chapter include parallel treebank induced phrase extraction, percolated dependencies, combining multiple translation models, and syntax-based reordering.

2.1 Syntax-aware Models in SMT

Incorporation of linguistic knowledge into the phrase extraction process has shown mixed results in recent years. For instance, Koehn et al. (2003), demonstrated that using syntax to constrain their phrase-based system actually harmed translation quality. In contrast, all of the following approaches have shown that augmenting the baseline string-based translation model with syntax-aware word and phrase alignments causes translation per-

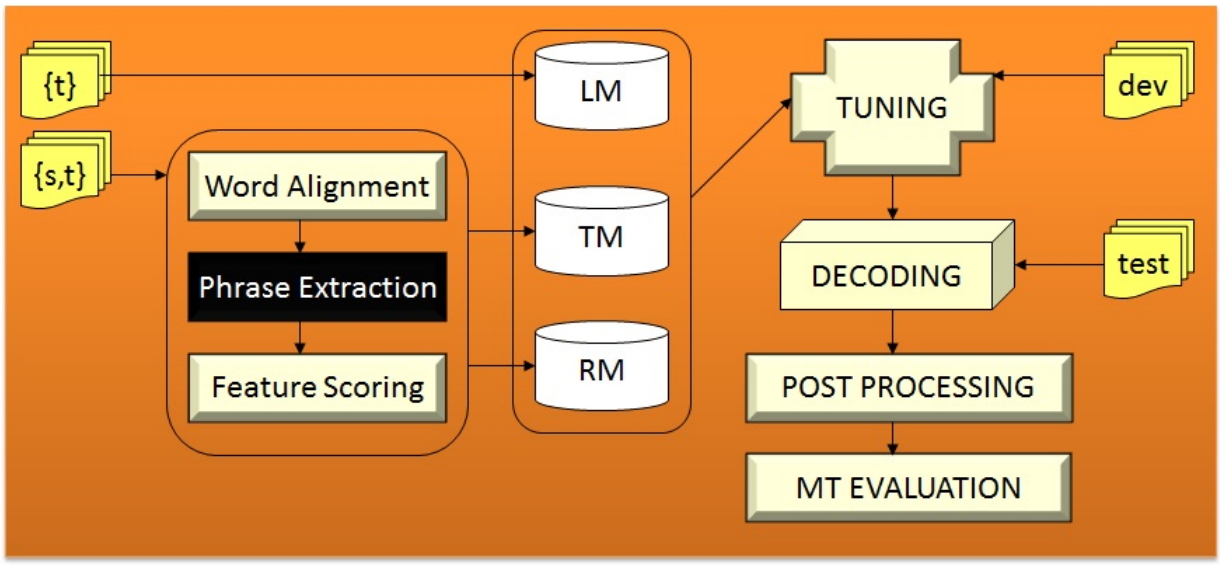


Figure 2.1: Schematic diagram of the modules in a Phrase-based Statistical Machine Translation System: *Phrase Extraction*.

formance to improve.

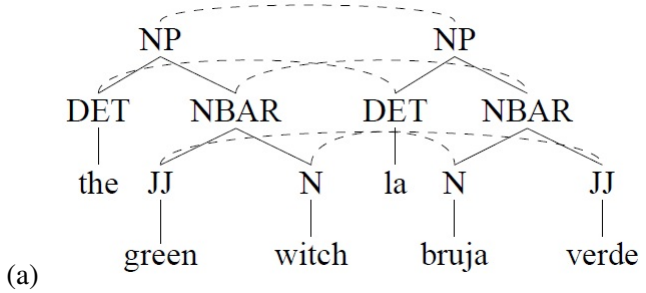
Groves and Way (2005) extract Example-based Machine Translation (EBMT) phrase pairs by monolingually chunking both the source and target sides using closed-class marker words (Green, 1979) and then aligning the resulting chunks using mutual information techniques.

Tinsley et al. (2007) extract phrase pairs by obtaining phrase structure parses for both the source and target sides using monolingual parsers and then aligning the subtrees using a statistical tree aligner. Hearne et al. (2008) go a step further by building on the work of Tinsley et al. (2007) and adding phrase pairs induced from dependency parse trees. Note that all these approaches work on string-based translation models, i.e. syntactic knowledge is merely used to extract linguistically motivated phrase pairs. The phrase translation tables[1] still contain unannotated translations of strings, just like in Moses (Koehn et al., 2007).

As is clear from this description, virtually all this work was done at Dublin City University (DCU). There also exist a number of other approaches (Chiang, 2005; Quirk et al., 2005; Galley et al., 2006) which have developed different models where the incorporation of syntax has shown itself to be beneficial. However such models are not restricted to the

[1] Phrase tables contain a list of source–target language phrase pairs with associated probabilities.

string-based translation model, but fall under tree-based SMT, and are thus beyond the scope of our research.



(a)

(b)

the green witch	↔	la bruja verde
green witch	↔	bruja verde
the	↔	la
green	↔	verde
bruja	↔	witch

Figure 2.2: Example of (a) a parallel treebank entry and (b) the set of extracted phrases; From Srivastava et al. (2009)

Figure 2.2 shows a constituency tree-aligned fragment from a parallel treebank and the resulting surface-level phrase pairs or chunks extracted. This technique of parallel treebank-induced phrase extraction has been used in a number of papers (Tinsley et al., 2007; Hearne et al., 2008) including our own. In Section 2.2, we extend the experiments of Hearne et al. (2008) by adding another syntax-aware phrase extraction methodology in the parallel treebank framework, namely *percolated dependencies* (Magerman, 1995). We also scale up the volume of the training data, and compare and contrast the resultant phrase tables and models (cf. sections 2.5.2 and 2.5.3).

2.2 Parallel Treebank-based Phrase Extraction

The standard method of extracting phrase-pairs from parallel data involves using union and intersection heuristics on both source-to-target and target-to-source word alignments (Och and Ney, 2003), in the Moses system (Koehn et al., 2007). This string-based extraction methodology gives rise to 'non-linguistic' chunk pairs, henceforth known as STR(ing).

In this section, we seek to investigate the performance of the baseline Moses MT system by changing one step only, namely the phrase extraction process (appearing as the

black shaded box in Figure 2.1). Specifically, this entails using three sets of syntactically motivated phrase pairs such as those extracted from node-aligned parallel treebanks. Tinsley et al. (2007) and Hearne et al. (2008) extracted phrase-pairs from constituency-aligned and dependency-aligned data, giving rise to two types of linguistic chunk pairs: CON and DEP respectively. Both these data sets were obtained by monolingual parsing of training sentences, subtree-aligning the parsed trees, and extracting word and phrase alignments. A prerequisite for this approach is the existence of constituency and dependency parsers for both the source and target languages.

Hearne et al. (2008) demonstrated on a very small set of training data that combining string-based extraction (baseline Moses) with either of the syntax-induced phrase extractions resulted in improved translation accuracy with a general trend toward preferring dependency-based over constituency-based phrases. However, there exist more robust and accurate phrase structure parsers than dependency structure parsers for most languages in NLP applications, which has led to alternate measures of automatically generating dependencies from phrase structure parses (cf. Nivre (2006) : 129–131).

In this piece of work, we heuristically obtain dependency parses by using lexical head information in constituency parse trees. While the head percolation tables themselves are nothing new (details in the following section (2.3)), the use of phrase pairs induced from them as a separate knowledge source in PB-SMT phrase tables is novel. This method of annotating and subsequently aligning percolated dependency parses gives rise to another set of aligned chunks: PERC. We then evaluate the uniqueness and utility of these alignments against STR, CON, and DEP alignments, and combinations thereof. A substantial portion of this research was previously published in Srivastava and Way (2009) and Srivastava et al. (2009).

2.3 Percolated Dependencies

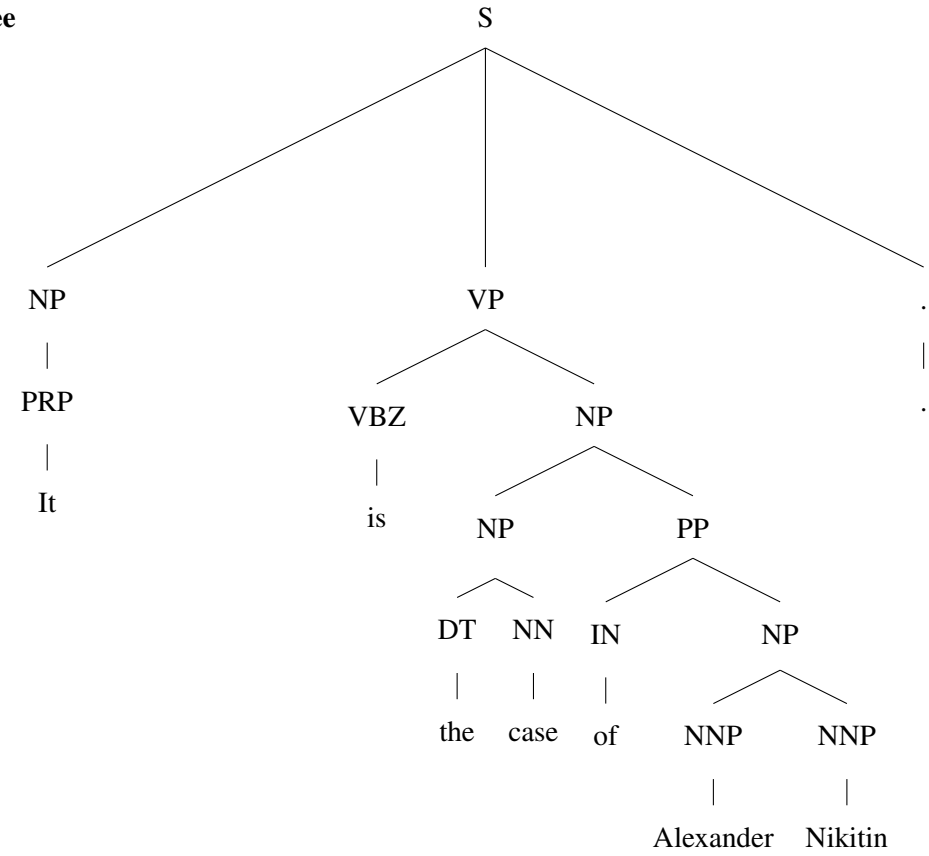
Syntactic theory deals with how sentences are structured or how the words are arranged in sentences (cf. van Valin Jr. (2001) : 1–5; 86–106; 110–142). There are two main ap-

proaches to describing syntactic structure, namely constituency grammar[2] (constituency tree in the example below) and dependency grammar[3] (dependency tree in the example below).

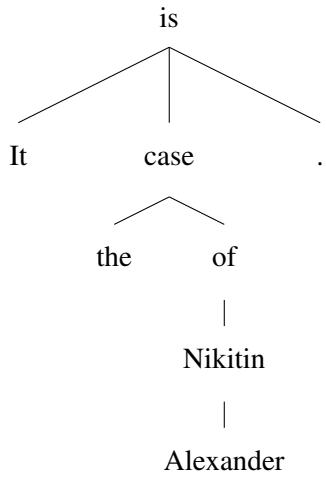
Example of a sentence and its corresponding dependency and constituency tree:

Sentence *It is the case of Alexander Nikitin .*

Constituency Tree



Dependency Tree



[2] Constituency grammar entails grouping words into units called constituents.

[3] Dependency grammar entails classifying the words into head (dominant element) and its dependents.

It is possible to obtain a dependency parse for a sentence from its constituency parse by exploiting lexicalized heads, i.e. head words of each phrase or constituent. In the absence of this information, a head percolation table (hand-coded rules) is used to select the head node in each constituent structure. For example, the syntactic head of a phrase *(NP (DET The) (NN box))* is the node *(NN box)*. Head percolation tables were first introduced in Magerman (1995) and implemented in Collins (1997).

Head percolation tables are so called because, to extract head-dependent information from a constituency parsed treebank, the lexical items are percolated like features from the heads to their parent projections. A head percolation table consists of hand-coded rules identifying the head-child of each node. We implemented the algorithm described in Xia (2001) to obtain head-dependent relations between words of a sentence. The head percolation algorithm will output the head or governor for each word in the sentence. In case the word is the head word of the sentence (e.g. *is* in the example above), it will be assigned a default value as its head.

Dependency trees can also have labels, i.e. classifying the relationship between each head and dependent word. For instance, the relationship between the dependent *the* and its head word *case* is *det*, short for determiner. Note that the above example of a dependency tree shows unlabelled dependencies which is what the output of the head percolation algorithm resembles. In our work, DEP dependency trees are obtained from a dependency parser (labelled dependencies) and the PERC dependency trees are obtained using the head percolation algorithm on constituency trees (unlabelled dependencies).

In order to further illustrate the workings of the head percolation algorithm, Figure 2.3 shows a constituency tree for the sentence. The head percolation algorithm applies the head percolation table to determine the head word for each word in the sentence and percolates these up the trees. Figure 2.4 shows the same constituency tree with each node subscripted with the head word of the corresponding subtree. For instance, the subtree *PP* is subscripted with the preposition *as* to reflect the fact that the noun phrase *a nonexecutive director* is dependent on the preposition *as*.

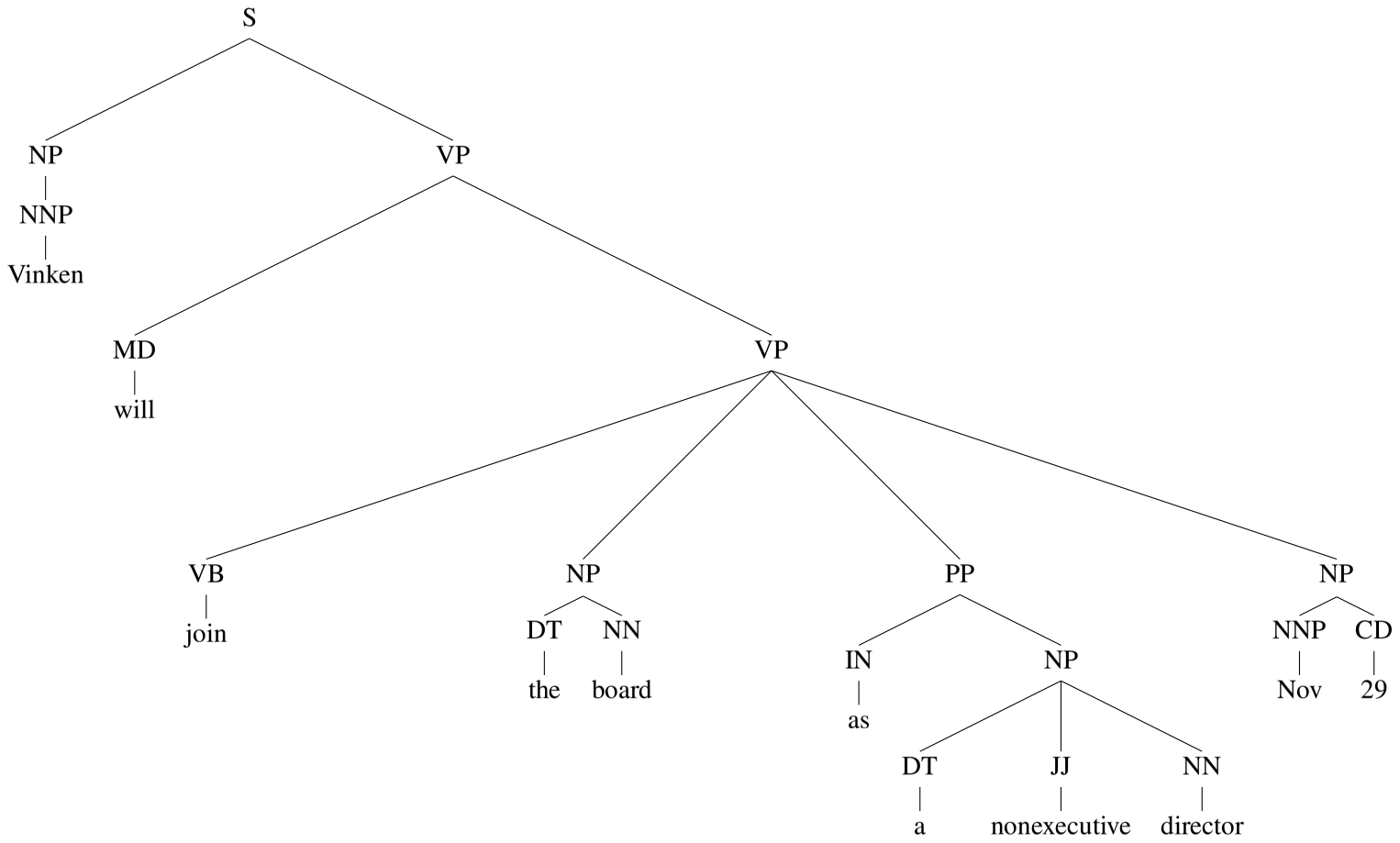


Figure 2.3: Constituency tree for the sentence *Vinken will join the board as a nonexecutive director Nov 29*

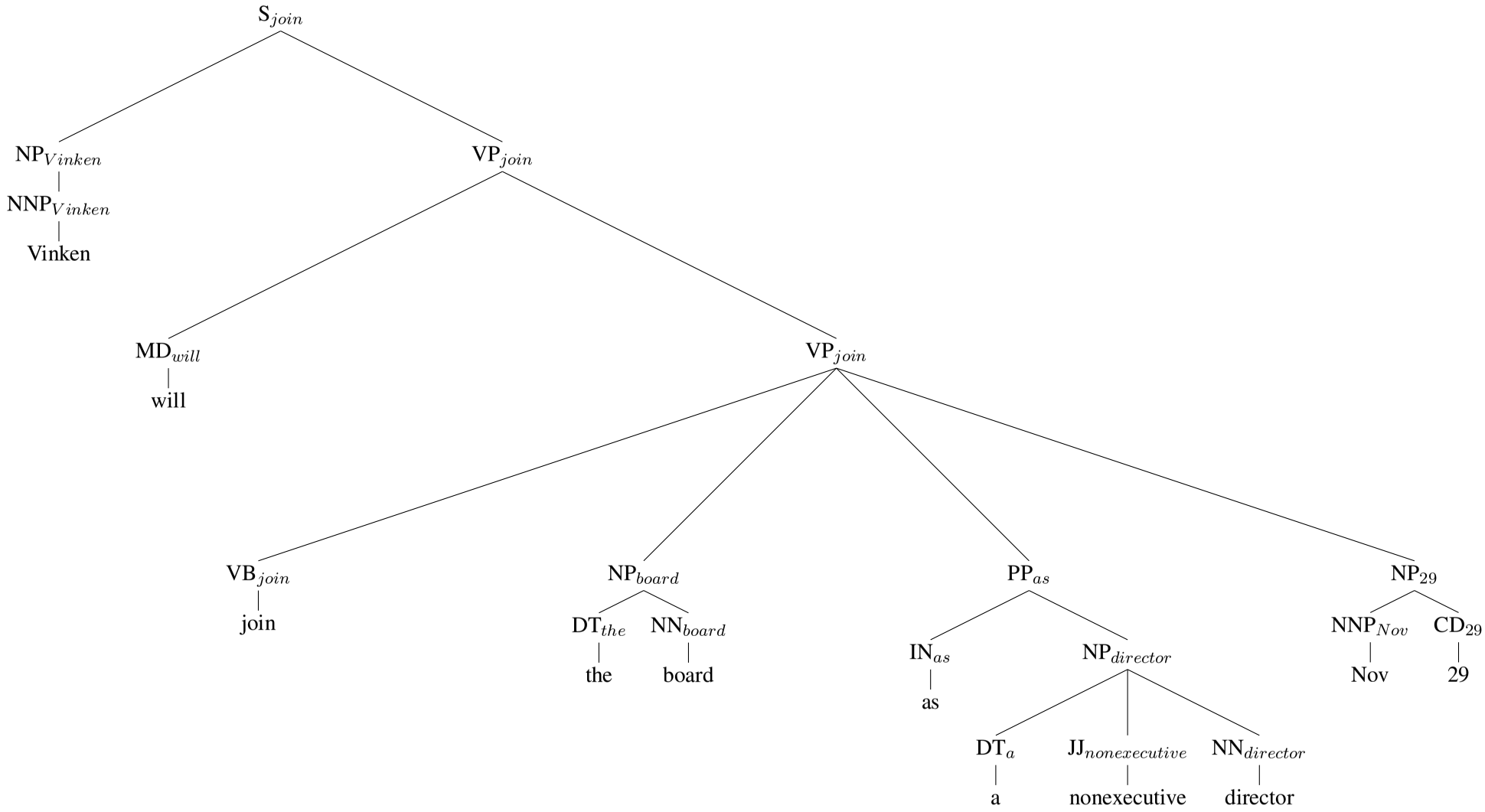


Figure 2.4: Constituency tree with labelled head words for the sentence *Vinken will join the board as a nonexecutive director Nov 29*

In our approach to phrase extraction, we construct translation units or phrase pairs from aligned source–target constituents (CON) and aligned source–target groupings of a head word with its dependents (DEP, PERC). The idea is to segment a sentence into meaningful syntactic units (syntax-aware phrase pairs) rather than any arbitrary sequence (STR).

Producing phrases via a semi-automatic and language-independent process of morphological and syntactic analysis may remove the need for compatible NLP tools per language pair, which generalises the approach to language pairs where no such tools exist.

2.4 Data and Tools

We outline the MT system and data used in our experiments before describing the multitude of techniques in Section 2.5 to evaluate the impact of phrase pairs extracted from percolated dependencies.

CORPORA	TRAIN	DEV	TEST
JOC (sentences)	7,723	400	599
EUROPARL (sentences)	100,000	1,889	2,000

Table 2.1: Statistics of French–English corpus used in treebanking experiments

We use two different datasets as shown in Table 2.1. We obtain results on a small parallel corpus of approximately 7,700 parallel sentences—the JOC English–French parallel corpus (Chiao et al., 2006) [7,723 train + 400 dev + 599 test sentences]—and a larger set of 100,000 parallel sentences extracted from the freely available Europarl corpus (Koehn, 2005) [100,000 train + 1,889 dev + 2,000 test sentences]. The JOC corpus contains excerpts from the Official Journal of the European Community and the Europarl corpus contains parliamentary proceedings of the European Union. Both datasets fall under the same domain. Experimenting on the JOC corpus allows us compare our results directly with those of Hearne et al. (2008), while at the same time we successfully scale up their work by almost 13 times in the larger experiment.

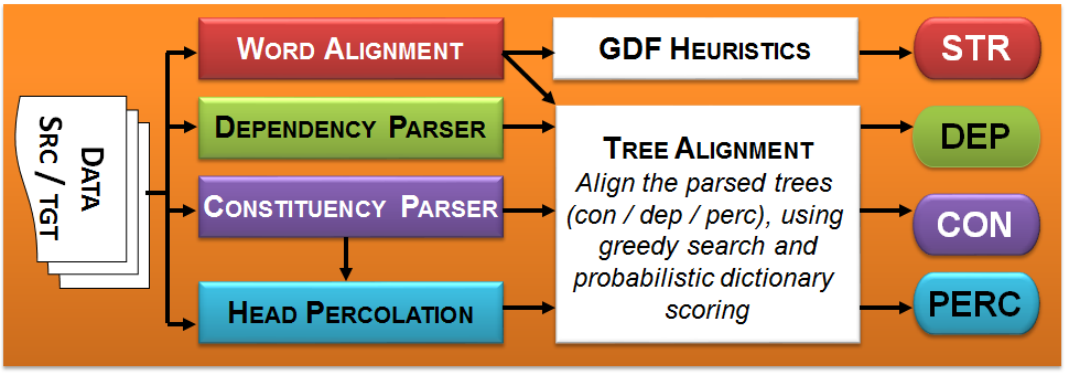


Figure 2.5: Four types of phrase extraction applied to PB-SMT.

We also used an open source tree aligner (Zhechev, 2009) to obtain subtree-alignments for the linguistic chunks CON, DEP, and PERC. The tree aligner works by performing a greedy search on all possible alignments between the tree pair nodes and scores using lexical probabilities to select the highest-scoring alignment hypothesis. Constituency parse trees were obtained by using the Berkeley parser (Petrov et al., 2006) for both the French and English sides, and dependency parse trees were obtained from the English and French versions of the Syntex parser (Bourigault et al., 2005). The dependency structures were converted into a bracketed format to enable use of the tree aligner. This is graphically depicted in Figure 2.5.

We used GIZA++ (Och and Ney, 2003)for word alignment, SRILM (Stolcke, 2002) for building a 5-gram language model, Minimum Error Rate Training (Och, 2003) for tuning, and the Moses beam search decoder (Koehn et al., 2007) in each of our systems. Thus the only difference between each system is in the phrase table used in the translation model.

2.5 Experiments

2.5.1 Vanilla Merge Translation Tables

For the purposes of our experiments, we create 15 possible combinations of translation tables from the four types of phrase extractions, namely STR, CON, DEP, and PERC. The combination of two or more systems is carried out by merging the individual phrase ta-

bles and re-estimating the phrase translation scores as defined in Moses. We label this method of combination **vanilla merge**.[4] For example, the translation table of the system C+D+P is computed by concatenating the extracted phrase tables CON, DEP, and PERC and then re-estimating the probabilities. Each of the 15 configurations were run on both the JOC and Europarl datasets in the French–English translation direction. The results are displayed in Tables 2.2 and 2.3 respectively. We evaluate the MT system performance using five evaluation metrics. These are BLEU (Papineni et al., 2002), NIST (Doddington, 2002), METEOR (Banerjee and Lavie, 2005), WER (Word Error Rate; (Niessen et al., 2000)) and PER (Position-independent WER; (Leusch et al., 2003)). Note that statistical significance tests on the different system performance for all experiments were computed using bootstrap resampling methods on BLEU described in Koehn (2004). An improvement in system performance at a confidence level above 95% (p-value = 0.05) is assumed to be statistically significant. The bold-faced figures in Tables 2.2 and 2.3 indicate the best -performing systems on a particular evaluation metric.

SYSTEM	BLEU	NIST	MET	WER	PER
STR (S)	31.29	6.31	63.91	61.09	47.34
CON (C)	30.64	6.34	63.82	60.72	45.99
DEP (D)	30.75	6.31	64.12	61.34	46.77
PERC (P)	29.19	6.09	62.12	62.69	48.21
S + C	**32.87**	**6.55**	**65.04**	58.70	44.93
S + D	32.69	**6.55**	64.98	58.66	44.81
S + P	32.34	6.48	64.56	59.42	45.51
C + D	31.24	6.41	64.40	60.28	45.76
C + P	30.99	6.36	63.84	60.47	45.81
D + P	31.40	6.41	64.41	60.28	45.87
S + C + D	32.70	6.53	64.86	58.45	**44.73**
S + C + P	32.49	6.48	64.65	58.82	45.22
S + D + P	32.62	6.51	64.82	58.72	45.07
C + D + P	31.46	6.41	64.33	59.90	45.58
S+C+D+P	32.82	**6.55**	65.03	**58.35**	44.77

Table 2.2: Summary of the results on JOC test data

[4] This is so named because it is the most direct way of combining multiple phrase translation tables wherein each table is weighted equally with no bias.

SYSTEM	BLEU	NIST	MET	WER	PER
STR (S)	28.50	7.00	57.83	57.43	44.11
CON (C)	25.64	6.55	55.26	60.77	46.82
DEP (D)	25.24	6.59	54.65	60.73	46.51
PERC (P)	25.87	6.59	55.63	60.76	46.48
S + C	**29.50**	**7.10**	**58.55**	56.62	**43.40**
S + D	29.30	7.08	58.43	56.84	43.62
S + P	29.45	**7.10**	58.54	56.73	43.43
C + D	26.32	6.69	55.56	59.97	45.90
C + P	26.37	6.62	56.05	60.41	46.40
D + P	26.57	6.74	55.83	59.53	45.62
S + C + D	29.29	7.09	58.48	56.70	43.41
S + C + P	29.49	**7.10**	58.50	**56.59**	43.45
S + D + P	29.39	7.09	58.49	56.80	43.65
C + D + P	26.90	6.75	56.14	59.38	45.53
S+C+D+P	29.40	7.09	58.49	56.67	43.49

Table 2.3: Summary of the results on Europarl test data

Automatic System-level Evaluation

What is quite clear from analysing the results on both the JOC and Europarl corpora is the very strong baseline[5] performance of the STR system. For the pairwise comparison, any system combination omitting STR-induced phrase pairs underperforms. Note that in their experiments, both Groves and Way (2005) and Tinsley et al. (2007) acknowledge, as we do here, that *n*-gram-induced phrase pairs are required for both improved translation performance and coverage.

Working on the JOC corpus allowed us to directly compare our novel phrase induction method against the work of Hearne et al. (2008). While we could not improve upon their results (when substituting D with P in any system in Table 2.2) for the JOC corpus, running experiments on the 13 times larger Europarl data set showed clear performance gains (a relative increase of as high as 2.49% in BLEU when replacing D with P in any system in Table 2.3) over their method when the PERC phrases were utilised. Even if our method did not outperform theirs, our method would still be of use if no separate dependency parser were available for either the source or target language or both.

[5] The term *strong* refers to the significant difference in the evaluation scores of STR against those of the syntax-aware systems CON, DEP and PERC.

While the best-performing system combination on both tasks was where STR and CON phrases were merged, for almost all metrics, the lowest WER rates were observed when PERC chunks were included.

Automatic Sentence-level Evaluation

In addition, there are quite a few sentences (when computing sentence-level WER scores for each of the four base systems, PERC ranked 2nd best with over 25% sentences on both JOC and Europarl datasets, or 546 out of 2000 sentences) where PERC performs better than any other system, as in (1) below.

(1) *Source*: La commission entend-elle garantir plus de transparence à cet égard?

Ref: Does the commission intend to *seek more* transparency in this area?

STR: Will the commission ensure that more than transparency in this respect?

CON: The commission will the commission ensure greater transparency in this respect?

DEP: The commission will the commission ensure greater transparency in this respect?

PERC: Does the commission intend to *ensure greater* transparency in this regard?

Note that the propensity of the baseline STR model to omit the verb can be seen to good effect here. Both CON and DEP phrases repeat the translation of the subject NP. In contrast, the translation using PERC phrases is both fluent and accurate, despite not mimicking exactly the reference translation and so is not considered a perfect translation by BLEU.

Manual Evaluation

The lexical differences between the outputs and the reference translation (*ensure* versus *seek* and *greater* versus *more* in (1)) lead us to speculate that the gains from PERC are not accurately reflected in the automatic evaluation scores. That is, perfectly acceptable target sentences are output via PERC which are unreasonably penalised by the MT evaluation metrics. Accordingly, we also performed a manual evaluation on a random selection of 100 sentences from the Europarl testset. A human annotator[6] was shown pairs of sentences along with the source and reference translations and asked to grade whether one system was better than the other or if they were of equal calibre. Table 2.4 shows the results where a human evaluator judged the translation quality of 100 random sentences between systems which contained and did not contain PERC chunks. The sum of each row amounts to 100 indicating the total number of sentences judged.

System Pairs	Equivalent Performance	1st System is Better	2nd System is Better
C & P	46	27	27
D & P	35	30	35
S+C & S+P	60	27	13
S+C & P	28	57	15
S+C & S+C+P	37	37	26
S+D & S+P	58	24	18

Table 2.4: Summary of Pairwise System Comparison (Number of Sentences) by a Human Annotator for select systems: Europarl data

To summarise, while PERC and CON systems performed better than each other on the same number of sentences (27%), PERC performed 5% better than DEP. When comparing systems S+C and S+C+P (where the automatic evaluation score differences were not statistically significant), the former system was 11% better. However, there were a number of sentences (26%) in which PERC was responsible for an output's superior quality, although no pattern was immediately discernible.

This leads credence to our belief that the gains from PERC-enabled systems are not accurately reflected in the automatic evaluation metrics (BLEU, NIST, METEOR, WER,

[6] The annotator was a bilingual speaker of French and English.

PER). We conducted a range of other tests in order to evaluate the uniqueness (degree of difference from other phrase extractions) and utility (contribution to MT system performance) of PERC chunks, as described in subsequent experiments.

2.5.2 Phrase pair Overlap

Phrase Types	Common to Both	Uniq. Align. in 1st type	Uniq. Align. in 2nd type
STR & CON	161,314	1,983,162	501,822
STR & DEP	144,834	1,999,642	438,698
STR & PERC	143,162	2,001,314	421,850
CON & DEP	399,220	263,916	184,312
CON & PERC	497,159	165,977	67,853
DEP & PERC	376,377	207,155	188,635

Table 2.5: Number of common and unique alignments (phrase pairs) for each method: Europarl data

The total number of entries in each of the four phrase tables (Europarl data) are STR: 2,144,476, CON: 663,136, DEP: 583,532, and PERC: 565,012. We can see that the CON t-table is just 31% of the size of the full STR t-table, with DEP just 27% and PERC even smaller at just 26% of the size. By correlating the t-table[7] sizes and the system performance in Table 2.3 of the four base systems,[8] it can be concluded that the much smaller pure syntactic systems (CON, DEP, and PERC) give a high-quality yield.

Table 2.5 compares pairs of phrase tables and displays the overlap as well as unique phrase pairs extracted under each of the four methods. It is interesting that despite the huge size of the STR phrase table, there is very little overlap with any of the other methods; the largest overlap with STR is using CON phrases, but this amounts to only 7.5% of the STR phrase table derived via CON, and only 24% of the CON phrase table derived via STR.

The largest overlap in pure numerical terms is between CON and PERC; 75% of the CON phrase table is common with PERC, whereas 88% of the PERC phrase pairs are

[7] T-table or translation table is another term for phrase table.
[8] These are STR, CON, DEP and PERC systems.

common with CON. Given that the PERC phrases are derived directly from the CON trees, one might have expected these two to have the biggest intersection. However, surprisingly, the output (translated sentences produced by CON and PERC systems) has a 30% overlap only. Therefore, it seems that despite a huge overlap in the phrase table configurations, the systems are different enough to produce different translations. We leave for future work an investigation into any bias here.[9] We also acknowledge the fact that different parsers could jointly produce even more new phrase pairs.

Note that the overlap numbers (column 2 (*Common to Both*) from Table 2.5) refer to identical phrase pairs in both phrase tables under consideration, i.e. overlap on both the source language (French) and the target language (English) side. The remaining phrase pairs in each of the two phrase tables under consideration (Unique Alignments in each type: columns 3 and 4 from Table 2.5) are bound to have phrase pairs wherein there is an overlap on the source language side but not on the target language side, and vice versa. We perform a deeper investigation into such one-sided overlaps between the six pairs of phrase tables and report in Table 2.6 on the number of phrase pairs[10] which have a common source language phrase aligned to a dissimilar target language phrase, i.e. overlap on the source side only. This will demonstrate whether we are extending existing phrases in one phrase table with additional translations from another phrase table.

Phrase Types	**Ext. Src. Alig. in 1st type**	**BLEU IMP. Over 2nd**	**Ext. Src. Alig. in 2nd type**	**BLEU IMP. Over 1st**
STR & CON	143,317	3.86	211,203	1.0
STR & DEP	141,112	4.06	203,363	0.80
STR & PERC	132,607	3.58	189,189	0.95
CON & DEP	88,706	1.08	80,279	0.68
CON & PERC	67,747	0.50	50,102	0.73
DEP & PERC	73,052	0.70	66,873	1.33

Table 2.6: Number of extended phrase pairs (overlap on source side only) and BLEU score improvements for combined system over single system for each method: Europarl data

Table 2.6 gives the number of phrase pairs that overlap on the source language side in

[9] Specific details are covered in Chapter 5, Section 5.3.1.

[10] Note the phrase pairs in Table 2.6 (Extended Source Alignments in each type: columns 2 and 4) are a subset of the phrase pairs in Table 2.5 (Unique Alignments in each type: columns 3 and 4), respectively.

a pairwise comparison of the four types of phrase tables. The BLEU IMP. (columns 3 and 5) are displayed to verify the impact of the extended phrases in terms of BLEU evaluation metric. As evident, there is a direct correlation between the number of unique phrase pairs on the target side and the system level performance. This is as expected – the norm being more data implies better performance.

For each of the four phrase extraction methods, the average number of phrase pairs per sentence and the highest number of phrase pairs in a sentence were computed as follows: JOC corpus– (STR: 35.37 (134), CON: 17.62 (71), DEP: 17.82 (71), PERC: 8.45 (53)) and Europarl corpus– (STR: 20.33 (45), CON: 10.82 (27), DEP: 10.67 (27), PERC: 10.66 (26)). Similar performance is seen between the three non-STR methods on Europarl, whereas on JOC our PERC model produces fewer alignments. The smaller number of phrase pair alignments might very well prove useful for systems with a smaller footprint requiring smaller t-tables (Sanchez-Martinez and Way, 2009).

STR Phrase Pairs			**CON Phrase Pairs**		
la commission	↔	the commission	la commission	↔	the commission
des	↔	of the	le conseil	↔	the council
, mais	↔	, but	ce rapport	↔	this report
, nous	↔	, we	le rapport	↔	the report
, je	↔	, i	en europe	↔	in europe
DEP Phrase Pairs			**PERC Phrase Pairs**		
la commission	↔	the commission	la commission	↔	the commission
le conseil	↔	the council	le conseil	↔	the council
ce rapport	↔	this report	ce rapport	↔	this report
le rapport	↔	the report	le rapport	↔	the report
l' union	↔	the union	l' ue	↔	the eu

Figure 2.6: Top 5 phrase pairs (with target length constrained to 2 words) for each of the four phrase extractions, namely STR, CON, DEP, and PERC: Europarl data

A small sample of the types of chunks produced by each of the four phrase extraction methodologies in Figure 2.6 gives a clearer picture of how STR phrase pairs differ from the linguistically motivated phrase pairs (CON, DEP, and PERC). For example, the STR t-table contains a large number of non-linguistic sequences of words, often containing punctuation marks.

Having investigated the differences between the chunking methods, the next, more

important step is to evaluate whether these unique chunks are of use in PB-SMT.

2.5.3 Phrase Type Used in Decoding

The PB-SMT decoder Moses translates (as is the norm) a sentence by segmenting the sentence into phrases and selecting their translation from the phrase table. Moses (Koehn et al., 2007) can be run in 'trace' mode (-t switch) in order to investigate what particular phrases are being selected to derive the translation at any particular time.

TABLE	JOC	EP
STR (S)	2090	3423
CON (C)	95	419
DEP (D)	111	402
PERC (P)	236	385
S & C	44	287
S & D	87	280
S & P	61	275
C & D	301	330
C & P	91	364
D & P	31	305
S & C & D	196	222
S & C & P	73	259
S & D & P	8	220
C & D & P	780	322
ALL	1261	238
NONE	656	4017

Table 2.7: Analysis of which phrases the decoder uses in decoding the test data, when trained on the S+C+D+P translation model

In Table 2.2, we demonstrated that all four sets of phrase pairs could be combined in one phrase table in what we called the 'S+C+D+P' system. In order to translate the Europarl test set of 2,000 sentences, 11,748 phrases were found to be of use. These comprised 5204 STR phrases (of which 3423 were unique, i.e. not produced by any of the other three phrase tables), 2441 CON (419), 2319 DEP (402), and 2368 PERC (385). When it came to a pairwise comparison, the biggest overlap was between CON and PERC. As with our finding regarding Table 2.5, we will investigate in further work whether there was any bias between these two phrase induction methods. In the case of the JOC corpus,

for a test set of 599 sentences, 6,121 phrases were found to be of use. These comprised 3820 STR (2090 unique), 2841 CON (95), 2775 DEP (111), and 2541 PERC (236). Note, however, that for the JOC corpus, we found the biggest overlap to be between the CON and DEP phrase tables. As far as triples are concerned, by far the greatest overlap was between CON, DEP and PERC, with an intersection of 780 phrase pairs (the next nearest was just 196). Overall, 1261 phrase pairs were found by each of the four methods. The details for both corpora can be found in Table 2.7.

