Effective Learning of Pre-ordering Rules for Machine Translation

Md Ahmar Hashmi
Roll No: 143050014

—

Guided By
Prof. Ganesh Ramakrishnan
and
Prof. Amitabha Sanyal

Computer Science and Engineering Department
Indian Institute of Technology Bombay

2nd April, 2016

Outline

Reordering in machine translation

- Machine translation heavily relies on appropriate reordering of sentences and their word order.
- Phrase based statistical systems lack ability of long-range reordering.
- This is because it does not consider any syntactic or linguistic phenomena specific to the language pair.

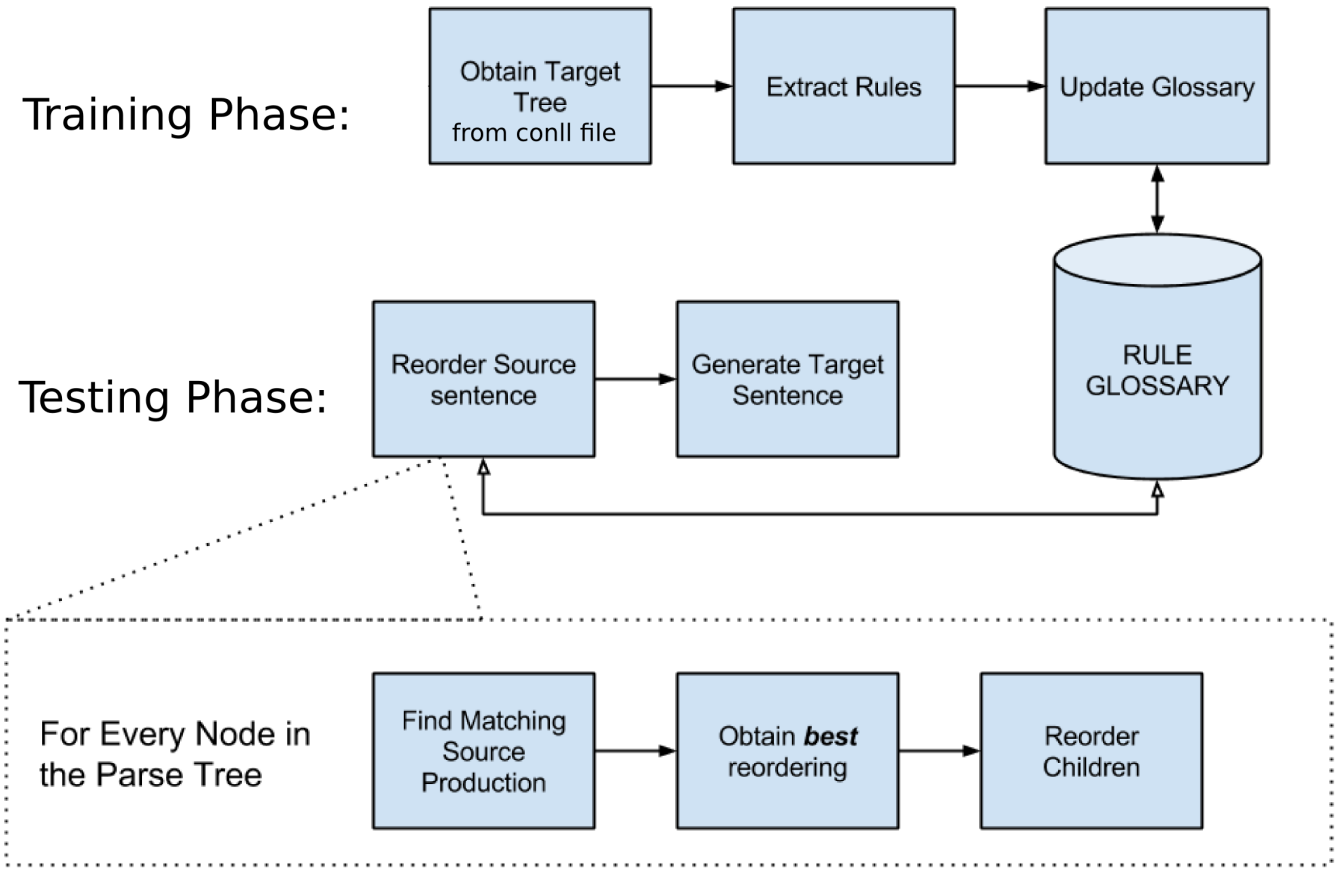
Learning Pre-ordering Rules

- A phrase based SMT can be complemented with a pre-ordering component
- Sentences are pre-ordered to match target-side word order
- System is trained and later tested with such pre-ordered sentences
- Manually writing pre-ordering rules is tedious, requires expertise
- Pre-ordering rules can be learnt using aligned sentence pairs.

Program Synthesis and Least General Generalisations

- We view the problem of learning pre-ordering rules as that of synthesizing a program to predict an appropriate target-side word-order given the source side sentence
