NLP Tools Contest - 2008: Summary

Sriram Venkatapathy
Language Technologies Research Centre,
International Institute of Information
Technology - Hyderabad,
Hyderabad, India.
{sriram}@research.iiit.ac.in

Abstract

NLP Tools Contest 2008 was held along with International Conference on Natural Language Processing. The goal of this contest was to explore statistical machine translation techniques from English to Indian Languages and vice versa.

12 teams registered in the contest out of which, 5 papers were accepted for oral presentation at the contest venue on 19th December, 2008. There were wide variety of ideas which were explored and were well-suited in the context of English-Hindi Machine Translation. In this document, I summarize the ideas and techniques tested by the participants along with the results.

1 Introduction

Machine translation involves automatic translation of sentences in one language to another. The statistical approaches for doing machine translation have proved effective in recent years, specially for translating between European languages. However, there is a need to explore its effectiveness in translating sentences from English to Indian languages or vice versa.

The aim of this shared task was to collectively explore a variety of ways of combining statistical techniques with linguistic inputs to improve a baseline statistical machine translation system from English to Hindi.

2 Training Data

The participants were provided two different datasets.

1. TIDES-IIIT Dataset

   This dataset was initially collected for the DARPA-TIDES surprise language contest on Statistical Machine Translation in 2002. This automatically collected corpus had 64k sentence pairs. To reduce the large amount of noise in the dataset, manual refinement was carried out at IIIT-Hyderabad. The size of this dataset is 54k sentence pairs of which, 50k sentence pairs are used for training, 1k as a development corpus and 1k as a test corpus.

   This corpus is a general domain dataset with news articles forming the greatest proportion. The sentences are not faithful translations ie., the target sentences conveyed only the meaning of the source sentences in the best possible way in the target language. So, they were really, the paraphrased translations.

2. EILMT-Tourism Corpus

   This dataset was provided by the EILMT consortium funded by DIT, Govt. of India. We were allowed to use only a subset of the entire dataset because of project considerations. This dataset contained 7K training sentence pairs, 500 sentence pairs for development and 500 sentence pairs for testing.

   This dataset is a domain-specific corpus developed purely to build machine translation system catering to tourism domain. This corpus

is extremely clean as all the translations were manually entered.

One of the participants **dcu** (See section 3) presented a comparision of the two datasets in their paper. I take the liberty of presenting their comparision in this summary (Table 1).

	TIDES	EILMT
Total sentences	50000	7000
|En|>100, or, |Hn|>100	123	1
Fertility>2	314	32
Highest E-H word ratio	16.5	4.7
Lowest E-H word ratio	.08	.17
Highest E-H char ratio	82.5	5.6
Lowest E-H char ratio	.17	.23

Table 1: Summary of training data

Table 1 shows that the TIDES dataset is relatively noisy when compared to EILMT dataset, and hence might yield lower machine translation scores.

3 Teams

In Table 2, I present the list of participating teams.

Team	Members	Institute
indians	Karthik Gali Avinesh PVS Taraka	IIIT-H
nlp_iitb	Om Damani Vasudevan Amit	IIT-B
hakunamatata	Sumit Goswami Nirav Shah Devshri Roy S Sarkar	IIT-Kgp
dcu	Andy Way Sudip Naskar Ankit Srivastava Rejwanul Haque	Dublin City University
ufal	Ondrej Bojar Pavel Stranak Daniel Zeman	Charles university in Prague

Table 2: Participating teams

4 Baseline

For a baseline statistical machine translation from English to Hindi, we suggested Moses to the participants[1]. Most of the participants, infact, made use of Moses and incorporated various pre-processing and post-processing steps in order to obtain better accuracies than the baselines. I summarize all the techniques adopted by the participants in Section 5.

5 Summary of Techniques

I will use the names of the teams given in table 2 to refer to the teams in this section.

The teams **indians**, **nlp_iitb**, **hakunamatata** and **ufal** used **MOSES** as a platform to conduct their experiments. The team **dcu** used their own system **MATREX** as a platform. This is a Hybrid EBMT/SMT System. However, they did not test the EBMT component in this contest and relied only of SMT component which functions in the same way as Moses.

In this section, wherever I refer to the translation accuracy, I mean BLEU score which is an automatic evaluation metric. Also, for general comparison, I consider the EILMT-Tourism corpus as it has relatively less noise.

5.1 Reordering

One of the major issues while translating from English to Hindi is large word-reordering. All the participants (except one) handled this by reordering the English parse tree according to the Hindi language word order. The team **ufal** did a shallow reordering of the source sentence.

5.1.1 ufal

They applied rule based shallow reordering during the pre-processing step. The rules that they applied are,

1. Move the finite verb to the end of sentence. A constraint is imposed not to cross punctuations, 'that' and wh-words.

2. Move the prepositions to the position immediately after the corresponding content word.

[1] http://www.statmt.org/wmt07/baseline.html

They observed that these reordering rules give marginal improvements in the translation accuracies.

5.1.2 dcu

They applied simple heuristic rules on the English phrase structure tree to obtain the Hindi word order. To obtain the phrase structure tree, they used the *Stanford Parser*. A list of sample rules is,

- VP NP → NP VP
- NP PP → PP NP
- Prep NP → NP Prep
- Vaux VP → VP Vaux

5.1.3 nlp_iitb

They applied well-formulated rules on the *Stanford Dependency Parser* to reorder it. Their reordering at every node of the source tree consists of two stages,

1. Computing parent-child order based on dependency relationship between them. For example, in Hindi, the direct object of a verb comes before the verb. Hence, their rules says, if *dobj*(Parent,Child), then Child-Position<Parent-Position.