In another experiment (Section 2.5.5), we extract each of these resources as separate phrase tables in the log-linear framework, as it should be the case that where a set of phrase pairs has been verified by all four methods, these can be considered to be of high quality, and worthy of a large weight in the combination of translation resources.

2.5.4 System Combination

An alternative to combining the phrase tables (either directly or via some prioritised weighting) is to use Minimum Bayes Risk and Confusion Network decoding (MBR-CN framework; (Du et al., 2009)) to combine phrase pairs at the system level (after decoding) rather than at the phrase table level (during training). This was evaluated by combining the four base systems – STR, CON, DEP, and PERC – as well as performing a system combination on the entire set of 15 systems. These results were published in Srivastava et al. (2009).

System	BLEU	NIST	METEOR
MBR (4 systems)	0.2952	6.85	0.5784
CN (4 systems)	0.3070	7.06	0.5852
MBR (15 systems)	**0.3260**	7.32	**0.6050**
CN (15 systems)	0.3251	**7.33**	0.6039

Table 2.8: Results of MBR-CN system combination on the systems in in Table 2.2: Europarl data

The results of these experiments are shown in Table 2.8. The results demonstrate

that, when combining only the four systems (against the STR, the best-performing system in this sub-group), there is a 7.16% relative improvement in BLEU score. Furthermore, when all 15 systems are passed through the Confusion Network (row 4 in Table 2.8), we see a 12.3% relative improvement in BLEU score. The improvements are reflected across all evaluation metrics. We attribute these gains to the fact that the translation output produced by the CON and PERC systems (which had the biggest overlap in their phrase pairs: cf. Section 2.5.2) has a mere 30% overlap (i.e. only 30% of the translations are identical). Therefore, it again seems that despite a huge overlap in the phrase table configurations, the systems are different enough to produce different translations. Consequently, the divergences between the phrase tables produced by the various phrase segmentation strategies can be successfully exploited using a system combination framework.

2.5.5 Multiple Translation Tables

In using multiple knowledge sources (STR, CON, DEP, PERC), so far we have used a single translation table while decoding. The different types of phrase pairs were all merged into one, with their relative frequencies recalculated. The drawback to this vanilla merging is that if induced by only one of the extraction methodologies, correct phrase alignments could be assigned a lower probability than incorrect alignments occurring in all extraction methodologies.

SYSTEM	BLEU	NIST	METEOR	WER	PER
BASE$_S$	26.43	6.66	55.63	60.16	46.57
BASE$_C$	23.27	6.24	53.38	63.56	49.08
BASE$_D$	22.88	6.23	52.99	63.81	49.06
BASE$_P$	23.13	6.25	53.14	63.58	48.95
BASE$_{PS}$	26.75	6.73	55.99	59.65	46.11
ALL$_{PS}$	22.66	6.15	52.86	64.13	49.50
ANY$_{PS}$	26.65	6.72	55.88	59.69	46.21
BASE$_{PCDS}$	26.60	6.72	55.98	59.86	46.26
ALL$_{PCDS}$	21.11	5.92	51.50	65.86	50.91
ANY$_{PCDS}$	26.58	6.66	55.65	59.97	46.53

Table 2.9: Summary of the results on multiple translation tables: Europarl data

One way to overcome this is not to merge but consider each knowledge source separately as a stand-alone translation table. We conducted experiments (shown in Table 2.9) on two techniques of using multiple translation tables: ALL and ANY. Under the ANY setting, the phrase pair can be present in any of the t-tables. However, the order in which the decoder views each t-table is determined by the user. Therefore, ANY_{PS} is different from the configuration $\text{ANY}_{SP.}$ Under the ALL setting, a phrase pair must be present in and scored by all the t-tables in order to be selected by the decoder. Hence, performance under the ALL combination is inferior to that under ANY.

Both configurations were compared against the baseline system in two scenarios: (1) Using two translation tables (STR and PERC); (2) Using four translation tables (STR, CON, DEP, and PERC). The multiple translation table strategy did not help as it was either worse or equal to the performance by a system using vanilla combination.

2.5.6 Other Decoder Configurations

Given an input string of words to translate, a number of phrase translations could be applied. Each such applicable phrase translation is called a translation option in standard SMT literature. We experimented with the method used by the decoder to select the translation options from a given phrase table. None yielded any gains. We used Minimum Bayes Risk (MBR) to determine the best translation instead of log-linear score computation in the baseline. Letting the decoder see all translation options (ALL-OPTS) rather than a fixed 20 options per phrase (standard setting in BASELINE) did not help either. The results for the S+C+D+P system are shown in Table 2.10. Most of these experiments either gave a lower system performance or a performance on a par with the baseline system.

SYSTEM	BLEU	NIST	METEOR	WER	PER
BASELINE	26.60	6.72	55.98	59.86	46.26
MBR-SCORE	26.56	6.70	56.13	59.91	46.14
ALL-OPTS	26.51	6.71	55.91	59.90	46.24

Table 2.10: Summary of the results on using MBR and ALL-OPTS on the SCDP system: Europarl data

2.5.7 Reordering Tables

So far we have only used parallel treebank-induced chunks in the phrase table, and not the reordering table. The reordering table retained the Moses baseline (STR) orientations even in systems which did not contain STR chunks in the phrase table. Theoretically, it is good practice for both the phrase and the reordering tables to contain the same phrase pairs. This led us to create reordering tables for phrase pairs induced from parallel treebanks. However, somewhat surprisingly, this modification proved to be yet another instance where we were not able to outperform the baseline. All systems in Table 2.11 were tested on STR-based reordering tables (Base) as well as reordering tables containing only syntactic phrases in the phrase table (Self).

In order to generate reordering tables for phrase pairs induced from parallel treebanks, the syntactic phrases were rendered into the same format as their non-syntactic counterpart. This implies generating a lexical weighting, i.e. a word alignment between source-language and target-language phrase pairs. Thus each phrase-pair extracted from the parse trees has its word alignment (source-language and target-language word mappings) generated. If a word alignment cannot be found the phrase pair is rejected.[11] This technique ensures that all systems are compatible and all phrase entries considered are in the orientation model. This also implies that the number of phrase pairs collected for translation and orientation (reordering) tables is often less than the initial number of phrase pairs extracted from parse trees. Thus all phrase entries considered are in the orientation model.

SYSTEM	BLEU	NIST	METEOR	WER	PER
CON$_{Base}$	23.27	6.24	53.38	63.56	49.08
CON$_{Self}$	22.78	6.13	51.86	64.60	50.55
DEP$_{Base}$	22.88	6.23	52.99	63.81	49.06
DEP$_{Self}$	22.24	6.00	50.95	65.19	51.46
PERC$_{Base}$	23.13	6.25	53.14	63.58	48.95
PERC$_{Self}$	22.50	6.10	51.58	64.59	50.69
SC$_{Base}$	26.70	6.72	55.88	59.76	46.23
SC$_{Self}$	26.70	6.69	55.49	60.05	46.53

Table 2.11: Summary of the results on creating reordering tables from phrases contained in the phrase table: Europarl data

[11] Note this same technique is used to create the translation tables.

It is apparent that the orientation models learnt from baseline phrases are extremely important to system performance. This once again reinforces our notion that syntax-based phrases can augment the performance of a string-based system, but not replace it altogether. Therefore, the differences between the reodering models of the S+C system (last two rows in Table 2.11) are considerably less than in stand-alone syntax-based systems (Base and Self configurations of CON, DEP, and PERC).

Perhaps the distance-based reordering model is not sufficient to take advantage of reordering information stored in syntactic phrases. One way to exploit syntactic reordering models would be to implement and use hierarchical reordering models in the PB-SMT framework, as shown in Galley and Manning (2008). At the time of this work, this model implementation[12] was not yet available. This is again outside the scope of our work as it implies using tree-based SMT models.

2.5.8 Oracles amongst SCDP

In order to address our second research question stating whether the PB-SMT system is able to optimally exploit linguistically motivated phrase pairs or the PB-SMT pipeline is flawed, we make some general observations on the afore-described experiments.

In Section 2.5.6 the failure of ALL-OPTS model to outperform the baseline system demonstrates that a system which takes into account all the translation options (and is therefore very time-consuming) instead of performing a non-exhaustive search, still fails the access translation options or phrase pairs which are linguistically motivated and hence more accurate.

In Section 2.5.1, we observe that accuracy is lost in the vanilla merge combining models, that is the performance of all four phrase extractions merged together (SCDP) is worse than when using just two of them (SC). An additional experiment is performed which shows that the system SCDP is worse than the system obtained by selecting the translation amongst S, C, D, P closest to the reference (has the best sentence-level score). This is similar to selecting an oracle translation amongst 4-best lists as shall be described

[12] The hierarchical reordering model was implemented in Moses in 2010.

in the next chapter.

This will help support two claims: (1) Different phrase extractions do not merge well in the decoder when used in combination implying that the PB-SMT system is flawed, and (2) Significant redesigning (reranking either post or during decoding) is required to profitably exploit multiple knowledge sources. Moreover, the SCDP table contains combined information from S, C, D, and P phrase tables and hence should not give sub-optimal performance because it has access to all the phrase pairs. This implies the phrase pairs are scored erroneously and must be rescored.

2.6 Conclusions

While producing smaller translation models and believed to contain more useful (syntax-aware) phrases than the standard string-based extraction, the syntax-based extractions may perform worse than the PB-SMT string-based baseline, especially as the amount of training data increases (cf. Zollmann et al. (2008)). Lopez (2009) argues that due to the lack of systematicity in MT system development, it is extremely difficult to compare systems purporting to be of different types, and nigh on impossible to pinpoint exactly to which component any gains in performance might accurately be attributed. However, it has been observed by many researchers that rather than replacing one with the other, combining both types of induced phrases into one translation model can significantly improve translation accuracy. Thus we can supplement SMT phrases with syntax-aware phrases.

Most system development today uses one particular approach to generate phrase pairs for use in translation, namely that of Koehn et al. (2003) (or perhaps more accurately, using the word- and phrase-alignment scripts in Moses (Koehn et al., 2007)). However, some researchers have pointed out that system performance can be increased when chunks induced by other methods (EBMT (Groves and Way, 2005); constituency parsers (Tinsley et al., 2007); dependency parsers (Hearne et al., 2008); percolated dependencies (Srivastava and Way, 2009)) are added to the SMT phrase table.

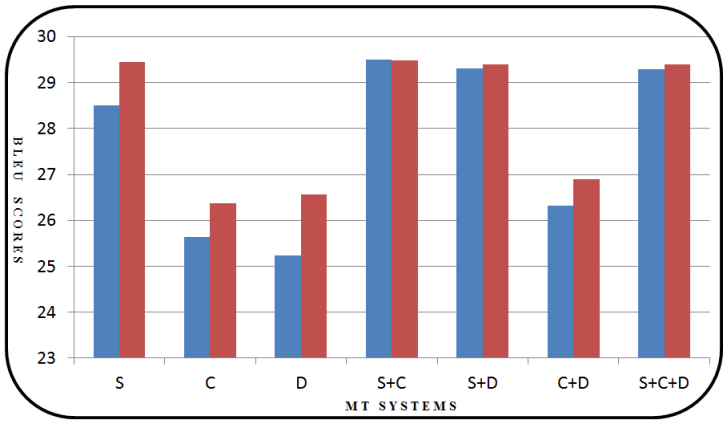


Figure 2.7: Bar graph to show that adding PERC chunks (red bar) to any system (blue bar) generally boosts the BLEU score: Europarl data. These systems are also reviewed in Table 2.3.

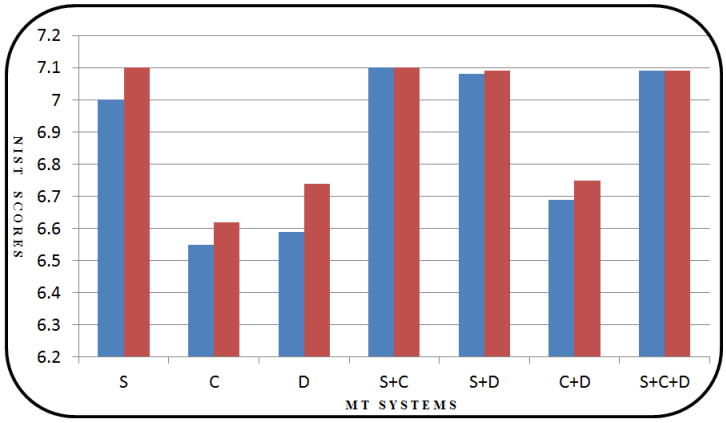


Figure 2.8: Bar graph to show that adding PERC chunks (red bar) to any system (blue bar) generally boosts the NIST score: Europarl data. These systems are also reviewed in Table 2.3.

The general point is the following: adherence to one approach may lead to sub-optimal system performance; if any one phrase pair induced by some other method proves to be useful, then ignoring other approaches will cause translation performance to deteriorate, even when the data size is increased (Srivastava and Way, 2009).

Accordingly, in this chapter we investigated whether phrase pairs induced via head percolation (Magerman, 1995) might prove useful in PB-SMT. In a number of experiments, we showed that the number of chunks, and their content, was different for each of the four methods: STR, CON, DEP, and PERC. Furthermore, we showed that system performance improved significantly when PERC phrases were added to the phrase table of any other system. Figure 2.7 shows that adding PERC chunks to any system shows a general trend towards boosting scores for BLEU. While we do not include similar graphs for the other automatic evaluation metrics, this tendency is confirmed across all evaluation metrics used in our experiments for both corpora. Figures 2.8, 2.9, 2.10, and 2.11 show the same for NIST, METEOR, WER, and PER scores respectively.

The utility of percolated dependencies in PB-SMT was validated on two tasks for

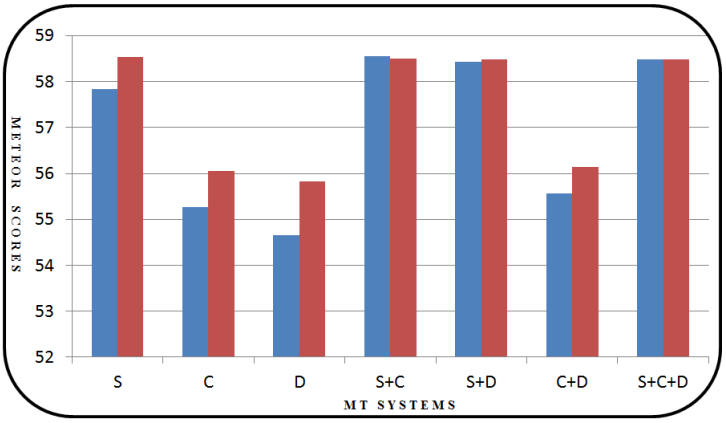


Figure 2.9: Bar graph to show that adding PERC chunks (red bar) to any system (blue bar) generally boosts the METEOR score: Europarl data. These systems are also reviewed in Table 2.3.

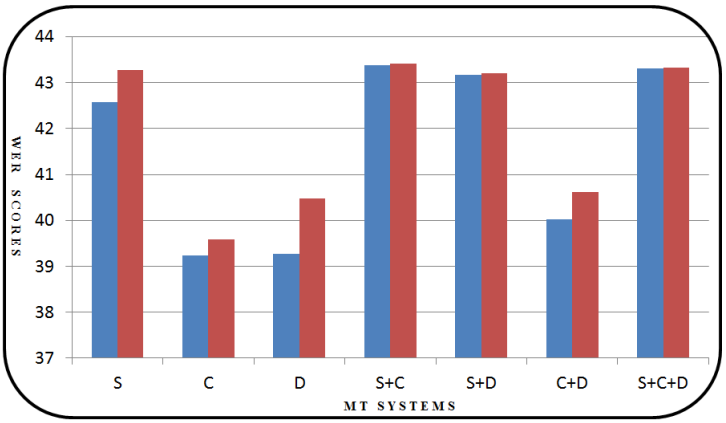


Figure 2.10: Bar graph to show that adding PERC chunks (red bar) to any system (blue bar) generally boosts the WER score (displayed as accuracy scores for uniformity with other metrics): Europarl data. These systems are also reviewed in Table 2.3.

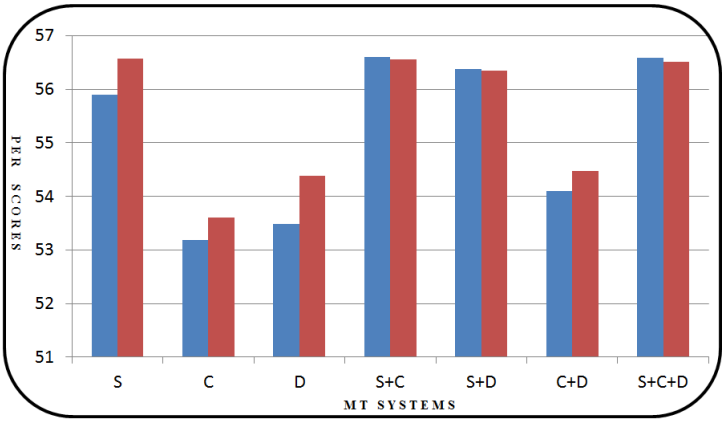


Figure 2.11: Bar graph to show that adding PERC chunks (red bar) to any system (blue bar) generally boosts the PER score (displayed as accuracy scores for uniformity with other metrics): Europarl data. These systems are also reviewed in Table 2.3.

French–English: a small (JOC) and a larger (Europarl) dataset. Working on the JOC corpus allowed us to directly compare our novel phrase induction method against the work of Hearne et al. (2008). While we could not improve upon their results for the JOC corpus, running experiments on the far larger Europarl data set showed clear performance gains over their method (dependencies using a parser) when the PERC phrases were utilised. In any case, our method would still be useful in language pairs for which no separate dependency parser was available.

It was also discovered through automatic evaluation measures that the S+C system gave the best performance. However, lack of statistical significance in the results and manual evaluation leads us to believe that PERC is useful enough to warrant further investigation. Therefore, percolated dependencies appear to be a useful knowledge source for PB-SMT.

2.7 Summary

In this chapter we observed that our syntax-aware models do not show consistent performance in automatic evaluation. However, a qualitative analysis leads us to believe that good phrase pairs fail to be selected by the decoder as the optimum translation.

After carrying out experiments on several additional configurations of the decoder, system combination, and the reordering models, we conclude that we have exhausted most approaches to utilize syntax-aware phrases. Instead, we need to conduct a diagnostic analysis of the PB-SMT decoding pipeline to trace how the syntax-aware phrases are picked up or dropped by the decoder. We put forth a new hypothesis: the PB-SMT modeling-optimization-decoding pipeline is flawed. This leads to our next area of research: oracle-based training covered in Chapter 3.

Chapter 3

Oracle-based System Diagnosis

A Statistical Machine Translation (SMT) decoder generates an *n*-best list of candidate translations for each sentence. The decoding task for the PB-SMT models considered in this thesis has been shown to be NP-hard (Knight, 1999). This exponential complexity of the search space implies that the decoder performs a non-exhaustive search such as the heuristic beam search[1] to find the best possible translation for a given input leading to a number of system errors namely search errors, model errors, and induction errors (Germann et al., 2004; Auli et al., 2009). **Model errors** occur when the highest-scoring translation according to the model (1-best) is not the most accurate translation to be generated as measured by its similarity to the human reference translation (an oracle).

In the last chapter, we concluded that the translation models composed from multiple knowledge sources (STR, CON, DEP, PERC) give sub-optimal performance. For example, it was observed that the combined system S+C+D+P scored 26.60 BLEU points as opposed to 28.77 BLEU points scored by a system that merely selects the best translation from amongst individual outputs of S, C, D, and P systems (cf. Section 2.5.8). This leads us to believe that the decoder fails to select more accurate phrase pairs in the construction of the optimum translation, as a result of the model errors.

Accordingly, in this chapter we analyse these model errors, investigate the parametric differences between the 1-best and the oracle translation and attempt to try and close

[1] It is true that even if beam search is not used, there may still be system errors.

this gap by proposing two rescoring strategies to push the oracle up the *n*-best list. We conduct a range of evaluations across several dimensions: *n*-best list sizes, language directions, and evaluation metrics. We observe modest improvements over the baseline SMT system trained on Europarl corpora (Koehn, 2005). We present a detailed analysis of the oracle rankings to determine the source of model errors, which in turn has the potential to improve the performance of the baseline (STR) system as well as syntax-aware models.

After an introduction to model errors in Section 3.1 and a brief overview of related approaches in Section 3.2, we present in Section 3.3 the baseline SMT system, corpora, and related tools used in all our experiments along with an overview of the parameters or features employed in the baseline system. We then describe in Section 3.4 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 3.5 outlines our algorithm on modifying the feature weights to help push the oracles up the *n*-best lists followed by detailed system-level evaluation in Section 3.6. In Sections 3.7 through 3.10, we report on additional analytical experiments followed by a contrastive analysis across all language directions and *n*-best list sizes in Section 3.11. 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 explored in the following chapter. Figure 3.1 is a reproduction of Figure 1.4 showing a schematic diagram of all the modules of a PB-SMT system. It highlights the tuning module (parameter estimation) as the main research focus of this chapter.

3.1 Model Errors in PB-SMT

Phrase-based SMT (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

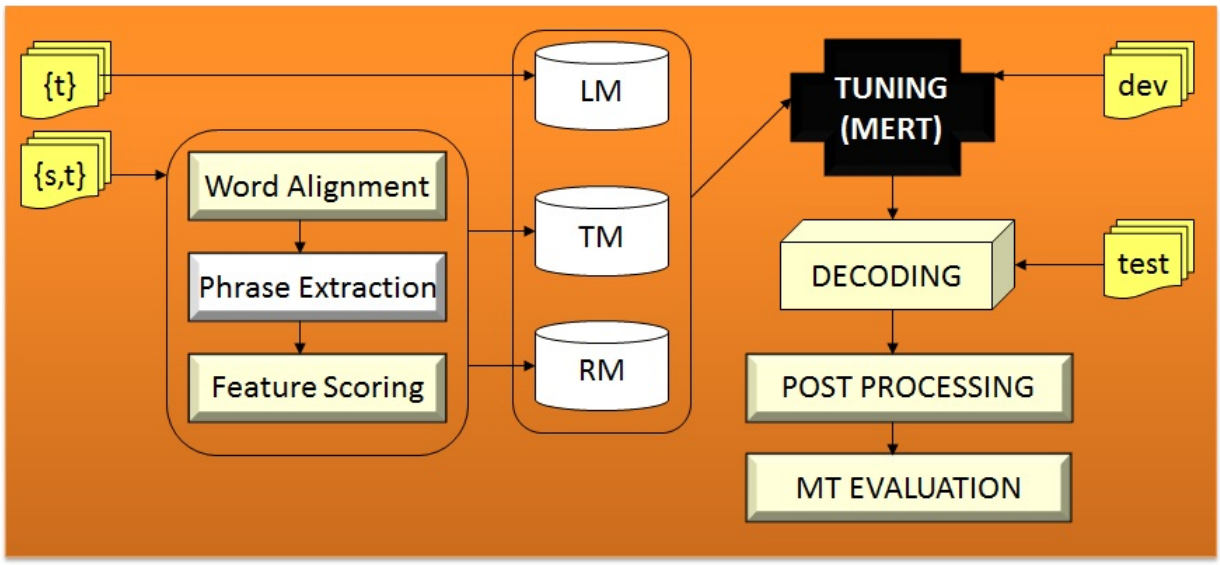


Figure 3.1: Schematic diagram of the modules in a Phrase-based Statistical Machine Translation System: *Tuning* or Parameter Estimation.

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 of Och and Ney (2002) in (3.1):

$$score(e|f) = \sum_{i=1}^{M} \lambda_i h_i(e, f) \tag{3.1}$$

The variable h denotes each of the M features (probabilities of target language phrases (e) given source language phrases (f) learned from language models, translation models, etc.) and λ 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 (Hasan et al., 2007) 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. This is demonstrated in Figure 3.2. Model errors (Germann et al., 2004) occur when the optimum translation (1-best) is not equivalent to the most accurate translation (oracle). This can be formulated as Equation 3.2. The aim of this part of the thesis is to investigate these model errors by quantifying the differences between the 1-best and the oracle translations, and to evaluate the impact of the features used in decoding (following parameter estimation via MERT) on the positioning of oracles in the *n*-best list.

$$\text{MODEL ERROR} = \begin{cases} 0 & : \quad \text{rank}_{oracle} = 1 \\ 1 & : \quad \text{rank}_{oracle} \neq 1 \end{cases} \tag{3.2}$$

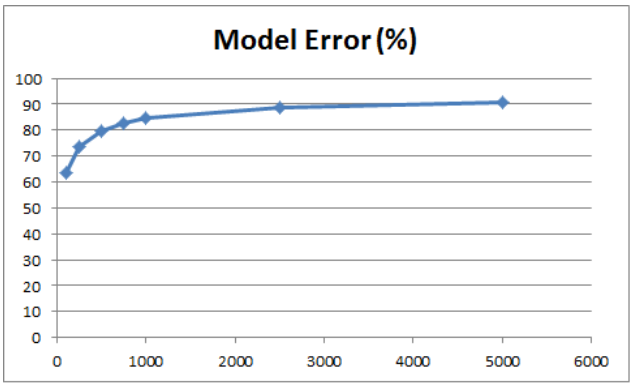


Figure 3.2: Number of model errors (as a percentage) with varying *n*-best list sizes for the devset of French→English WMT 2009 system

.

3.2 Approaches to Minimizing Model Errors

One way 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 at the 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 chapter 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 a 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.

3.3 Baseline System: Data & Tools

CATEGORY	TRAIN	DEV	TEST
DATASET	Europarlv3	test2006	test2008
SENTENCES	1,050,398	2,000	2,000

Table 3.1: Statistics of French→English corpus used in oracle-based training experiments

The set of parallel sentences for all our experiments is extracted from the WMT 2009[2] 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 also experiment on 3 additional language pairs in order to validate the analyses of our oracle-based system diagnosis: English→French, Spanish→English, and German→English. The training sentences for all three language pairs are extracted from the same WMT 2009 dataset as that used in the French→English language pair. Note that the devset and testset remain the same.

Note that French→English dataset is our primary translation pair throughout our thesis.[3] The other translation pairs are used for the purpose of vetting our rescoring strategies and evaluating whether the language pair has any bearing on the failure or success of our methods.

CATEGORY	TRAIN	DEV	TEST
DATASET	Europarlv3	test2006	test2008
SENTENCES	1,050,398	2,000	2,000

Table 3.2: Statistics of English→French corpus used in oracle-based training experiments

CATEGORY	TRAIN	DEV	TEST
DATASET	Europarlv3	test2006	test2008
SENTENCES	1,083,773	2,000	2,000

Table 3.3: Statistics of Spanish→English corpus used in oracle-based training experiments

CATEGORY	TRAIN	DEV	TEST
DATASET	Europarlv3	test2006	test2008
SENTENCES	1,118,399	2,000	2,000

Table 3.4: Statistics of German→English corpus used in oracle-based training experiments

We train a 5-gram language model using SRILM (Stolcke, 2002)[4] with Kneser-Ney

[2] `http://www.statmt.org/wmt09/`

[3] In Chapter 2, we only used a subset (100,000 sentence pairs) of the French→English Europarl dataset.

[4] `http://www-speech.sri.com/projects/srilm/`

smoothing (Kneser and Ney, 1995). We train the translation model using GIZA++ (Och and Ney, 2003)[5] for word alignment in both directions followed by phrase-pair extraction using the 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 the previous and next phrases.

3.3.1 Baseline Features

LABEL	DESCRIPTION
d1	Distortion: distance-based reordering
d2	Distortion: monotone previous
d3	Distortion: swap previous
d4	Distortion: discontinuous previous
d5	Distortion: monotone following
d6	Distortion: swap following
d7	Distortion: discontinuous following
lm	Language Model feature
w	Word penalty feature
tm1	Translation: Phrase Translation (s | t)
tm2	Translation: Lexical Weighting (s | t)
tm3	Translation: Phrase Translation (t | s)
tm4	Translation: Lexical Weighting (t | s)
tm5	Translation: Phrase penalty feature

Table 3.5: Features used in the Moses PB-SMT Decoder

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 (see Table 3.5), 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 feature which penalises for any source-language word absent from the phrase table applies uniformly to all the candidate translations of a sentence, and is therefore dropped from consideration in our experiments.

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

3.4 Oracle-based Training

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.

3.4.1 *N*-best Lists and Oracles

The oracle sentence is selected by picking the candidate translation from an *n*-best list which is 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 (3.3).

$$\textsc{Sbleu} = \sum_{i=1}^{4} \frac{\textsc{Bleu}_i(cand, ref)}{2^{4-i+1}} \tag{3.3}$$

Figure 3.3 shows a sample of 10 candidate English translations from an *n*-best list for a French sentence. The first column gives the relative rank number of each entry corresponding to its decoder cost. The second column gives the respective decoder cost (log-linear score) used to rank an *n*-best list and the fourth 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 sBLEU score.

Rank	Decoder Cost	Sentence	sBLEU
1	-5.32	is there not here two weights , two measures ?	0.0188
2	-5.50	is there not here double standards ?	0.147
3	-5.66	are there not here two weights , two measures ?	0.0125
4	-6.06	is there not double here ?	0.025
5	-6.15	is there not here double ?	0.025
6	-6.17	is it not here two sets of standards ?	0.0677
7	**-6.28**	**is there not a case of double standards here ?**	**0.563**
8	-6.37	is there not here two weights and two yardsticks ?	0.0188
9	-6.38	is there no double here ?	0.0190
10	-6.82	is there not here a case of double standards ?	0.563

Figure 3.3: 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?*

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-ranked sentences have the same sBLEU score). Such issues are discussed and dealt with in Section 3.9. 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.

SYSTEM	BLEU	NIST	METEOR	WER	PER
BASELINE	32.17	7.70	61.34	57.10	40.96
ORACLE$_{B100}$	34.90	8.08	63.65	54.78	38.52
ORACLE$_{M100}$	34.32	8.02	63.63	55.13	38.88
ORACLE$_{B250}$	35.75	8.19	64.22	54.09	37.93
ORACLE$_{M250}$	34.99	8.11	64.20	54.56	38.37
ORACLE$_{B500}$	36.45	8.28	64.70	53.63	37.44
ORACLE$_{M500}$	35.57	8.19	64.74	54.11	37.87
ORACLE$_{B750}$	36.80	8.32	64.95	53.32	37.17
ORACLE$_{M750}$	35.81	8.22	65.01	53.88	37.62
ORACLE$_{B1000}$	37.05	8.35	65.14	53.08	36.97
ORACLE$_{M1000}$	36.01	8.25	65.18	53.69	37.49
ORACLE$_{B2500}$	37.97	8.47	65.83	52.38	36.31
ORACLE$_{M2500}$	36.73	8.34	65.89	53.12	36.84
ORACLE$_{B5000}$	38.75	8.56	66.32	51.84	35.81
ORACLE$_{M5000}$	37.19	8.41	66.36	52.71	36.46

Table 3.6: Summary of the French→English oracle-best systems for 100-best, 250-best, 500-best, 750-best, 1000-best, 2500-best, and 5000-best lists: devset

3.4.2 Recalculating Lambdas

In contrast to the main reranking approaches in the literature 3.2, this work analyzes the 14 remaining **baseline features** (outlined in Table 3.5) 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.

$$\lambda_{new} = \lambda_{old} + (\bar{f}_{oracle} - \bar{f}_{1best}) \tag{3.4}$$

- 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.

$$\lambda_{new} = \lambda_{old} * \frac{\bar{f}_{oracle}}{\bar{f}_{1best}} \tag{3.5}$$

Both formulae aim to close the gap between the 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 3.5.

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).

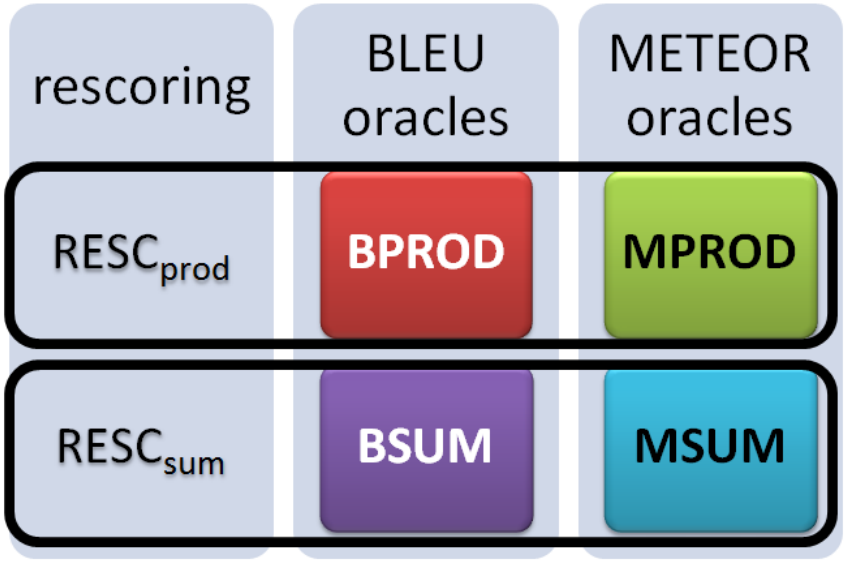


Figure 3.4: Four types of rescoring strategies used to push oracles up the *n*-best lists.

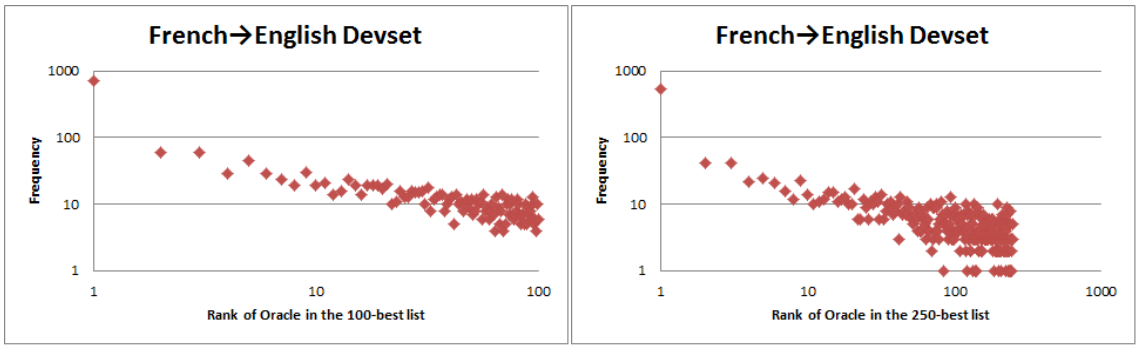
.

Figure 3.4 illustrates the resulting four types of rescoring systems (RESCORED$_{BPROD}$, RESCORED$_{MPROD}$, RESCORED$_{BSUM}$, and RESCORED$_{MSUM}$) arising out of the two rescoring strategies (RESC$_{prod}$ and RESC$_{sum}$) and two types of oracles (BLEU-oracles and METEOR-oracles).

3.5 Experimental Design

Our analyses of the differences between the 1-best and the oracle translations follows. We perform all our diagnostic experiments on 7 different *n*-best list sizes across four language directions. Before embarking on the evaluation in Section 3.6, we briefly outline our experimental scheme below followed by a discussion on the distribution of oracles in a *n*-best list to further elucidate the task at hand.

We extract the 14 baseline features for sentences from the devset of 2000 sentences using the WMT test2006 dataset. For each of these sentences, we compare the 1-best



(a) 100-best **(b)** 250-best

Figure 3.5: Plotting oracle rank (logarithmic scale) against frequency (logarithmic scale) for *n*-best list on the devset of French→English WMT 2009 systems.

and oracle-best features and compute the mean value per feature. This is then used to compute two new sets of weights using the RESC$_{sum}$ and RESC$_{prod}$ rescoring strategies from Equations (3.4) and (3.5), respectively. We implemented our rescoring strategies on the devset and then applied the two new sets of weights computed on the testset of *n*-bests.

3.5.1 Distribution of Oracles

The research in this chapter focusses on methods for moving the oracle translations up the *n*-best list. Before proceeding with our rescoring experiments, it is therefore important to determine how the oracle translations are distributed for the baseline system, i.e. the position of the oracle candidate in the *n*-best list for each sentence. Table 3.7 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. Figure 3.5 gives a graphical representation of the number of oracle candidates at each position or rank in an *n*-best list of French → English devsets for (a) 100-best lists and (b) 250-best lists.

It is evident that with increasing *n*-best list size, the number of oracles in the top ranks decreases. The oracle distribution changes with increasing *n*-best list sizes because when the size of the list is expanded, better translations are generated which are positioned lower in the *n*-best list. This is alarming as this increases the complexity of our problem with increasing *n*-best list sizes. One possible method to contain this is to filter the oracles, i.e. only use those sentences whose oracles are different enough from its peers. This

is described in Section 3.9 as a proof-of-concept experiment. We will see from oracle filtering that not all sentences have a good quality oracle. The filtering will balance the tendency of high-ranking translations to be placed lower in the list.

	(a) DEVSET			(b) TESTSET		
RANGE	**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 3.7: Number of times an oracle occurs in a particular range of ranks in the *n*-best lists of (a) DEVSET and (b) TESTSET

3.6 System-level Evaluation

In this section, we report on performance of the baseline system against four rescored PB-SMT systems for four language pairs by pushing oracles up seven different *n*-best list sizes using seven MT system evaluation scores.

Specifically, we report our rescoring results on four sets of PB-SMT systems using four different language directions (each occupying its own sub-section): (1) French $\rightarrow$ English, (2) German $\rightarrow$ English, (3) Spanish $\rightarrow$ English, and (4) English $\rightarrow$ French.

The *n*-best lists for each PB-SMT system refer to the maximum number of translation hypotheses generated by translating each source-language sentence. We have evaluated using seven such sizes of *n*: 100-best, 250-best, 500-best, 750-best, 1000-best, 2500-best, and 5000-best. The MT system evaluation scores for each of the five PB-SMT systems (itemized in the next paragraph) are presented in seven tables (one for each *n*-best list size). Each table is further divided into two sections: (a) devset: Development data on which the lambdas are rescored and new set of weights are computed, and (b) testset: Blind dataset used to test the effectiveness of the new set of weights.

We evaluate by comparing a baseline system against two implementations of each of the two rescoring strategies RESC$_{sum}$ RESC$_{prod}$. Hence there are five separate systems tested, accounting for five rows of values under each dataset:

- BASELINE: System using weights computed using MERT with no rescoring
- RESCORED$_{BPROD}$: System in which the MERT weights are recalculated as per Equation (3.5) based on Oracles with respect to sentence-level BLEU score
- RESCORED$_{MPROD}$: System in which the MERT weights are recalculated as per Equation (3.5) based on Oracles with respect to sentence-level METEOR score
- RESCORED$_{BSUM}$: System in which the MERT weights are recalculated as per Equation (3.4) based on Oracles with respect to sentence-level BLEU score
- RESCORED$_{MSUM}$: System in which the MERT weights are recalculated as per Equation (3.4) based on Oracles with respect to sentence-level METEOR score

In each table, each of the two sections (devset and testset) reports evaluation results using 7 system-level evaluation metrics accounting for the seven columns of values in each table. The first five are standard MT system evaluation metrics[6] used in all experiments throughout this thesis. The remaining two measure the number of oracle translations positioned at the top ranks.