- We club similar rules together using the concept of LGGs,ie,the least upper bound in the subsumption lattice
- Kitzelmann, 2011 [1] uses Least General Generalisations to learn declarative rules that are program segments
- Example of least general generalisations of list expressions: $LGG(x1 : (x2 : (x3 : [])), (x1 : (x2 : [])) = (x1 : (x2 : xs))$

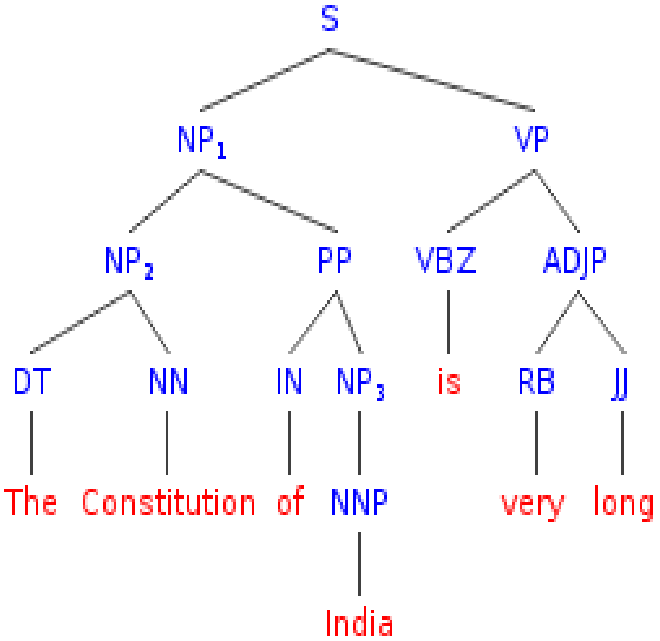
Architecture of our system

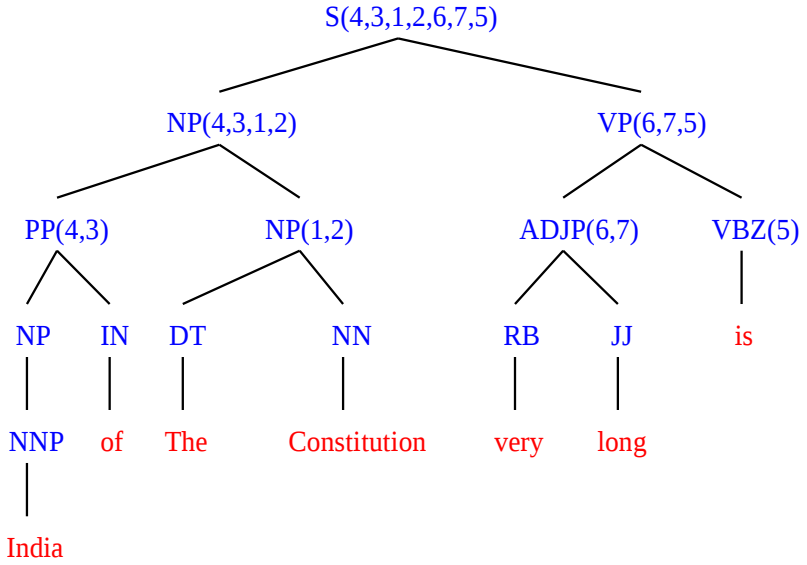


System details

1. Stanford Parser [2] used to parse the input sentence, with tokenization turned off, since POS tags also given along with each word.
2. Alignment of training dataset also given.
3. Each rule contains
   - Source production, say $A \rightarrow \alpha$
   - List of possible reorderings for the production $\{ A \rightarrow \alpha_1 , A \rightarrow \alpha_2, \ldots, A \rightarrow \alpha_n \}$
   - Each reordering contains:
     - T_A, a parse tree rooted at A as a reference to context
     - Frequency count of the reordering
     - Array storing the depths in the source tree at which this rule occured
4. The rule learning process is incremental in nature.