2. Computing relative order of siblings in case of multiple children. They use a preference table to order the siblings. For example, if a parent has two children with relationships *nsubj* and *dobj* respectively, their preference table would indicate that *nsubj*<*dobj*.

They obtained a large improvement in translation accuracy when they applied reordering on the English parse structure. Stanford parser seems appropriate for this task, as the relation-types are large allowing formulation of detailed and precise rules.

5.1.4 indians

In contrast to the other systems, they attempted to learn the transfer rules automatically from the training corpus. The steps that they follow are,

1. Run a dependency parser on English training sentences. They used Libin Shen's parser for their experiments.

2. Run GIZA++ (word-alignment tool) to get the alignment of English words with the Hindi words.

3. At every node in the English parser, observe the relative order of source words in Hindi sentence to infer a transfer rule. These rules are based on part-of-speech tags of the words.

A sample rule learnt by their system is,

- IN1_NN&_VB2 → NN&_IN1_VB2

Here, '&' marks the part-of-speech tag of the root. The above rules are applied for reordering the test sentences. In addition to these rules, they also tested an heuristic of moving verbs to the end of clauses.

5.2 Transliteration

All the participants suggested that transliteration is a good option for improving the system, none except **dcu** were able to conduct experiments in this direction. It gave a marginal improvement in the translation accuracy.

5.3 Factored Approach

The English-Hindi corpus is extremely small as compared to the corpora between European languages. Hence, to address the issue of sparsity, factorizing words in their morphological features is expected to give better results. Most of the teams attempted this approach. The team **ufal** used an unsupervised stemmer to break words into stem-suffix pairs. Other teams **dcu**, **indians** and **hakunamatata** also tried this but this did not yield fruitful results for any team. The probable reason could be that the participating teams could not use an accurate morphological analyzer for Hindi. Other reasons need to be explored.

5.4 Lexicon

The teams **dcu**, **indians** and **hakunamatata** augmented their systems with a manually developed English-Hindi dictonaries. **hakunamatata** developed a methodology to include lexicon into the phrase-table which is produced by Moses. They present heuristics to assign probabilities and lexical weightages to these entries. **indians** add the lexicon to the parallel corpora. **hakunamatata** used

English-Hindi Shabdakosh[2] while **indians** used English-Hindi Shabdanjali[3] for their experiments.

indians also used a tense dictionary (mapping from English tense markers to Hindi tense markers). All the teams only obtained a marginal improvement in their translation accuracies.

5.5 Larger language model

The teams **ufal** and **indians** experiment with large language models but the trend was inconclusive. This could be because the additional monolingual data that was used to build these larger models came for other domains.

However, **ufal** showed that larger parallel data to build the phrase table was quite helpful.

5.6 Miscellaneous tricks

Here, I present some of the other methods adopted by teams in order to get good improvements in the translation accuracy.

dcu applied the strategy of removing noisy sentences from the corpora. They removed sentences above sentence length of 100 from the training corpora. Also, those sentence pairs were removed whose sentence-lengths-ratio was greater than 2:1. This gave them large improvement, specially for the TIDES-IIIT corpus.

indians removed all the phrase pairs which had the lexical item '.' on either side except the entry '.' $\Rightarrow$ '.'.

ufal discovered an interesting case. When the alignment heuristic 'grow-diag-final' (in Moses) was used instead of 'grow-diag-final-and', the translation accuracy decreased by 5 BLEU points. This was not the case for other language pairs such as English-French or English-Czech which makes it interesting.

6 Evaluation & Results

The teams were evaluated using the BLEU scores. BLEU (Papineni et al., 2002) is a standard n-gram metric used by the machine translation community to compare the systems' outputs with the reference sentences. We first present the best results of various teams as well as the baseline results that they obtained using Moses.

[2]http://www.Sabdhakosh.com/

[3]http://ltrc.iiit.ac.in/downloads/

There is a difference in baseline scores as different teams used different parameter settings to define their baseline.

Team	Baseline		Best
indians	-		-
nlp_iitb	0.0842		0.0853
hakunamatata	-		-
dcu	0.0487		0.1049
ufal	0.1006		0.1029

Table 3: BLEU on TIDES-IIIT corpus

Team	Baseline		Best
indians	0.2018		0.2260
nlp_iitb	0.1450		0.1751
hakunamatata	0.1649		0.1873
dcu	0.1635		0.1741
ufal	0.1888		0.2101

Table 4: BLEU on EILMT-Tourism corpus

Evaluation using BLEU scores is not very effective while measuring the translation accuracies while translating into Indian Languages. The reason is that Indian Languages are relatively free-word ordered and the n-gram based metric may not be the best. Hence, we plan to measure the subjective evaluation score for each system and update those scores in this document.

The criteria proposed for subjective evaluaton involves rating each translation according to the following scale.

1. Doesn't make sense
2. Bad translation
3. Average translation with major errors
4. Good translation with minor errors
5. Good translation

References

Kishore Papineni, Salim Roukos, Todd Ward, and W.J. Zhu. 2002. Bleu: A method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association of Computational Linguistics*, pages 313–318, Philadelphia, PA, July.