- BLEU (Papineni et al., 2002): The values are presented as a percentage with higher values implying higher accuracy
- NIST (Doddington, 2002): The values are presented on a scale of 10 with higher values implying higher accuracy
- METEOR (Banerjee and Lavie, 2005): The values are presented as a percentage with higher values implying higher accuracy
- WER (Word Error Rate; (Niessen et al., 2000)): The error rates are presented as a percentage with lower values implying higher accuracy
- PER (Position-independent WER; (Leusch et al., 2003)): The error rates are presented as a percentage with lower values implying higher accuracy

[6] For a broader introduction to the MT evaluation metrics like BLEU, NIST, METEOR, WER, PER see Tinsley (2010) : 37–43 and Owczarzak (2008) : 14–22.

- OBLEU: % sentences containing the oracle translations at rank 1 (Oracles are identified with respect to sentence-level BLEU score)

- OMET: % sentences containing the oracle translations at rank 1 (Oracles are identified with respect to sentence-level METEOR score)

3.6.1 French to English

Herein, we report on rescoring *n*-best lists when translating the test2006 (devset) and test2008 (testset) datasets from French into English. We evaluate the performance of our RESC$_{prod}$ and RESC$_{sum}$ rescoring strategies by pitting the translation system scores of the four rescoring systems (RESCORED$_{BPROD}$, RESCORED$_{MPROD}$, RESCORED$_{BSUM}$, and RESCORED$_{MSUM}$) against the BASELINE system. We also make observations on whether the RESC$_{prod}$ yields better results than RESC$_{sum}$ or vice versa. Another comparison criterion is whether the BLEU-oracles are more effective in rescoring than METEOR-oracles or vice versa. These results are tested using seven evaluation metrics (BLEU, NIST, METEOR, WER, PER, OBLEU, OMET) on seven *n*-best lists of increasing sizes from 100-best to 5000-best. We will end this section with a summary by commenting on the general trend, if any, seen in all seven *n*-best lists. The motivation for experimenting on a range of *n*-best list sizes is that there is no consensus in literature on the optimal size of the *n*-best list. We analyse this across the language pairs in Table 3.48 in Section 3.11.

100-BEST LIST

Table 3.8 (a) gives system evaluation scores on oracle-based rescoring of 100-best lists for the French–English devset. The BASELINE system outperforms all four rescored systems on BLEU (with a *statistically significant*[7] score of 32.17) and METEOR (with a *statistically significant* score of 61.34) scores. The BASELINE system in turn is outperformed by all the four rescored systems on NIST, WER, and PER. The RESCORED$_{BSUM}$ system demonstrates the lowest (i.e. best) WER (absolute difference of 0.16 over base-

[7] All statistical significance tests were performed using bootstrap resampling described in Koehn (2004). The p-values used were 0.05, i.e. the scores are significantly different with a 95% confidence interval.

line) and PER (absolute difference of 0.09 over baseline). Additionally all four rescored systems outperform the BASELINE system as far as the percentage of oracles (with respect to BLEU and METEOR) in rank 1, i.e. OBLEU and OMET scores are concerned.

The PROD rescoring (2nd and 3rd rows in Table 3.8 (a)) yields better translation results than the SUM rescoring with respect to BLEU, OMET, and OBLEU scores while the SUM rescoring (4th and 5th rows in Table 3.8 (a)) beats the PROD rescoring on all other metrics, namely NIST, METEOR, WER, and PER scores. When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.8 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.8 (a)), both sets of PROD rescoring systems (RESCORED$_{BPROD}$ Vs RESCORED$_{MPROD}$) and SUM rescoring systems (RESCORED$_{BSUM}$ Vs RESCORED$_{MSUM}$) give similar performance across all metrics.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.17**	7.70	**61.34**	57.10	40.96	36.25	36.90
RESCORED$_{BPROD}$	31.94	7.71	61.04	57.04	40.94	37.05	38.90
RESCORED$_{MPROD}$	31.94	7.71	61.04	57.04	40.94	**37.10**	**38.95**
RESCORED$_{BSUM}$	31.90	**7.72**	61.11	**56.94**	**40.87**	36.65	37.75
RESCORED$_{MSUM}$	31.89	7.71	61.10	56.96	40.88	36.30	37.30
(b) testset							
BASELINE	**32.47**	7.81	**61.80**	56.43	40.34	36.25	37.30
RESCORED$_{BPROD}$	32.20	7.81	61.50	56.38	40.37	37.50	39.80
RESCORED$_{MPROD}$	32.21	7.81	61.51	56.37	40.36	37.50	**39.85**
RESCORED$_{BSUM}$	32.33	**7.82**	61.61	**56.21**	**40.26**	**37.85**	38.70
RESCORED$_{MSUM}$	32.31	**7.82**	61.58	56.24	40.28	37.40	37.95

Table 3.8: Summary of the French→English translation system results for 100-best list: (a) devset and (b) testset

Table 3.8 (b) gives translation results on oracle-based rescoring of 100-best lists for the French–English testset. Again, as on the devset, the BASELINE system is outperformed by one or more of the four rescored systems across all evaluation metrics except the BLEU and METEOR scores. The RESCORED$_{BSUM}$ scores the lowest (i.e. best) WER and PER scores (e.g. 56.21 WER points *statistically significant*) with an absolute difference of 0.22 or 0.39% over the BASELINE system).

All four rescoring systems record ranking 1% to 2% more oracles in the first position

than the BASELINE system, with RESCORED$_{MPROD}$ and textscRescored$_{BSUM}$ scoring the highest OMET and OBLEU scores, respectively.

Unlike the devset, the SUM rescoring systems (4th and 5th rows in Table 3.8 (b)) perform slightly better than the PROD rescoring systems (2nd and 3rd rows in Table 3.8 (b)) across all five system evaluation metrics: BLEU, NIST, METEOR, WER, and PER.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.8 (b)) with METEOR-oracles (rows 3 and 5 in Table 3.8 (b)), the PROD rescoring gives similar performance (e.g. 61.50 METEOR points in RESCORED$_{BPROD}$ against 61.51 METEOR points in RESCORED$_{MPROD}$) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{BSUM}$ as slightly better than RESCORED$_{MSUM}$ (e.g. 61.61 METEOR points in RESCORED$_{BSUM}$ against 61.58 METEOR points in RESCORED$_{MSUM}$). Note that this observation is unlike that seen on the devset in Table 3.17 (a).

Overall, the RESCORED$_{BSUM}$ system gives the best performance amongst all rescoring systems, especially on the testset. As stated above, this system outperforms the BASELINE on all metrics except BLEU and METEOR on both the devset and the testset.

CATEGORY	DEVSET	TESTSET
BELU	0.84% ↓	0.43% ↓
METEOR	0.38% ↓	0.30% ↓

Table 3.9: Margin of difference in the BLEU and METEOR performance scores of RESCORED$_{BSUM}$ and BASELINE: French–English 100-best list

An important observation is that although BLEU and METEOR scores favour the BASELINE system, the difference in these scores between the baseline and RESCORED$_{BSUM}$ is reduced when moving from devset to testset. Table 3.9 summarises this: the values in the testset column are lower than the corresponding values in the devset column. This lends credence to our claim that our rescoring strategies have a good learnability.

250-BEST LIST

Note that from here onwards, the BASELINE system remains unchanged with regards to its translation output and therefore gives the same results on BLEU, NIST, METEOR,

WER, and PER scores. The only difference is in the size of the *n*-best list which in turn is reflected in the number of sentences ranking the oracle translation in the top position resulting in OBLEU and OMET scores different from the BASELINE system in 100-best list. In fact, all systems show a decrease in their OBLEU and OMET scores implying a decrease in the number of oracles in the top ranks with increasing *n*-best list sizes.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.17**	7.70	**61.34**	57.10	40.96	26.30	26.85
RESCORED$_{BPROD}$	31.84	7.70	60.88	57.04	41.04	**28.25**	**28.85**
RESCORED$_{MPROD}$	31.84	7.70	60.88	57.04	41.04	**28.25**	**28.85**
RESCORED$_{BSUM}$	31.88	**7.72**	61.09	**56.89**	**40.81**	26.50	28.50
RESCORED$_{MSUM}$	31.88	**7.72**	61.07	**56.89**	40.83	26.45	28.25
(b) testset							
BASELINE	**32.47**	7.81	**61.80**	56.43	40.34	26.80	28.10
RESCORED$_{BPROD}$	32.07	7.80	61.38	56.46	40.44	27.60	30.20
RESCORED$_{MPROD}$	32.07	7.80	61.39	56.46	40.44	27.65	**30.25**
RESCORED$_{BSUM}$	32.31	**7.83**	61.56	**56.17**	**40.26**	**28.70**	29.80
RESCORED$_{MSUM}$	32.28	**7.83**	61.54	**56.17**	40.27	28.35	29.55

Table 3.10: Summary of the French→English translation system results for 250-best list: (a) devset and (b) testset

Table 3.10 (a) gives system evaluation scores on oracle-based rescoring of 250-best lists for French–English devset. Just like in the 100-best list, the BASELINE system outperforms all four rescored systems on BLEU (with a *statistically significant* score of 32.17) and METEOR (with a *statistically significant* score of 61.34) scores. However, one or more of our rescoring systems performed well on the remaining five metrics:

- NIST: RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ perform slightly better than the BASELINE with an absolute difference of 0.02 points
- WER: RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ perform significantly better than the BASELINE with an absolute difference of 0.21 points. Both the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems outperform the baseline as well.
- PER: RESCORED$_{BSUM}$ performs significantly better than the BASELINE with an absolute difference of 0.15 points

- OBLEU: RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ have the highest percentage of BLEU-oracles in the 1-best position (39 more sentences than the baseline)

- OMET: RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ have the highest percentage of METEOR-oracles in the 1-best position (34 more sentences than the baseline).

Note that on comparing the BLEU and METEOR scores of all the four rescoring systems on 100-best list with their corresponding scores on 250-best list, we find that the scores decrease. On the other hand, the WER and PER scores improve while the NIST scores remain the same for the most part. All other observations were similar to that of 100-best lists described previously.

The results for rescoring 250-best outputs of testset appear in Table 3.10 (b). Like on the devset, the BASELINE system gives the best scores on BLEU and METEOR, while the RESCORED$_{BSUM}$ system beats the BASELINE system as well as giving the best scores on all the remaining metrics: NIST (absolute difference of 0.02), WER (absolute difference of 0.26 (*statistically significant*)), and PER (absolute difference of 0.18 (*statistically significant*)).

As observed in the 100-best list, although BLEU and METEOR scores favour the BASELINE system, the difference in these scores between the baseline and RESCORED$_{BSUM}$ is reduced when moving from devset to testset. All other observations followed the same pattern as the 100-best list as well. An analysis of the metric score trend with increasing *n*-best list size is made after Table 3.15.

500-BEST LIST

In Tables 3.11 (a) and (b), we report on rescoring 500-best lists for devset and testset, respectively. It can be seen that the BASELINE system gives the best scores on BLEU and METEOR. However, just like on 100-best and 250-best lists, RESCORED$_{BSUM}$ system beats the BASELINE system as well as gives the best scores for devset on all the remaining metrics: NIST (absolute difference of 0.03), WER (absolute difference of 0.13 (*statistically significant*)), and PER (absolute difference of 0.14 (*statistically significant*)). For the testset as well, the RESCORED$_{BSUM}$ system beats the BASELINE system on the

following metrics: NIST (absolute difference of 0.03), WER (absolute difference of 0.33 (*statistically significant*)), and PER (absolute difference of 0.10 (*statistically significant*)).

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.17**	7.70	**61.34**	57.10	40.96	20.10	20.25
RESCORED$_{BPROD}$	31.72	7.70	60.80	57.10	41.05	21.75	22.40
RESCORED$_{MPROD}$	31.72	7.70	60.81	57.09	41.04	**21.85**	**22.55**
RESCORED$_{BSUM}$	31.87	**7.73**	61.04	**56.87**	**40.82**	20.25	21.60
RESCORED$_{MSUM}$	31.84	**7.73**	61.03	**56.87**	40.83	20.25	21.50
(b) testset							
BASELINE	**32.47**	7.81	**61.80**	56.43	40.34	20.75	21.75
RESCORED$_{BPROD}$	31.95	7.78	61.18	56.57	40.58	21.15	23.65
RESCORED$_{MPROD}$	31.95	7.78	61.18	56.56	40.57	20.90	23.45
RESCORED$_{BSUM}$	32.33	**7.84**	61.57	**56.10**	**40.24**	**22.80**	**23.75**
RESCORED$_{MSUM}$	32.27	7.83	61.54	56.13	40.27	21.95	23.35

Table 3.11: Summary of the French→English translation system results for 500-best list: (a) devset and (b) testset

As far as the percentage of oracles in the 1-best position is concerned, RESCORED$_{MPROD}$ scores the highest number in both OBLEU (32 more sentences than RESCORED$_{BSUM}$) and OMET (19 more sentences than RESCORED$_{BSUM}$) on the devset. This implies that even though a significant number of SUM system segments match with the reference translation segments or significantly fewer edits needed to be taken between reference and SUM system sentences, as per the system-level NIST, WER, PER scores, a majority of them are not oracles (resulting in low OBLEU and OMET scores). More light will be shed on this in the analysis section after describing the results of 5000-best lists. On the other hand, RESCORED$_{BSUM}$ achieves the highest OBLEU and OMET scores on testset.

Once again, our rescoring systems seem to fare better on the testset than on the devset,[8] as observed in 100-best and 250-best lists.

750-BEST LIST

Table 3.12 (a) gives system evaluation scores on oracle-based rescoring of 750-best lists for French–English devset. Unlike the preceding *n*-best lists of French→English,

[8] This behaviour is quite standard in SMT.

the best-performing system according to METEOR is RESCORED$_{MSUM}$ with a *statistically significant* value of 61.64 points (0.5% improvement over the baseline). Note that the RESCORED$_{MSUM}$ was also found to give the worst performance on all other metrics. A manual analysis on why the METEOR scoring metric alone seems to favour RESCORED$_{BSUM}$ system will be covered towards the end of this Section 3.6.1. The PROD systems perform on a par with the BASELINE on NIST and outperform the BASELINE on OBLEU and OMET. However, the highest percentage of METEOR-oracles in 1-best position (OMET) scores is achieved by RESCORED$_{MSUM}$.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.17**	**7.70**	61.34	**57.10**	**40.96**	17.35	18.05
RESCORED$_{BPROD}$	32.12	**7.70**	61.33	57.12	40.98	**17.50**	17.95
RESCORED$_{MPROD}$	32.11	**7.70**	61.33	57.12	40.98	17.45	18.10
RESCORED$_{BSUM}$	31.56	7.58	61.51	58.07	41.50	17.35	**18.30**
RESCORED$_{MSUM}$	31.52	7.57	**61.64**	58.21	41.58	17.15	18.25
(b) testset							
BASELINE	**32.47**	**7.81**	61.80	56.43	**40.34**	**18.05**	**18.70**
RESCORED$_{BPROD}$	32.46	7.80	61.76	56.41	40.41	17.90	18.60
RESCORED$_{MPROD}$	**32.47**	7.80	61.76	**56.40**	40.40	17.90	18.60
RESCORED$_{BSUM}$	31.93	7.67	61.86	57.28	40.99	17.45	17.40
RESCORED$_{MSUM}$	31.89	7.66	**61.95**	57.47	41.12	17.60	17.70

Table 3.12: Summary of the French→English translation system results for 750-best list: (a) devset and (b) testset

The PROD rescoring system (2nd and 3rd rows in Table 3.12 (a)) yields significantly better translation results than the SUM rescoring (4th and 5th rows in Table 3.12 (a)) across all metrics except METEOR scores.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.12 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.12 (a)), the PROD rescoring gives similar performance (e.g. 61.33 METEOR points in both RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$), while the SUM rescoring approach shows a larger degree of variation with the RESC$_{BSUM}$ as significantly better than RESCORED$_{MSUM}$ on all metrics except the METEOR scores.

The results for rescoring 750-best outputs of testset appear in Table 3.12 (b). The rescoring system with the lowest evaluation scores is RESCORED$_{MSUM}$ as per all metrics

except the METEOR score. As observed in the devset, the best-performing system is also RESCORED$_{MSUM}$ as scored by METEOR (a 0.24% improvement over the baseline). In contrast to the devset, the RESCORED$_{MPROD}$ and RESCORED$_{BPROD}$ systems show slight improvements or give on par performance with the baseline system on BLEU, NIST, and WER.

All other observations were similar to that seen on the devset. Moreover, when comparing performance of the rescoring systems on devset versus testset, we find that the testset again gives better performance over the BASELINE system. For example, the RESCORED$_{BPROD}$ records a 0.03 WER points improvement over the BASELINE system on the testset as opposed to an underperforming 0.02 WER points on the devset.

Note that in contrast to preceding *n*-best lists of French→English, the PROD systems perform better than the SUM systems on all scores except METEOR. Whether this behaviour is an anomaly or will be observed in subsequent *n*-best lists remains to be seen.

1000-BEST LIST

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.17**	**7.70**	61.34	**57.10**	**40.96**	15.40	16.20
RESCORED$_{BPROD}$	32.11	**7.70**	61.32	57.11	40.98	15.55	16.10
RESCORED$_{MPROD}$	32.10	**7.70**	61.33	57.11	40.97	**15.60**	16.05
RESCORED$_{BSUM}$	31.55	7.58	61.57	58.16	41.57	**15.60**	**16.35**
RESCORED$_{MSUM}$	31.48	7.56	**61.65**	58.33	41.72	15.10	15.95
(b) testset							
BASELINE	32.47	**7.81**	61.80	56.43	**40.34**	16.20	16.25
RESCORED$_{BPROD}$	**32.48**	7.80	61.77	**56.40**	40.40	16.20	16.30
RESCORED$_{MPROD}$	32.44	7.80	61.74	56.42	40.42	16.20	**16.35**
RESCORED$_{BSUM}$	31.84	7.65	61.86	57.39	41.08	15.90	15.30
RESCORED$_{MSUM}$	31.88	7.65	**62.01**	57.50	41.17	**16.35**	15.40

Table 3.13: Summary of the French→English translation system results for 1000-best list: (a) devset and (b) testset

Table 3.13 (a) gives system evaluation scores on oracle-based rescoring of 1000-best lists for French–English devset. Just like the preceding *n*-best lists, BASELINE achieves the best score on BLEU, NIST, WER, and PER. However, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems perform as well as BASELINE on NIST, METEOR, WER,

and PER. Similar to 750-best list, the best-performing system on METEOR is RESC$_{MSUM}$ with a *statistically significant* value of 61.65 points (0.5% relative above baseline). With 61.57 METEOR points, RESCORED$_{BSUM}$ performs 0.37% relative above baseline. The highest OBLEU and OMET scores were achieved by RESCORED$_{BSUM}$.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.13 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.13 (a)), the SUM rescoring systems show a larger degree of mutual variation on all metrics: BLEU (favouring BLEU-oracles), NIST (favouring BLEU-oracles), METEOR (favouring METEOR-oracles), WER (favouring BLEU-oracles), and PER (favouring BLEU-oracles).

The results for rescoring 1000-best outputs of testset appear in Table 3.13 (b). As observed on the devset, the best-performing system is RESCORED$_{MSUM}$ as scored by METEOR (a 0.34% improvement over the baseline, *statistically significant*). In contrast to the devset, RESCORED$_{BPROD}$ system shows slight improvement / give at par performance with the baseline system on BLEU, NIST, and WER.

All other observations were similar to that seen in the devset. Moreover, when comparing performance of the rescoring systems on devset versus testset, surprisingly, we find that the devset gives better performance over the BASELINE system than on the testset. For example, the RESCORED$_{MSUM}$ records a 0.31 METEOR points improvement over the BASELINE system on the devset as opposed to an improvement of 0.21 METEOR points on the devset. This observations is recorded on 750-best lists as well but not on the other *n*-best lists evaluated so far.

On 750-best lists, we also noted that in contrast to preceding *n*-best lists of French→ English, the PROD systems perform better than the SUM systems on all scores except METEOR. This behaviour is also observed in 1000-best lists. This might indicate that RESC$_{sum}$ unlike RESC$_{prod}$ rescoring strategy improves with increasing *n*-best list sizes.

2500-BEST LIST

Table 3.14 (a) gives system evaluation scores on oracle-based rescoring of 2500-best lists for French–English devset. Just like the preceding *n*-best lists, BASELINE achieves

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.17**	**7.70**	61.34	**57.10**	**40.96**	11.50	**12.70**
RESCORED$_{BPROD}$	32.10	7.70	61.33	57.14	40.98	**11.70**	12.45
RESCORED$_{MPROD}$	32.09	7.70	61.33	57.12	40.97	11.65	12.65
RESCORED$_{BSUM}$	31.40	7.54	61.60	58.33	41.77	11.40	11.60
RESCORED$_{MSUM}$	31.10	7.49	**61.73**	58.94	42.18	11.60	12.05
(b) testset							
BASELINE	**32.47**	**7.81**	61.80	56.43	**40.34**	11.80	11.95
RESCORED$_{BPROD}$	32.44	7.80	61.75	56.44	40.40	11.65	12.00
RESCORED$_{MPROD}$	32.44	7.80	61.75	**56.41**	40.40	11.70	**12.10**
RESCORED$_{BSUM}$	31.69	7.61	61.89	57.63	41.28	11.40	10.75
RESCORED$_{MSUM}$	31.43	7.57	**61.95**	58.06	41.60	**11.95**	11.60

Table 3.14: Summary of the French→English translation system results for 2500-best list: (a) devset and (b) testset

the best score on BLEU, NIST, WER, and PER. However, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems perform as well as BASELINE on NIST, METEOR, and PER. Similar to 1000-best list, the best-performing system on METEOR is RESCORED$_{MSUM}$ with a *statistically significant* value of 61.73 points (0.64% above baseline). With 61.60 METEOR points, RESCORED$_{BSUM}$ performs 0.42% above baseline. Both these scores are better than the corresponding system scores on 1000-best list. Unlike the preceding *n*-best lists, the highest OMET scores were achieved by BASELINE system.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.14 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.14 (a)), the SUM rescoring systems show a larger degree of mutual variation on all metrics: BLEU (favouring BLEU-oracles), NIST (favouring BLEU-oracles), METEOR (favouring METEOR-oracles), WER (favouring BLEU-oracles), and PER (favouring BLEU-oracles). This was seen on 1000-best lists as well.

The results for rescoring 2500-best outputs of testset appear in Table 3.14 (b). As observed on the devset, the best-performing system is RESCORED$_{MSUM}$ as scored by METEOR (a 0.24% improvement over the baseline, *statistically significant*). This score is lower than the corresponding system on 1000-best list (opposite observation on devset). In contrast to the devset, RESCORED$_{MPROD}$ system shows slight improvement over the

baseline system on WER (absolute difference of 0.02 over baseline).

All other observations were similar to that seen in the devset. Moreover, when comparing performance of the rescoring systems on devset versus testset, we record the same observations as on 750-best and 1000-best lists.

On 750-best and 1000-best lists, we also noted that in contrast to preceding *n*-best lists of French→English, the PROD systems perform better than the SUM systems on all scores except METEOR. This behaviour is also observed in 2500-best lists. This lends credence to our claim that unlike RESC$_{prod}$, the RESC$_{sum}$ rescoring strategy improves with increasing *n*-best list sizes.

5000-BEST LIST

It is observed that the performance of rescoring systems on 5000-best list follows a similar pattern to that observed on 100-best, 250-best, and 500-best lists. Unlike the remaining *n*-best lists (750-best, 1000-best, 2500-best), the BASELINE system achieves best scores on BLEU and METEOR scores and is outperformed by one or more of the rescoring systems (usually SUM systems) on all other metrics.

Table 3.15 (a) gives system evaluation scores on oracle-based rescoring of 5000-best lists for French–English devset. The BASELINE system achieves the best score (*statistically significant*) on BLEU (32.17 points) and METEOR (61.34 points). However, just like on 100-best, 250-best, and 500-best lists, RESCORED$_{BSUM}$ system beats the BASELINE system as well as gives the best scores for devset on all the remaining metrics: NIST (absolute difference of 0.5% (*statistically significant*)), WER (absolute difference of 0.67% (*statistically significant*)), and PER (absolute difference of 0.46% (*statistically significant*)). The RESCORED$_{MSUM}$ system beats the baseline too and performs slightly worse than RESCORED$_{BSUM}$. The worst performing systems are RESCORED$_{MPROD}$ and RESCORED$_{BPROD}$ as per the five MT evaluation metrics: BLEU, NIST, METEOR, WER, and PER.

As far as the percentage of oracles in the 1-best position is concerned, RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ obtain the highest scores for both OBLEU (31 more sentences

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.17**	7.70	**61.34**	57.10	40.96	9.10	10.15
RESCORED$_{BPROD}$	31.29	7.67	60.43	57.19	41.22	**10.65**	**12.05**
RESCORED$_{MPROD}$	31.28	7.67	60.43	57.21	41.23	10.60	**12.05**
RESCORED$_{BSUM}$	31.72	**7.74**	60.89	**56.72**	**40.77**	10.40	11.05
RESCORED$_{MSUM}$	31.54	7.73	60.76	56.74	40.79	10.35	11.00
(b) testset							
BASELINE	**32.47**	7.81	**61.80**	56.43	40.34	8.85	9.6
RESCORED$_{BPROD}$	31.69	7.78	60.93	56.51	40.58	**11.85**	11.90
RESCORED$_{MPROD}$	31.72	7.78	60.94	56.51	40.56	11.80	**11.95**
RESCORED$_{BSUM}$	32.22	**7.85**	61.40	**55.99**	**40.16**	10.25	10.50
RESCORED$_{MSUM}$	32.06	**7.85**	61.26	56.00	**40.16**	10.90	11.60

Table 3.15: Summary of the French→English translation system results for 5000-best list: (a) devset and (b) testset

than BASELINE and 5 more sentences than RESCORED$_{BSUM}$) and OMET (38 more sentences than BASELINE and 20 more sentences than RESCORED$_{BSUM}$) on the devset. This implies that even though a significant number of SUM system segments match with the reference translation segments or significantly less edits needed to be taken between reference and SUM system sentences, as per the system-level NIST, WER, PER scores, a majority of them are not oracles (resulting in low OBLEU and OMET scores).

The results for rescoring 5000-best outputs of testset appear in Table 3.15 (b). As observed on the devset, the BASELINE system achieves the best score (*statistically significant*) on BLEU (32.47 points) and METEOR (61.80 points). However, both the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems beat the BASELINE system as well as gives the best scores for devset on all the remaining metrics: NIST (absolute difference of 0.51% (*statistically significant*)), WER (absolute difference of 0.78% (*statistically significant*)), and PER (absolute difference of 0.45% (*statistically significant*)). The worst performing systems are RESCORED$_{MPROD}$ and RESCORED$_{BPROD}$ as per the five MT evaluation metrics: BLEU, NIST, METEOR, WER, and PER.

As far as the percentage of oracles in the 1-best position is concerned, RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ score the highest number in both OBLEU (60 more sentences than BASELINE and 32 more sentences than RESCORED$_{BSUM}$) and OMET (47 more sen-

tences than BASELINE and 29 more sentences than RESCORED$_{BSUM}$) on the testset. This behavious was observed on the devset as well.

Our rescoring systems seem to fare better on the testset than on the devset, as observed in 100-best, 250-best and 500-best lists. Additionally, the PROD systems easily outperform the SUM systems. Both these bahaviours are contrary to that observed on larger *n*-best lists: 750-best, 1000-best, and 2500-best. More on such trends will be handled below.

SUMMARY

Having reported on the performance of rescoring systems in individual *n*-best lists, we would now like to comment on any general trends observed in French–English translation systems as a whole. Table 3.16 summarises the performance of our rescoring systems on French→English data by listing the best-performing systems in each of the seven *n*-best lists (rows: 100-best, 250-best, 500-best, 750-best, 1000-best, 2500-best, and 5000-best) for each of the seven evaluation metrics (columns: BLEU, NIST, METEOR, WER, PER, OBLEU, and OMET). The table is divided into two sections: (a) devset and (b) testset. The abbreviations used for each of the five systems are as follows: B (BASELINE), bP (RESCORED$_{BPROD}$), mP (RESCORED$_{MPROD}$), bS (RESCORED$_{BSUM}$), and mS (RESCORED$_{MSUM}$).

There is no discernible pattern visible in the type of system (B, bP, mP, bS, mS) scoring the best scores on French→English *n*-best lists. Table 3.16 demonstrates that each of the seven metrics favour different systems for different *n*-best list sizes. However, some facts can still be gleaned from the evaluation results.

One or more of the rescoring systems almost always achieve the highest percentage of oracles in the 1-best position (OBLEU, OMET). This shows that our rescoring strategies have been successful in their primary aim of moving oracles up the *n*-best lists.

The BASELINE system remains unbeatable on BLEU score. However, one or both the SUM systems perform at par or outperform the baseline on NIST scores.

The RESCORED$_{MSUM}$ system is consistently the best-performing system on ME-

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
100-BEST	B	bS	B	bS	bS	mP	mP
250-BEST	B	bS, mS	B	bS, mS	bS	bP, mP	bP, mP
500-BEST	B	bS, mS	B	bS, mS	bS	mP	mP
750-BEST	B	B, bP, mP	mS	B	B	bP	bS
1000-BEST	B	B, bP, mP	mS	B	B	mP, bS	bS
2500-BEST	B	B	mS	B	B	bP	B
5000-BEST	B	bS	B	bS	bS	bP	bP, mP
(b) testset							
100-BEST	B	bS, mS	B	bS	bS	bS	mP
250-BEST	B	bS, mS	B	bS, mS	bS	bS	mP
500-BEST	B	bS	B	bS	bS	bS	bS
750-BEST	B, mP	B	mS	mP	B	B	B
1000-BEST	bP	B	mS	bP	B	mS	mP
2500-BEST	B	B	mS	mP	B	mS	mP
5000-BEST	B	bS, mS	B	bS	bS, mS	bP	mP

Table 3.16: Summary of the best-performing French→English translation systems across all *n*-best lists: (a) devset and (b) testset

TEOR from 750-best to 1000-best lists on both the devset and testset. We also note that one or more of the rescoring systems (most often RESCORED$_{MSUM}$) outperform the baseline on WER and PER across most *n*-best list sizes.

Another important observation is that there are two groups of best systems. 100-best, 250-best, 500-best and 5000-best appear to rank the same systems at top while 750-best, 1000-best, and 2500-best lists rank another type of systems at the top. On further investigation, it was found that 750-best, 1000-best, and 2500-best lists appear to have the same range of lambdas while the other group adheres to another range much closer to the baseline system.

In addition to identifying the best-performing systems, we also note the general trend of a metric with increasing *n*-best list sizes for each of the five systems. Figure 3.6 shows this phenomenon for the BLEU score on the testset. Both RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ give similar performances and score at a lower level than the remaining three systems. Also, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems give similar performances to the BASELINE system (black line), even outperforming on larger *n*-best lists. While the SUM systems consistently deteriorate with increasing *n*-best list sizes,

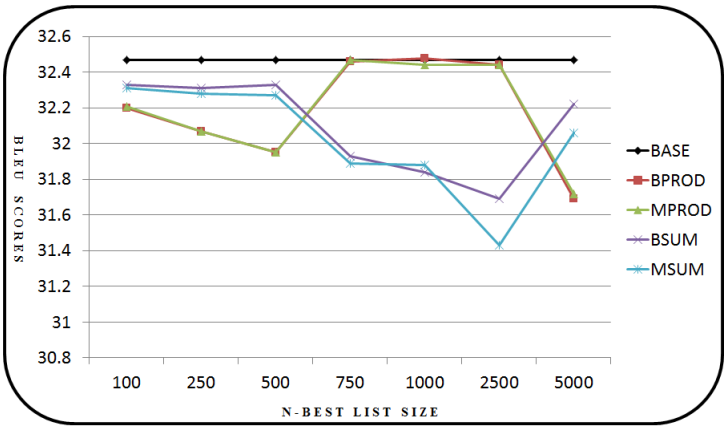


Figure 3.6: Line graph to show the trends of the five PB-SMT systems for BLEU score with increasing *n*-best list sizes: Europarl data, French–English, testset.

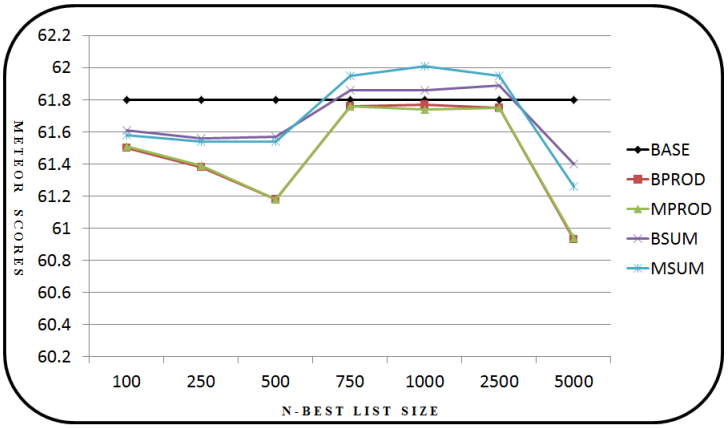


Figure 3.7: Line graph to show the trends of the five PB-SMT systems for METEOR score with increasing *n*-best list sizes: Europarl data, French–English, testset.

the PROD systems remain relatively constant and even improve in accuracy.

In Figure 3.7, we plot the METEOR performance of the four rescoring systems and the baseline system for each of the seven *n*-best list sizes. Here, the behaviour is erratic and somewhat reverse of what was observed on the BLEU metric. All four rescoring systems perform below baseline on all *n*-best list sizes except from 750-best to 2500-best lists. The RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ give much better performance than the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems, which is opposite that on the BLEU metric in Figure 3.6.

Figures 3.8 and 3.9 plot the percentage of oracles with respect to BLEU and METEOR, respectively against *n*-best list sizes for all five systems. Note that the phenomenon of decreasing values with increasing *n*-best lists is seen across all metrics but is particu-

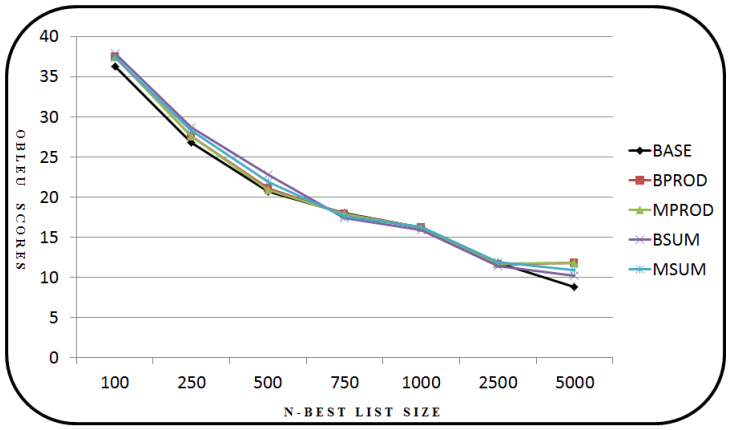


Figure 3.8: Line graph to show the trends of the five PB-SMT systems for OBLEU score with increasing *n*-best list sizes: Europarl data, French–English, testset.

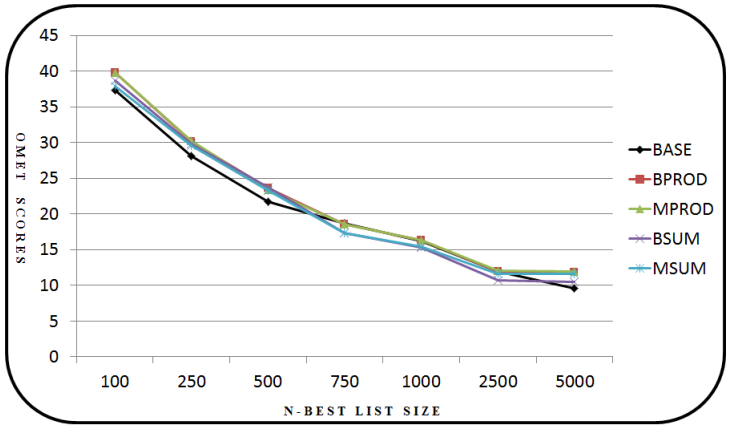


Figure 3.9: Line graph to show the trends of the five PB-SMT systems for OMET score with increasing *n*-best list sizes: Europarl data, French–English, testset.

larly prominent in the OBLEU and OMET trends. the number of oracles in the 1-best position severely reduces as we approach 5000-best lists.

All five systems intersect each other many times but the baseline system is the worst performing system. The SUM systems are initially the best but are gradually overtaken by the PROD systems towards the end. We can conclude that as the *n*-best list size is increased, more accurate oracle translations are discovered at much lower ranks and hence prove to be more difficult in being rescored to the top ranks. This implies that the complexity of the rescoring problem increases with the *n*-best list size.[9]

It is important to keep in mind that the OBLEU and OMET scores measure different things than the other five metrics. While BLEU, NIST, METEOR, WER, and PER

[9] In contrast, it could be argued that instead of increasing the complexity in terms of the expanding search space with increasing *n*-best list sizes, the parameters are simply more poorly estimated.

measure the similarity of the output (sentence ranked at the top position) to the reference translation, OMET and OBLEU merely count how many of these first ranked sentences are most similar to the reference from amongst the entire *n*-best list search space. Hence, a system can give a high similarity score (system level METEOR) and still have only a low percentage of these sentences as oracles (most similar to the reference translation).

Overall, the RESCORED$_{MSUM}$ system is the best rescoring system on French→ English data.

3.6.2 German to English

Herein, we report on rescoring *n*-best lists when translating the test2006 (devset) and test2008 (testset) datasets from German into English. We evaluate the performance of our RESC$_{prod}$ and RESC$_{sum}$ rescoring strategies by pitting the translation system scores of the four rescoring systems (RESCORED$_{BPROD}$, RESCORED$_{MPROD}$, RESCORED$_{BSUM}$, and RESCORED$_{MSUM}$) against the BASELINE system. We also make observations on whether the RESC$_{prod}$ yields better results than RESC$_{sum}$ or vice versa. Another comparison criterion is whether the BLEU-oracles are more effective in rescoring than METEOR-oracles or vice versa. These results are tested using seven evaluation metrics (BLEU, NIST, METEOR, WER, PER, OBLEU, OMET) on seven *n*-best lists of increasing sizes from 100-best to 5000-best. We will end this section with a summary by commenting on the general trend, if any, seen in all seven *n*-best lists.

100-BEST LIST

Table 3.17 (a) gives system evaluation scores on oracle-based rescoring of 100-best lists for German–English devset. The BASELINE system outperforms all four rescored systems on BLEU (with a *statistically significant* score of 26.93) and PER (with an error rate of 44.89) scores, gives similar performance to the RESC$_{BPROD}$ and RESC$_{MPROD}$ systems on NIST and WER scores, and is outperformed by the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems on METEOR AND OBLEU scores. Additionally all four rescored systems outperform the BASELINE system as far as the percentage of oracles (with respect to METEOR) in rank 1, i.e. OMET score is concerned.