Obtaining reordered tree: example



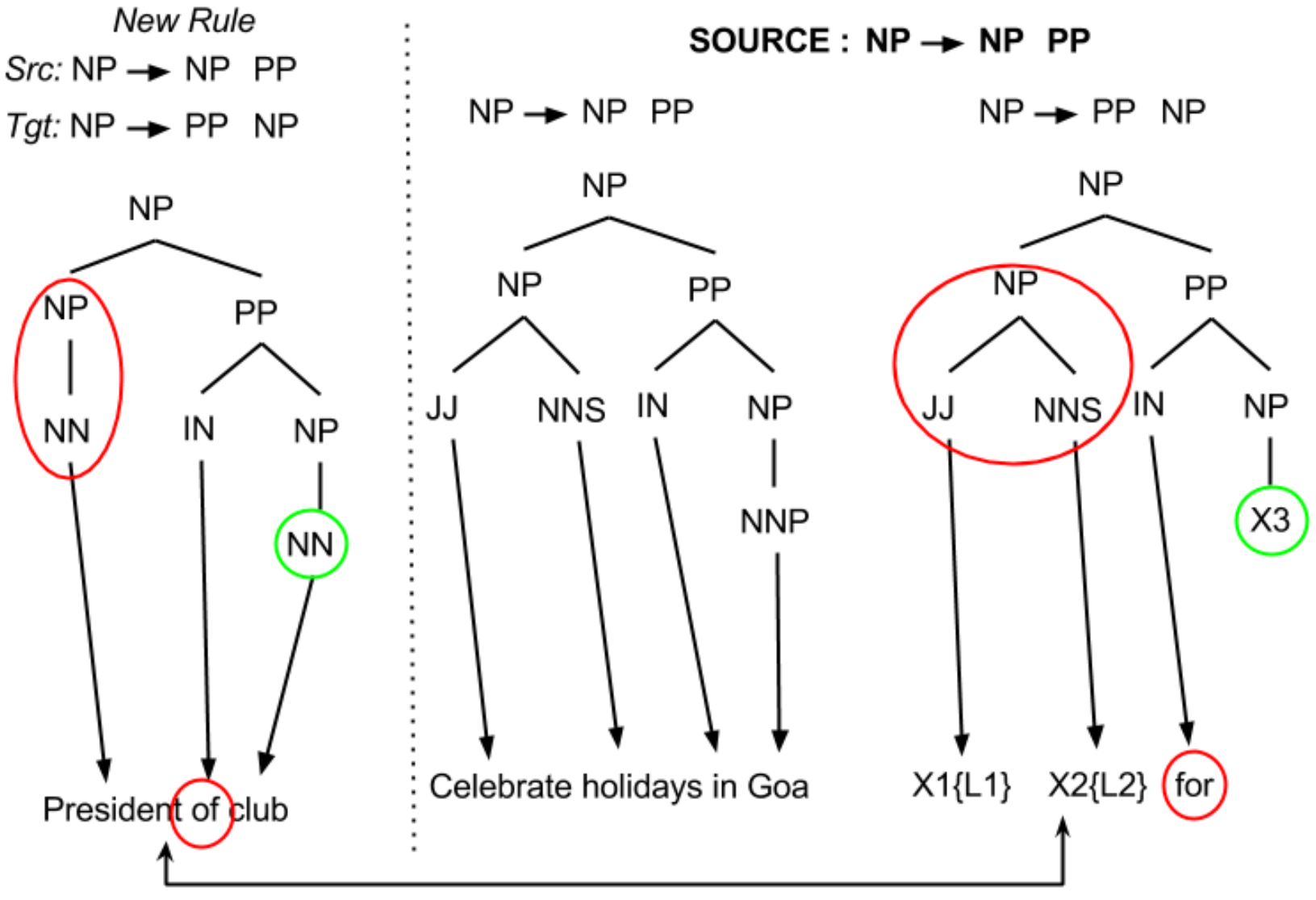


Extracting rules: example

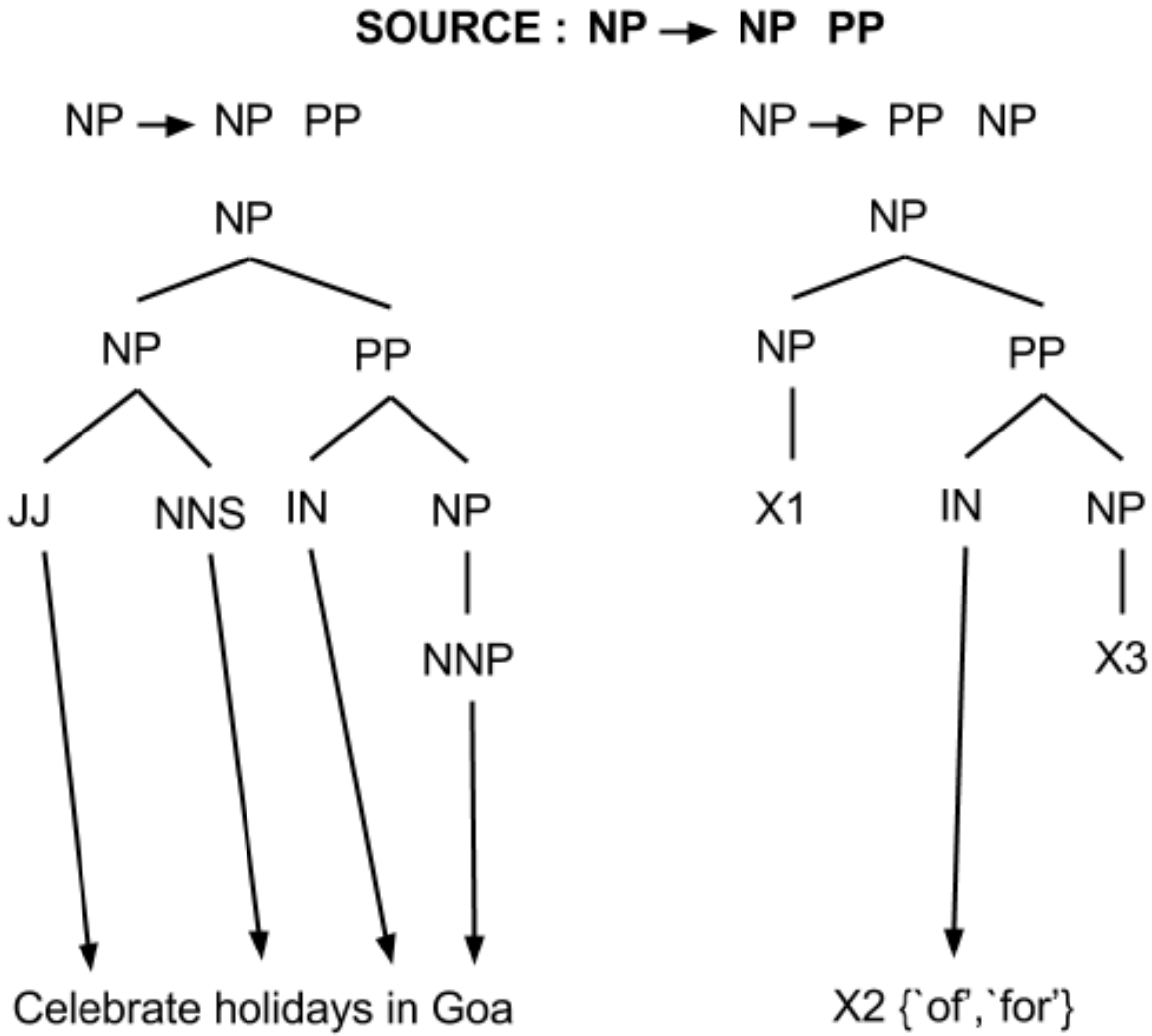
Source production	Target production	Context Tree
$NP \rightarrow NP\ PP$	$NP \rightarrow PP\ NP$	[NP [NP [DT [The]] [NN [Constitution]]] [PP [IN [of]] [NP [NNP [India]]]]]
$PP \rightarrow IN\ NP$	$PP \rightarrow NP\ IN$	[PP [IN [of]] [NP [NNP [India]]]]
$VP \rightarrow VBZ\ ADJP$	$VP \rightarrow ADJP\ VBZ$	[VP [VBZ [is]] [ADJP [RB [very]] [JJ [long]]]]

