English–Hindi Translation in 21 Days

Ondřej Bojar, Pavel Straňák, Daniel Zeman*
Univerzita Karlova v Praze, ÚFAL,
Malostranské nám. 25, 118 00, Praha, Czech Republic
{bojar,stranak,zeman}@ufal.mff.cuni.cz

Abstract

We present experiments with a Moses-based English-to-Hindi translation system. We evaluate the impact of additional out-of-domain training data, both parallel and Hindi-only, and experiment with three methods for improving word order: standard Moses reordering model, rule-based pre-processing and language-independent suffix identification.

1 Data

1.1 Parallel

We tried to obtain as much parallel data as possible:

EILMT+TIDES: The parallel data provided by the organizers; 7k and 50k sentences.[2]

Emille: The parallel part of this corpus consists of 200k words of text in English and its accompanying translations in Hindi and other languages.

Agriculture domain parallel corpus: Resource Center for Indian Language Technology Solutions English-Hindi-Marathi-UNL parallel corpus. It contains 17,105 English and 13,248 Hindi words.

Daniel Pipes: D. Pipes' website:[3] a limited-domain articles about the Middle East. 322 texts are available in Hindi.

1.2 Hindi

For Hindi language model, we generally use the target side of the parallel data mentioned above. To extend the data, we could use e.g. the monolingual sections of the Emille corpus.

These data are still quite small compared to data regularly used for e.g. English language models, so we decided to create a big monolingual corpus of Hindi ourselves. Starting from Hindi news sites lists we downloaded 27 websites, mostly news portals. After clean-up this amounted for 18.1M sentences and 309M tokens.

Downloading the data itself was done very simply.[4] We then proceeded to clean-up the HTML and classify the texts by languages. The language classification is based on a model comparing frequencies of three-character suffixes of word forms with known suffix frequencies for each language. If no clearly-winning language is found, we back off to three simple models counting 1-, 2-, and 3-grams of characters.

[1]This work was supported by grants of Czech Academy of Sciences: 1ET201120505, 1ET101470416, European Union: FP6-IST-5-034291-STP, and Ministry of Education: MSM0021620838.

[2]We acknowledge the Department of IT (DIT), Government of India and the English-to-Indian Languages (EILMT) consortium for making the EILMT tourism dataset available.

[3]http://www.danielpipes.org/

[4]wget -r –user-agent=”” –timestamping –no-parent ”$1”

Finally we filtered the texts classified as Hindi by deleting all the lines that did not contain any Devanagari.

1.3 Tokenization

For the data supplied by the organizers, we stick to their sentence segmentation and tokenization. For the additional data, we use a trainable tokenizer (Klyueva and Bojar, 2008) that can be easily adapted to a new language simply by providing a few instances of sentence and token breaks.

2 Translation Setup

2.1 Baseline and Evaluation

We use a variation of Moses (Koehn et al., 2007) standard pipeline. Word alignments are obtained using GIZA++ (grow-diag-final-and heuristic). To reduce data sparseness for word-alignment estimation, we consider only first four letters (lowercased) of each English and Hindi (UTF-8) word form.

The baseline setup is single-factored (i.e. plain phrase-based) wordform-to-wordform translation with distance-based reordering model. As a basic contrast experiment, we use a standard reordering model.[5]

We evaluate the translation quality using our implementation of BLEU with 95% confidence intervals obtained using the bootstrapping method by (Koehn, 2004). Due to the built-in tokenization rules in `mteval-v11b` (and not in our system), the scores differ slightly.

We use Moses standard MERT (Och, 2003) to tune weights of the individual models in our setup. The final scores reported are evaluated on the unseen official Test set. Unless stated otherwise, we use only the EILMT Development and Test sets.[6].

Our baseline results as well as the impact of additional parallel data are given in Table 1.