The PROD rescoring (2nd and 3rd rows in Table 3.17 (a)) yields better translation results than the SUM rescoring with respect to BLEU, NIST, WER, and PER scores while the SUM rescoring (4th and 5th rows in Table 3.17 (a)) beats the PROD rescoring on the remaining metrics, namely METEOR, OBLEU, and OMET scores. When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.17 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.17 (a)), the PROD rescoring gives similar performance (e.g. 26.87 BLEU points in RESCORED$_{BPROD}$ against 26.86 BLEU points in RESCORED$_{MPROD}$) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{BSUM}$ as significantly better than RESCORED$_{MSUM}$. The RESCORED$_{BSUM}$ scores the highest METEOR score with an absolute difference of 0.31 over the BASELINE system.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**26.93**	**7.00**	57.01	65.52	**44.89**	13.30	13.65
RESCORED$_{BPROD}$	26.87	6.99	56.94	65.52	44.92	12.85	13.70
RESCORED$_{MPROD}$	26.86	6.99	56.93	**65.51**	44.92	12.95	13.80
RESCORED$_{BSUM}$	25.85	6.79	**57.32**	67.56	46.21	**14.05**	**14.25**
RESCORED$_{MSUM}$	25.78	6.79	57.19	67.45	46.19	13.90	13.70
(b) testset							
BASELINE	27.02	**7.01**	57.11	65.25	45.01	11.75	11.95
RESCORED$_{BPROD}$	27.02	**7.01**	57.12	65.19	44.96	12.00	12.25
RESCORED$_{MPROD}$	**27.03**	**7.01**	57.10	**65.17**	**44.95**	11.95	12.30
RESCORED$_{BSUM}$	26.46	6.86	**57.83**	66.99	46.14	**13.15**	**14.10**
RESCORED$_{MSUM}$	26.35	6.86	57.56	66.90	46.11	12.75	14.00

Table 3.17: Summary of the German→English translation system results for 100-best list: (a) devset and (b) testset

Table 3.17 (b) gives translation results on oracle-based rescoring of 100-best lists for German–English testset. The BASELINE system is outperformed by one or more of the four rescored systems across all evaluation metrics except the BLEU and NIST scores. However, both the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems perform at the same level as the BASELINE on BLEU (with 27.02 or 27.03 points) and NIST (with 7.01 points). The RESCORED$_{BSUM}$ scores the highest METEOR score (57.83 points *statistically significant*) with an absolute difference of 0.72 or 1.3% over the BASELINE system and an absolute difference of 0.27 or 0.5% over the second highest performing

system (RESCORED$_{MSUM}$ with 57.56 METEOR points). Both the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems record ranking 1% to 2% more oracles in the first position than the BASELINE system. For example, the RESCORED$_{BSUM}$ system was reported to have 43 more sentences with the oracles as 1-best than the BASELINE. Further analysis can be found in Section 3.8.

The PROD rescoring (2nd and 3rd rows in Table 3.17 (b)) gives better translation results than the SUM rescoring with respect to BLEU, NIST, WER, and PER scores while the SUM rescoring (4th and 5th rows in Table 3.17 (b)) beats the PROD rescoring on the remaining metrics, namely METEOR, OBLEU, and OMET scores. When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.17 (b)) with METEOR-oracles (rows 3 and 5 in Table 3.17 (b)), the PROD rescoring gives similar performance (e.g. 57.12 METEOR points in RESCORED$_{BPROD}$ against 57.10 METEOR points in RESCORED$_{MPROD}$) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{BSUM}$ as significantly better than RESCORED$_{MSUM}$ (e.g. 57.83 METEOR points in RESCORED$_{BSUM}$ against 57.56 METEOR points in RESCORED$_{MSUM}$) . Both these observations are the same as that seen on the devset in Table 3.17 (a).

An important observation is that although BLEU and NIST scores favour the BASELINE system, the difference in these scores between the baseline and the rescoring systems is reduced when moving from devset to testset. Moreover, the RESCORED$_{BSUM}$ shows a greater improvement over the BASELINE METEOR scores in testset (absolute difference of 0.72) than in devset (absolute difference of 0.31). This leads credence to our claim that our rescoring strategies have a good learnability.

250-BEST LIST

As observed in French $\rightarrow$ English (Section 3.6.1), the BASELINE system remains unchanged with regards to its translation output and therefore gives the same results on BLEU, NIST, METEOR, WER, and PER scores. The only difference is in the size of the *n*-best list which in turn is reflected in the number of sentences ranking the oracle translation in the top position resulting in OBLEU and OMET scores different from the BASELINE system in 100-best list. In fact, all systems show a decrease in their OBLEU

and OMET scores, as observed previously in Section 3.6.1.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**26.93**	**7.00**	57.01	65.52	**44.89**	**8.60**	8.55
RESCORED$_{BPROD}$	26.85	6.99	56.92	**65.48**	44.92	8.30	8.70
RESCORED$_{MPROD}$	26.86	6.99	56.93	65.49	44.92	8.35	**8.80**
RESCORED$_{BSUM}$	25.29	6.71	**57.26**	68.46	46.97	8.40	8.35
RESCORED$_{MSUM}$	25.25	6.71	57.12	68.29	46.92	7.65	7.70
(b) testset							
BASELINE	27.02	7.01	57.11	65.25	45.01	8.15	8.35
RESCORED$_{BPROD}$	**27.04**	**7.02**	57.12	**65.14**	44.93	**8.25**	8.75
RESCORED$_{MPROD}$	26.84	7.00	56.92	65.47	**44.89**	8.20	8.70
RESCORED$_{BSUM}$	25.90	6.78	**57.74**	67.83	46.83	7.35	**8.95**
RESCORED$_{MSUM}$	25.84	6.79	57.60	67.67	46.69	6.65	8.30

Table 3.18: Summary of the German→English translation system results for 250-best list: (a) devset and (b) testset

Table 3.18 (a) gives system evaluation scores on oracle-based rescoring of 250-best lists for German–English devset. Just like in the 100-best list, the BASELINE system outperforms all four rescored systems on BLEU (with a *statistically significant* score of 26.93) and PER (with an error rate of 44.89) scores. Differing from the 100-best list, the OBLEU also ranks the BASELINE system at the top with 8.60% sentences (i.e. 172 out of 2000 sentences) having BLEU-oracle translations in the 1-best position, followed by the RESCORED$_{BSUM}$, RESCORED$_{MPROD}$, and RESCORED$_{BPROD}$ systems having 168, 167, and 166 BLEU-oracles in the 1-best position, respectively. However, one or more of our rescoring systems performed well on the remaining four metrics:

- NIST: RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ perform at par with the BASELINE
- METEOR: RESCORED$_{BSUM}$ is the best-performing system with a *statistically significant* value of 57.26
- WER: RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ perform slightly better than the BASELINE

- OMET: RESCORED$_{MPROD}$ has the highest percentage of oracles in the 1-best position.

In contrast to 100-best list, the SUM rescoring (4th and 5th rows in Table 3.18 (a)) only beats the PROD rescoring (2nd and 3rd rows in Table 3.18 (a)) on METEOR scores with a *statistically significant* value of 57.26. Unexpectedly, the OMET score seems to favour the PROD systems which have lower METEOR scores than the SUM systems. This observation will be discussed in some detail after observations of all the seven *n*-best lists for German–English data. All other observations were similar to that of 100-best list described above.

The results for rescoring 250-best outputs of testset appear in Table 3.18 (b). The RESCORED$_{BPROD}$ system performs at par (BLEU, NIST, METEOR) or significantly better (WER, PER, OBLEU, OMET) than the BASELINE system across all metrics. The RESCORED$_{BSUM}$ system beats the BASELINE system as well as gives the best scores on METEOR (*statistically significant* 57.74) and OMET (8.95%).

As observed in the 100-best list, the RESCORED$_{BSUM}$ shows a greater improvement over the BASELINE METEOR scores in testset (absolute difference of 0.63) than in devset (absolute difference of 0.25). All other observations followed the same pattern as the 100-best list as well. An analysis of the metric score trend with increasing *n*-best list size is made after Table 3.23.

500-BEST LIST

In Tables 3.19 (a) and (b), we report on rescoring 500-best lists for devset and testset, respectively. RESCORED$_{BSUM}$ is the best-performing system (*statistically significant*) on the METEOR metric with an absolute difference of 0.34 and 0.74 points from the BASELINE on devset and testset, respectively. The second best-performing system on the METEOR metric for both devset and testset is RESCORED$_{MSUM}$ (devset: 57.08 points, testset: 57.66 points). However, the SUM systems fail to outperform any other system on any other metric, not even OMET and/or OBLEU as seen in 100-best and 250-best German–English data.

This implies that even though a significant number of SUM system segments match

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**26.93**	**7.00**	57.01	65.52	**44.89**	**6.25**	7.05
RESCORED$_{BPROD}$	26.84	**7.00**	56.93	**65.46**	44.90	6.15	**7.40**
RESCORED$_{MPROD}$	26.84	**7.00**	56.92	65.47	**44.89**	6.10	**7.40**
RESCORED$_{BSUM}$	24.86	6.62	**57.35**	69.44	47.85	5.20	4.90
RESCORED$_{MSUM}$	24.77	6.63	57.08	69.15	47.59	5.15	5.05
(b) testset							
BASELINE	27.02	7.01	57.11	65.25	45.01	6.55	7.05
RESCORED$_{BPROD}$	27.04	**7.02**	57.10	65.11	44.93	**6.65**	**7.50**
RESCORED$_{MPROD}$	**27.07**	**7.02**	57.12	**65.10**	**44.92**	6.60	7.45
RESCORED$_{BSUM}$	25.52	6.70	**57.85**	68.66	47.60	5.05	5.60
RESCORED$_{MSUM}$	25.50	6.72	57.66	68.36	47.34	4.35	4.85

Table 3.19: Summary of the German→English translation system results for 500-best list: (a) devset and (b) testset

with the reference translation segments as per the system-level METEOR score, a majority of them are not oracles (resulting in low OBLEU and OMET scores). More light will be shed on this in the analysis section after describing the results of 5000-best lists.

On the devset (Table 3.19 (a)), we observe that RESC$_{BPROD}$ and RESC$_{MPROD}$ systems beat the BASELINE and give the best scores on WER and OMET, and perform at par with the BASELINE system on NIST and PER. The BLEU score continues to be favouring our baseline system. On the other hand, on the testset (Table 3.19 (b)), apart from METEOR scores, the RESCORED$_{MPROD}$ system seems to be the best system across all metrics including BLEU.

Note since the BASELINE system remains unchanged across the varying *n*-best list sizes (apart from the decreasing %age of oracles in rank 1: OBLEU, OMET scores), the fact that the different rescoring systems are either performing at the same level or outperforming (by more and more greater margins) our baseline system with increasing *n*-best list sizes, lends support to the effectiveness of RESC$_{sum}$ and RESC$_{prod}$ rescoring strategies.

Once again, our rescoring systems seem to fare better on the testset than on the devset, as observed in 100-best and 250-best lists.

750-BEST LIST

Table 3.20 (a) gives system evaluation scores on oracle-based rescoring of 750-best lists for German–English devset. The best-performing system is RESCORED$_{BSUM}$ as scored by METEOR with a *statistically significant* value of 57.36 points. The PROD systems perform at par with the BASELINE on NIST and PER and outperform the BASELINE on WER and OMET. The largest improvement in pure numerical terms was seen in RESCORED$_{BSUM}$ with an absolute difference of 0.35 METEOR points (0.6%). Note that the RESCORED$_{BSUM}$ was also found to give worst performance on all other metrics. A manual analysis on why the METEOR scoring metric alone seems to favour RESCORED$_{BSUM}$ system shall be covered towards the end of this Section 3.6.2.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**26.93**	**7.00**	57.01	65.52	44.89	**5.55**	6.55
RESCORED$_{BPROD}$	26.83	**7.00**	56.94	65.46	**44.88**	5.25	6.65
RESCORED$_{MPROD}$	26.84	**7.00**	56.91	**65.45**	44.89	5.35	**6.75**
RESCORED$_{BSUM}$	24.66	6.56	**57.36**	70.08	48.41	3.45	4.20
RESCORED$_{MSUM}$	24.76	6.61	57.12	69.46	47.85	3.75	4.55
(b) testset							
BASELINE	27.02	7.01	57.11	65.25	45.01	5.90	6.05
RESCORED$_{BPROD}$	27.08	**7.03**	57.12	65.08	**44.88**	5.95	**6.40**
RESCORED$_{MPROD}$	**27.09**	**7.03**	57.12	**65.06**	44.90	**6.05**	**6.40**
RESCORED$_{BSUM}$	25.16	6.63	**57.78**	69.40	48.29	3.75	4.05
RESCORED$_{MSUM}$	25.27	6.68	57.54	68.79	47.76	4.05	3.65

Table 3.20: Summary of the German→English translation system results for 750-best list: (a) devset and (b) testset

The PROD rescoring (2nd and 3rd rows in Table 3.20 (a)) yields significantly better translation results than the SUM rescoring (4th and 5th rows in Table 3.20 (a)) across all metrics except METEOR scores.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.20 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.20 (a)), the PROD rescoring gives similar performance (e.g. 56.94 METEOR points in RESCORED$_{BPROD}$ against 56.91 METEOR points in RESCORED$_{MPROD}$) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{MSUM}$ as significantly better than RESCORED$_{BSUM}$ on all metrics

except the METEOR scores.

The results for rescoring 750-best outputs of testset appear in Table 3.20 (b). The rescoring system with the lowest evaluation scores is RESCORED$_{BSUM}$ as per all metrics except the METEOR score. As observed in the devset, the best-performing system is also RESCORED$_{BSUM}$ as scored by METEOR (a 1.17% improvement over the baseline). In contrast to the devset, RESCORED$_{MPROD}$ and RESCORED$_{BPROD}$ systems show slight improvement / give at par performance with the baseline system on all metrics including the BLEU score (0.07 points absolute improvement).

All other observations were similar to that seen in the devset. Moreover, when comparing performance of the rescoring systems on devset versus testset, we find that the testset again gives better performance over the BASELINE system. For example, the RESCORED$_{BPROD}$ records a 0.13 PER points improvement over the BASELINE system on the testset as opposed to a mere 0.01 PER points on the devset.

Note that as expected, most of these observations followed the same pattern as the preceding *n*-best list sizes on German–English data.

1000-BEST LIST

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**26.93**	**7.00**	57.01	65.52	44.89	**5.15**	6.15
RESCORED$_{BPROD}$	26.81	**7.00**	56.92	**65.44**	**44.86**	4.80	6.10
RESCORED$_{MPROD}$	26.82	**7.00**	56.92	**65.44**	**44.86**	4.95	**6.20**
RESCORED$_{BSUM}$	24.56	6.53	**57.39**	70.37	48.72	2.65	3.35
RESCORED$_{MSUM}$	24.53	6.57	57.14	69.83	48.20	2.75	3.10
(b) testset							
BASELINE	27.02	7.01	57.11	65.25	45.01	5.50	5.50
RESCORED$_{BPROD}$	27.07	7.02	57.10	65.03	44.87	**5.60**	**5.85**
RESCORED$_{MPROD}$	**27.10**	**7.03**	57.13	**65.02**	**44.85**	**5.60**	**5.85**
RESCORED$_{BSUM}$	25.08	6.60	**57.79**	69.72	48.57	3.30	3.35
RESCORED$_{MSUM}$	25.17	6.66	57.58	69.09	47.98	3.55	3.30

Table 3.21: Summary of the German→English translation system results for 1000-best list: (a) devset and (b) testset

Table 3.21 (a) gives system evaluation scores on oracle-based rescoring of 1000-best lists for German–English devset. The best-performing systems are RESCORED$_{BPROD}$

and RESCORED$_{MPROD}$ because they give slightly better (WER, PER, OMET) or similar performance (NIST) to the BASELINE system on all metrics except BLEU and METEOR. Still, the largest improvement was seen in RESCORED$_{BSUM}$ with an absolute difference of 0.38 METEOR points (57.39 Vs BASELINE's 57.01). However, overall (barring the METEOR system scores), the SUM systems fared worse than the other systems.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.21 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.21 (a)), the SUM rescoring systems show a larger degree of variation especially on the METEOR (favouring BLEU-oracles), WER (favouring METEOR-oracles), and PER (favouring METEOR-oracles) metrics.

The results for rescoring 1000-best outputs of testset appear in Table 3.21 (b). The best-performing system is RESCORED$_{MPROD}$ with respect to BLEU, NIST, WER, PER, OBLEU, OMET scores, and RESCORED$_{BSUM}$ with respect to METEOR scores (*statistically significant*). The largest improvement was seen in RESCORED$_{BSUM}$ with an absolute difference of 0.68 METEOR points (1.2% improvement over the baseline).

Note that all these observations followed the same pattern as the preceding *n*-best list sizes on German–English data.

2500-BEST LIST

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**26.93**	**7.00**	57.01	65.52	44.89	**4.25**	**4.85**
RESCORED$_{BPROD}$	26.83	**7.00**	56.94	65.38	44.81	3.85	4.70
RESCORED$_{MPROD}$	26.82	**7.00**	56.92	**65.37**	**44.80**	3.90	4.75
RESCORED$_{BSUM}$	24.02	6.44	**57.21**	71.37	49.55	2.00	1.45
RESCORED$_{MSUM}$	24.17	6.49	57.11	70.78	49.08	2.15	1.50
(b) testset							
BASELINE	27.02	7.01	57.11	65.25	45.01	4.40	4.50
RESCORED$_{BPROD}$	27.09	**7.03**	57.10	64.99	44.84	**4.45**	**4.65**
RESCORED$_{MPROD}$	**27.11**	**7.03**	57.11	**64.97**	**44.82**	**4.45**	**4.65**
RESCORED$_{BSUM}$	24.66	6.53	**57.74**	70.70	49.31	1.85	2.75
RESCORED$_{MSUM}$	24.68	6.56	57.54	70.26	48.89	1.85	2.90

Table 3.22: Summary of the German→English translation system results for 2500-best list: (a) devset and (b) testset

Table 3.22 (a) gives system evaluation scores on oracle-based rescoring of 2500-best

lists for German–English devset. The RESCORED$_{BSUM}$ scores the highest on METEOR (57.21 points) while RESCORED$_{MPROD}$ scores the best on WER (65.37 points) and PER (44.80 points) metrics.

Again, the PROD rescoring (2nd and 3rd rows in Table 3.22 (a)) yields better translation results than the SUM rescoring (4th and 5th rows in Table 3.22 (a)) across all metrics (BLEU, NIST, WER, PER, OBLEU, and OMET) except METEOR scores.

The results for rescoring 2500-best outputs of testset appear in Table 3.22 (b). The RESCORED$_{BPROD}$ system gives a statistically significant improvement over BASELINE on BLEU (0.3%), WER (0.43%), and PER (0.42%) metrics, while the RESCORED$_{BSUM}$ gives a statistically significant improvement over BASELINE on METEOR metric (1.1%).

When comparing performance of the rescoring systems on devset versus testset, we find that again the testset shows a greater degree of improvement over the BASELINE system than the devset.

Note that all these observations followed the same pattern as the preceding *n*-best list sizes on German–English data, even if some metrics record a slightly lower score than the corresponding systems in a smaller *n*-best list.

5000-BEST LIST

Table 3.23 (a) and (b) give system evaluation scores on oracle-based rescoring of 5000-best lists for German–English devset and testset, respectively. All PB-SMT system behaviour adhere to the same pattern as that seen in the preceding *n*-best list sizes of German-English data.

The RESCORED$_{BSUM}$ system once again gives the best performance on METEOR (0.38 points improvement over the BASELINE on devset and 0.68 points improvement over the BASELINE on testset). While the RESCORED$_{MPROD}$ system is unable to outperform the BASELINE on devset as per BLEU and NIST, it successfully beats the BASELINE system on testset across all metrics.

SUMMARY

Having reported on the performance of rescoring systems in individual *n*-best lists, we would now like to comment on any general trends observed in German–English trans-

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**26.93**	7.00	57.01	65.52	44.89	**3.95**	**4.30**
RESCORED$_{BPROD}$	26.83	**7.01**	56.93	**65.32**	**44.78**	3.60	4.05
RESCORED$_{MPROD}$	26.81	7.00	56.92	65.36	44.80	3.60	4.05
RESCORED$_{BSUM}$	23.63	6.36	**57.23**	72.34	50.56	1.10	1.15
RESCORED$_{MSUM}$	23.78	6.43	57.07	71.39	49.67	1.25	1.20
(b) testset							
BASELINE	27.02	7.01	57.11	65.25	45.01	3.80	4.10
RESCORED$_{BPROD}$	27.11	**7.04**	57.11	**64.95**	**44.79**	**3.85**	**4.25**
RESCORED$_{MPROD}$	**27.12**	**7.04**	57.12	**64.95**	44.81	3.80	4.15
RESCORED$_{BSUM}$	24.20	6.44	**57.62**	71.74	50.30	1.25	1.75
RESCORED$_{MSUM}$	24.32	6.50	57.42	70.97	49.53	1.25	1.60

Table 3.23: Summary of the German→English translation system results for 5000-best list: (a) devset and (b) testset

lation systems as a whole. Table 3.24 summarises the performance of our rescoring systems on German–English data by listing the best-performing systems in each of the seven *n*-best lists (rows: 100-best, 250-best, 500-best, 750-best, 1000-best, 2500-best, and 5000-best) for each of the seven evaluation metrics (columns: BLEU, NIST, METEOR, WER, PER, OBLEU, and OMET). The table is divided into two sections: (a) devset and (b) testset. The abbreviations used for each of the five systems are as follows: B (BASELINE), bP (RESCORED$_{BPROD}$), mP (RESCORED$_{MPROD}$), bS (RESCORED$_{BSUM}$), and mS (RESCORED$_{MSUM}$).

It is clearly evident from Table 3.24, that a general pattern is followed across increasing *n*-best lists and that the performance is consistent for a specific metric. For example, the RESCORED$_{BSUM}$ is consistently the best-performing system on METEOR across all *n*-best list sizes on both the devset and testset. We also note that although the RESCORED$_{BSUM}$ is the best-performing system as per METEOR scores, it gives one of the lowest performances on the BLEU score. One possible reason for this is that the METEOR scores are computed using both precision and recall while the BLEU score is purely a precision-based metric. It was discovered that RESCORED$_{BSUM}$ tends to have a higher recall and a lower precision than the RESCORED$_{BPROD}$ system.

Another important observation is that more rescoring systems outperform the baseline

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
100-BEST	B	B	bS	mP	B	bS	bS
250-BEST	B	B	bS	bP	B	B	mP
500-BEST	B	B, bP, mP	bS	bP	B, mP	B	bP, mP
750-BEST	B	B, bP, mP	bS	mP	bP	B	mP
1000-BEST	B	B, bP, mP	bS	bP, mP	bP, mP	B	mP
2500-BEST	B	B, bP, mP	bS	mP	mP	B	B
5000-BEST	B	bP	bS	bP	bP	B	B
(b) testset							
100-BEST	mP	B, bP, mP	bS	mP	mP	bS	bS
250-BEST	bP	bP	bS	bP	mP	bP	bS
500-BEST	mP	bP, mP	bS	mP	mP	bP	bP
750-BEST	mP	bP, mP	bS	mP	bP	mP	bP, mP
1000-BEST	mP	mP	bS	mP	mP	bP, mP	bP, mP
2500-BEST	mP	bP, mP	bS	mP	mP	bP, mP	bP, mP
5000-BEST	mP	bP, mP	bS	bP, mP	bP	bP	bP

Table 3.24: Summary of the best-performing German→English translation systems across all *n*-best lists: (a) devset and (b) testset

(B) on the testset than on the devset.

In addition to identifying the best-performing systems, we also note the general trend of a metric with increasing *n*-best list sizes for each of the five systems. Figure 3.10 shows this phenomenon for the BLEU score on the testset. Both RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ perform at a lower level than the remaining three systems. Also, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems are give similar performance to the BASELINE system (black line), even outperforming on larger *n*-best lists.

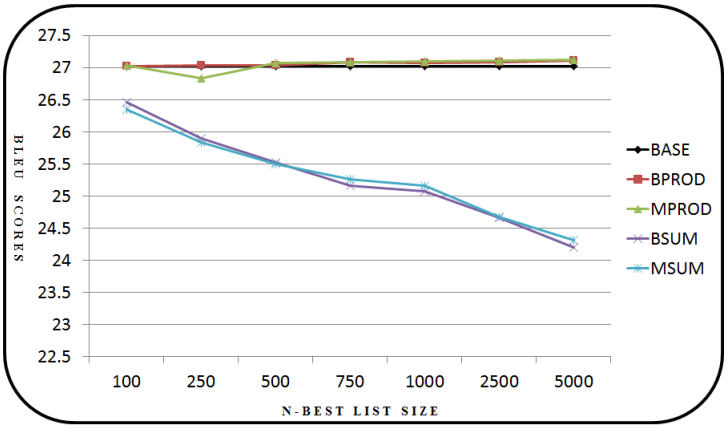


Figure 3.10: Line graph to show the trends of the five PB-SMT systems for BLEU score with increasing *n*-best list sizes: Europarl data, German–English, testset.

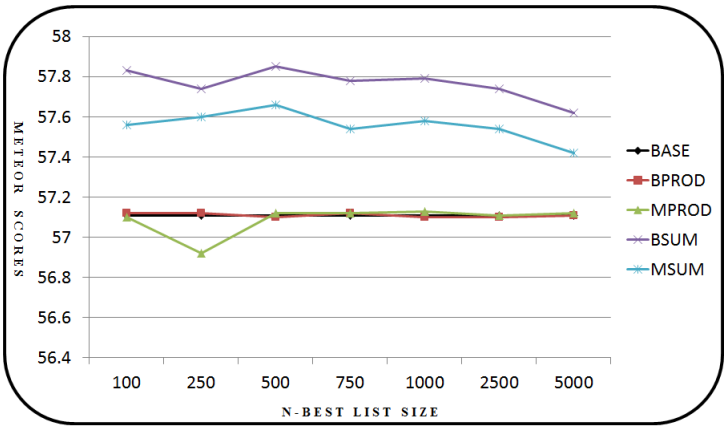


Figure 3.11: Line graph to show the trends of the five PB-SMT systems for METEOR score with increasing *n*-best list sizes: Europarl data, German–English, testset.

In Figure 3.11, we plot the METEOR performance of the four rescoring systems and the baseline system for each of the seven *n*-best list sizes. Here, the behaviour is somewhat reverse what was observed on the BLEU metric. The RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ give once again similar performance to the BASELINE but perform at a lower level than the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems, which is opposite that on the BLEU metric in Figure 3.10.

Figures 3.12 and 3.13 plot the percentage of oracles with respect to BLEU and METEOR, respectively against *n*-best list sizes for all five systems. Note that the phenomenon of decreasing values with increasing *n*-best lists is seen across all metrics but is particularly prominent in the OBLEU and OMET trends. The number of oracles in the 1-best position severely reduces as we approach 5000-best lists. However, since this trend is uniform across all systems and will be confirmed in other language pairs, we can conclude that as the *n*-best list size is increased, more accurate translations (the oracles) are discovered. This implies that the complexity of the rescoring problem increases with the *n*-best list size.

The RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems were outperforming the remaining three systems for 100-best lists and gradually dropped much below as the *n*-best list increased. This is despite the fact that these systems continued outperforming the other systems on METEOR scores. We reiterate that the OBLEU and OMET scores measure different things than the other five metrics. While BLEU, NIST, METEOR, WER, and

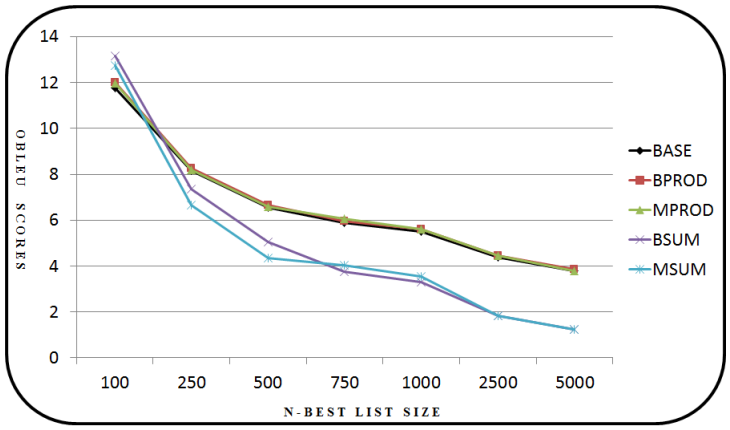


Figure 3.12: Line graph to show the trends of the five PB-SMT systems for OBLEU score with increasing *n*-best list sizes: Europarl data, German–English, testset.

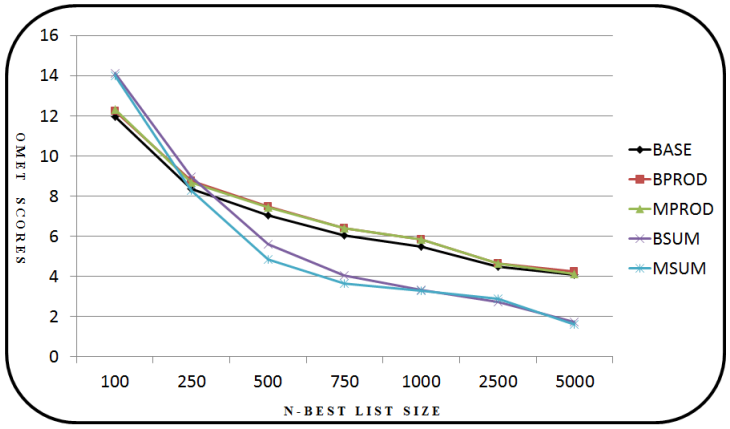


Figure 3.13: Line graph to show the trends of the five PB-SMT systems for OMET score with increasing *n*-best list sizes: Europarl data, German–English, testset.

PER measure the similarity of the output (sentence ranked at the top position) to the reference translation, OMET and OBLEU merely count how many of these first ranked sentences are most similar to the reference from amongst the entire *n*-best list search space. Hence, a system can give a high similarity score (system level METEOR) and still have only a low percentage of these sentences as oracles (most similar to the reference translation).

3.6.3 Spanish to English

In this section, we report on rescoring *n*-best lists when translating the test2006 (devset) and test2008 (testset) datasets from Spanish into English. We evaluate the performance of our RESC$_{prod}$ and RESC$_{sum}$ rescoring strategies by pitting the translation system scores of the four rescoring systems (RESCORED$_{BPROD}$, RESCORED$_{MPROD}$, RESCORED$_{BSUM}$, and RESCORED$_{MSUM}$) against the BASELINE system. We also make observations on whether the RESC$_{prod}$ yields better results than RESC$_{sum}$ or vice versa. Another comparison criterion is whether the BLEU-oracles are more effective in rescoring than METEOR-oracles or vice versa. These results are tested using seven evaluation metrics (BLEU, NIST, METEOR, WER, PER, OBLEU, OMET) on seven *n*-best lists of increasing sizes from 100-best to 5000-best. We will end this section with a summary by commenting on the general trend, if any, seen in all seven *n*-best lists and whether this trend is similar to that seen in the PB-SMT systems form French $\rightarrow$ English (Section 3.6.1) and German $\rightarrow$English (Section 3.6.2) language directions.

100-BEST LIST

Table 3.25 (a) gives system evaluation scores on oracle-based rescoring of 100-best lists for Spanish–English devset. The BASELINE system outperforms all four rescored systems on BLEU (with a *statistically significant* score of 32.98) and WER (with an error rate of 56.50) scores, gives similar performance to the RESC$_{BPROD}$ and RESC$_{MPROD}$ systems on NIST (7.80 Vs 7.79)and PER (40.68 Vs 40.69) scores, and is outperformed by the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems on METEOR. The RESCORED$_{BSUM}$ system outperforms the BASELINE on METEOR with an absolute difference of 0.08

points.

Surprisingly, none of the four rescored systems outperform the BASELINE system as far as the percentage of oracles (with respect to BLEU and METEOR) in rank 1, i.e. OBLEU and OMET respectively, is concerned.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.98**	**7.80**	61.99	**56.50**	**40.68**	**32.45**	**33.80**
RESCORED$_{BPROD}$	32.91	7.79	61.94	56.55	40.69	32.20	33.40
RESCORED$_{MPROD}$	32.91	7.79	61.94	56.55	40.69	32.30	33.50
RESCORED$_{BSUM}$	32.35	7.68	**62.07**	57.26	41.21	31.80	31.70
RESCORED$_{MSUM}$	32.46	7.70	61.95	57.18	41.15	31.25	31.90
(b) testset							
BASELINE	32.88	**7.88**	61.95	56.03	**40.12**	**33.05**	**33.20**
RESCORED$_{BPROD}$	32.88	**7.88**	61.95	**56.02**	40.15	32.80	32.70
RESCORED$_{MPROD}$	32.88	7.87	61.95	56.03	40.15	32.95	32.85
RESCORED$_{BSUM}$	32.85	7.79	**62.17**	56.58	40.54	31.85	32.25
RESCORED$_{MSUM}$	**32.94**	7.82	62.11	56.36	40.36	31.50	31.90

Table 3.25: Summary of the Spanish→English translation system results for 100-best list: (a) devset and (b) testset

The PROD rescoring (2nd and 3rd rows in Table 3.25 (a)) yields better translation results than the SUM rescoring (4th and 5th rows in Table 3.25 (a)) on all metrics (BLEU, NIST, WER, PER, OBLEU, OMET) except the METEOR scores.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.25 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.25 (a)), the PROD rescoring (RESCORED$_{BPROD}$ Vs RESCORED$_{MPROD}$) gives similar performance (i.e. exactly same figures for all metrics) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{MSUM}$ as significantly better than RESCORED$_{BSUM}$ on most metrics.

Table 3.25 (b) gives translation results on oracle-based rescoring of 100-best lists for Spanish–English testset. The BASELINE system is outperformed by one or more of the four rescored systems across all evaluation metrics except the PER (*not statistically significant*) score. In contrast to 100-best list of German–English data, the BASELINE system records the highest number of oracles in the 1-best position (highest OBLEU and OMET scores). Whether this is an anomaly or is characteristic of the Spanish–English data re-

mains to be seen.

Both the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems perform at the same level as the BASELINE on BLEU (with 32.88 points), NIST (with 7.88 points), and WER (56.03 points). However, the SUM rescoring systems score the highest on both BLEU and METEOR with the RESCORED$_{MSUM}$ outperforming the BASELINE with 32.94 BLEU points (0.2 % difference) and the RESCORED$_{BSUM}$ reporting the highest METEOR score (62.17 points *statistically significant*) with an absolute difference of 0.22 or 0.35% over the BASELINE system

The PROD rescoring (2nd and 3rd rows in Table 3.25 (b)) gives better translation results than the SUM rescoring with respect to NIST, WER, PER, OBLEU, and OMET scores while the SUM rescoring (4th and 5th rows in Table 3.25 (b)) beats the PROD rescoring on the remaining metrics, namely BLEU and METEOR scores. When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.25 (b)) with METEOR-oracles (rows 3 and 5 in Table 3.25 (b)), the PROD rescoring gives similar performance (e.g. 56.02 WER points in RESCORED$_{BPROD}$ against 56.03 WER points in RESCORED$_{MPROD}$) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{MSUM}$ as significantly better than RESCORED$_{BSUM}$ (e.g. 32.94 BLEU points in RESCORED$_{MSUM}$ against 32.85 BLEU points in RESCORED$_{BSUM}$) . Both these observations are the same as that seen on the devset in Table 3.25 (a).

An important observation is that as observed on German–English, the difference in these scores between the baseline and the rescoring systems is reduced when moving from devset to testset. Moreover, the RESCORED$_{BSUM}$ shows a greater improvement over the BASELINE METEOR scores in testset (absolute difference of 0.22) than in devset (absolute difference of 0.08). This leads credence to our claim that our rescoring strategies have a good learnability. Additionally, one of our rescoring systems (RESCORED$_{MSUM}$) achieves the best BLEU score on the testset, which was not seen on other language pairs on the 100-best list. Most of the other observations however adhered to the same pattern of preceding language pairs.

250-BEST LIST

As observed in French → English (Section 3.6.1) and German → English (Section 3.6.2), the BASELINE system remains unchanged with regards to its translation output and therefore gives the same results on BLEU, NIST, METEOR, WER, and PER scores. The only difference is in the size of the *n*-best list which in turn is reflected in the number of sentences ranking the oracle translation in the top position resulting in OBLEU and OMET scores different from the BASELINE system in 100-best list. In fact, all systems show nearly a 10% decrease in their OBLEU and OMET scores, which is similar to the pattern (albeit a higher difference in Spanish→ English) observed on French→English and German→English in Section 3.6.1 and Section 3.6.2, respectively.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.98**	**7.80**	61.99	**56.50**	**40.68**	22.90	**24.10**
RESCORED$_{BPROD}$	32.92	7.79	61.93	56.52	**40.68**	23.00	**24.10**
RESCORED$_{MPROD}$	32.91	7.79	61.93	56.52	**40.68**	**23.10**	**24.10**
RESCORED$_{BSUM}$	32.17	7.64	**62.10**	57.56	41.40	23.00	22.55
RESCORED$_{MSUM}$	32.30	7.66	62.04	57.43	41.30	22.45	23.45
(b) testset							
BASELINE	32.88	**7.88**	61.95	56.03	**40.12**	**23.95**	**23.70**
RESCORED$_{BPROD}$	**32.91**	**7.88**	61.99	**56.00**	40.15	**23.95**	**23.70**
RESCORED$_{MPROD}$	32.88	7.87	61.96	56.01	40.17	23.75	23.40
RESCORED$_{BSUM}$	32.69	7.76	**62.24**	56.77	40.72	21.95	22.35
RESCORED$_{MSUM}$	32.74	7.78	62.18	56.68	40.61	21.75	22.55

Table 3.26: Summary of the Spanish→English translation system results for 250-best list: (a) devset and (b) testset

Table 3.26 (a) gives system evaluation scores on oracle-based rescoring of 250-best lists for Spanish–English devset. Just like in the 100-best list, the BASELINE system outperforms all four rescored systems on BLEU (with a *statistically significant* score of 32.98) and WER (with an error rate of 56.50) scores. Differing from the 100-best list, the BASELINE system no longer exclusively holds the highest OBLEU and OMET scores. The RESCORED$_{MPROD}$ appears to have the highest percentage of both BLEU-oracles (OBLEU: 23.10%) and METEOR-oracles (OMET: 24.10%) in the 1-best position.

As expected from preceding datasets and language pairs, RESCORED$_{BSUM}$ is the is

the best-performing system on METEOR with a *statistically significant* value of 62.10. RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ perform at par with the BASELINE on NIST and PER.

Similar to 100-best list, the SUM rescoring (4th and 5th rows in Table 3.26 (a)) only beats the PROD rescoring (2nd and 3rd rows in Table 3.26 (a)) on METEOR scores. Also note that from 100-best to 250-best, the SUM systems appear to give worse scores on all metrics except METEOR, while the PROD systems essentially demonstrate similar evaluation scores. For example, the RESCORED$_{BSUM}$ system scores 32.17 BLEU points on 250-best and 32.35 BLEU points on 100-best. A possible explanation for such behaviour may be found in contrasting the individual parameters (lambdas) of specific features for these systems. This is dealt with in Section 3.7. All other observations were similar to that of 100-best list described above.

The results for rescoring 250-best outputs of testset appear in Table 3.26 (b). The RESCORED$_{BPROD}$ system performs at par with the BASELINE system (NIST, OBLEU, OMET) or slightly better than the BASELINE system (BLEU, METEOR, WER) across all metrics. The RESCORED$_{BSUM}$ system beats the BASELINE system as well as gives the best scores on METEOR (*statistically significant* 62.24).