Table : Rules learnt

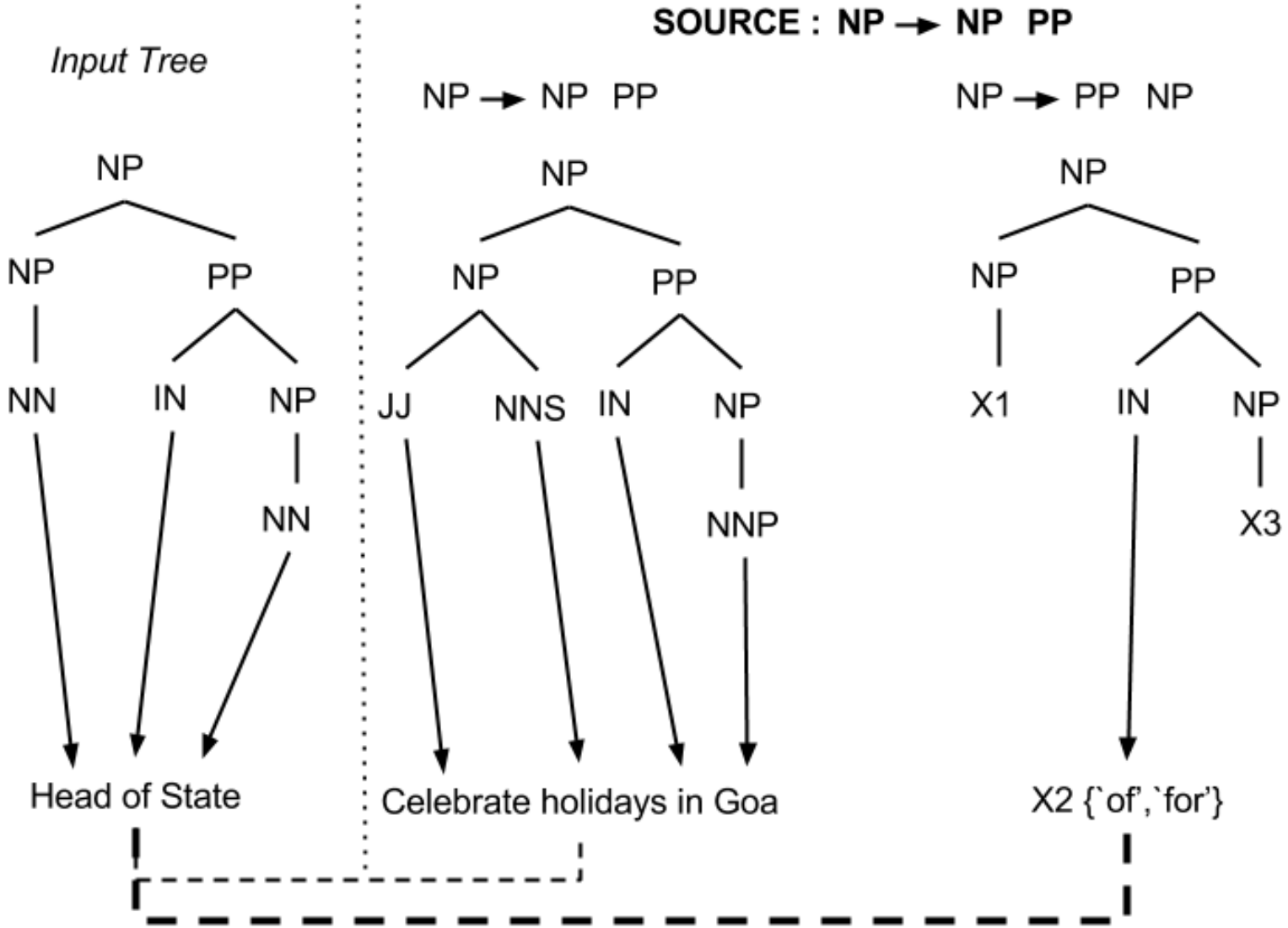
Merging rules to glossary: example



Rules generated



Reordering an input tree: example



Rule selection method

1. If a given source production, has multiple target reorderings in the glossary, we need to select the *best* out of them.
2. For doing this, we take the help of a tool called RankLib which basically ranks the target reorderings on the basis of their feature values.
3. The chosen features are:
    1. Subsumption check
    2. Relative MatchScore
    3. Relative frequency
    4. Relative depth
4. During validation phase, a dataset containing 500 sentences is used to learn the parameters for the model chosen.
5. During testing phase, we find the feature vector for every source side production, and based on the rules available in the glossary, we find the scores for each. The rule with the highest score is chosen.

Experiment-Results

System	En-Fa mBLEU	En-It mBLEU	En-Ur mBLEU
Baseline	50.0	65.1	38.3
Kunchukuttan[3]	46.4	64.7	37.8
Gupta et al. (2012)	55.7	73.0	44.7
Our system	**61.57**	**72.05**	**56.42**
Dlougach [4]	65.6	76.7	55.8
Visweswariah et al.[5]	68.7	83.0	63.3

Table : Results on the reordering shared task on test dataset

Critical Comments

- Above baseline translation scores achieved on test data for all three language pairs
- Italian is seen to perform the best amongst all the three languages, owing, to its similarity with the english language, as both belong to the same family of European languages.

Comparison with other existing works

- Dlougach and Galinskaya, 2013 [4]:
  - Syntax based reordering system using Moses
  - Uses quite a similar approach as ours, except that the rules learnt are flattened out to reorder a span labeled word sequence rather than tree nodes.
  - Takes advantage of both syntactic reordering as well as lexical features.