In our original experiments we used grow-diag-final (no -and) heuristic but this lead to about 4 times smaller set of extracted phrases and BLEU scores lower by about 5 points (e.g. 13.82±1.46 instead of 18.88±2.05). We are grateful to the organizers for helping us find the difference in the configuration. The exact reason of this loss in extracted phrases seems related to reorderings necessary for English-Hindi pair and has yet to be clarified.

2.2 Web LM

Table 2 summarizes the impact of using a 4-gram language model (LM) based on 309M Hindi tokens (see Section 1.2), called webLM, in addition to the 3-gram model based on the Hindi side of EILMT. We use a separate weight for webLM in the MERT training to give Moses a chance of a weak domain adaptation.

Unfortunately, our results indicate that for the given domain of EILMT Test data the additional LM brings no improvement. This could be caused by three reasons: 1) the domain of EILMT is very specific and different from webLM, 2) webLM contains too much noise despite our attempts to keep only real Hindi sentences, 3) the tokenization of webLM is different from EILMT data.[7]

2.3 Unsupervised Stem-Suffix Segmentation

To lessen the impact of data sparseness, we can use Moses' ability to work with factors. Splitting each word into the lexical stem and the grammatical suffix (thus defining two separate factors) is the obvious option here. However, being unfamiliar with Hindi morpholog-

[5] Orientation-bidirectional based on both source and target side tokens.

[6] 500 sentences each, 1 ref. translation per sent.

[7] The tokenizer we use can be easily trained to mimic any tokenization style given *both* the original non-tokenized text and the intended tokenized form, but we had no access to the non-tokenized version of EILMT.

Parallel data	Distance Reordering		Reordering Using en+hi Forms	
	`mteval-v11b`	BLEU	`mteval-v11b`	BLEU
EILMT (7k sents)	19.07	18.88±2.05	19.98	19.77±2.03
EILMT+Tides (57k sents)	19.41	19.27±2.22	20.39	20.29±2.17
EILMT+Tides+Our Additions (77k sents)	**20.20**	**20.07±2.21**	**20.72**	**20.57±2.15**

Table 1: Impact of reordering model and parallel data size.

Parallel Data	Language Models	Distance Reordering	Reordering Using en+hi Forms
EILMT	EILMT+webLM	19.62±2.06	19.31±2.10
EILMT	EILMT	18.88±2.05	19.77±2.03
EILMT+Tides+Additions	EILMT+webLM	18.82±2.13	19.39±2.11
EILMT+Tides+Additions	EILMT	**20.07±2.21**	**20.57±2.15**

Table 2: Impact of additional language model.

ical analysers and taggers, we decided to employ a tool for unsupervised segmentation of words into morphemes. The tool was originally published in the context of information retrieval (Zeman, 2008).

The tool has been trained on the word types of the Hindi side of the Tides corpus. For every word the algorithm searches for positions where it can be cut in two parts: the stem and the suffix. Then it tries to filter the stem and suffix candidates so that real stems and suffixes remain. The core idea is that real stems occur with multiple suffixes and real suffixes occur with multiple stems. For the purpose of filtering, a collection of stem and suffix candidates that have been observed together is called a *paradigm*.

Various techniques are applied to filter out spurious candidates:

1. If there are more suffixes than stems in a paradigm, the paradigm is removed.
2. If all suffixes in a paradigm begin with the same letter, there is another paradigm where the letter is part of the stem. The former paradigm is removed. Example:
   suffixes: त, ति, तियां, तियों
   stems: अनुभू, अभिव्‌यक्, अभिव्यक्, सस्‌कृ
3. If the suffixes of paradigm B form a subset of suffixes of paradigm A ($A \subset B$) and there is no C, different from A, such that B is also subset of C: $\forall C \neq A : (B \not\subset C)$, we add the stems of B to the stems of A, and remove B. A subset paradigm is merged with its superset, as long as there is only one superset candidate.
4. Paradigms with only one suffix are removed.