As observed in the 100-best list, the RESCORED$_{BSUM}$ shows a greater improvement over the BASELINE METEOR scores in testset (absolute difference of 0.29) than in devset (absolute difference of 0.11). However, on all other metrics the scores recorded for the SUM systems are lower on the 250-best testset than on 100-best testset, similar to the pattern seen in 250-best devset from 100-best devset.

Apart from the fact that RESCORED$_{MSUM}$ is not the best-performing system for testset on the BELU metric, all other observations followed the same pattern as the 100-best list as well. An analysis of the metric score trend with increasing *n*-best list size is made after Table 3.32.

500-BEST LIST

In Tables 3.27 (a) and (b), we report on rescoring 500-best lists for devset and testset, respectively. RESCORED$_{BSUM}$ is the best-performing system (*statistically significant*)

on the METEOR metric with an absolute difference of 0.10 and 0.31 points over the BASELINE on devset and testset, respectively.

Surprisingly, the second best-performing system on the METEOR metric for devset is BASELINE and not RESCORED$_{MSUM}$, as seen previously. The difference in the scores of BASELINE and RESCORED$_{MSUM}$ is mere 0.02 METEOR points. However, on testset, the second best-performing system as per METEOR metric is RESCORED$_{MSUM}$ (62.10 points; absolute difference of 0.15 over the BASELINE). In contrast to 100-best and 250-best lists, the RESCORED$_{MSUM}$ system also scores one of the highest percentage of METEOR-oracles in the 1-best position (devset: 18.55%; testset: 20.00%).

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.98**	**7.80**	61.99	**56.50**	40.68	**18.35**	18.35
RESCORED$_{BPROD}$	32.90	7.78	61.93	56.52	40.67	18.20	**18.55**
RESCORED$_{MPROD}$	32.92	7.79	61.95	56.51	**40.66**	**18.35**	**18.55**
RESCORED$_{BSUM}$	32.06	7.62	**62.09**	57.67	41.49	17.35	17.85
RESCORED$_{MSUM}$	32.26	7.65	61.97	57.46	41.38	17.70	**18.55**
(b) testset							
BASELINE	**32.88**	**7.88**	61.95	56.03	**40.12**	**18.95**	19.45
RESCORED$_{BPROD}$	**32.88**	7.87	61.98	**56.01**	40.16	18.45	19.15
RESCORED$_{MPROD}$	**32.88**	7.87	61.97	56.02	40.18	18.40	19.00
RESCORED$_{BSUM}$	32.67	7.75	**62.26**	56.89	40.82	17.60	19.05
RESCORED$_{MSUM}$	32.71	7.77	62.10	56.73	40.68	17.80	**20.00**

Table 3.27: Summary of the Spanish→English translation system results for 500-best list: (a) devset and (b) testset

On the devset (Table 3.27 (a)), we observe that RESC$_{BPROD}$ and RESC$_{MPROD}$ systems beat the BASELINE (give the best scores as well) on PER and OMET, and perform slightly worse (*not statistically significant*) than the BASELINE system on NIST and WER.

The BASELINE system outperforms all other systems on the BLEU score (32.98 points; absolute difference of 0.06 points over RESC$_{MPROD}$, the second best-performing system). On the other hand, on the testset (Table 3.27 (b)), both the RESC$_{MPROD}$ and RESC$_{BPROD}$ systems achieve the same BLEU accuracy as the BASELINE system at 32.88 points.

The highest percentage of BLEU-oracles in the 1-best position (OBLEU) was recorded by the BASELINE on both devset and testset, together with the RESCORED$_{MPROD}$ system on devset.

Note that the BASELINE system remains unchanged across the varying *n*-best list sizes (apart from the decreasing percentage of oracles in rank 1: OBLEU, OMET scores). The RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems give similar performance from 100-best to 500-best lists. On the contrary, the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems appear to worsen on all metrics except METEOR scores from 100-best to 250-best to 500-best lists. A possible reason is that the RESC$_{sum}$ rescoring strategy unlike RESC$_{prod}$ rescoring strategy is not robust enough for increasing *n*-best lists. Whether this pattern is visible in other language pairs and datasets remains to be seen and shall be addressed under comparative analysis (cf. Section 3.11).

Once again, our rescoring systems seem to fare better on the testset than on the devset, as observed in 100-best and 250-best lists.

750-BEST LIST

Table 3.28 (a) gives system evaluation scores on oracle-based rescoring of 750-best lists for Spanish–English devset. The best-performing system is RESCORED$_{BSUM}$ as scored by METEOR with a *statistically significant* value of 62.09 points. The largest improvement in pure numerical terms was also seen in RESCORED$_{BSUM}$ with an absolute difference of 0.10 METEOR points (0.16%) over the baseline. Note that despite this, the RESCORED$_{BSUM}$ was found to give worst performance on all other metrics. This particular trait has also been observed on the German–English data and will be analysed under summary at the end of this Section 3.6.3.

From the Table 3.28 (a), it can be seen that the PROD systems perform at par with the BASELINE on NIST and WER, and outperform the BASELINE on PER and OMET.

The PROD rescoring (2nd and 3rd rows in Table 3.28 (a)) yields significantly better translation results than the SUM rescoring (4th and 5th rows in Table 3.28 (a)) across all metrics except METEOR scores. For example, the RESCORED$_{BPROD}$ system outperforms the RESCORED$_{BSUM}$ with an absolute difference of 0.95 BLEU points (2.9%). On

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.98**	**7.80**	61.99	**56.50**	40.68	**15.75**	15.95
RESCORED$_{BPROD}$	32.92	7.79	61.93	**56.50**	**40.66**	15.65	16.30
RESCORED$_{MPROD}$	32.92	7.79	61.95	56.51	**40.66**	15.70	**16.35**
RESCORED$_{BSUM}$	31.97	7.60	**62.09**	57.84	41.63	15.10	16.05
RESCORED$_{MSUM}$	32.05	7.61	62.02	57.72	41.53	15.25	16.25
(b) testset							
BASELINE	**32.88**	**7.88**	61.95	**56.03**	**40.12**	**16.00**	**17.35**
RESCORED$_{BPROD}$	32.86	7.87	61.95	56.04	40.19	15.75	17.25
RESCORED$_{MPROD}$	32.87	7.87	61.96	**56.03**	40.19	15.75	17.15
RESCORED$_{BSUM}$	32.51	7.72	**62.23**	57.06	40.97	15.50	16.75
RESCORED$_{MSUM}$	32.48	7.73	62.06	56.99	40.91	15.20	16.75

Table 3.28: Summary of the Spanish→English translation system results for 750-best list: (a) devset and (b) testset

the other hand, the RESCORED$_{BSUM}$ improves over the METEOR scores of RESC$_{BPROD}$ by a mere 0.16 points (0.25%). Therefore on 750-best lists, the RESC$_{prod}$ rescoring strategy is far more effective than the RESC$_{sum}$ rescoring strategy.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.28 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.28 (a)), the PROD rescoring gives similar performance (e.g. 61.93 METEOR points in RESCORED$_{BPROD}$ against 61.95 METEOR points in RESCORED$_{MPROD}$) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{MSUM}$ as significantly better than RESCORED$_{BSUM}$ on all metrics except the METEOR scores.

The results for rescoring 750-best outputs of testset appear in Table 3.28 (b). As observed in the devset, the best-performing system is RESCORED$_{BSUM}$ as scored by METEOR (a 0.45% improvement over the baseline). In contrast to the devset, RESC$_{MPROD}$ and RESCORED$_{BPROD}$ systems give at par performance with the baseline system on all similarity-based metrics except the Position-independent Word Error Rate (PER).

All other observations were similar to that seen in the devset. Moreover, when comparing performance of the rescoring systems on devset versus testset, we find that the testset again gives better performance over the BASELINE system. For example, the RESCORED$_{BSUM}$ is only 0.37 BLEU points below the BASELINE system on the testset

as opposed to a whole 1 BLEU point on the devset.

Note that as expected, most of these observations followed the same pattern as the preceding *n*-best list sizes on Spanish–English data.

1000-BEST LIST

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.98**	**7.80**	61.99	**56.50**	40.68	13.95	14.40
RESCORED$_{BPROD}$	32.92	7.79	61.92	56.52	40.67	**14.05**	**14.80**
RESCORED$_{MPROD}$	32.92	7.79	61.93	56.52	**40.66**	14.00	14.70
RESCORED$_{BSUM}$	31.83	7.57	**62.15**	58.06	41.79	13.65	14.35
RESCORED$_{MSUM}$	31.99	7.60	61.99	57.84	41.62	13.70	14.65
(b) testset							
BASELINE	**32.88**	**7.88**	61.95	**56.03**	**40.12**	**14.60**	**16.00**
RESCORED$_{BPROD}$	32.84	7.87	61.94	56.04	40.21	14.10	15.70
RESCORED$_{MPROD}$	32.85	7.87	61.94	56.04	40.19	14.10	15.55
RESCORED$_{BSUM}$	32.29	7.68	**62.21**	57.31	41.19	14.15	15.60
RESCORED$_{MSUM}$	32.46	7.71	62.07	57.07	41.00	13.95	15.70

Table 3.29: Summary of the Spanish→English translation system results for 1000-best list: (a) devset and (b) testset

Table 3.29 (a) gives system evaluation scores on oracle-based rescoring of 1000-best lists for Spanish–English devset. Apart from the RESCORED$_{BSUM}$ on METEOR, none of our four rescoring systems seem to outperform the BASELINE system. This behaviour appears to be prevalent in Spanish–English data unlike German–English data. However, the BASELINE system remained the top performing system on the BLEU metric alone.

The RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems give similar performance to the BASELINE on NIST, WER, and PER. This is also reflected in these two systems scoring the highest percentage of oracles at rank 1 (OBLEU, OMET). The largest improvement on the MT evaluation metrics was seen in RESCORED$_{BSUM}$ with an absolute difference of 0.16 METEOR points (62.15.39 Vs BASELINE's 61.99). However, overall (barring the METEOR system scores), the SUM systems fared worse than the remaining three systems, another common observation across *n*-best lists of Spanish–English and German–English data.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.29 (a)) with

METEOR-oracles (rows 3 and 5 in Table 3.29 (a)), the SUM rescoring systems show a larger degree of variation especially on the METEOR (favouring BLEU-oracles), WER (favouring METEOR-oracles), and PER (favouring METEOR-oracles) metrics.

The results for rescoring 1000-best outputs of testset appear in Table 3.29 (b). The best-performing system on BLEU, PER, OBLEU, and OMET is the BASELINE, while the best-performing system on METEOR is RESCORED$_{BSUM}$ (*statistically significant*). The largest improvement was seen in RESCORED$_{BSUM}$ with an absolute difference of 0.26 METEOR points (0.4% improvement over the baseline).

Note that all these observations followed the same pattern as the preceding 750-best list on Spanish–English data.

2500-BEST LIST

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.98**	**7.80**	61.99	**56.50**	40.68	9.25	10.30
RESCORED$_{BPROD}$	32.91	7.78	61.91	56.54	40.68	**9.60**	10.85
RESCORED$_{MPROD}$	32.92	7.79	61.92	**56.50**	**40.66**	**9.60**	**10.95**
RESCORED$_{BSUM}$	31.40	7.49	**62.08**	58.73	42.35	9.30	10.10
RESCORED$_{MSUM}$	31.42	7.48	62.07	58.73	42.41	9.25	10.20
(b) testset							
BASELINE	**32.88**	**7.88**	61.95	**56.03**	**40.12**	**10.40**	**11.65**
RESCORED$_{BPROD}$	32.83	7.86	61.94	56.06	40.20	10.30	11.40
RESCORED$_{MPROD}$	32.85	7.87	61.95	**56.03**	40.18	10.30	11.50
RESCORED$_{BSUM}$	31.81	7.60	**62.10**	57.94	41.72	9.15	9.50
RESCORED$_{MSUM}$	31.78	7.59	62.06	58.03	41.83	9.05	9.70

Table 3.30: Summary of the Spanish→English translation system results for 2500-best list: (a) devset and (b) testset

Table 3.30 (a) gives system evaluation scores on oracle-based rescoring of 2500-best lists for Spanish–English devset. The RESCORED$_{BSUM}$ scores the highest on METEOR (62.08 points) while RESCORED$_{MPROD}$ scores the best on PER (40.66 points). The RESCORED$_{MPROD}$ also scores the highest percentage of BLEU-oracles (OBLEU: 9.60%) and METEOR-oracle (OMET: 10.95%).

Again, the PROD rescoring (2nd and 3rd rows in Table 3.30 (a)) yields better translation results than the SUM rescoring (4th and 5th rows in Table 3.30 (a)) across all metrics

(BLEU, NIST, WER, PER, OBLEU, and OMET) except METEOR scores.

The results for rescoring 2500-best outputs of testset appear in Table 3.30 (b). As with other *n*-best lists on Spanish–English data, BASELINE remains the top scoring system as per the BLEU and PER metrics. The RESCORED$_{BSUM}$ gives a statistically significant improvement over BASELINE on METEOR metric (0.24%).

When comparing performance of the rescoring systems on devset versus testset, we find that again the testset shows a greater degree of improvement against the BASELINE system than the devset. Table 3.31 illustrates this by comparing the percentage of increase or decrease in the scores of RESCORED$_{BSUM}$ over the BASELINE. This implies that under-performing rescoring systems score closer to the baseline score on the testset than on the devset (contrast a margin of 3.25% BLEU on testset with 4.79% below BASELINE's BLEU on devset), and superior rescoring systems beat the baseline by a larger margin on the testset than on the devset (contrast an improvement margin of 0.24% METEOR on testset with 0.15% above BASELINE's METEOR on devset).

CATEGORY	DEVSET	TESTSET
BELU	4.79% ↓	3.25% ↓
METEOR	0.15% ↑	0.24% ↑

Table 3.31: Margin of difference in the BLEU and METEOR performance scores of RESCORED$_{BSUM}$ and BASELINE: Spanish–English 2500-best list

Note that all these observations followed the same pattern as the preceding *n*-best list sizes on Spanish–English data, even if some metrics record a slightly lower score than the corresponding systems in a smaller *n*-best list.

5000-BEST LIST

Table 3.32 (a) and (b) give system evaluation scores on oracle-based rescoring of 5000-best lists for Spanish–English devset and testset, respectively. All PB-SMT system behaviour adhere to the same pattern as that seen in the preceding *n*-best list sizes of Spanish-English data.

The RESCORED$_{BSUM}$ system once again gives the best performance on METEOR (0.18 points improvement over the BASELINE on devset and 0.17 points improvement

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**32.98**	**7.80**	61.99	**56.50**	40.68	7.20	8.00
RESCORED$_{BPROD}$	32.91	7.79	61.90	56.52	**40.67**	7.60	**8.65**
RESCORED$_{MPROD}$	32.91	7.79	61.90	56.51	**40.67**	**7.65**	8.60
RESCORED$_{BSUM}$	30.84	7.39	**62.17**	59.67	43.19	5.05	6.00
RESCORED$_{MSUM}$	30.89	7.40	**62.17**	59.60	43.17	5.10	6.05
(b) testset							
BASELINE	**32.88**	**7.88**	61.95	**56.03**	**40.12**	**8.50**	**9.20**
RESCORED$_{BPROD}$	32.79	7.86	61.92	56.07	40.23	8.40	8.95
RESCORED$_{MPROD}$	32.80	7.87	61.93	56.04	40.18	8.40	8.90
RESCORED$_{BSUM}$	31.23	7.48	**62.12**	58.86	42.55	5.90	5.40
RESCORED$_{MSUM}$	31.22	7.48	62.05	58.92	42.59	5.80	5.70

Table 3.32: Summary of the Spanish→English translation system results for 5000-best list: (a) devset and (b) testset

over the BASELINE on testset). Note that the improvement on testset is of the same margin as on devset and not greater unlike on all smaller *n*-best lists.

The RESCORED$_{MPROD}$ system achieves similar scores to the BASELINE system on devset as per the NIST, WER, and PER. The RESCORED$_{MPROD}$ system achieves similar scores to the BASELINE system on testset as per the NIST and WER. The PROD systems successfully beat the BASELINE system on devset in the percentage of both BLEU-oracles and METEOR-oracles in the 1-best position. However, on the testset, the BASELINE system retains the highest OBLEU and OMET percentages albeit by a lower margin.

SUMMARY

Having reported on the performance of rescoring systems in individual *n*-best lists, we would now comment on any general trends observed in Spanish–English translation systems as a whole. The empirical evidence in the form of MT evaluation results on seven *n*-best lists for two datasets (devset, testset) would help us in answering the following questions about rescoring in Spanish–English:

- Which is the best-performing and worst performing system across the *n*-best lists of Spanish–English devset and testset?
- Do one or more of the rescoring systems improve over the baseline consistently?

- Which metric favours which type of system: RESC$_{sum}$ or RESC$_{prod}$?
- How do the evaluation scores fare with increasing *n*-best list sizes for all systems?

Note that this subsection only deals with Spanish–English performance and all comparisons across different language pairs shall be addressed in Section 3.11.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
100-BEST	B	B	bS	B	B	B	B
250-BEST	B	B	bS	B	B, bP, mP	mP	B, bP, mP
500-BEST	B	B	bS	B	mP	B, mP	bP, mP, mS
750-BEST	B	B	bS	B, bP	bP, mP	B	mP
1000-BEST	B	B	bS	B	mP	bP	bP
2500-BEST	B	B	bS	B, mP	mP	bP, mP	mP
5000-BEST	B	B	bS, mS	B	bP, mP	mP	bP
(b) testset							
100-BEST	mS	B, bP	bS	bP	B	B	B
250-BEST	bP	B, bP	bS	bP	B	B, bP	B, bP
500-BEST	B, bP, mP	B	bS	bP	B	B	mS
750-BEST	B	B	bS	B, mP	B	B	B
1000-BEST	B	B	bS	B	B	B	B
2500-BEST	B	B	bS	B, mP	B	B	B
5000-BEST	B	B	bS	B	B	B	B

Table 3.33: Summary of the best-performing Spanish→English translation systems across all *n*-best lists: (a) devset and (b) testset

Table 3.33 summarises the performance of our rescoring systems on Spanish–English data by listing the best-performing systems in each of the seven *n*-best lists (rows: 100-best, 250-best, 500-best, 750-best, 1000-best, 2500-best, and 5000-best) for each of the seven evaluation metrics (columns: BLEU, NIST, METEOR, WER, PER, OBLEU, and OMET). The table is divided into two sections: (a) devset and (b) testset. The abbreviations used for each of the five systems are as follows: B (BASELINE), bP (RESCORED$_{BPROD}$), mP (RESCORED$_{MPROD}$), bS (RESCORED$_{BSUM}$), and mS (RESCORED$_{MSUM}$).

It is clearly evident from Table 3.33, that a general pattern is followed across increasing *n*-best lists and that the performance is consistent for a specific metric, i.e. the same system(s) are ranked as the best on that particular metric (with a few exceptions). For example, the RESCORED$_{BSUM}$ is consistently the best-performing system for METEOR

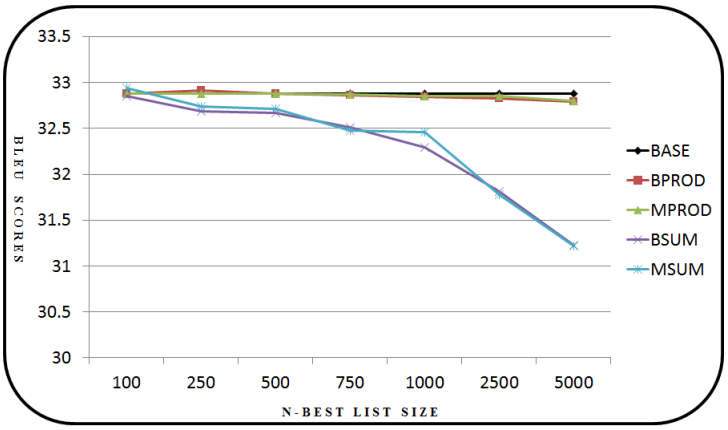


Figure 3.14: Line graph to show the trends of the five PB-SMT systems for BLEU score with increasing *n*-best list sizes: Europarl data, Spanish–English, testset.

across all *n*-best list sizes on both the devset and testset. We also note that although the RESCORED$_{BSUM}$ is the best-performing system as per METEOR scores, it gives one of the lowest performances on the BLEU score. One possible reason for this is that the METEOR scores are computed using both precision and recall while the BLEU score is purely a precision-based metric. It was discovered that RESCORED$_{BSUM}$ tends to have a higher recall and a lower precision than the RESCORED$_{BPROD}$ system. Secondly, METEOR scores take into account reordering issues when comparing a translation hypothesis to a reference translation, unlike the *n*-gram based metrics like BLEU and NIST.

The PER scores favour the BASELINE on testset while ranking one or both PROD systems at the top on devset. The PROD systems are ranked at the top more than the SUM systems (cf. WER and PER scores in Table 3.33). The testset favours baseline on larger *n*-best list sizes especially for BLEU, NIST, OBLEU, and OMET.

In addition to identifying the best-performing systems, we also note the general trend of a metric with increasing *n*-best list sizes for each of the five systems. Figure 3.14 shows this phenomenon for the BLEU score on the testset. Both RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ perform at a lower level than the remaining three systems after 100-best lists. Also, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems are give similar performance to the BASELINE system (black line), even outperforming on smaller *n*-best lists.

In Figure 3.15, we plot the METEOR performance of the four rescoring systems

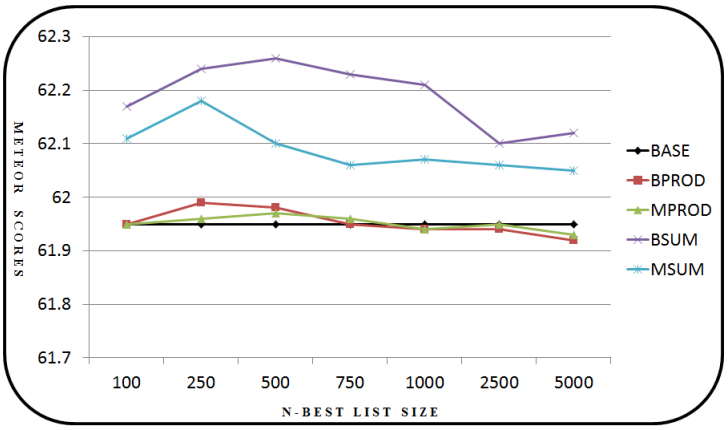


Figure 3.15: Line graph to show the trends of the five PB-SMT systems for METEOR score with increasing *n*-best list sizes: Europarl data, Spanish–English, testset.

and the baseline system for each of the seven *n*-best list sizes. Here, the behaviour is somewhat reverse what was observed on the BLEU metric. The RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ give once again similar performance to the BASELINE but perform at a lower level than the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems, which is opposite that on the BLEU metric in Figure 3.10. The RESCORED$_{BSUM}$ is the best METEOR-performing system. There is a wider gap between the SUM systems than between the PROD systems. This trend is evident in the BLEU scores as well.

Figures 3.16 and 3.17 plot the percentage of oracles with respect to BLEU and METEOR, respectively against *n*-best list sizes for all five systems. Note that the phenomenon of decreasing values with increasing *n*-best lists is seen across all metrics but is particularly prominent in the OBLEU and OMET trends. The number of oracles in the 1-best position severely reduces as we approach 5000-best lists. However, since this trend is uniform across all systems and will be confirmed in other language pairs, we can conclude that as the *n*-best list size is increased, more accurate translations (the oracles) are discovered. This implies that the complexity of the rescoring problem increases with the *n*-best list size.

The RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems were outperforming the remaining three systems for 100-best lists and gradually dropped much below as the *n*-best list increased. This is despite the fact that these systems continued outperforming the other systems on METEOR scores.

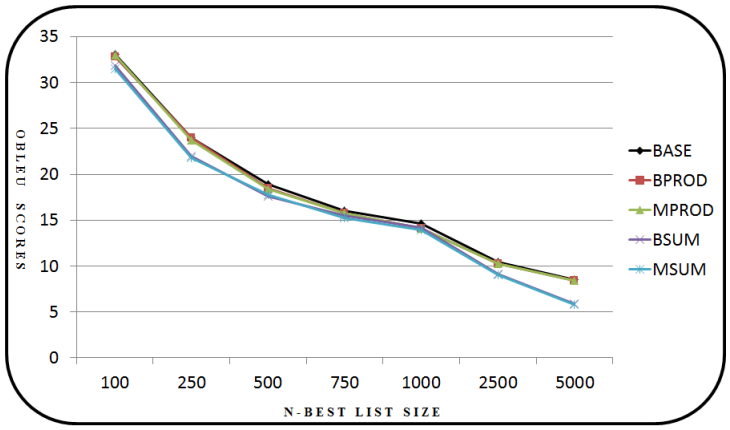


Figure 3.16: Line graph to show the trends of the five PB-SMT systems for OBLEU score with increasing *n*-best list sizes: Europarl data, Spanish–English, testset.

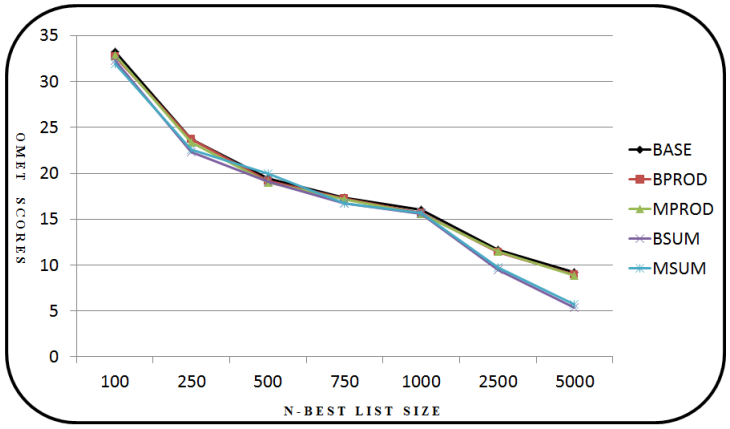


Figure 3.17: Line graph to show the trends of the five PB-SMT systems for OMET score with increasing *n*-best list sizes: Europarl data, Spanish–English, testset.

To conclude, amongst the rescoring systems, the RESC$_{MPROD}$ and the RESC$_{BPROD}$ systems are the best-performing systems on Spanish–English datasets.

3.6.4 English to French

So far, we have evaluated the rescoring and pushing oracles up the *n*-best lists of PB-SMT systems translating from different languages into English. Herein, we report on rescoring *n*-best lists when translating the test2006 (devset) and test2008 (testset) datasets from English into French. The experiments in this set have two purposes:

- To evaluate the effectiveness of rescoring *n*-best lists on a language other than English.
- To facilitate comparison of the rescoring of *n*-best lists for a dataset in both translation directions: French→English and English→French.

We evaluate the performance of our RESC$_{prod}$ and RESC$_{sum}$ rescoring strategies by pitting the translation system scores of the four rescoring systems (RESCORED$_{BPROD}$, RESCORED$_{MPROD}$, RESCORED$_{BSUM}$, and RESCORED$_{MSUM}$) against the BASELINE system. We also make observations on whether the RESC$_{prod}$ yields better results than RESC$_{sum}$ or vice versa. Another comparison criterion is whether the BLEU-oracles are more effective in rescoring than METEOR-oracles or vice versa. These results are tested using seven evaluation metrics (BLEU, NIST, METEOR, WER, PER, OBLEU, OMET) on seven *n*-best lists of increasing sizes from 100-best to 5000-best. We will end this section with a summary by commenting on the general trend, if any, seen in all seven *n*-best lists.

100-BEST LIST

Table 3.34 (a) gives system evaluation scores on oracle-based rescoring of 100-best lists for English–French devset. The BASELINE system is bested by the rescoring systems on the METEOR metric alone (textscRescored$_{BSUM}$ outperforms baseline with an absolute difference of 0.06 points). This improvement is also reflected in the highest percentage of METEOR-oracles moved up to the 1-best position by the textscRescored$_{BSUM}$

system (15 sentences). However the BASELINE system achieves the highest percentage of BLEU-oracles in the 1-best position, albeit by a nearly 50% lower margin of improvement (8 sentences). The BASELINE system also gives slightly better results than all the four rescored systems on the BLEU, WER, PER, and achieves similar NIST score as the textscRescored$_{BPROD}$ and textscRescored$_{MPROD}$ systems.

The PROD rescoring (2nd and 3rd rows in Table 3.34 (a)) yields better translation results than the SUM rescoring with respect to BLEU, NIST, WER, and PER scores while the SUM rescoring (4th and 5th rows in Table 3.34 (a)) beats the PROD rescoring on the remaining metric, i.e. METEOR score. When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.34 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.34 (a)), both sets of PROD rescoring systems (RESCORED$_{BPROD}$ Vs RESCORED$_{MPROD}$) and SUM rescoring systems (RESCORED$_{BSUM}$ Vs RESCORED$_{MSUM}$) give similar performance across all metrics.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**31.05**	**7.48**	60.88	**60.04**	**44.93**	**30.30**	30.85
RESCORED$_{BPROD}$	30.99	7.47	60.85	60.08	44.99	29.60	30.25
RESCORED$_{MPROD}$	31.00	7.47	60.87	60.06	44.97	29.90	30.55
RESCORED$_{BSUM}$	30.88	7.42	**60.94**	60.36	45.22	29.65	30.80
RESCORED$_{MSUM}$	30.86	7.42	60.91	60.38	45.27	29.90	**31.60**
(b) testset							
BASELINE	28.14	**7.01**	57.77	62.54	47.42	24.90	**27.55**
RESCORED$_{BPROD}$	28.17	**7.01**	57.76	**62.49**	47.42	24.95	27.25
RESCORED$_{MPROD}$	**28.18**	**7.01**	**57.78**	**62.49**	**47.39**	24.95	27.50
RESCORED$_{BSUM}$	27.96	6.95	57.71	62.96	47.73	24.55	26.65
RESCORED$_{MSUM}$	27.98	6.95	57.72	62.93	47.73	**25.20**	27.00

Table 3.34: Summary of the English→French translation system results for 100-best list: (a) devset and (b) testset

Table 3.34 (b) gives translation results on oracle-based rescoring of 100-best lists for English–French testset. Unlike the devset, one or more of our rescoring systems outperform the baseline across all metrics. RESCORED$_{MPROD}$ bests the BASELINE on BLEU (absolute difference of 0.04), WER (absolute difference of 0.05), and PER (absolute difference of 0.03). On NIST and METEOR, the BASELINE and RESCORED$_{MPROD}$ give

similar performance. The RESCORED$_{BSUM}$ system is able to score more BLEU-oracles than the BASELINE, but not METEOR-oracles.

Unlike the devset, the SUM rescoring systems (4th and 5th rows in Table 3.34 (b)) perform worse than the PROD rescoring systems (2nd and 3rd rows in Table 3.34 (b)) on METEOR as well as the other system evaluation metrics.

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.34 (b)) with METEOR-oracles (rows 3 and 5 in Table 3.34 (b)), both sets of PROD rescoring systems (RESC$_{BPROD}$ Vs RESC$_{MPROD}$) and SUM rescoring systems (RESC$_{BSUM}$ Vs RESC$_{MSUM}$) give similar performance across all metrics, just like in devset.

Overall, the BASELINE system remains the best (except for METEOR: RESC$_{BSUM}$) on devset, while the RESC$_{MPROD}$ system gives the best performance on the testset.

250-BEST LIST

As observed in French $\rightarrow$ English (Section 3.6.1), the BASELINE system remains unchanged with regards to its translation output and therefore gives the same results on BLEU, NIST, METEOR, WER, and PER scores. The only difference is in the size of the *n*-best list which in turn is reflected in the number of sentences ranking the oracle translation in the top position resulting in OBLEU and OMET scores different from the BASELINE system in 100-best list. In fact, all systems show a decrease in their OBLEU and OMET scores, as observed previously in Section 3.6.1.

Table 3.35 (a) gives system evaluation scores on oracle-based rescoring of 250-best lists for English–French devset. Similar to the 100-best list, the BASELINE system gives slightly better results than all the four rescored systems on the BLEU, METEOR, WER, PER, and achieves similar NIST score as the RESCORED$_{BPROD}$ and textscRescored$_{MPROD}$ systems. The textscRescored$_{MPROD}$ system is in fact the second best-performing system and the best rescoring system on devset. However, the RESCORED$_{MSUM}$ system achieves the highest percentage of BLEU-oracles and METEOR-oracles in the 1-best position. All other observations were similar to that on 100-best devset.

The results for rescoring 250-best outputs of testset appear in Table 3.35 (b). Unlike the devset, the BASELINE system is outperformed by one or more of the rescoring systems

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**31.05**	**7.48**	**60.88**	**60.04**	**44.93**	20.90	22.30
RESCORED$_{BPROD}$	30.98	7.47	60.85	60.07	44.99	20.30	22.10
RESCORED$_{MPROD}$	31.00	7.47	60.86	60.06	44.98	20.40	22.00
RESCORED$_{BSUM}$	30.68	7.37	60.81	60.70	45.53	21.00	22.05
RESCORED$_{MSUM}$	30.63	7.38	60.81	60.68	45.52	**21.05**	**22.35**
(b) testset							
BASELINE	28.14	**7.01**	57.77	62.54	47.42	17.55	19.50
RESCORED$_{BPROD}$	**28.18**	**7.01**	57.76	62.49	47.41	**17.75**	19.45
RESCORED$_{MPROD}$	**28.18**	**7.01**	**57.78**	**62.48**	**47.40**	17.70	**19.55**
RESCORED$_{BSUM}$	27.88	6.91	57.69	63.15	47.98	17.45	18.40
RESCORED$_{MSUM}$	27.83	6.91	57.60	63.16	47.98	17.50	18.10

Table 3.35: Summary of the English→French translation system results for 250-best list: (a) devset and (b) testset

on all metrics. The RESCORED$_{MPROD}$ system beats the BASELINE system as well as gives the best scores on most metrics: BLEU (absolute difference of 0.04), NIST (same score), METEOR (absolute difference of 0.01), WER (absolute difference of 0.06), and PER (absolute difference of 0.02).

All other observations followed the same pattern as the 100-best list. An analysis of the metric score trend with increasing *n*-best list size is made after Table 3.15.

Note that on comparing the BLEU, NIST, METEOR, WER, PER scores of all the four rescoring systems on 100-best list with their corresponding scores on 250-best list, we find that the scores decrease on the SUM systems (RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$) and remain relatively unchanged on the PROD systems (RESC$_{BPROD}$ and RESC$_{MPROD}$). This implies that the difference in performance quality of PROD and SUM rescoring widens from 100-best to 250-best lists, with the PROD systems as better than SUM systems.

500-BEST LIST

Table 3.36 (a) gives system evaluation scores on oracle-based rescoring of 500-best lists for English–French devset. Similar to the 100-best and 250-best lists, the BASELINE system gives slightly better results than all the four rescored systems on the BLEU, PER, and achieves similar NIST and WER scores as the RESC$_{BPROD}$ and RESC$_{MPROD}$

systems. The RESCORED$_{BPROD}$ system is in fact the second best-performing system and the best rescoring system on devset according to four out of seven metrics. The RESCORED$_{MSUM}$ system achieves the best METEOR score (absolute difference of 0.03 over the baseline) and the highest percentage of BLEU-oracles and METEOR-oracles in the 1-best position. All other observations such as the relative differences between PROD and SUM systems were similar to that on 100-best and 250-best lists.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**31.05**	**7.48**	60.88	**60.04**	**44.93**	16.05	17.05
RESCORED$_{BPROD}$	30.99	7.47	60.86	60.06	44.98	15.50	16.95
RESCORED$_{MPROD}$	30.96	7.47	60.85	60.07	44.99	15.50	16.95
RESCORED$_{BSUM}$	30.60	7.36	**60.91**	60.81	45.60	16.50	**17.60**
RESCORED$_{MSUM}$	30.60	7.36	60.88	60.83	45.62	**16.60**	16.95
(b) testset							
BASELINE	28.14	**7.01**	**57.77**	62.54	47.42	13.70	14.90
RESCORED$_{BPROD}$	28.17	**7.01**	**57.77**	62.51	**47.41**	**13.75**	14.90
RESCORED$_{MPROD}$	**28.18**	**7.01**	**57.77**	**62.50**	47.42	**13.75**	**14.95**
RESCORED$_{BSUM}$	27.63	6.87	57.58	63.47	48.26	12.55	13.30
RESCORED$_{MSUM}$	27.79	6.89	57.70	63.31	48.12	12.95	13.80

Table 3.36: Summary of the English→French translation system results for 500-best list: (a) devset and (b) testset

The results for rescoring 250-best outputs of testset appear in Table 3.35 (b). Unlike the devset, the BASELINE system is outperformed by one or more of the rescoring systems on all metrics. This is reflected in the 100-best and 250-best lists as well. The RESCORED$_{MPROD}$ system performs slightly better than the BASELINE system as well as gives the best scores on most metrics: BLEU (absolute difference of 0.04), NIST (same score), METEOR (same score), WER (absolute difference of 0.02), PER (same score). Note that the SUM systems fail to outperform any other system on any metric, not even OMET and/or OBLEU as seen on devset.

Once again, our rescoring systems seem to fare better on the testset than on the devset, as observed in 100-best and 250-best lists.

750-BEST LIST

Table 3.37 (a) gives system evaluation scores on oracle-based rescoring of 750-best lists for English–French devset. Similar to the preceding *n*-best lists, the BASELINE system gives slightly better results than all the four rescored systems on the BLEU, WER, PER, and achieves similar NIST scores as the RESCORED$_{BPROD}$ system. The RESCORED$_{MSUM}$ system achieves the best METEOR score (absolute difference of 0.04 over the baseline) and the highest percentage of BLEU-oracles (24 more sentences than the baseline) and METEOR-oracles (7 more sentences than the baseline) in the 1-best position. All other observations such as the relative differences between PROD and SUM systems were similar to that on 100-best, 250-best, and 500-best lists.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**31.05**	**7.48**	60.88	**60.04**	**44.93**	13.85	14.55
RESCORED$_{BPROD}$	30.96	7.47	60.85	60.08	44.99	13.50	14.50
RESCORED$_{MPROD}$	30.96	7.46	60.85	60.07	45.01	13.55	14.65
RESCORED$_{BSUM}$	30.41	7.34	60.84	60.98	45.74	14.10	14.05
RESCORED$_{MSUM}$	30.52	7.35	**60.92**	60.92	45.66	**15.05**	**14.90**
(b) testset							
BASELINE	28.14	**7.01**	**57.77**	62.54	**47.42**	**11.70**	**12.65**
RESCORED$_{BPROD}$	**28.16**	**7.01**	57.76	**62.53**	**47.42**	11.65	12.60
RESCORED$_{MPROD}$	28.15	**7.01**	57.74	**62.53**	47.44	11.65	12.60
RESCORED$_{BSUM}$	27.50	6.84	57.44	63.68	48.43	10.00	10.55
RESCORED$_{MSUM}$	27.60	6.86	57.52	63.57	48.32	10.25	11.15

Table 3.37: Summary of the English→French translation system results for 750-best list: (a) devset and (b) testset

When comparing rescoring of BLEU-oracles (rows 2 and 4 in Table 3.37 (a)) with METEOR-oracles (rows 3 and 5 in Table 3.37 (a)), the PROD rescoring gives similar performance (e.g. 60.85 METEOR points in both RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$) while the SUM rescoring shows a larger degree of variation with the RESCORED$_{MSUM}$ as significantly better than RESCORED$_{BSUM}$ on all metrics.