- Visweswariah et al.[5] :
  - Views the reordering problem as Asymmetrical Travelling Salesman Problem(ATSP) with words as cities and pairwise costs as edge weights
  - The tour with the least cost is the predicted reordering of the sentence
  - Objective is to learn the edge weights
  - Found to perform the best till now.

Scope for future work

- Including lexical features or word hierarchies in LGGs at lexical level. Eg. LGG of pen,pencil is also NN, LGG of pen,football is also NN.
- Extending same approach to work with dependency parses.

References I

Emanuel Kitzelmann.
A combined analytical and search-based approach for the inductive synthesis of functional programs.
KI-Künstliche Intelligenz, 25(2):179–182, 2011.

Dan Klein and Christopher D Manning.
Accurate unlexicalized parsing.
In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics-Volume 1*, pages 423–430. Association for Computational Linguistics, 2003.

Kunchukuttan and Bhattacharya.
Partially modelling word reordering as a sequence labelling problem.
In *24th International Conference on Computational Linguistics*, page 47, 2012.

References II

Jacob Dlougach and Irina Galinskaya.
Building a reordering system using tree-to-string hierarchical model.
arXiv preprint arXiv:1302.3057, 2013.

Karthik Visweswariah, Rajakrishnan Rajkumar, Ankur Gandhe, Ananthakrishnan Ramanathan, and Jiri Navratil.
A word reordering model for improved machine translation.
In *proceedings of the conference on empirical methods in natural language processing*, pages 486–496. Association for Computational Linguistics, 2011.

Acknowledgements

- Prof. Ganesh Ramakrishnan and Prof. Amitabha Sanyal for their valuable guidance
- Mr. Anamay Tengse for clearing ad-hoc doubts with patience
- Mrs. Simoni Shah for much needed help in code debugging and with experiment setups

Thank You!