The following is an example of a paradigm that survived the filtering:

- suffixes: 0,ा,े,ों
- stems: अहात खांच घुटन चढ़ाव ज्हटक दायर बचन भाल मदरस मसाल मेमन सितार ख़र्च

Given the lists of stems and suffixes obtained during training, we want to find the stem-suffix boundary in a word of the same language. Theoretically, we could use the learned stem-suffix combinations to require that both stem and suffix be known. However, this approach proved too restrictive, so we ended up in using just the list of suffixes. If a word ends in a string equal to a known suffix, the morpheme boundary is placed at the beginning of that substring.

2.4 Rule-based Reordering

Although none of the authors speaks Hindi, we have obtained some information about its word order. We decided to experiment with two reordering transformations of the English

	EILMT	TIDES
Baseline Moses, Distance Reordering	18.88±2.05	10.06±0.76
Baseline Moses, Reordering Using en+hi Forms	19.77±2.03	**10.95±0.75**
Suffix LM+Reord	20.09±2.18	10.18±0.74
Rule-based Reordering + Suffix LM+Reord	**21.01±2.18**	10.29±0.69

Table 3: Impact of rule-based reordering and suffix LM and reordering on EILMT and TIDES datasets.

sentences, so that their word order gets closer to Hindi. We first tagged the English side using Morče (Votrubec, 2005), than parsed dependencies using the MST parser (McDonald et al., 2005), and finally moved parts of the sentence, based on POS tags and the dependency tree. The following transformations have been applied:

1. Move finite verb forms to the end of the sentence (not crossing punctuation, “that”, WH-words).
2. Transform prepositions to postpositions

Table 3 displays some of the Moses configurations using our rule-based reordering and the unsupervised stem-suffix segmentation evaluated both on EILMT domain (7k training, 500 dev, 500 test sentences) and TIDES (50k training, 1k dev, 1k test sentences). The rule-based reordering is used before Moses training and translation. The stem-suffix segmentation is applied to the Hindi side of the training data. We use two output factors and two decoding steps: 1) English word forms are translated to Hindi word forms, and 2) suffixes are generated from the hypothesized output word forms. We apply two language models: 3-grams of Hindi forms and 10-grams of suffixes. We also use the target-side suffix factor in the reordering model. We see that a similar improvement can be achieved by various methods with no clear winning combination.

3 Conclusion

In our experiments we were able to achieve improvements of BLEU using stem-suffix segmentation and rule-based reordering for EILMT data. The result is not confirmed on TIDES data where standard bi-directional reordering performs better. Similarly mixed are the effects of additional parallel and/or monolingual data.

One notable observation is that the grow-diag-final (no -and) heuristic for alignment symmetrization is highly unsuitable for the English-Hindi language pair and can degrade BLEU scores by up to 5 points absolute.

References

Natalia Klyueva and Ondřej Bojar. 2008. UMC 0.1: Czech-Russian-English Multilingual Corpus. In *Proc. of International Conference Corpus Linguistics*, pages 188–195, October.

Philipp Koehn et al. 2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *Proceedings of ACL 2007*, pages 177–180, Prague, Czech Republic, June.

Philipp Koehn. 2004. Statistical Significance Tests for Machine Translation Evaluation. In *Proceedings of EMNLP 2004*, Barcelona, Spain.

Ryan McDonald, Fernando Pereira, Kiril Ribarov, and Jan Hajič. 2005. Non-Projective Dependency Parsing using Spanning Tree Algorithms. In *Proceedings of HLT/EMNLP 2005*, October.

Franz Josef Och. 2003. Minimum Error Rate Training in Statistical Machine Translation. In *Proc. of the Association for Computational Linguistics*, Sapporo, Japan, July 6-7.

Jan Votrubec. 2005. Volba vhodné sady rysů pro morfologické značkování češtiny. MSc. thesis.

Daniel Zeman. 2008. Unsupervised acquiring of morphological paradigms from tokenized text. In Carol Peters et al., editor, *Advances in Multilingual and Multimodal Information Retrieval, CLEF 2007*, volume 5152/2008, pages 892–899. Springer.