The results for rescoring 750-best outputs of testset appear in Table 3.37 (b). The rescoring system with the lowest evaluation scores is RESCORED$_{BSUM}$ as per all metrics. In contrast to the devset, RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems show slight

improvement / give at par performance with the baseline system on all metrics.

Moreover, when comparing performance of the rescoring systems on devset versus testset, we find that the testset again gives better performance over the BASELINE system,, especially with respect to PROD systems. For example, the RESCORED$_{BPROD}$ records the same PER score as the BASELINE system on the testset as opposed to 0.06 PER points below baseline on the devset.

Note that as expected, most of these observations followed the same pattern as the preceding *n*-best list sizes on English→French data.

1000-BEST LIST

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**31.05**	**7.48**	**60.88**	**60.04**	**44.93**	12.75	13.20
RESCORED$_{BPROD}$	30.96	7.46	60.84	60.07	44.99	12.50	13.15
RESCORED$_{MPROD}$	30.95	7.46	60.85	60.07	45.00	12.50	13.30
RESCORED$_{BSUM}$	30.32	7.32	60.84	61.15	45.85	**12.95**	**13.35**
RESCORED$_{MSUM}$	30.36	7.32	60.85	61.12	45.80	12.85	**13.35**
(b) testset							
BASELINE	28.14	**7.01**	**57.77**	**62.54**	**47.42**	10.40	**11.25**
RESCORED$_{BPROD}$	**28.16**	**7.01**	57.75	62.56	47.44	**10.50**	**11.25**
RESCORED$_{MPROD}$	**28.16**	**7.01**	57.74	**62.54**	47.45	10.40	**11.25**
RESCORED$_{BSUM}$	27.34	6.81	57.40	63.90	48.59	8.60	9.70
RESCORED$_{MSUM}$	27.40	6.83	57.43	63.80	48.53	8.65	9.90

Table 3.38: Summary of the English→French translation system results for 1000-best list: (a) devset and (b) testset

Table 3.38 (a) gives system evaluation scores on oracle-based rescoring of 1000-best lists for English–French devset. Similar to the preceding *n*-best lists, the BASELINE system gives slightly better results than all the four rescored systems on the BLEU, NIST, METEOR, WER, and PER. The RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems are the second best-performing systems and the best rescoring systems on devset. However, the RESCORED$_{BSUM}$ system achieves the highest percentage of BLEU-oracles and METEOR-oracles in the 1-best position.

The results for rescoring 1000-best outputs of testset appear in Table 3.38 (b). Unlike the devset, the BASELINE system no longer has the best score on any metric except ME-

TEOR and PER. The RESCORED$_{MPROD}$ system performs at the same level as BASELINE system on most metrics: BLEU (absolute difference of 0.02), NIST (same score), and WER (same score), OBLEU (absolute difference of 0.10), and OMET (same score).

All other observations followed the same pattern as the 750-best list. An analysis of the metric score trend with increasing *n*-best list size is made after Table 3.15.

Note that on comparing the BLEU, NIST, METEOR, WER, PER scores of all the four rescoring systems on 750-best list with their corresponding scores on 1000-best list, we find that the accuracy decreases on the SUM systems (RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$) and remain relatively unchanged on the PROD systems (RESC$_{BPROD}$ and RESC$_{MPROD}$). This implies that the difference in performance quality of PROD and SUM rescoring widens as the *n*-best list size increases, with the PROD systems as better than SUM systems.

Note that all these observations followed the same pattern as the preceding *n*-best list sizes on English–French data.

2500-BEST LIST

Table 3.39 (a) gives system evaluation scores on oracle-based rescoring of 2500-best lists for English–French devset. The BASELINE system fails to be outperformed by any other rescoring system. The RESCORED$_{MPROD}$ system is the second-best system across all metrics.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**31.05**	**7.48**	**60.88**	**60.04**	**44.93**	**8.45**	8.95
RESCORED$_{BPROD}$	30.95	7.46	60.83	60.09	45.02	8.40	**9.20**
RESCORED$_{MPROD}$	30.94	7.46	60.84	60.08	45.01	8.40	9.15
RESCORED$_{BSUM}$	29.76	7.24	60.77	61.78	46.32	8.00	8.75
RESCORED$_{MSUM}$	29.47	7.20	60.47	61.98	46.53	6.50	7.90
(b) testset							
BASELINE	28.14	**7.01**	**57.77**	**62.54**	**47.42**	**7.70**	8.70
RESCORED$_{BPROD}$	28.14	7.00	57.74	62.58	47.47	7.60	8.65
RESCORED$_{MPROD}$	**28.16**	7.00	57.75	62.57	47.46	**7.70**	**8.80**
RESCORED$_{BSUM}$	26.86	6.75	57.34	64.36	49.03	5.80	6.55
RESCORED$_{MSUM}$	26.69	6.73	57.16	64.49	49.13	5.10	5.95

Table 3.39: Summary of the English→French translation system results for 2500-best list: (a) devset and (b) testset

Again, the PROD rescoring (2nd and 3rd rows in Table 3.39 (a)) yields better translation results than the SUM rescoring (4th and 5th rows in Table 3.39 (a)) across all metrics (BLEU, NIST, WER, PER, OBLEU, and OMET) including METEOR scores.

The results for rescoring 2500-best outputs of testset appear in Table 3.39 (b). The RESCORED$_{MPROD}$ system gives a slight improvement over BASELINE on BLEU (0.02 points absolute improvement), while the BASELINE system appears to perform the best.

When comparing performance of the rescoring systems on devset versus testset, we find that again the testset shows a greater degree of improvement over the BASELINE system or smaller margin of difference from the BASELINE than the devset.

Note that all these observations followed the same pattern as the preceding *n*-best list sizes on English–French data, even if some metrics record a slightly lower score than the corresponding systems in a smaller *n*-best list.

5000-BEST LIST

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	**31.05**	**7.48**	**60.88**	**60.04**	**44.93**	7.25	7.50
RESCORED$_{BPROD}$	30.93	7.46	60.82	60.10	45.04	**7.40**	**7.90**
RESCORED$_{MPROD}$	30.96	7.46	60.84	60.07	45.00	**7.40**	7.80
RESCORED$_{BSUM}$	29.33	7.16	60.57	62.40	46.87	4.95	5.80
RESCORED$_{MSUM}$	29.18	7.15	60.33	62.36	46.89	4.40	5.45
(b) testset							
BASELINE	**28.14**	**7.01**	**57.77**	**62.54**	**47.42**	5.60	**6.40**
RESCORED$_{BPROD}$	28.10	7.00	57.72	62.60	47.48	5.60	6.30
RESCORED$_{MPROD}$	**28.14**	7.00	57.75	62.58	47.47	**5.65**	**6.40**
RESCORED$_{BSUM}$	26.36	6.67	57.25	65.11	49.71	2.50	3.00
RESCORED$_{MSUM}$	26.42	6.68	57.09	64.88	49.54	3.05	3.95

Table 3.40: Summary of the English→French translation system results for 5000-best list: (a) devset and (b) testset

Table 3.40 (a) and (b) give system evaluation scores on oracle-based rescoring of 5000-best lists for English–French devset and testset, respectively. All PB-SMT system behaviour adhere to the same pattern as that seen in the preceding *n*-best list sizes of English–French data.

All rescored systems once again fail to best the BASELINE system on either devset or testset. Although, the RESCORED$_{MPROD}$ system achieves the same BLEU score as baseline on testset.

As far as the percentage of oracles at first rank is concerned, RESCORED$_{MPROD}$ and RESCORED$_{BPROD}$ systems record slighly higher number of sentences in both OBLEU and OMET scores.

SUMMARY

Having reported on the performance of rescoring systems for individual *n*-best lists, we would now like to comment on any general trends observed in English–French translation systems as a whole. Table 3.41 summarises the performance of our rescoring systems on English–French data by listing the best-performing systems in each of the seven *n*-best lists (rows: 100-best, 250-best, 500-best, 750-best, 1000-best, 2500-best, and 5000-best) for each of the seven evaluation metrics (columns: BLEU, NIST, METEOR, WER, PER, OBLEU, and OMET). The table is divided into two sections: (a) devset and (b) testset. The abbreviations used for each of the five systems are as follows: B (BASELINE), bP (RESCORED$_{BPROD}$), mP (RESCORED$_{MPROD}$), bS (RESCORED$_{BSUM}$), and mS (RESCORED$_{MSUM}$).

It is clearly evident from Table 3.41, that while the BASELINE system remains relatively unbeaten on devset, the rescoring systems easily outperform the baseline on testset, especially for lower *n*-best list sizes.

One or more of the rescoring systems almost always achieve the highest percentage of oracles in the 1-best position (OBLEU, OMET) on both devset and testset. This shows that our rescoring strategies, especially the RESC$_{prod}$ have been successful in their primary aim of moving oracles up the *n*-best lists.

While the metrics may not agree with each other on the best-performing system, a general pattern is followed across increasing *n*-best lists that is consistent for a specific metric. For example, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems are consistently the best-performing systems on testset across all metrics from 100-best to 1000-best lists.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
100-BEST	B	B	bS	B	B	B	mS
250-BEST	B	B	B	B	B	mS	mS
500-BEST	B	B	bS	B	B	mS	bS
750-BEST	B	B	mS	B	B	mS	mS
1000-BEST	B	B	B	B	B	bS	bS, mS
2500-BEST	B	B	B	B	B	B	bP
5000-BEST	B	B	B	B	B	bP, mP	bP
(b) testset							
100-BEST	mP	B, bP, mP	mP	bP, mP	mP	mS	B
250-BEST	bP, mP	B, bP, mP	mP	mP	mP	bP	mP
500-BEST	mP	B, bP, mP	B, bP, mP	mP	bP	bP, mP	mP
750-BEST	bP	B, bP, mP	B	bP, mP	B, bP	B	B
1000-BEST	bP, mP	B, bP, mP	B	B, mP	B	bP	B, bP, mP
2500-BEST	mP	B	B	B	B	B, mP	mP
5000-BEST	B, mP	B	B	B	B	mP	B, mP

Table 3.41: Summary of the best-performing English→French translation systems across all *n*-best lists: (a) devset and (b) testset

Another important observation is that more rescoring systems outperform the baseline (B) on the testset than on the devset. We also note that the SUM systems perform better than PROD systems on devset, while the opposite is true and on a much larger scale on testset. This implies that unlike on French→English, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems are the best rescoring systems.

In addition to identifying the best-performing systems, we also note the general trend of a metric with increasing *n*-best list sizes for each of the five systems. Figure 3.18

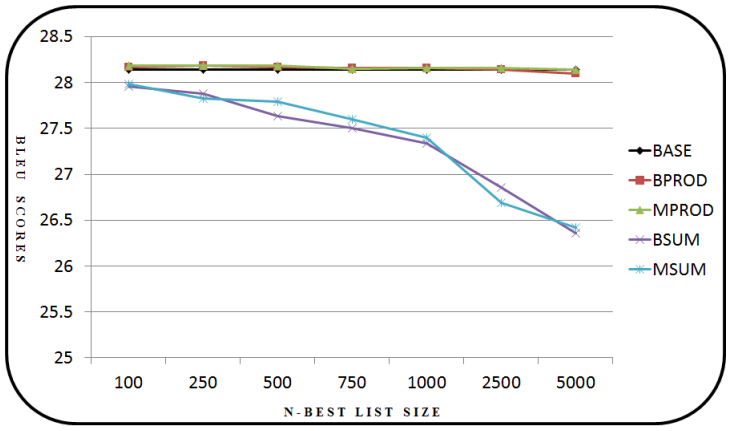


Figure 3.18: Line graph to show the trends of the five PB-SMT systems for BLEU score with increasing *n*-best list sizes: Europarl data, English–French, testset.

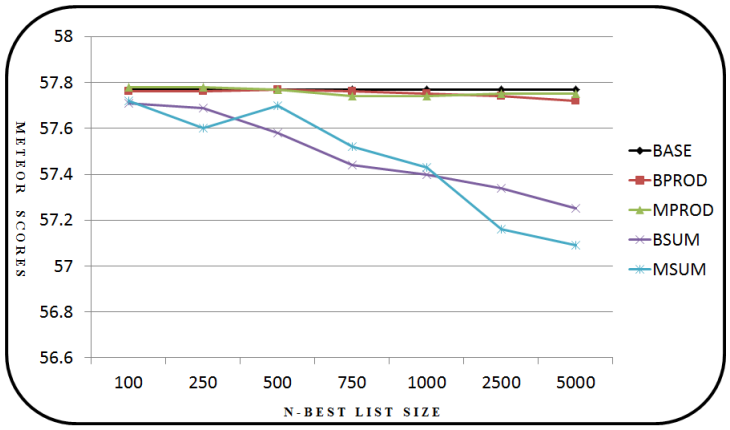


Figure 3.19: Line graph to show the trends of the five PB-SMT systems for METEOR score with increasing *n*-best list sizes: Europarl data, English–French, testset.

shows this phenomenon for the BLEU score on the testset. Both RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems perform at a lower level than the remaining three systems. This difference widens with increasing *n*-best list sizes. Also, the RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ systems give similar performance to the BASELINE system (black line), even outperforming on larger *n*-best lists.

In Figure 3.19, we plot the METEOR performance of the four rescoring systems and the baseline system for each of the seven *n*-best list sizes. Here, unlike other language pairs, the behaviour is similar to what was observed on the BLEU metric. The RESCORED$_{BPROD}$ and RESCORED$_{MPROD}$ give once again similar performance (relatively constant) to the BASELINE and perform at a higher level than the RESCORED$_{BSUM}$ and RESCORED$_{MSUM}$ systems.

Figures 3.20 and 3.21 plot the percentage of oracles with respect to BLEU and METEOR, respectively against *n*-best list sizes for all five systems. Note that the phenomenon of decreasing values with increasing *n*-best lists is seen across all metrics but is particularly prominent in the OBLEU and OMET trends. The number of oracles in the 1-best position severely reduces as we approach 5000-best lists. However, since this trend is uniform across all systems and has been confirmed in other language pairs, we can conclude that the increasing complexity of the *n*-best list search space accounts for the decreasing oracle-ranking accuracy.

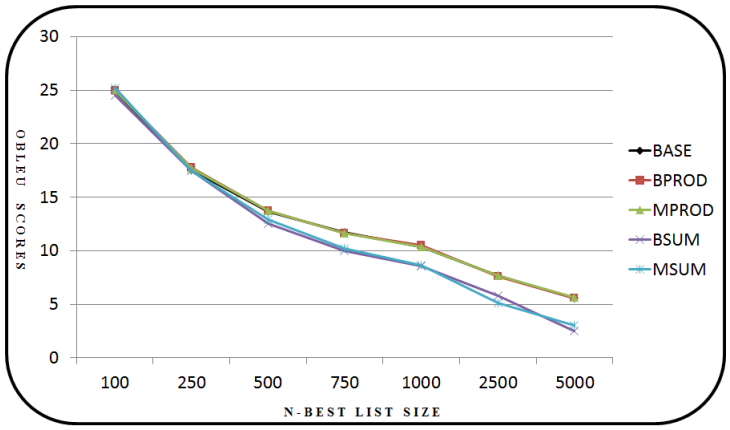


Figure 3.20: Line graph to show the trends of the five PB-SMT systems for OBLEU score with increasing *n*-best list sizes: Europarl data, English–French, testset.

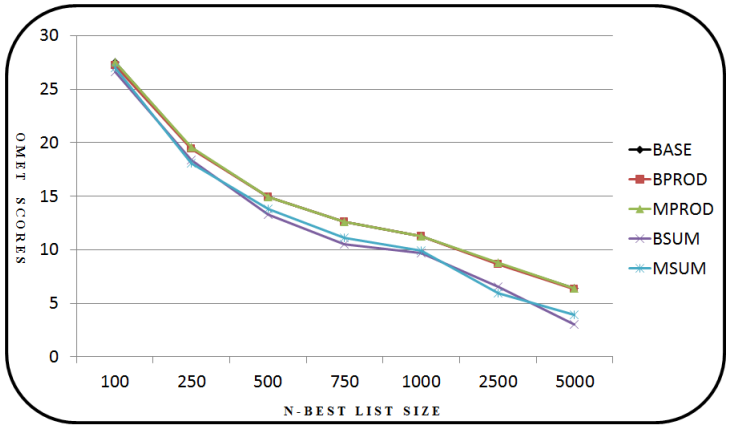


Figure 3.21: Line graph to show the trends of the five PB-SMT systems for OMET score with increasing *n*-best list sizes: Europarl data, English–French, testset.

To conclude, the French→English datasets demonstrated different behaviour from other language pairs, especially on METEOR scores. The BASELINE system was the best-performing system on larger *n*-best list sizes and the RESCORED$_{MPROD}$ system outperformed the RESCORED$_{BSUM}$ system on the testset. This anomalous behaviour is addressed in Section 3.11 when we contrast performances across language pairs.

3.7 Per feature Comparison

Moving on from the system-level evaluation, we now perform a deeper analysis by looking at feature values of the oracle translation and the 1-best translation for each of the sentences in a dataset. Here onwards, all experiments have been performed on French → English data only. Figure 3.22 analyses which features (outlined in Table 3.5) 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.

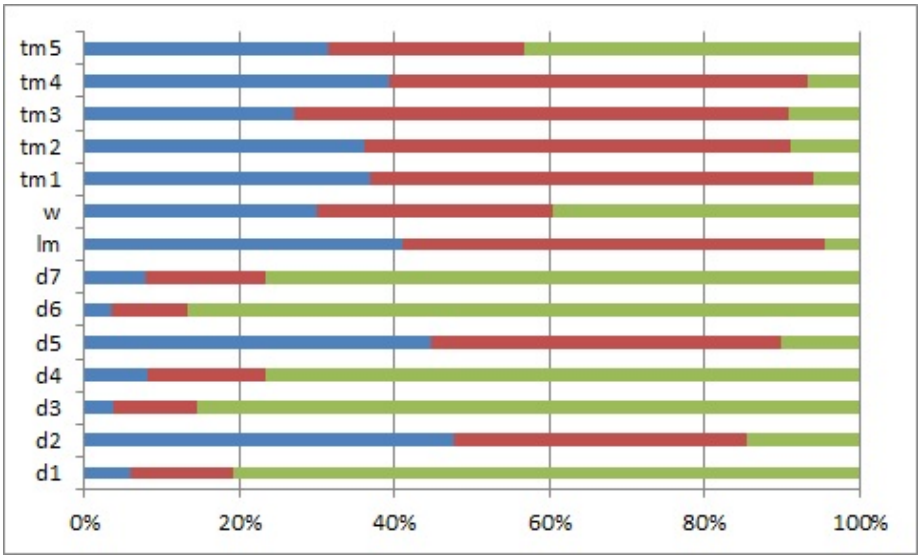


Figure 3.22: Results for a 1000-best list of filtered oracles: For how many sentences (% given on the X-axis) does a baseline feature (given on the Y-axis) favour the oracle translation (blue bar) over the 1-best translation (red bar). The green bar (third band in each bar) denotes percentage of sentences having the same value for its oracle and 1-best hypothesis.

The oracles seem 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 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.

3.8 Movement in Rankings

Table 3.42 shows the number (n) of sentences (out of 2000) which were moved up (↑), moved up to a position in the top-5, moved down (↓), 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$_{sum}$ is more effective in promoting oracles than RESC$_{prod}$. Perhaps it is no surprise that the RESC$_{sum}$ 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.

	(a) *DEVSET*						**(b)** *TESTSET*					
SYS	**n**↑	**p**↑	**n**$_5$ ↑	**n**↓	**p**↓	**n**$_5$ ↓	**n**↑	**p**↑	**n**$_5$ ↑	**n**↓	**p**↓	**n**$_5$ ↓
	rescored on 100-best list											
R$_{sum}$	637	24	267	776	23	278	627	24	260	794	22	278
R$_{prod}$	590	10	94	534	11	89	559	10	93	587	12	93
	rescored on 500-best list											
R$_{sum}$	840	122	212	875	121	185	869	129	277	850	111	199
R$_{prod}$	856	54	75	722	74	64	831	55	84	739	69	80
	rescored on 1000-best list											
R$_{sum}$	908	237	180	878	248	147	933	247	198	870	215	176
R$_{prod}$	918	114	63	758	163	51	895	117	73	785	148	66

Table 3.42: Movement of oracles in *n*-bests of (a) development set and (b) test set after rescoring the baseline system with weights learned from RESC$_{sum}$ and RESC$_{prod}$: how many & how much?

3.9 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 (or very similar) score to sentences differing in the order or lexical choice of one or two words only.

Revisiting the *n*-best list given in Figure 3.3, 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.

***N*-BEST**	**DIFF**	**DIVERSE**	**ACCEPTED**
100	62.10%	48.55%	27.10%
500	55.50%	57.75%	30.50%
1000	54.05%	61.40%	32.80%

Table 3.43: 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 often 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 3.43 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.

3.10 Top 5

We also perform a Top 5 BLEU-oracle evaluation (shown in Table 3.45). The difference between the evaluations in Tables 3.44 and 3.45 is that the latter 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 3.44 are present in the top 1 position of sentences used in Table 3.45. 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

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

Table 3.44: Summary of the French→English 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 with respect to BLEU metric.

translations together (Koehn and Haddow, 2009). We only use 5 in our evaluation.

The baseline system performance i.e. the standard evaluation (top-1) is shown in Table 3.44. 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 observe that overall the 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 3.44) to top-5 (Table 3.45) is more obvious in the rescored systems RESC$_{sum}$ and RESC$_{prod}$ compared to the baseline. This gives weight to our observation that the oracles have moved up, just not to the top position.

	(a) *DEVSET*			(b) *TESTSET*		
SYSTEM	**BLEU**	**MET**	**ORC**	**BLEU**	**MET**	**ORC**
rescored on 100-best list						
BASE$_5$	32.83	61.95	45.95	33.17	62.34	45.05
RESC$_{sum5}$	32.72	**62.04**	45.75	33.08	**62.40**	45.65
RESC$_{prod5}$	32.78	61.92	45.80	33.16	62.34	45.00
ORACLE	34.90	63.65	100	35.26	64.01	100
rescored on 500-best list						
BASE$_5$	32.83	61.95	24.45	33.17	62.34	25.50
RESC$_{sum5}$	32.49	**62.31**	27.20	32.95	**62.71**	27.90
RESC$_{prod5}$	32.74	61.89	24.75	33.12	62.30	25.80
ORACLE	36.45	64.70	100	36.80	65.12	100
rescored on 1000-best list						
BASE$_5$	32.83	61.95	18.80	33.17	62.34	19.65
RESC$_{sum5}$	32.45	**62.27**	20.90	32.85	**62.68**	21.85
RESC$_{prod5}$	32.70	61.88	18.60	33.13	62.30	19.85
ORACLE	37.05	65.14	100	37.50	65.65	100

Table 3.45: Top5 Eval: Summary of the French→English 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 with respect to BLEU metric.

3.11 Comparison and Contrastive Analyses

While we have previously summarised rescoring system performance trends in each language pair,[10] we have not yet contrasted trends across the language pairs. In this chapter on exploring suitable oracle reranking algorithms, we rescored *n*-best lists using two rescoring methods (Section 3.4.2) as follows:

- RESC$_{sum}$: The feature weights estimated via MERT (Minimum Error Rate Training, Och (2003)) are recomputed using the difference between mean feature values of oracle and 1-best sentences as defined in Equation (3.4);
- RESC$_{prod}$: The feature weights estimated via MERT are recomputed using the ratio of mean feature values of oracle and 1-best sentences as defined in Equation (3.5).

We identified the oracle translations using two metrics, namely sentence-level BLEU and sentence-level METEOR. Thus each of the two rescoring methods can be classified into two subtypes. This gives rise to four rescored systems in addition to a fifth (baseline system, i.e. a translation system with no rescoring), as follows:

[10] cf. Summary subsections under each of the Sections 3.6.1 through 3.6.4.

- BASELINE [B]: System using weights computed using MERT with no rescoring
- RESCORED$_{BPROD}$ [bP]: System in which the MERT weights are recomputed on the RESC$_{prod}$ strategy based on Oracles with respect to sentence-level BLEU score
- RESCORED$_{MPROD}$ [mP]: System in which the MERT weights are recomputed on the RESC$_{prod}$ strategy based on Oracles with respect to sentence-level METEOR score
- RESCORED$_{BSUM}$ [bS]: System in which the MERT weights are recomputed on the RESC$_{sum}$ strategy based on Oracles with respect to sentence-level BLEU score
- RESCORED$_{MSUM}$ [mS]: System in which the MERT weights are recomputed on the RESC$_{sum}$ strategy based on Oracles with respect to sentence-level METEOR score

We conducted experiments on the French→English language direction to maintain the continuity with experiments in Chapter 2 on treebank-based phrase extraction.[11] In order to test the language independence of our rescoring methods, we experimented on two additional languages, German and Spanish. We also experimented in the reverse direction, English→French. Thus four language directions were explored as follows:

- FR→EN: Translation system translating from French into English
- DE→EN: Translation system translating from German into English
- ES→EN: Translation system translation from Spanish into English
- EN→FR: Translation system translating from English into French

This helps us conduct contrastive analysis in two ways: (a) comparison of rescoring *n*-best lists when translating from English versus translating into English, and (b) comparison of rescoring *n*-best lists when translating from different languages (French, German, Spanish) into the same language (English) (Tables 3.46 and 3.47).

[11] Note that we have scaled up from 100,000 sentence pairs to approximately 1 million sentence pairs from Europarl as we no longer have any syntactic parser contraints.

There is no consensus in reranking literature on what *n*-best list size of the translation hypotheses should be used. In order to test the optimal *n*-best list size for our rescoring methods (Table 3.48), all four language directions have each of the five translation systems (baseline and 4 rescored systems) rescored on seven *n*-best list sizes as follows:

- 100-BEST: Rescoring MT systems where each sentence has at most 100 alternate translations
- 250-BEST: Rescoring MT systems where each sentence has at most 250 alternate translations
- 500-BEST: Rescoring MT systems where each sentence has at most 500 alternate translations
- 750-BEST: Rescoring MT systems where each sentence has at most 750 alternate translations
- 1000-BEST: Rescoring MT systems where each sentence has at most 1000 alternate translations
- 2500-BEST: Rescoring MT systems where each sentence has at most 2500 alternate translations
- 5000-BEST: Rescoring MT systems where each sentence has at most 5000 alternate translations

Note that all our experiments were performed on two translation datasets: (a) devset and (b) testset. The parameters (feature weights) for rescoring the *n*-best lists were trained on the devset and tested on the testset. This implies that in the course of our extensive multi-dimensional experiments, we created a total of 140 different MT systems. For each of the 4 language pairs, we created 35 translation systems (5 types of MT systems [baseline and 4 rescored systems] for each of the 7 *n*-best list sizes). Additionally, we evaluated the system performances on 7 different evaluation metrics: BLEU, NIST, METEOR, WER, PER, OBLEU, and OMET (previously described in Section 3.7). We

computed oracles using two different metrics: sentence-level BLEU and sentence-level METEOR. Hence, we have focussed on the contrastive analysis of translations as per these two metrics. This also helped us compare the two metrics across language pairs. The purpose of this section on contrastive analyses is to try and draw some discernable patterns across the 140 MT systems.

LANG PAIR	100	250	500	750	1000	2500	5000
(a) devset							
EN→FR	B	B	B	B	B	B	B
FR→EN	B	B	B	B	B	B	B
DE→EN	B	B	B	B	B	B	B
ES→EN	B	B	B	B	B	B	B
(b) testset							
EN→FR	mP	bP, mP	mP	bP	bP, mP	mP	B, mP
FR→EN	B	B	B	B, mP	bP	B	B
DE→EN	mP	bP	mP	mP	mP	mP	mP
ES→EN	mS	bP	B, bP, mP	B	B	B	B

Table 3.46: Summary of the best-performing translation systems across all *n*-best lists and all language directions as per the BLEU evaluation metric: (a) devset and (b) testset

Table 3.46 summarises the best-performing systems across all language directions (rows: English →French, French→English, German→English, and Spanish→English) in each of the seven *n*-best list sizes (columns: 100-best, 250-best, 500-best, 750-best, 1000-best, 2500-best, and 5000-best) for the BLEU evaluation metric. The table is divided into two sections: (a) devset and (b) testset. The abbreviations used for each of the five systems are as follows: B (BASELINE), bP (RESCORED$_{BPROD}$), mP (RESCORED$_{MPROD}$), bS (RESCORED$_{BSUM}$), and mS (RESCORED$_{MSUM}$). We have made the following observations.

- The BASELINE system is the best-performing system across all *n*-best list sizes on the devset as per the BLEU evaluation metric because all the rescoring systems either underperformed or gave a similar performance (including statistically insignificant results) to the baseline. We hoped to see similar patterns across all four language directions and although none of the rescored systems outperformed the baseline, all four language directions show the same pattern.

- On the testset, in contrast to the above observation, one or more of our rescored systems is the best-performing system 19 out of 28 times (68%). Note that in any evaluation campaign it is on the testset and not on the devset that competing system performances are compared. In cases where one of the rescored MT systems gives a similar performance or statistically insignificant improvement to the BASELINE system, multiple systems are reported in the table.

- The RESC$_{prod}$ method is the dominant rescoring strategy across all language pairs and *n*-best list sizes: 18 out of 28 times (64%). A possible reason is that the RESC$_{sum}$ rescoring method is similar to that of a perceptron and most likely requires multiple iterations to stabilise. All our rescoring methods were computed in just a single iteration post-MERT framework. Note that this observation pertains to the BLEU evaluation metric alone and may not follow the pattern shown by other metrics, especially METEOR (addressed below in Table 3.47).

- There is a distinct mismatch in performance betwen the devset and testset as reported in the first two observations. As the same set of feature weights were used to rescore both datasets, this may just be down to the variable nature of the data itself and deficiencies in the BLEU metric regarding non *n*-gram-based matching between the system translation and the reference translation (Ye et al., 2007).

- The recommendation for both EN→FR and DE→EN language directions is to always use the RESCORED$_{MPROD}$ MT system as they have been proven the most effective. Each of the five competing systems only differ in the feature weights which lead to a different ranking of the *n*-best lists producing a different set of translations and hence a different evaluation score. A closer inspection of these parameters revealed that the language model feature weight was significantly lower for RESC$_{sum}$ systems. This is the most likely reason for the distinct lower performance of RESCORED$_{MSUM}$ and RESCORED$_{BSUM}$ systems. While there were variations in the remaining 13 features[12] as well, none were as diverse as the language model

[12] These features are described in Table 3.5 in Section 3.3.1.

feature.

- There are anomalous cases in both FR→EN and ES→EN where the BASELINE system starts outperforming the rescored systems as the *n*-best list size increases. We were not able to find a definite cause for this and further experimentation is required, but it may be down to the fact that quite simply, smaller *n*-best list sizes suit these language directions better. We will discuss this in more detail in Table 3.48 below.

LANG PAIR	100	250	500	750	1000	2500	5000
(a) devset							
EN→FR	bS	B	bS	mS	B	B	B
FR→EN	B	B	B	mS	mS	mS	B
DE→EN	bS	bS	bS	bS	bS	bS	bS
ES→EN	bS	bS	bS	bS	bS	bS	bS, mS
(b) testset							
EN→FR	mP	mP	B, bP, mP	B	B	B	B
FR→EN	B	B	B	mS	mS	mS	B
DE→EN	bS	bS	bS	bS	bS	bS	bS
ES→EN	bS	bS	bS	bS	bS	bS	bS

Table 3.47: Summary of the best-performing translation systems across all *n*-best lists and all language directions as per the METEOR evaluation metric: (a) devset and (b) testset

Table 3.47 summarises the best-performing systems across all language pairs (English →French, French→English, German→English, and Spanish→English) for the METEOR evaluation metric. We do this because we have observed in individual language pairs that the BLEU and METEOR metrics do not agree with each other possibly due to lack of recall in *n*-gram-based BLEU while the METEOR considers both precision and recall, as well as language-dependent processing.

We have made the following observations.

- On both the devset and testset, one or more of our rescored systems outperformed the baseline 40 out of 56 times (71% times). Compared to the BLEU metric, this percentage of systems is similar (68%) although the devset did not figure in there.

- All three FR→EN, DE→EN, and ES→EN present a similar pattern individually across both devset and testset. This is as expected. The EN→FR system demonstrated an anomaly on the testset by having the RESCORED$_{MPROD}$ system outperform all other systems barring the BASELINE system. Perhaps this is mostly because it is translation into French and METEOR scores are language-dependent.

- The RESC$_{sum}$ method is the dominant rescoring strategy across all language pairs and *n*-best list sizes: 37 out of 56 times (66%). Note that this observation contrasts with the BLEU metric above and we speculate that the technical differences between METEOR and BLEU render one to favour one type of rescoring over other. Any analysis on a bias will require further experimentation and comparison with oracles generated by more metrics than sentence-level BLEU and sentence-level METEOR.

- The recommendation for the ES→EN direction is to always use the RESCORED$_{BSUM}$ MT system as they have proven the most effective. As before, each of the five competing systems only differ in the feature weights which lead to a different ranking of the *n*-best lists producing a different set of translations and hence a different evaluation score. We speculate that a combination of the five translation model features is the most likely cause as they were observed to be the most impacting on inspecting the MERT weights.

Table 3.48 shows which *n*-best list size is the top-performing system in each language pair across all the evaluation metrics. We have made the following observations.

- 5000-best list sizes lead to the best-performing system the most number of times across all language directions and all metrics.

- Despite the aforementioned observation, there are cases when a smaller *n*-best list size suffices. Discernible patterns are visible when considering a particular metric (any one column) in isolation. For example, the METEOR metric on the testset

LANG PAIR	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
EN→FR	n/a	n/a	100	n/a	n/a	750	100
FR→EN	n/a	5000	2500	5000	5000	250	500
DE→EN	n/a	5000	1000	5000	5000	100	100
ES→EN	n/a	n/a	5000	n/a	750	5000	2500
(b) testset							
EN→FR	250	n/a	100	250	100	100	2500
FR→EN	n/a	5000	1000	5000	5000	5000	100
DE→EN	5000	5000	500	5000	5000	100	100
ES→EN	100	n/a	500	250	n/a	n/a	500

Table 3.48: Summary of the best-performing *n*-best list across all language pairs and all the evaluation metrics: (a) devset and (b) testset; A n/a implies none of the rescoring methods outperformed the BASELINE system thus nullifying *n-best list*

especially favours a smaller size *n*-best list. This is most likely because with increasing *n*-best list sizes, the complexity in terms of the search space of the number of positions to move up an oracle increases.

- Our recommendation for the EN→FR system especially is to use smaller than 500-best lists because they gave the best performance. It seems to be the case that translating into English requires a larger *n*-best list size than while translating from English.

3.12 Discussion

3.12.1 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 rescoring. 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 implement 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 3.22). However, we believe that there is scope for improvement on the baseline features (used in decoding) before extracting more complex features for reranking.

3.12.2 Role of oracles in boosting translation accuracy

We believe oracle-based training to be a viable method. In the next chapter, we explore additional 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 have used here sentence-level BLEU as opposed to system-level BLEU as used in MERT for oracle identification. We have also demonstrated the effectiveness using sentence-level METEOR.

3.13 Conclusion

We analyze the relative position of oracle translations in the *n*-best list of translation hypotheses to help boost oracles up a PB-SMT system. We propose two novel simple rescoring strategies (RESC$_{sum}$ and RESC$_{prod}$) which differ in only the manner they update the feature weights. In general, the improvements provided by oracle-based training of the *n*-best lists is dependent on the size of *n* and the type of translations produced in the *n*-best list. For example, the RESC$_{sum}$ strategy is dominant on the METEOR evaluation metric while the RESC$_{prod}$ strategy favours the BLEU evaluation metric. Translating into English seems to require a larger *n*-best list size than translating from English. Part of this work (French $\rightarrow$ English rescoring on BLEU-oracles) was published in Srivastava et al. (2011). To conclude, oracles have much to contribute to the ranking of better translations,

feature selection, and reducing the model errors. We will conduct these experiments with additional syntactic features in a reranking framework in Chapter 4.

3.14 Summary

In this chapter we analyse model errors, investigated the parametric differences between the 1-best and the oracle translation and attempted to try and close this gap by proposing two rescoring strategies to push the oracle up the *n*-best list: RESC$_{sum}$ and RESC$_{prod}$. We generated 140 MT systems and we observed modest improvements in METEOR scores (Banerjee and Lavie, 2005) over the baseline SMT system trained on French $\rightarrow$ English, German $\rightarrow$ English, Spanish $\rightarrow$ English, and English $\rightarrow$ French Europarl corpora (Koehn, 2005). We also reported on the patterns observed across the four language directions and the seven *n*-best list sizes. We presented a detailed analysis of the oracle rankings to determine the source of model errors, which in turn has the potential to improve the performance of the baseline (STR) system as well as syntax-aware models. In Chapter 4 we will incorporate our methods in mainstream reranking paradigms with additional features.

Chapter 4

Feature-based Sentence Reranking

Typically, an SMT system undergoes a second pass decoding, wherein a number of sophisticated features like higher-gram and syntactic language model scores, posterior probabilities, etc. are used to rescore the *n*-best list of translations in a process called reranking. This is done post-decoding (as shown by the black shaded box in Figure 4.1) and can therefore include a number of translation model and language model features which would otherwise massively increase the decoding complexity.

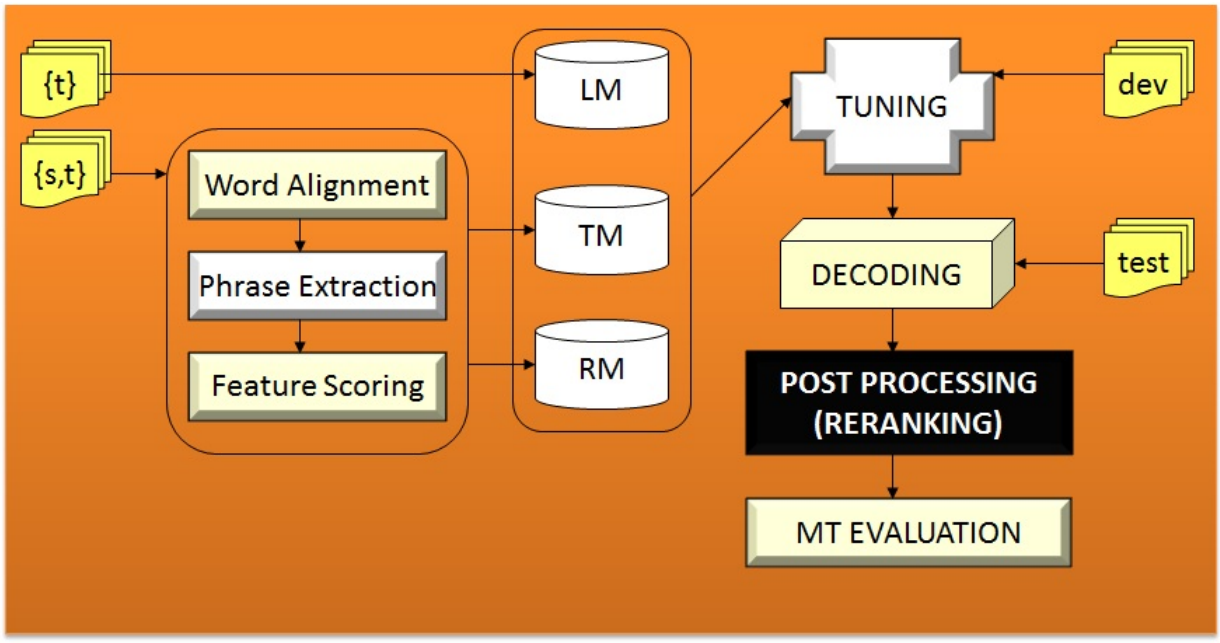


Figure 4.1: Schematic diagram of the modules in a Phrase-based Statistical Machine Translation System: *Reranking*.

The reranking task in Machine Translation can be defined as pushing oracles up the

n-best list. Having analysed the existence of model errors and rescored *n*-best lists on decoding features using oracle-based training (OBT) in the previous chapter, here we extend our approach to a reranking framework using more fine-grained features (described in Section 4.4). We also compare our approach to a vanilla implementation of using uniform weights for the new features as well as to a baseline system of using MERT-optimized weights for the new features. Finally, we discuss the possibility of using oracle-based reranking in improving syntax-aware MT systems from Chapter 2.

The purpose of this chapter is two-fold: to evaluate the oracle-based analytical algorithm from Chapter 3 on reranking features (i.e. those not used in decoding). The second objective is to analyse how reranking in general helps minimize model errors while maintaining a balance at the double-edged sword of the model overfitting (too many features) and model underfitting (too few features) in MT.

4.1 Reranking *n*-best Lists in PB-SMT

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 like BLEU (Papineni et al., 2002), METEOR (Banerjee and Lavie, 2005), etc.) occur much lower in the list. A baseline system (first-pass decoding) is thus reranked (second-pass decoding) by extracting additional features from the *n*-best candidates for each sentence and rescoring them. Figure 4.2 shows a typical *n*-best list with numerous repetitions in the hypotheses that differ only in the internal phrase segmentation. As most of our reranking features (monolingual or bag-of-words in case of bilingual) would generate the same result for the same surface-level hypothesis, we work with distinct *n*-best lists in this chapter as opposed to regular *n*-best lists in Chapter 3.

Note that we focus on *n*-best list reranking, i.e. the translation hypothesis are represented as a sorted list of sentences. Another popular format is lattice-based rescoring (Li and Khudanpur, 2009), wherein the translation hypothesis are represented as a con-

Rank	Sentence	Decoder Cost
2:0	any other approach would lead to a major democratic deficit .	-7.17474
2:1	any other approach would be a major democratic deficit .	-7.19585
2:2	any other approach would lead to a major democratic deficit .	-7.20819
2:3	any other approach would lead to a major democratic deficit .	-7.37001
2:4	any other approach would be a major democratic deficit .	-7.39112
2:5	any other approach would lead to a major democratic deficit .	-7.40346
2:6	any other approach would be a major democratic deficit .	-7.42145
2:7	any other approach would lead to a major democratic deficit .	-7.42274
2:8	any other approach would lead to a major democratic deficit .	-7.44014
2:9	any other approach would be a major democratic deficit .	-7.44384
2:10	any other way of doing this would lead to a major democratic deficit .	-7.45051
2:11	any other approach would lead to a major democratic deficit .	-7.45619
2:12	any other way of doing this would be a major democratic deficit .	-7.45701
2:13	any other approach would be a major democratic deficit .	-7.46125
2:14	any other approach would lead to a major democratic deficit .	-7.47359
2:15	any other way of doing this would lead to a major democratic deficit .	-7.48396
2:16	any other approach would have a major democratic deficit .	-7.53336
2:17	any other approach would entail a major democratic deficit .	-7.54438
2:18	any other approach would be a major democratic deficit .	-7.61672
2:19	any other approach would lead to a major democratic deficit .	-7.61801
2:20	any other approach would result in a major democratic deficit .	-7.62037
2:21	any other approach would lead to a major democratic deficit .	-7.63541
2:22	any other approach would be a major democratic deficit .	-7.63911
2:23	any other way of doing this would lead to a major democratic deficit .	-7.64577
2:24	any other approach would lead to a major democratic deficit .	-7.65145
2:25	any other way of doing this would be a major democratic deficit .	-7.65228
2:26	any other approach would be a major democratic deficit .	-7.65652
2:27	**any other approach would mean a major democratic deficit .**	-7.65906
2:28	any other approach would lead to a major democratic deficit .	-7.66886
2:29	any other approach would be a major democratic deficit .	-7.66945

Figure 4.2: Sample (3rd sentence in testset) from an *n*-best list of 30 translation candidates for the input sentence *toute autre façon de faire entraînerait un déficit démocratique majeur.*, whose reference translation is: *any other procedure would mean a huge democratic deficit.*

Rank	Sentence	Decoder Cost
2:0	any other approach would lead to a major democratic deficit .	-7.17474
2:1	any other approach would be a major democratic deficit .	-7.19585
2:10	any other way of doing this would lead to a major democratic deficit .	-7.45051
2:12	any other way of doing this would be a major democratic deficit .	-7.45701
2:16	any other approach would have a major democratic deficit .	-7.53336
2:17	any other approach would entail a major democratic deficit .	-7.54438
2:20	any other approach would result in a major democratic deficit .	-7.62037
2:27	**any other approach would mean a major democratic deficit .**	-7.65906

Figure 4.3: Sample (3rd sentence in testset) from an *n*-best list of 30 translation candidates after duplicate candidates have been filtered out (distinct). The input sentence is: *toute autre façon de faire entraînerait un déficit démocratique majeur.*, whose reference translation is: *any other procedure would mean a huge democratic deficit.*

nected graph. While lattices are more compact and can store a relatively larger number of candidates, they have the distinct disadvantage of incurring complexity costs in feature extraction (especially for global features dependent on the entire surface-level sentence).

The main objective of the reranking approach to MT is to obtain a better translation than the one initially generated by the decoder. Most approaches in the literature (reviewed in Section 4.2) extract a host of fine-grained and coarse features from the *n*-best lists and rescore them discriminatively. A key research question in this domain is which features benefit reranking of MT outputs.

4.2 Mainstream Approaches to Reranking

A number of strategies have been suggested to minimize the low ranking of higher quality translation candidates in the *n*-best list. These differ mainly in the type of features used for reranking and the training algorithm used to determine the weights needed to combine these features. Note that the parameter estimation methods are the same as reviewed in Chapter 3 for determining the weights of features used in decoding. The main difference is in the sheer number of features trained at the reranking stage in contrast to a limited number of features (about 15 to 20) exploited in the decoding stage.

Och et al. (2004) employed nearly 450 syntactic features extracted from tagged and parsed *n*-best lists in a log-linear model optimized on the BLEU score (using MERT) to

rerank translation candidates.

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).

While the afore-mentioned approaches learned features from *n*-best lists of size up to 1000 Yamada and Muslea (2009) trained a perceptron-based classifier on millions of features extracted from *n*-best lists of size 200 of the entire training set for reranking. Note that while both MERT and the perceptron-like algorithms use BLEU as the objective function for optimization, Yamada and Muslea (2009) compute BLEU on a sentence level rather than corpus level.

Hasan et al. (2007) observed that even after the reference translations were included in the *n*-best list, only 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.

Olteanu et al. (2006) use a host of complex language models in reranking since they cannot be used to score partial hypotheses while decoding. In addition to standard language models, they also include binary features based on presence of *n*-grams of individual hypothesis in the Gigaword corpus.[1] They also used the log probability of parse tree extracted from the Charniak parser. The novel contribution of this work is that voting might help with minimizing the overfitting by combining output of multiple local maxima. Instead of simply using the weights assigned by MERT for the 21 reranking features, a set of 4-10 distinct lambda weight vectors were generated. Each vector picks a different hypothesis for each sentence's *n*-best. The best hypothesis is computed by using a voting mechanism (incurring low-cost).

To summarize, the reranking task involves improving an existing ranking of candidates that were created using local features in an underlying generative model (used in MERT and decoding). The general approach is to extract global properties and additional features from the *n*-best candidates in order to train a discriminative reranking model

[1] `https://catalog.ldc.upenn.edu/LDC2005T12`

(Shen et al., 2004).

An important issue in this area is feature selection: What kind of features or how many features are useful for improving the system performance significantly? We use some of the best proven features in our reranking model. We also conduct an oracle-based analysis of these features similar to that carried out in the previous chapter on the MERT features.

4.3 Baseline System: Data & Tools

LANGUAGE PAIR	TRAINING	DEVELOPMENT	TESTING
French→English	1,050,398	2,000	2,000
German→English	1,118,399	2,000	2,000
Spanish→English	1,083,773	2,000	2,000

Table 4.1: Statistics of corpora used in reranking experiments

All our reranking experiments are performed on the French–English, German–English, and Spanish–English WMT 2009 datasets as used in our oracle-based system diagnosis in Chapter 3. The set of parallel sentences for all our experiments is extracted from the WMT 2009[2] Europarl (Koehn, 2005) training dataset for all three language pairs (Europarl v3) after filtering out sentences longer than 40 words. An additional 2,000 sentences each are taken for development (test2006 dataset) and testing (test2008 dataset). This is summarised in Table 4.1.

We train a 5-gram language model using SRILM (Stolcke, 2002)[3] with Kneser-Ney smoothing (Kneser and Ney, 1995). We train the translation model using GIZA++ (Och and Ney, 2003)[4] for word alignment in both directions followed by phrase-pair extraction using the 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 the previous and next phrases.

[2] http://www.statmt.org/wmt09/
[3] http://www-speech.sri.com/projects/srilm/
[4] http://code.google.com/p/giza-pp/

LABEL	DESCRIPTION
d1	Distortion: distance-based reordering
d2	Distortion: monotone previous
d3	Distortion: swap previous
d4	Distortion: discontinuous previous
d5	Distortion: monotone following
d6	Distortion: swap following
d7	Distortion: discontinuous following
lm	Language Model feature
w	Word penalty feature
tm1	Translation: Phrase Translation (s | t)
tm2	Translation: Lexical Weighting (s | t)
tm3	Translation: Phrase Translation (t | s)
tm4	Translation: Lexical Weighting (t | s)
tm5	Translation: Phrase penalty feature

Table 4.2: Features used in the Moses PB-SMT Decoder

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	32.17	7.70	61.34	57.10	40.96	12.55	13.75
ORACLE$_{BLEU}$	37.62	8.42	65.56	52.64	36.57	100.00	52.90
ORACLE$_{METEOR}$	36.43	8.30	65.59	53.35	37.09	52.90	100.00
(b) testset							
BASELINE	32.47	7.81	61.80	56.43	40.34	12.75	12.70
ORACLE$_{BLEU}$	38.15	8.53	65.98	51.90	36.13	100.00	42.50
ORACLE$_{METEOR}$	37.01	8.43	66.07	52.47	36.62	42.50	100.00

Table 4.3: 1-best and Oracle-best systems for 100-best distinct *n*-best lists on French→English: (a) devset and (b) testset

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 (see Table 4.2), 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 feature applies uniformly to all the candidate translations of a sentence, and is therefore dropped from consideration in our experiments.

Tables 4.3 through 4.5 show the results for the upper-bound of each system (on both devset and testset), that is when all the oracles (selected using sentence-level BLEU and sentence-level METEOR) are placed at the top of *n*-best lists. The systems ORACLE$_{BLEU}$

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	26.93	7.00	57.01	65.52	44.89	4.85	5.40
ORACLE$_{BLEU}$	33.66	7.93	62.56	59.61	38.73	100.00	36.90
ORACLE$_{METEOR}$	31.92	7.77	62.94	60.83	39.54	36.90	100.00
(b) testset							
BASELINE	27.02	7.01	57.11	65.25	45.01	5.05	5.10
ORACLE$_{BLEU}$	34.04	7.96	62.79	59.20	38.60	100.00	38.35
ORACLE$_{METEOR}$	32.26	7.80	63.15	60.37	39.51	38.35	100.00

Table 4.4: 1-best and Oracle-best systems for 100-best distinct *n*-best lists on German→English: (a) devset and (b) testset

and ORACLE$_{METEOR}$ consistently outperform the baseline system by a significant difference of at least 5 BLEU points.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
(a) devset							
BASELINE	32.98	7.80	61.99	56.50	40.68	10.35	11.20
ORACLE$_{BLEU}$	38.65	8.56	66.43	51.73	35.95	100.00	50.30
ORACLE$_{METEOR}$	37.41	8.44	66.53	52.38	36.44	50.30	100.00
(b) testset							
BASELINE	32.88	7.88	61.95	56.03	40.12	11.30	12.20
ORACLE$_{BLEU}$	38.72	8.65	66.43	51.17	35.46	100.00	50.20
ORACLE$_{METEOR}$	37.50	8.55	66.54	51.86	36.00	50.20	100.00

Table 4.5: 1-best and oracle-best systems for 100-best distinct *n*-best lists on Spanish→English: (a) devset and (b) testset

4.4 Reranking Features

In this section we briefly describe the features employed in our reranking framework. We use distinct *n*-best list because the 13 features give the same value if the surface string is identical. We have used nearly a dozen additional features in our reranking framework to enable discrimination between good and bad translations as well as pushing of oracles up the *n*-best lists. These features were selected mainly due to their proven usefulness in the literature. Moreover, they have been used in MATREX, our in-house machine translation system presented at WMT shared tasks (Du et al., 2009; Penkale et al., 2010).

In this section, we describe these features (*at-a-glance* list in Table 4.6) modeling

translations with the help of a worked example and test their advantage over the baseline system via system-level automatic MT evaluation metrics. We also evaluate their prediction power by comparing the individual values for the oracle and the 1-best candidates, as computed in Chapter 3.

LABEL	DESCRIPTION
lm3	Language Model: 3-gram
ppl3	Language Model Perplexity: 3-gram
lm4	Language Model: 4-gram
ppl4	Language Model Perplexity: 4-gram
lm5	Language Model: 5-gram
ppl5	Language Model Perplexity: 5-gram
lm6	Language Model: 6-gram
ppl6	Language Model Perplexity: 6-gram
lm7	Language Model: 7-gram
ppl7	Language Model Perplexity: 7-gram
plm3	Part-of-Speech Language Model: 3-gram
plm4	Part-of-Speech Language Model: 4-gram
plm5	Part-of-Speech Language Model: 5-gram
mbr	Cost: Minimum Bayes Risk Score
np	Posterior Probability: *n*-gram
len	Posterior Probability: Sentence Length
lenr	Source–Target Sentence Length Ratio feature
ibm	Cost: IBM Model 1 Score (t | s)
ibm^{-1}	Cost: Inverse IBM Model 1 Score (s | t)

Table 4.6: Features used in the Reranker after decoding

4.4.1 Language Models

The aim of language model feature is to measure the fluency of a translation. Our baseline system uses a single language model feature (*lm* in Table 4.2), namely 5-gram language model score estimated on the target language training data. We augment this with four *n*-gram (3-gram, 4-gram, 6-gram, and 7-gram) and three part-of-speech (3-gram, 4-gram, and 5-gram) language models (first seven features in Table 4.6).

The labels in the part-of-speech language model were obtained using an off-the-shelf Maximum Entropy part-of-speech tagger (Ratnaparkhi, 1996). The only difference between regular *n*-gram and part-of-speech language models is that the latter models on

sequences of part-of-speech tags instead of word sequences. All language models used herein as well as in other experiments throughout this thesis are smoothed with modified Kneser-Ney discounting (Kneser and Ney, 1995) interpolated with lower-order estimates as described in Chen and Goodman (1998) and implemented in the SRILM toolkit (Stolcke, 2002).

The reason for using *n*-gram language models which are up to 2 orders higher and lower than the 5-grams used in the baseline is two-fold. Firstly, we wished to observe the impact of *n*-gram size on the ranking of translation hypotheses. More importantly, this allowed us to observe whether the baseline language model was deficient (i.e. context window size of 5 was insufficient) or overzealous (i.e. the context window size of 5 was too large to distinguish between good and bad hypotheses). By using part-of-speech language models, we were able to incorporate syntax in the candidate selection process in a very simple manner.

We also use perplexity model feature in addition to the language model probabilities because the perplexity score is normalized over the sentence length, i.e. it does not favour sentences of relatively shorter length (Ye at al., 2007 WMT).

4.4.2 Minimum Bayes Risk

The minimum bayes risk probability (*mbr* feature in Table 4.6) is an alternative to the maximum a posteriori (MAP) translation modeled by the baseline decoder score. The MBR score takes into account not only the likelihood of a candidate but also its similarity to the other very probable translations. The similarity is measured using the BLEU metric as described in Section 3.4.1 in the previous chapter.

$$e_{mbr} = \sum_{e\prime} BLEU(e, e\prime) \Pr(e\prime \mid f) \tag{4.1}$$

Equation 4.1 gives the formula for computing the MBR score for a translation candidate e. It traverses through a pairwise comparison of the candidate e with every other candidate e$\prime$. The baseline decoder score is a special case of the MBR probability score

when the loss function (indicated by the BLEU metric in our formula) simply assigns a value 1 when e is equivalent to e′ and 0 otherwise. This feature is considered useful because it seeks to take into account the relative position of a hypothesis in its respective *n*-best list.

4.4.3 Posterior Probabilities

Zens and Ney (2006) define a range of posterior probability measures for SMT. We have used the *n*-gram posterior probability (Equation 4.2) and the sentence length posterior probability (Equation 4.3) as features in our reranking framework.

$$\Pr(e_1^n \mid f_1^J) = \frac{C(e_1^n, f_1^J)}{\sum_{e'^n_1} C(e'^m_1, f_1^J)} \tag{4.2}$$

$$\Pr(I \mid f_1^J) = \sum_{e_1^I} \Pr(e_1^I \mid f_1^J) \tag{4.3}$$

These posterior probabilities for all candidate translations of a particular sentence are estimated over the search space covered by the *n*-best list for this sentence.

4.4.4 Source–Target Length Ratio

This feature takes into account the relative lengths of the source language and target language sentences.

4.4.5 IBM Model 1 Score

Herein we describe two sets of features: IBM and IBM inverse. The IBM model feature is computed using Equation 4.4 which focusses on the source–target word conditional probabilities.

$$\Pr(f \mid e) = \frac{\epsilon}{(l+1)^m} \prod_{j=1}^{m} \sum_{i=0}^{l} t(f_j \mid e_i) \tag{4.4}$$

For each feature, we also give their performance on reranking the baseline system using each of the five weighting schemes (MERT and 4 rescoring algorithms).

4.5 Experiments: Feature Combination

In order to evaluate whether the reranking features described in Section 4.4 help improve the performance of PB-SMT systems over the baseline (reported in Section 4.3), we build five reranking systems (RERANK$_{MERT}$, RERANK$_{BPROD}$, RERANK$_{MPROD}$, RERANK$_{BSUM}$, and RERANK$_{MSUM}$) for each of the three language pairs (French→English, German→English, and Spanish→English) and contrast their translation accuracies across five MT system evaluation metrics (BLEU, NIST, METEOR, WER, and PER). We also observe how each of the six PB-SMT systems (baseline plus five reranking systems) vary in terms of the percentage of oracles placed at the top rank (OBLEU and OMET). Note that the OBLEU and OMET scores are a direct indication of the model errors (number of sentences having oracles at the top position of an *n*-best list) in a system.

The devset is used to estimate the parameters (weights or lambdas) of the features in a particular system and the testset is used to validate the effectiveness of the features and their associated weights on 'unseen' sentences. The specifics of the six PB-SMT systems evaluated in this section follow:

- **BASELINE**: This is a basic PB-SMT system consists of 14 decoding features with weights optimised on system-level BLEU using MERT
- **RERANK**$_{MERT}$: This is a reranking system (i.e. the PB-SMT system undergoes a second-stage decoding step wherein the *n*-best list is reranked) with 14 decoding + 19 reranking features optimised on system-level BLEU using MERT
- **RERANK**$_{BPROD}$: This is a reranking system with 14 decoding + 19 reranking features optimised on system-level BLEU using MERT followed by a weight adjustment algorithm using the RESC$_{prod}$ rescoring formula on BLEU-oracles

- **RERANK**$_{MPROD}$:This is a reranking system with 14 decoding + 19 reranking features optimised on system-level BLEU using MERT followed by a weight adjustment algorithm using the RESC$_{prod}$ rescoring formula on METEOR-oracles

- **RERANK**$_{BSUM}$: This is a reranking system with 14 decoding + 19 reranking features optimised on system-level BLEU using MERT followed by a weight adjustment algorithm using the RESC$_{sum}$ rescoring formula on BLEU-oracles

- **RERANK**$_{MSUM}$: This is a reranking system with 14 decoding + 19 reranking features optimised on system-level BLEU using MERT followed by a weight adjustment algorithm using the RESC$_{sum}$ rescoring formula on METEOR-oracles

Thus each of the five reranking MT systems is composed of the same 33 features differing in merely the method used to optimise the weights for combining the features in a log-linear model (parameter estimation). The purpose of this system-level evaluation of decoding and reranking feature combination reported in Tables 4.7 through 4.12 is two-fold: (1) To demonstrate the utility of RESC$_{prod}$ and RESC$_{sum}$ rescoring formulas defined in Chapter 3 on additional (reranking) features as opposed to mere baseline (decoding) features (2) To evaluate the impact of reranking features on model errors in a PB-SMT system.

4.5.1 French–English

Table 4.7 shows the results for reranking 100-best distinct lists on French–English devset, evaluated using the five metrics: BLEU, NIST, METEOR, WER, and PER. We compare the performance of five reranking systems (33 features: RERANK$_{MERT}$, RERANK$_{BPROD}$, RERANK$_{MPROD}$, RERANK$_{BSUM}$, RERANK$_{MSUM}$) against the BASELINE system (14 features).

Three out of five reranking systems (RERANK$_{MERT}$, RERANK$_{BPROD}$, and RERANK$_{MPROD}$) outperform the baseline across all metrics. The remaining two reranking systems RERANK$_{BSUM}$ and RERANK$_{MSUM}$ perform slightly worse than the baseline on all metrics except METEOR score.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
BASELINE	32.17	7.70	61.34	57.10	40.96	12.55	13.75
RERANK$_{MERT}$	**32.58**	**7.74**	**61.63**	**56.89**	**40.76**	14.80	**14.65**
RERANK$_{BPROD}$	32.49	7.73	61.51	56.97	40.86	**14.90**	14.60
RERANK$_{MPROD}$	32.49	7.73	61.51	56.96	40.85	14.85	14.60
RERANK$_{BSUM}$	31.95	7.65	61.42	57.57	41.28	12.60	12.80
RERANK$_{MSUM}$	31.89	7.63	61.46	57.71	41.33	12.70	12.40

Table 4.7: Summary of the results on reranking 100-best distinct *n*-best lists: French→English devset.

The best-performing system is RERANK$_{MERT}$ which bests the BASELINE system on BLEU with a *statistically significant* score of 32.58 points (absolute difference of 1.27% over the baseline), NIST with a *statistically significant* score of 7.73 points (absolute difference of 0.39% over the baseline), METEOR with a *statistically significant* score of 61.63 points (absolute difference of 0.47% over the baseline), WER with a *statistically significant* score of 56.89 points (absolute difference of 0.37% over the baseline), PER with a *statistically significant* score of 40.76 points (absolute difference of 0.49% over the baseline), OBLEU (45 more sentences than the baseline) and OMET (18 more sentences than the baseline).

Both RERANK$_{BPROD}$ and RERANK$_{MPROD}$ systems give similar performance and perform slightly worse than RERANK$_{MERT}$, but significantly better than the baseline (1% improvement over the baseline on BLEU). There is a larger degree of variation between RERANK$_{BSUM}$ and RERANK$_{MSUM}$ systems with RERANK$_{BSUM}$ as the better performing system of the two. Note that almost all reranking systems rank more oracles in the top position than the baseline.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
BASELINE	32.47	**7.81**	61.80	56.43	40.34	12.75	12.70
RERANK$_{MERT}$	**32.67**	**7.81**	61.91	**56.38**	**40.33**	13.50	14.50
RERANK$_{BPROD}$	32.64	7.80	61.89	56.44	40.38	**14.00**	**15.00**
RERANK$_{MPROD}$	32.65	7.80	61.88	56.43	40.38	13.95	**15.00**
RERANK$_{BSUM}$	32.35	7.74	61.93	56.86	40.68	11.75	13.10
RERANK$_{MSUM}$	32.29	7.73	**61.96**	56.91	40.72	11.65	12.45

Table 4.8: Summary of the results on reranking 100-best distinct *n*-best lists: French→English testset.

On the 100-best distinct lists on French–English testset (Table 4.8), we see similar performance as on devset. Again, the three reranking systems (RERANK$_{MERT}$, RERANK$_{BPROD}$, and RERANK$_{MPROD}$) outperform the BASELINE with RERANK$_{MERT}$ giving the best scores on BLEU (32.67 points: absolute difference of 0.6% over the baseline), WER (56.38 points: absolute difference of 0.1% over the baseline), and PER (40.33 points: absolute difference of 0.01 over the baseline). However, unlike the devset,RERANK$_{MSUM}$ gives the best scores on METEOR with an absolute difference of 0.25% over the baseline and nearly 0.1% over the RERANK$_{MERT}$. The largest number of oracles in the top-most position were recorded by RERANK$_{BPROD}$ and RERANK$_{MPROD}$) systems.

Thus, on French→English, the log-linear combination of decoding features (Section 3.3.1) and reranking features (Section 4.4) significantly improves the system. RERANK$_{MERT}$ is the best reranking system followed by RERANK$_{BPROD}$ and RERANK$_{MPROD}$ systems. As observed in the system-level evaluation results for oracle-based rescoring in Chapter 3, the METEOR scores seem to favour the SUM systems. The reranking systems record a higher margin of improvement over the baseline on the devset than on the testset. Whether this pattern is adhered to by the other language pairs (in sections 4.5.2 and 4.5.3) remains to be seen.

4.5.2 German–English

Table 4.9 shows the MT system evaluation results for reranking 100-best distinct lists on German–English devset. Just like in Section 4.5.2 we compare the performance of five reranking systems (33 features: RERANK$_{MERT}$, RERANK$_{BPROD}$, RERANK$_{MPROD}$, RERANK$_{BSUM}$, RERANK$_{MSUM}$) against the BASELINE system (14 features).

Three out of five reranking systems (RERANK$_{MERT}$, RERANK$_{BPROD}$, and RERANK$_{MPROD}$) outperform the baseline across all metrics. The remaining two reranking systems RERANK$_{BSUM}$ and RERANK$_{MSUM}$ perform below the baseline on all metrics except METEOR score. In fact the RERANK$_{BSUM}$ system performs nearly as well as the baseline on BLEU and NIST, and outperforms the baseline on PER as well as METEOR scores.

The best-performing system is RERANK$_{MERT}$ which bests the BASELINE system on

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
BASELINE	26.93	7.00	57.01	65.52	44.89	**4.85**	**5.40**
RERANK$_{MERT}$	**27.23**	**7.05**	**57.39**	**65.19**	**44.40**	4.65	5.15
RERANK$_{BPROD}$	27.09	7.04	57.29	65.29	44.47	4.60	5.00
RERANK$_{MPROD}$	27.09	7.04	57.29	65.29	44.46	4.65	5.00
RERANK$_{BSUM}$	26.92	6.99	57.26	65.66	44.75	3.90	4.30
RERANK$_{MSUM}$	26.78	6.96	57.25	65.94	44.98	3.65	4.15

Table 4.9: Summary of the results on reranking 100-best distinct *n*-best lists: German→English devset.

BLEU with a *statistically significant* score of 27.23 points (absolute difference of 1.11% over the baseline), NIST with a *statistically significant* score of 7.05 points (absolute difference of 0.7% over the baseline), METEOR with a *statistically significant* score of 57.39 points (absolute difference of 0.67% over the baseline), WER with a *statistically significant* score of 65.19 points (absolute difference of 0.51% over the baseline), PER with a *statistically significant* score of 44.40 points (absolute difference of 1.1% over the baseline).

However unlike French–English, the BASELINE system records the highest number of BLEU-oracles (OBLEU) and METEOR-oracles (OMET) at the top of the *n*-best lists. This implies that although one or more of the reranking systems score more accurately than the baseline, a lower percentage of these translations are actually oracles in the reranking systems than in the baseline. Note that since the highest number of oracles in the top rank is 5.4% (BASELINE OMET score), i.e. merely 108 out of 2000 sentences, there is a strong possibility that in the remaining 1892 sentences, the reranking systems fare better than the baseline.

Both RERANK$_{BPROD}$ and RERANK$_{MPROD}$ systems give similar performance and perform slightly worse than RERANK$_{MERT}$, but significantly better than the baseline (0.5% improvement over the baseline on METEOR). There is a larger degree of variation between RERANK$_{BSUM}$ and RERANK$_{MSUM}$ systems with RERANK$_{BSUM}$ as the better performing system of the two.

On the 100-best distinct lists on German–English testset (Table 4.10), we see better performance from our reranking systems than on devset in that all the five rerank-

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
BASELINE	27.02	7.01	57.11	65.25	45.01	**5.05**	**5.10**
RERANK$_{MERT}$	27.36	7.05	57.48	**64.91**	44.64	4.50	4.55
RERANK$_{BPROD}$	27.37	**7.06**	**57.49**	64.92	44.61	4.65	4.55
RERANK$_{MPROD}$	**27.38**	**7.06**	**57.49**	**64.91**	**44.60**	4.70	4.55
RERANK$_{BSUM}$	27.21	7.02	57.39	65.27	44.88	4.30	4.25
RERANK$_{MSUM}$	27.13	7.00	57.46	65.48	45.05	3.80	3.75

Table 4.10: Summary of the results on reranking 100-best distinct *n*-best lists: German→English testset.

ing systems outperform the BASELINE as per most metrics. Unlike the devset, both RERANK$_{MPROD}$ and RERANK$_{BPROD}$ are the best-performing systems giving the best scores on BLEU (*statistically significant* 27.38 points: absolute difference of 1.33% over the baseline), NIST (*statistically significant* 7.06 points: absolute difference of 0.7% over the baseline), METEOR (*statistically significant* 57.49 points: absolute difference of 0.67% over the baseline), WER (64.91 points: absolute difference of 0.52% over the baseline), and PER (44.60 points: absolute difference of 0.9% over the baseline).

Just like on the devset, the largest number of oracles in the top-most position (OBLEU and OMET scores) were recorded by the BASELINE system.

Thus, on German→English, the log-linear combination of decoding features (Section 3.3.1) and reranking features (Section 4.4) significantly improves the system. RERANK$_{MERT}$, RERANK$_{BPROD}$ and RERANK$_{MPROD}$ systems are the best-performing systems. The reranking systems record a similar margin of improvement over the baseline on both the devset and testset, unlike on French–English (Section 4.5.1).

4.5.3 Spanish–English

Table 4.11 shows the MT system evaluation results for reranking 100-best distinct lists on Spanish–English devset. Just like in sections 4.5.1 and 4.5.2, we compare the performance of five reranking systems (33 features: RERANK$_{MERT}$, RERANK$_{BPROD}$, RERANK$_{MPROD}$, RERANK$_{BSUM}$, RERANK$_{MSUM}$) against the BASELINE system (14 features).

Three out of five reranking systems (RERANK$_{MERT}$, RERANK$_{BPROD}$, and RERANK$_{MPROD}$) outperform the baseline across all metrics. The remaining two reranking systems

RERANK$_{BSUM}$ and RERANK$_{MSUM}$ perform better than BASELINE on METEOR and at par with the BASELINE system on BLEU and PER.

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
BASELINE	32.98	7.80	61.99	56.50	40.68	10.35	11.20
RERANK$_{MERT}$	**33.22**	**7.82**	**62.13**	**56.27**	**40.52**	**10.75**	**11.50**
RERANK$_{BPROD}$	33.16	7.81	62.06	56.32	40.57	10.65	11.35
RERANK$_{MPROD}$	33.16	7.81	62.06	56.32	40.56	10.65	11.35
RERANK$_{BSUM}$	32.98	7.78	62.09	56.57	40.70	9.15	10.35
RERANK$_{MSUM}$	32.96	7.78	62.11	56.57	40.69	9.10	10.45

Table 4.11: Summary of the results on reranking 100-best distinct *n*-best lists: Spanish→English devset.

The best-performing system is RERANK$_{MERT}$ which bests the BASELINE system on BLEU with a *statistically significant* score of 33.22 points (absolute difference of 0.73% over the baseline), NIST with a score of 7.82 points (absolute difference of 0.26% over the baseline), METEOR with a score of 62.13 points (absolute difference of 0.23% over the baseline), WER with a *statistically significant* score of 56.27 points (absolute difference of 0.41% over the baseline), PER with a score of 40.52 points (absolute difference of 0.4% over the baseline).

Similar to French–English devset and unlike German–English devset, the RERANK$_{MERT}$ system records the highest number of BLEU-oracles (OBLEU) and METEOR-oracles (OMET) at the top of the *n*-best lists.

Both RERANK$_{BPROD}$ and RERANK$_{MPROD}$ systems give similar performance and perform slightly worse than RERANK$_{MERT}$, but significantly better than the baseline (0.55% improvement over the baseline on BLEU). Surprisingly, even RERANK$_{BSUM}$ and RERANK$_{MSUM}$ systems have negligible variation in their evaluation scores.

On the 100-best distinct lists on Spanish–English testset (Table 4.12), we see better performance from our reranking systems than on devset in that all the five reranking systems outperform the BASELINE as per most metrics. Both RERANK$_{BSUM}$ and RERANK$_{MSUM}$ are the best-performing systems according to BLEU (33.00 points: absolute difference of 0.37% over the baseline) and METEOR (62.09 points: absolute difference of 0.23% over the baseline). However, both RERANK$_{BPROD}$ and RERANK$_{MPROD}$

SYSTEM	BLEU	NIST	METEOR	WER	PER	OBLEU	OMET
BASELINE	32.88	7.88	61.95	56.03	40.12	**11.30**	**12.20**
RERANK$_{MERT}$	32.98	**7.89**	62.01	55.88	40.07	10.05	11.40
RERANK$_{BPROD}$	32.98	**7.89**	62.03	**55.84**	**40.05**	10.35	11.50
RERANK$_{MPROD}$	32.98	**7.89**	62.02	**55.84**	**40.05**	10.35	11.50
RERANK$_{BSUM}$	**33.00**	7.87	**62.09**	56.02	40.15	9.65	11.05
RERANK$_{MSUM}$	32.99	7.87	**62.09**	56.03	40.15	9.60	10.80

Table 4.12: Summary of the results on reranking 100-best distinct *n*-best lists: Spanish→English testset.

are the best-performing systems according to NIST (7.89 points: absolute difference of 0.13% over the baseline), WER (55.84 points: absolute difference of 0.34% over the baseline), and PER (40.05 points: absolute difference of 0.17% over the baseline).

Unlike on the devset, the largest number of oracles in the top-most position (OBLEU and OMET scores) were recorded by the BASELINE system. This implies that although one or more of the reranking systems score more accurately than the baseline, a lower percentage of these translations are actually oracles in the reranking systems than in the baseline. Note that since the highest number of oracles in the top rank is 12.2% (BASELINE OMET score), i.e. merely 244 out of 2000 sentences, there is a strong possibility that in the remaining 1756 sentences, the reranking systems fare better than the baseline.

Thus, on Spanish→English, the log-linear combination of decoding features (Section 3.3.1) and reranking features (Section 4.4) significantly improves the system. RERANK$_{MERT}$ is the best-performing system on devset while all four oracle-based reranking systems (RERANK$_{BPROD}$, RERANK$_{MPROD}$, RERANK$_{BSUM}$, and RERANK$_{MSUM}$) perform slightly better than the baseline on testset.

4.6 Contrasting Decoding with Reranking

Table 4.13 contrasts the oracle distribution of French → English 100-best testset for decoding (using the 14 baseline features) and using 13 additional reranking features.[5] Thus by using more sophisticated features we are able to reduce model errors by ranking 283

[5]These 13 features are a subset of the actual 19 features used for reranking. This is done as the remaining 6 features showed nearly equal proportion distribution for oracles (w.r.t. BLEU metric) and 1-bests.

	(a) DECODING	(b) RERANKING
RANGE	**100-BEST DISTINCT**	**100-BEST DISTINCT**
Rank 1	251 (251)	283 (283)
Rank 2 to 5	539 (790)	570 (853)
Rank 6 to 10	216 (1006)	281 (1134)
Rank 11 to 24	517 (1523)	446 (1580)
Rank 25 to 50	278 (1801)	245 (1825)
Rank 51 to 75	128 (1929)	113 (1938)
Rank 76 to 100	71 (2000)	62 (2000)

Table 4.13: Number of times an oracle occurs in a particular range of ranks in the *n*-best lists of (a) DECODING and (b) RERANKING features. The numbers in brackets give the corresponding cumulative frequencies.

oracles at rank 1 in contrast to 251 oracles with the baseline system.

FEATURE	% ORACLE	% 1-BEST	% BOTH
d1	7.10	10.85	82.05
d2	41.80	37.60	20.60
d3	3.95	7.90	88.15
d4	8.65	12.55	78.80
d5	41.95	41.60	16.45
d6	3.45	7.70	88.85
d7	8.95	12.90	78.15
lm	38.65	48.75	12.60
w	28.30	30.50	41.20
tm1	34.20	52.20	13.60
tm2	34.85	49.25	15.90
tm3	29.90	54.80	15.30
tm4	37.30	48.20	14.50
tm5	29.20	23.50	47.30

Table 4.14: % of sentence in which a feature favours an oracle (2nd column), a 1-best (3rd column) or favours both equally (last column) in the *n*-best lists of DECODING features

The features which actually make this impact and their individual effect on oracle ranking are shown in Tables 4.14 (decoding features) and 4.15 (reranking features). In both tables, the first column gives the feature name (described in Table 4.2 for decoding features and in Table 4.6 for reranking features). For each of the 2000 sentences in the testset,[6] the individual feature values were compared for the 1-best translation (baseline system output) and the oracle translation (output most similar to the reference as per the BLEU metric). The second column shows the percentage of these sentences for which

[6]Only those sentences were considered for which the 1-best translation was *not* the oracle translation.

FEATURE	% ORACLE	% 1-BEST	% BOTH
lm3	46.4	43.1	10.5
lm4	45.35	44.2	10.45
lm6	45.75	43.85	10.4
lm7	45.15	44.45	10.4
plm3	43.85	36.4	19.75
plm4	43.75	36.55	19.7
plm5	43.4	36.9	19.7
mbr	40.5	47.5	12
np	43.15	45.4	11.45
len	26.9	26.7	46.4
lenr	29.6	28.55	41.85
ibm	56.2	28.9	14.9
ibm-1	28.85	56.25	14.9

Table 4.15: % of sentence in which a feature favours an oracle (2nd column), a 1-best (3rd column) or favours both equally (last column) in the *n*-best lists of RERANKING features

any given feature has a greater value for oracle. The third column shows the same for the 1-best translation. The fourth column displays the percentage of sentences for which both oracle and 1-best possessed the same value. If a majority proportion of sentences had the equivalent feature value for both oracle and 1-best as in the case of d1 feature (82.05%) in Table 4.14, then it implies that this feature is not very good in discriminating between good and bad translations.

Figure 4.4 gives a graphical representation of Table 4.14 for decoding features on French–English testset. We observe that for the most part, distortion features (d1 through d7) are unable to discriminate between oracle and 1-best translations. This is because they either generate the same value for both types of sentences or favour one over the other evenly. It might perhaps be a good idea to not consider distortion features while reranking. We leave this for future work.

Figure 4.5 gives a graphical representation of Table 4.15 for reranking features on French–English testset. While the language models are mostly evenly matched on the oracle and 1-best translations, the IBM Model 1 feature favours the oracles heavily. This is as expected because most approaches in the literature report IBM Model 1 scores to be highly successful in discriminating between good and bad translations.

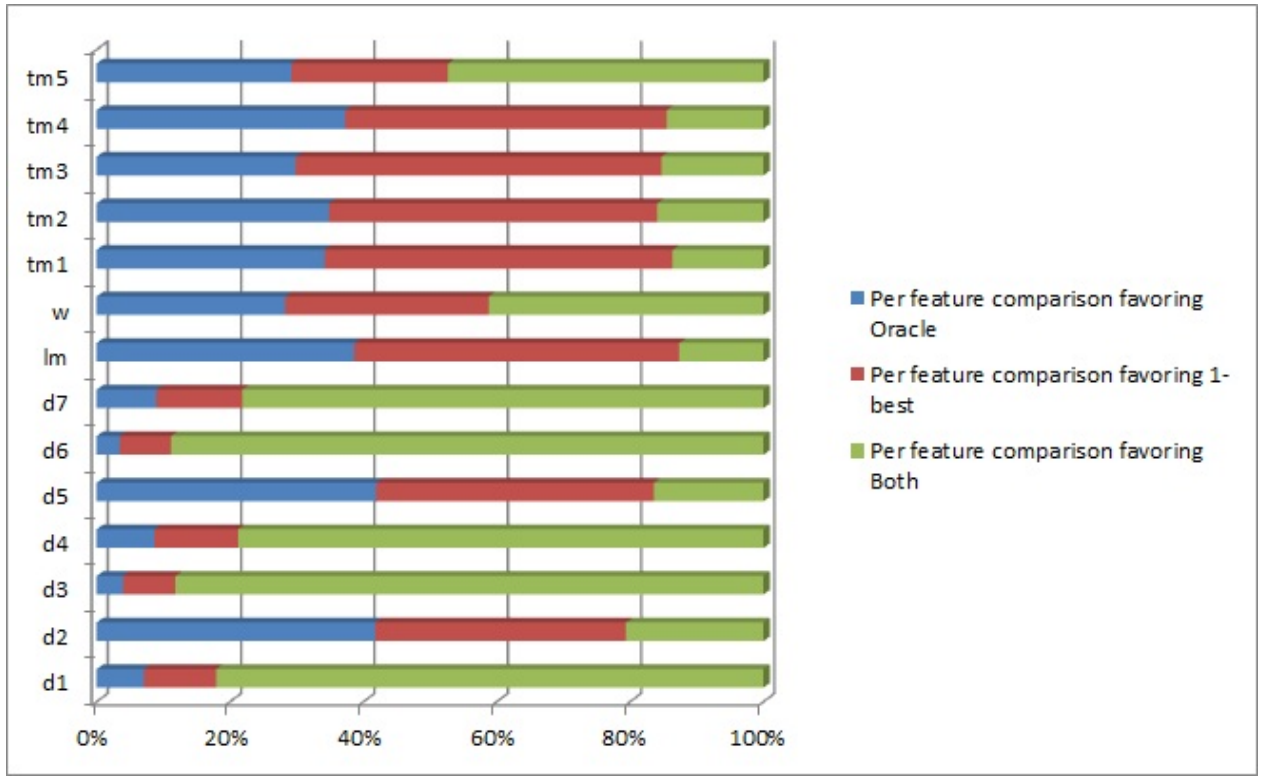


Figure 4.4: Results for a 100-best list of oracles: For how many sentences (% given on the X-axis) does a baseline feature (given on the Y-axis) favour the oracle translation (blue bar) over the 1-best translation (red bar). The green bar (third band in each bar) denotes percentage of sentences having the same value for its oracle and 1-best hypothesis

.

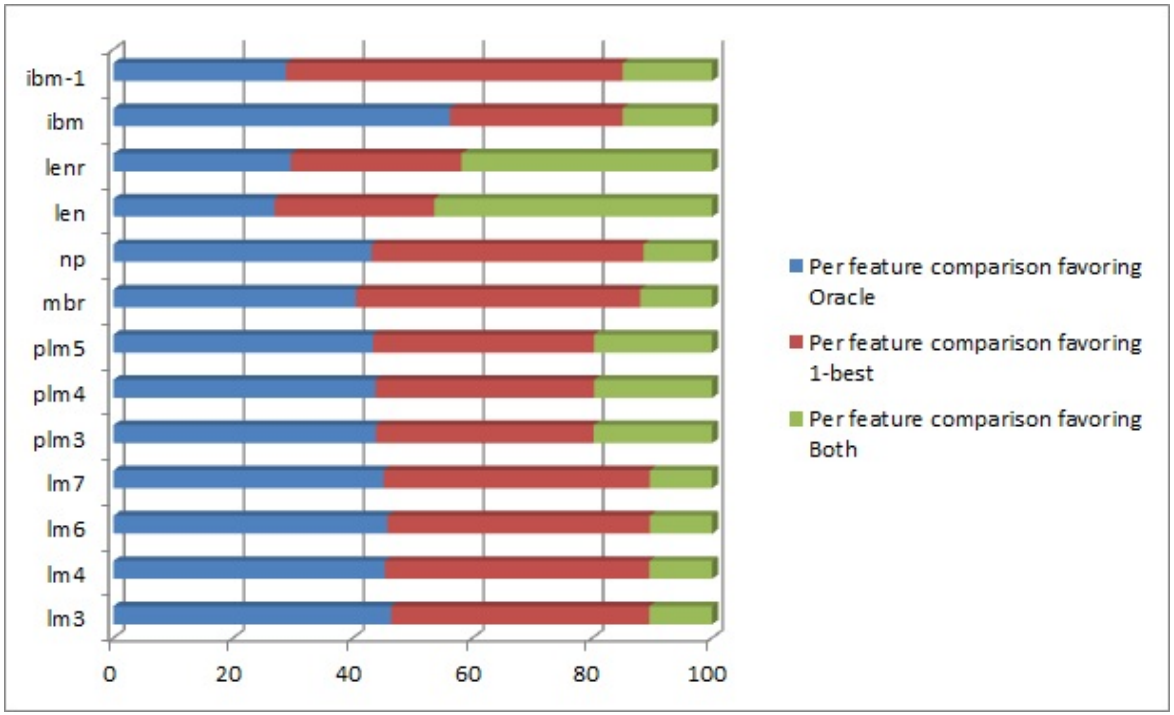


Figure 4.5: Results for a 100-best list of oracles: For how many sentences (% given on the X-axis) does a reranking feature (given on the Y-axis) favour the oracle translation (blue bar) over the 1-best translation (red bar). The green bar (third band in each bar) denotes percentage of sentences having the same value for its oracle and 1-best hypothesis

.

4.7 Conclusion

Having experimented on three language pairs (French→ English, German→ English, and Spanish→ English), we evaluated the application of Oracle-based Training (OBT) rescoring methods on *n*-best list reranking. We observed that the OBT scoring helps reranking on German→ English data more than any other language pair. We also observed that OBT is a viable method for feature selection for reranking in large feature spaces. For example, subsequent experiments may benefit from filtering out features like the distortion since they were ineffective in discriminating between good and bad translations.

4.8 Summary

In this chapter, we experimented on 3 different language pairs (French, Spanish, German) to English on 100-best distinct lists using 13 reranking features in addition to 14 baseline features. We pitted the baseline system against 5 different reranking systems (optimised using MERT (1), and optimised using OBT (4)). All reranking systems outperform the baseline as expected. However our OBT systems outperformed MERT-reranking only on German → English data. We believe that the variation between the source and target languages proved to be a factor. In the next chapter, we summarize our thesis and revisit our research questions.

Chapter 5

Conclusions

5.1 Research Questions Answered

In this thesis we have posed five research questions (RQ1 through RQ5, listed below) which have attempted to answer.

(RQ1) *Are phrase pairs extracted from percolated dependency treebanks a useful knowledge source for PB-SMT?*

(RQ2) *Can the PB-SMT system obtain optimal performance out of linguistically motivated phrase pairs?*

In Chapter 2 we explored the phrase pairs extracted from percolated dependencies-induced treebanks and through experiments on 15 syntax-aware PB-SMT systems (one or more combination of the four base systems: S, C, D, P), we demonstrated the effectiveness of PERC-induced phrase pairs on translation performance. The automatic MT system evaluation scores on French–English translations show that the S+C system gave the best performance. However, lack of statistical significance in the results and manual evaluation leads us to believe that PERC is useful enough to warrant further investigation. Therefore, percolated dependencies appear to be a useful knowledge source for PB-SMT.

Using four other decoder configurations (all-options, reordering models, MBR, multiple translation models) and a qualitative analysis leads us to believe that good phrase

pairs fail to be selected by the decoder as the optimum translation. We therefore also demonstrated that the PB-SMT system gives sub-optimal performance in case of multiple phrase tables.

The lessons learned from this chapter are that there are multiple techniques of performing the same function (in this case phrase extraction). It is far more useful to find an optimal way to combine each of these techniques (S, C, D, P and their combinations) rather than relying on merely one way (e.g. S (non-linguistic)).

(RQ3) *Does pushing oracles up the* n*-best list minimise model errors and improve performance of a PB-SMT system?*

In Chapter 3, we explore the concepts of rescoring n-best lists by identifying oracles to help reduce model errors which in turn improves the system performance. We postulated two rescoring strategies: RESC$_{sum}$ and RESC$_{prod}$. We demonstrated their effectiveness on French–English (both directions) as well as on German $\rightarrow$ English, Spanish $\rightarrow$ English PB-SMT systems. We can definitely conclude that pushing oracles up the n-best list reduces model errors (as demonstrated by the OMET and OBLEU scores) which in turn improves system performance as demonstrated by automatic evaluation measures. Generally speaking, the RESC$_{sum}$ systems were similar to the BASELINE and were more likely to beat the baseline system on all metrics except the METEOR scores. The METEOR score were favoured by the RESC$_{prod}$ systems. This implies that the ratio-based parameter estimation is more suitable for metrics which take into account both the precision and recall. While the difference-based parameter estimation (similar to perceptrons) favour the precision-based metrics like BLEU and NIST.

(RQ4) *Can additional features in a reranking framework help minimise model errors?*

(RQ5) *Can the oracle-reranked system help the PB-SMT system to better exploit linguistically motivated chunks?*

The remaining two research questions are answered in Chapter 4. By supplementing the 14 decoding features with 19 additional features, we successfully demonstrated the

utility of our rescoring strategies in a reranking (post-decoding) framework. We built PB-SMT systems for three language pairs and demonstrated that although MERT estimation of reranking features is the best-performing system, our oracle-based reranked systems perform on par. One must keep in mind that OBT (Oracle-based Training) is a single iteration algorithm as opposed to MERT.

With regards to the last research question, we conclude that although it is possible to utilise our OBT strategies to help optimise multiple phrase extractions, it is outside the scope of this research to demonstrate this. This is left for future work.

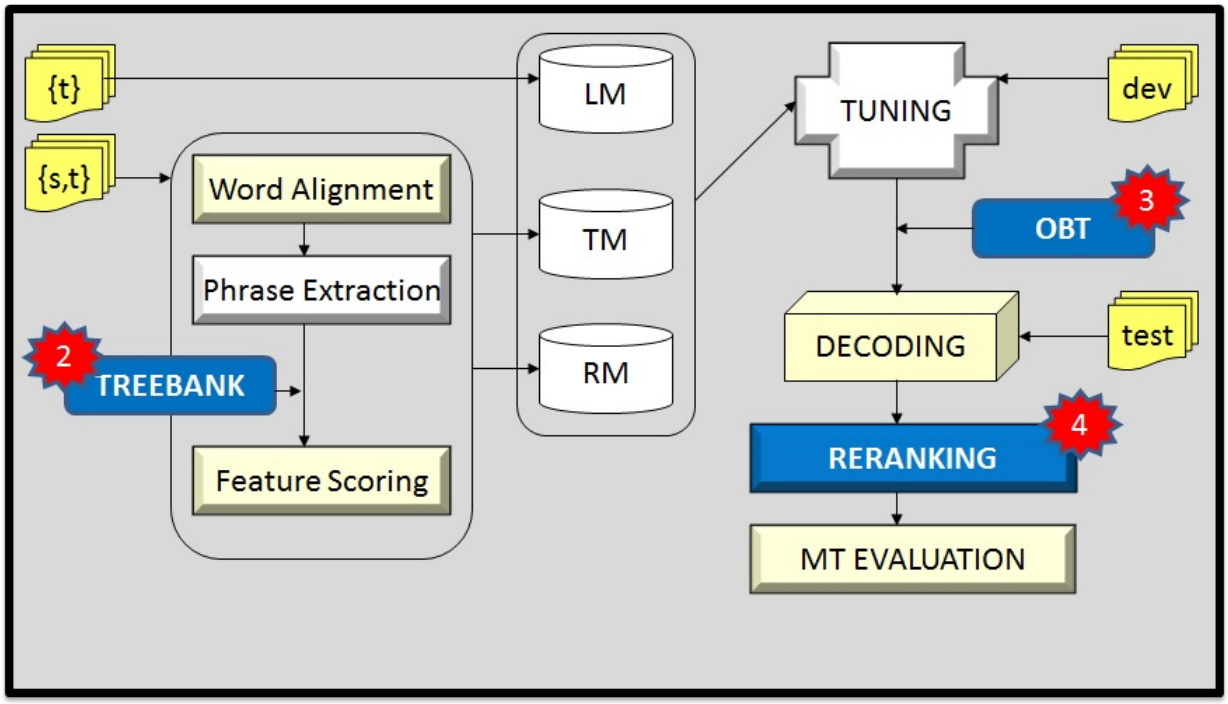


Figure 5.1: Schematic diagram of the modules in a Phrase-based Statistical Machine Translation System: *Thesis Overview*.

Figure 5.1 summarises the PB-SMT modules investigated in each chapter. Thus in this thesis, we have successfully explored useful modifications to three modules of a standard PB-SMT system: phrase extraction, tuning, and reranking. The major contributions of this thesis are as follows:

- It is possible to standardise outputs of different parsers for source and target languages used in syntax-aware PB-SMT systems by incorporating a head percolation based induction on constituency treebanks. This implies that the head percolation

algorithm can help minimise the differences in parser outputs when using different softwares for different languages (example Berkeley Parser for source language and Stanford Parser for target language).

- Percolated dependency-induced phrase pairs are significantly different from direct dependency-induced phrase pairs. Thus we have found a hitherto unutilised knowledge source for PB-SMT phrase pairs.

- It is beneficial to combine multiple techniques (phrase extractions) rather than use a single one. The S+C+D+P system gives better translations than individual systems S, C, D, or P.

- The PB-SMT system has not been fully exhausted as demonstrated by the existence of the huge amount of model errors. Therefore, it is our belief that before venturing into tree-based SMT, better optimisation algorithms and combination strategies for phrase-based SMT must be experimented upon. This thesis is a step in this direction.

- Simple intuitive rescoring strategies like RESC$_{prod}$ and RESC$_{sum}$ can help improve the deficiencies of Minimum Error Rate Training parameter estimation algorithm.

- In an age where there are numerous approaches of performing the same Natural Language Processing task, one must find ways to exploit the benefits of each of these techniques by combining the outputs in a smart manner or using rescoring / reranking techniques in a post-processing step.

5.2 Design Decisions

The SMT research field is very diverse and active. As such, there were numerous instances of where we were presented with a set of alternatives, and we chose a specific direction to demonstrate the effectiveness of our ideas. Thus in addition to the research questions stated above, we also address some of our design decisions. For example,

- *Why discard the syntactic labels when using treebank-based phrase extraction in syntax-aware systems?* We are focussing on linguistically-motivated phrase boundaries and not a specific phrase type (as indicated by the labels).

- *Why use* n*-best lists in rescoring and reranking instead of lattices (a graphical representation of translation hypotheses)?* It is our belief that lattices are a nifty space-saving mechanism or representation format and the *n*-best lists are easier to modify in our experiments which rely on having access to the whole sentence rather than parts of it. Lattices, though more compact, are computationally more complex than *n*-best lists by increasing the difficulty of extracting features for each complete sentence or hypothesis in the translation space. Future work will however address this issue.

- *Why employ sentence-level* BLEU *and not other metrics for oracle selection?* BLEU is the most popular metric and we also evaluated selected experiments using sentence-level METEOR scores.

- *Why only experiment on French-English data for additional experiments?* This was done for uniformity throughout experiments when correlating with percolated-dependency induced phrase extraction system.

- *Why use the specific features we use in reranking?* This was done while keeping ease of availability in mind as our primary purpose was not the innovativeness of features but the applicability of OBT on reranking features.

- *Why not use perceptron in rescoring?* Our RESC-sum and RESC-prod strategies postulate something resembling a perceptron, simpler than a perceptron or rather a 1-iteration flavour.

- *Why use distinct* n*-best lists for reranking?* We use distinct when the same surface level string will give same scores for features.

- *Why not translate from English into a foreign language in the reranking experiments?* This was because target language will require part-of-speech taggers for

POS-language models which we did not have access to.

5.3 Potential Research Avenues

We have categorized the future work on the three main research strands (phrase pair extraction, rescoring, and reranking).

5.3.1 Phrase Pair Extraction

We plan to work on scaling up the syntax-aware systems from 100,000 sentences upwards, and on other language pairs for treebank phrases. We intend to work on more types of phrase extractions and their combinations, e.g. marker-based, discriminative phrase alignment, alignments instead of grow-diag-final. We will also explore other models like str-con alignment or con-dep alignment rather than con-con, dep-dep for source-target language pairings. Another investigative scenario to be explored is using Google's OpenFST[1] (weighted finiste state transducers) to combine multiple phrase extractions.

For further future work, we plan to introduce a segmentation model in syntax-aware systems so that the phrases to be decoded are linguistically segmented. An interesting research question to be answered herein is whether it helps improve the performance by decoding syntactically well-formed phrases.

It was discovered that the largest overlap in pure numerical terms was between CON and PERC phrase tables. A useful exercise would be to conduct an investigation into any bias here. By using two different constituency parsers to produce two sets of PERC chunks, we plan to study the correlation between the overlap of phrase pairs in the four phrase tables (two constituency types and two percolated dependency types) as a measure of bias.

[1] `http://www.openfst.org/twiki/bin/view/FST/WebHome`

5.3.2 Rescoring (Parameter Estimation)

An avenue we did not explore (as we used Europarl data with 1 reference only) was how does OBT vary when using a dataset with multiple references and how is the oracle selection affected.

The rationale behind logistic regression is to rely on the power of discriminative learning to optimally use all available features to predict the final target (Liu et al., 2011). But we think that we can benefit from a tiered or hierarchical learning paradigm where in more reliable features are used to score the *n*-best lists followed by a second set of less reliable features and so on and so forth, rather than using all features at once.

Another avenue is to implement an on-the-fly MERT where the tuning is done depending on a sentence and is not done for the entire set. Hence the weights are more specific to each sentence (cluster of features rather than the entire set), rather than for the entire devset and testset. OBT would be helpful in a feature-selection strategy. This implies moving towards non-linear modeling where the same set of weights and features do not apply to all the sentences equally.

We plan to use metrics better suited for sentence-level evaluation like TER (Snover et al., 2006). On the issue of using other metrics, we plan to use more sophisticated (syntax-based) machine translation evaluation methods for oracle selection. A helpful avenue will be the MT Evaluation Metric Campaigns and to choose metrics which have the the highest correlation with human evaluation.

For future work, we will implement different flavors of perceptron for a discriminative reranking strategy and contrast it with MERT-based and oracle-based training strategies implemented in Chapter 3.

There are approaches in the field of optimisation which employ multi-objective functions. This implied tuning weights based on not just BLEU but BLEU and METEOR, etc. We wish to explore the applications of OBT in the same vein as employed by Duh et al. (2012) in learning to translate with multiple objectives.

An interesting experiment is oracle-based MERT: where instead of the true reference sentences we use oracles as a reference (i.e. translations which are reachable). We will

need to find a way of computing this after every intermediate decoding step in the MERT and before MERT's weight estimation algorithm.

5.3.3 Reranking

We also plan to use a host of reranking features (Shen et al., 2004; Carter and Monz, 2011) and couple them with our RESC$_{sum}$ and RESC$_{prod}$ rescoring strategy. We will also generate a feature based on our rescoring formula and use it as an additional feature in discriminative reranking framework.

We seek to trace through decoder and search graph where *n*-best oracles are pushed up or down. We wish to introduce phrase pair reranking rather than sentence reranking. This will tie in with investigating lattices as *n*-best lists will no longer be sufficient. There has been recent work done on oracle-based decoding (Wisniewski and Yvon, 2013) and we wish to introduce reranking during decoding instead of post-decoding.

While outside the scope of this thesis, an idea is to use each reranking feature to get the best ranking sentence(s) individually and then use system combination to combine the highest ranking hypotheses into one. This will be combining both reranking features and system combination, while doing away with the parameter estimation step.

Bibliography

Auli, M., Lopez, A., Hoang, H., and Koehn, P. (2009). A Systematic Analysis of Translation Model Search Spaces. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 224–232, Athens, Greece.

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

Bourigault, D., Fabre, C., Fréot, C., Jacques, M.-P., and Ozdowska, S. (2005). Syntex, analyseur syntaxique de corpus. In *Actes des 12èmes journées sur le Traitement Automatique des Langues Naturelles (TALN '05)*, Dourdan, France.

Brown, P., Cocke, J., Della Pietra, S., Della Pietra, V., Jelinek, F., Mercer, R., and Roossin, P. (1990). A Statistical Approach to Machine Translation. *Computational Linguistics*, **16**:79–85.

Brown, P. F., Pietra, S. A. D., Pietra, V. J. D., and Mercer, R. L. (1993). The Mathematics of Statistical Machine Translation: Parameter Estimation. *Computational Linguistics*, **19:2**:263–311.

Callison-Burch, C., Osborne, M., and Koehn, P. (2006). Re-evaluating the Role of BLEU in Machine Translation Research. In *Proceedings of the 11th Conference of the Eu-*

ropean Chapter of the Association for Computational Linguistics (EACL-06), pages 249–256, Trento, Italy.

Carl, M. and Way, A. (2003). *Recent Advances in Example-based Machine Translation*. Kluwer Academic Publishers, Dordrecht, The Netherlands.

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

Chen, S. F. and Goodman, J. T. (1998). An Empirical Study of Smoothing Techniques for Language Modeling. Technical report tr-10-98, Computer Science Group, Harvard University, Boston, MA.

Chiang, D. (2005). A Hierarchical Phrase-Based Model for Statistical Machine Translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pages 263–270, Ann Arbor, MI.

Chiang, D. (2007). Hierarchical Phrase-Based Translation. *Computational Linguistics*, **33:2**:201–228.

Chiao, Y.-C., Kraif, O., Laurent, D., Nguyen, T. M. H., Semmar, N., Stuck, F., Véronis, J., and Zaghouani, W. (2006). Evaluation of multilingual text alignment systems: the ARCADE II project. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC '06)*, pages 1975–1978, Genoa, Italy.

Collins, M. (1997). Three Generative, Lexicalised Models for Statistical Parsing. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics (ACL '97)*, pages 16–23, Madrid, Spain.

Doddington, G. (2002). Automatic Evaluation of MT Quality Using N-gram Co-occurrence Statistics. In *Proceedings of the Human Language Technology Conference*, pages 138–145, San Diego, CA.

Du, J., He, Y., Penkale, S., and Way, A. (2009). MATREX: The DCU MT system for WMT 2009. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*

at the 47th Annual Meeting of the Association for Computational Linguistics (ACL '09), pages 95–99, Athens, Greece.

Duh, K. and Kirchhoff, K. (2008). Beyond Log-Linear Models: Boosted Minimum Error Rate Training for N-best Re-ranking. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics Short Papers (ACL '08)*, pages 37–40, Columbus, OH.

Duh, K., Sudoh, K., Wu, X., Tsukada, H., and Nagata, M. (2012). Learning to translate with multiple objectives. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (ACL '12)*, pages 1–10, Jeju, Korea.

Forcada, M. L., Tyers, F. M., and Ramirez-Sanchez, G. (2009). The Free / Open-Source Machine Translation Platform Apertium: Five Years On. In *Proceedings of the 1st International Workshop on Free / Open-Source Rule-Based Machine Translation (FreeRBMT '09)*, pages 3–10, Alacant, Spain.

Galley, M., Graehl, J., Knight, K., Marcu, D., DeNeefe, S., Wang, W., and Thayer, I. (2006). Scalable Inference and Training of Context-Rich Syntactic Models. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics(COLING-ACL '06)*, pages 961–968, Sydney, Australia.

Galley, M. and Manning, C. D. (2008). A Simple and Effective Hierarchical Phrase Reordering Model. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP '08)*, pages 848–856, Honolulu, Hawaii, USA.

Germann, U., Jahr, M., Knight, K., Marcu, D., and Yamada, K. (2004). Fast and Optimal Decoding for Machine Translation. *Artificial Intelligence*, **154**:127–143.

Green, T. (1979). The Necessity of Syntax Markers. Two experiments with artificial languages. *Journal of Verbal Learning and Behavior*, **18**:481–496.

Groves, D. and Way, A. (2005). Hybrid Example-Based SMT: the Best of Both Worlds? In *Proceedings of the Workshop on Building and Using Parallel Texts: Data-Driven Machine Translation and Beyond, at the 43rd Annual Meeting of the Association of Computational Linguistics (ACL '05)*, pages 183–190, Ann Arbor, MI.

Hajič, J., Böhmová, A., Hajičová, E., and Vidová-Hladká, B. (2000). The Prague Dependency Treebank: A Three-Level Annotation Scenario. In Abeillé, A., editor, *Treebanks: Building and Using Parsed Corpora*, pages 103–127. Amsterdam:Kluwer.

Hasan, S., Zens, R., and Ney, H. (2007). Are Very Large N-best List Useful for SMT? In *Proceedings of the Combined Human Language Technology Conference Series and the North American Chapter of the Association for Computational Linguistics Conference Series (HLT-NAACL-07)*, pages 57–60, Rochester, NY.

He, Y. and Way, A. (2009). Improving the Objective Function in Minimum Error Rate Training. In *Proceedings of the Machine Translation Summit XII*, pages 238–245, Ottawa, Canada.

Hearne, M., Ozdowska, S., and Tinsley, J. (2008). Comparing Constituency and Depedency Representations for SMT Phrase-Extraction. In *15ème Conférence sur le Traitement Automatique des Langues Naturelles (TALN '08)*, Avignon, France.

Hutchins, J. (2000). *John W. Hutchins (Eds.), Early Years in Machine Translation*, chapter The first decades of Machine Translation: overview, chronology, sources, pages 1–16. John Benjamins B. V.

Kneser, R. and Ney, H. (1995). Improved Backing-Off for n-gram Language Modelling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing. Volume 1*, pages 181–184, Detroit, MI.

Knight, K. (1999). Decoding complexity in word-replacement translation models. *Computational Linguistics*, **25:4**:607–615.

Koehn, P. (2004). Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP '04)*, pages 388–395, Barcelona, Spain.

Koehn, P. (2005). Europarl: A Parallel Corpus for Statistical Machine Translation. In *Proceedings of the Machine Translation Summit X*, pages 79–86, Atlanta, GA.

Koehn, P. (2010). *Statistical Machine Translation*. Cambridge University Press, Cambridge, MA.

Koehn, P. and Haddow, B. (2009). Interactive Assistance to Human Translators using Statistical Machine Translation Methods. In *Proceedings of the Machine Translation Summit XII*, pages 73–80, Ottawa, Canada.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: open source toolkit for statistical machine translation. In *Proceedings of Demonstration and Poster Sessions at the 45th Annual Meeting of the Association of Computational Linguistics (ACL-07)*, Prague, Czech Republic.

Koehn, P., Och, F., and Marcu, D. (2003). Statistical Phrase-Based Translation. In *Proceedings of the Combined Human Language Technology Conference Series and the North American Chapter of the Association for Computational Linguistics Conference Series (HLT-NAACL '03)*, pages 48–54, Edmonton, Canada.

Leusch, G., Ueffing, N., and Ney, H. (2003). A novel string-to-string distance measure with applications to Machine Translation evaluation. In *Proceedings of the Machine Translation Summit IX*, pages 240–247, New Orleans, LO.

Li, Z. and Khudanpur, S. (2009). Efficient Extraction of Oracle-based Translations from Hypergraphs. In *Proceedings of the Combined Human Language Technology Conference Series and the North American Chapter of the Association for Computational Linguistics Conference Series (HLT-NAACL '09)*, pages 9–12, Boulder, Colorado, USA.

Liang, P., Bouchard-Cote, A., Klein, D., and Taskar, B. (2006). An end-to-end Discriminative Approach to Machine Translation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics(COLING-ACL '06)*, pages 761–768, Sydney, Australia.

Lin, C. Y. and Och, F. J. (2004). ORANGE: a Method for Evaluating Automatic Evaluation Metrics for Machine Translation. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, pages 501–507, Geneva, Switzerland.

Lopez, A. (2009). Translation as weighted deduction. In *Proceedings of the 12th Conference of the European Chapter of the Association for Computational Linguistics (EACL '09)*, pages 532–540, Athens, Greece.

Ma, W.-Y. and McKeown, K. (2009). Where's the verb? Correcting Machine Translation during Question Answering. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics and the 4th International Joint Conference on Natural Language Processing of the AFNLP (ACL-IJCNLP '09)*, pages 333–336, Suntec, Singapore.

Magerman, D. (1995). Statistical Decision-Tree Models for Parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computation Linguistics (ACL '95)*, pages 276–283, Cambridge, Massachusetts, USA.

Marcu, D. and Wong, W. (2002). A Phrase-based, Joint Probability Model for Statistical Machine Translation. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP '02)*, pages 133–139, Philadelphia, Pennsylvania, USA.

Marcus, M., Santorini, B., and Marcinkiewicz, M. (1993). Building a Large Annotated Corpus of English: The Penn Treebank. *Computational Linguistics*, **19:2**:313–330.

Nagao, M. (1984). A Framework of a Mechanical Translation between Japanese and English by Analogy Principle. In Elithom, A. and Banerji, R., editors, *Artificial and*

Human Intelligence: Edited Review Papers presented at the International NATO Symposium, October 181, pages 173–180. North Holland, Amsterdam.

Niessen, S., Och, F. J., Leusch, G., and Ney, H. (2000). An evaluation tool for machine translation: Fast evaluation for mt research. In *Proceedings of the 2nd International Conference on Language Resources and Evaluation (LREC '00)*, pages 39–45, Athens, Greece.

Nivre, J. (2006). *Inductive Dependency Parsing*. Springer Publishers, Netherlands.

Och, F. (2003). Minimum Error Rate Training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting of the Association for Computation Linguistics (ACL-03)*, pages 160–167, Sapporo, Japan.

Och, F. and Ney, H. (2003). A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*, **29:1**:19–51.

Och, F. J., Gildea, D., Khudanpur, S., Sarkar, A., Yamada, K., Fraser, A., Kumar, S., Shen, L., Smith, D., Eng, K., Jain, V., Jin, Z., and Radev, D. (2004). A Smorgasbord of Features for Statistical Machine Translation. In *Proceedings of the Combined Human Language Technology Conference Series and the North American Chapter of the Association for Computational Linguistics Conference Series (HLT-NAACL '04)*, pages 161–168, Boston, MA.

Och, F. J. and Ney, H. (2002). Discriminative Training and Maximum Entropy Models for Statistical Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computation Linguistics (ACL '02)*, pages 295–302, Philadelphia, PA.

Okita, T. (2012). *Word Alignment and Smoothing Method in Statistical Machine Translation: Noise, Prior Knowledge, and Overfitting*. PhD thesis, Dublin City University, Dublin, Ireland.

Olteanu, M., Suriyentrakorn, P., and Moldovan, D. (2006). Language Models and Reranking for Machine Translation. In *Proceedings of the First Workshop on Statistical Machine Translation (WMT '06)*, pages 150–153, New York City, New York, USA.

Owczarzak, K. (2008). *A Novel Dependency-based Evaluation Metric for Machine Translation*. PhD thesis, Dublin City University, Dublin, Ireland.

Papineni, K., Roukos, S., Ward, T., and Zhu, W. (2002). BLEU: A Method for Automatic Evaluation of Machine Translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics (ACL '02)*, pages 311–318, Philadelphia, PA.

Penkale, S., Haque, R., Dandapat, S., Banerjee, P., Srivastava, A. K., Du, J., Pecina, P., Naskar, S. K., Forcada, M. L., and Way, A. (2010). MATREX: The DCU MT system for WMT 2010. In *Proceedings of the Joint Fifth Workshop on Statistical Machine Translation and MetricsMATR at the 48th Annual Meeting of the Association for Computational Linguistics (ACL '10)*, pages 143–148, Uppsala, Sweden.

Petrov, S., Barrett, L., Thibaux, R., and Klein, D. (2006). Learning Accurate, Compact, and Interpretable Tree Annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics(COLING-ACL '06)*, pages 433–440, Sydney, Australia.

Probst, K., Levin, L., Peterson, E., Lavie, A., and Carbonell, J. (2002). MT for minority languages using elicitationa-based learning of syntactic transfer rules. *Machine Translation*, **17:4**:245–270.

Quirk, C., Menezes, A., and Cherry, C. (2005). Dependency Treelet Translation: Syntactically-informed Phrasal SMT. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics (ACL '05)*, pages 271–279, Ann Arbor, MI.

Ratnaparkhi, A. (1996). A Maximum Entropy Model for Part-Of-Speech Tagging. In *Proceedings of the Empirical Methods in Natural Language Processing (EMNLP '96)*, pages 133–142, Philadelphia, Pennsylvania, USA.

Sanchez-Martinez, F. and Forcada, M. (2007). Automatic Induction of Shallow-transfer rules for Open-source Machine Translation. In *Proceedings of the Eleventh Conference on Theoretical and Methodological Issues in Machine Translation (TMI-07)*, pages 181–190, Skovde, Sweden.

Sanchez-Martinez, F. and Way, A. (2009). Marker-based Filtering of Biingual Phrase Pairs for SMT. In *Proceedings of the 13th Conference of the European Association for Machine Translation (EAMT '06)*, pages 144–151, Barcelona, Spain.

Shen, L., Sarkar, A., and Och, F. J. (2004). Discriminative Reranking for Machine Transation. In *Proceedings of the Combined Human Language Technology Conference Series and the North American Chapter of the Association for Computational Linguistics Conference Series (HLT-NAACL '04)*, pages 177–184, Boston, MA.

Shen, L., Xu, J., and Weischedel, R. (2008). A New String-to-Dependency Machine Translation Algorithm with a Target Dependency Language Model. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics Short Papers (ACL '08)*, pages 577–585, Columbus, OH.

Snover, M., Dorr, B., Schwartz, R., Micciula, L., and Makhoul, J. (2006). A Study of Translation Edit Rate with targeted Human Annotation. In *Proceedings of the 7th biennial conference of the Association for Machine Translation in the Americas (AMTA '06)*, pages 223–231, Cambridge, MA.

Srivastava, A., Penkale, S., Groves, D., and Tinsley, J. (2009). Evaluating Syntax-Driven Approaches to Phrase Extraction for MT. In *Proceedings of the 3rd International Workshop on Example-based Machine Translation*, pages 19–28, Dublin, Ireland.

Srivastava, A. K., Ma, Y., and Way, A. (2011). Oracle-based Training for Phrase-based Statistical Machine Translation. In *Proceedings of the 15th Annual Conference of the European Association for Machine Translation*, pages 169–176, Leuven, Belgium.

Srivastava, A. K. and Way, A. (2009). Using Percolated Dependencies for Phrase Extraction in SMT. In *Proceedings of the Machine Translation Summit XII*, pages 316–323, Ottawa, Canada.

Stolcke, A. (2002). SRILM - An Extensible Language Modeling Toolkit. In *Proceedings of the International Conference on Spoken Language Processing*, pages 901–904, Denver, Colorado, USA.

Tiedemann, J. (2009). News from OPUS – A Collection of Multilingual Parallel Corpora with Tools and Interfaces. In N.Nicolov, Bontcheva, K., G.Angelova, and Mitkov, R., editors, *Recent Advances in Natural Language Processing Volume 5*, pages 237–248. John Benjamins, Philadelphia, PA.

Tinsley, J. (2010). *Resourcing Machine Translation with Parallel Treebanks*. PhD thesis, Dublin City University, Dublin, Ireland.

Tinsley, J., Hearne, M., and Way, A. (2007). Exploiting Parallel Treebanks to Improve Phrase-Based Statistical Machine Translation. In *Proceedings of the Sixth International Workshop on Treebanks and Linguistic Theories (TLT '07)*, pages 175–187, Bergen, Norway.

Toral, A., Naskar, S. K., Gaspari, F., and Groves, D. (2012). DELiC4MT: A Tool for Diagnostic MT Evaluation over User-defined Linguistic Phenomena. *Prague Bulletin of Mathematical Linguistics, Special Issue: Open Source Tools for MT*, **98**:121–132.

Tu, Z., Liu, Y., Hwang, Y., Liu, Q., and Lin, S. (2010). Dependency Forest for Statistical Machine Translation. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING 2010)*, pages 1092–1100, Beijing, China.

van Valin Jr., R. (2001). *An Introduction to Syntax*. Cambridge University Press, United Kingdom.

Vilar, D., Xu, J., D'Haro, L. F., and Ney, H. (2006). Error analysis of statistical machine translation output. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC '06)*, pages 697–702, Genoa, Italy.

Wang, W., May, J., Knight, K., and Marcu, D. (2010). Re-structuring, Re-labeling, and Re-aligning for Syntax-based Machine Translation. *Computational Linguistics*, **36:2**:247–278.

Weaver, W. (1949). *Recent Contributions to the Mathematical Theory of Communication. In Shannon, C.E. and Weaver, W., (Eds.), The Mathematical Theory of Communication*, pages 94–117. The University of Illinois Press, Urbana, IL.

Wisniewski, G. and Yvon, F. (2013). Oracle Decoding as a New Way to Analyze Phrase-based Machine Translation. *Machine Translation*, **27**:115–138.

Xia, F. (2001). *Automatic Grammar Generation from Two Different Perspectives*. PhD thesis, University of Pennsylvania, Philadelphia, PA.

Yamada, K. and Muslea, I. (2009). Reranking for Large-Scale Statistical Machine Translation. In Goutte, C., Cancedda, N., Dymetman, M., and Foster, G., editors, *Learning Machine Translation*, pages 151–168. MIT Press, Cambridge, MA.

Ye, Y., Zhou, M., and Lin, C.-Y. (2007). Sentence-level Machine Translation Evaluation as a Ranking Problem: One Step Aside from Bleu. In *Proceedings of the Second Workshop on Statistical Machine Translation at the 45th Annual Meeting of the Association for Computational Linguistics (ACL '07)*, pages 240–247, Prague, Czech Republic.

Zens, R. and Ney, H. (2006). N-gram Posterior Probabilities for Statistical Machine Translation. In *Proceedings of the Combined Human Language Technology Conference Series and the North American Chapter of the Association for Computational*

Linguistics Conference Series (HLT-NAACL '06), pages 72–77, New York City, New York, USA.

Zhechev, V. (2009). Unsupervised Generation of Parallel Treebank through Sub-Tree Alignment. *Prague Bulletin of Mathematical Linguistics, Special Issue: Open Source Tools for MT*, **91**:89–98.

Zollmann, A., Venugopal, A., Och, F., and Ponte, J. (2008). A Systematic Comparison of Phrase-Based, Hierarchical and Syntax-Augmented Statistical MT. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING '08)*, pages 1145–1152, Manchester, UK.