DCU-Lingo24 Participation in WMT 2014 Hindi-English Translation task

Xiaofeng Wu, Rejwanul Haque*, Tsuyoshi Okita
Piyush Arora, Andy Way, Qun Liu
CNGL, Centre for Global Intelligent Content
School of Computing, Dublin City University
Dublin 9, Ireland
{xf.wu,tokita,parora,away,qliu}@computing.dcu.ie
*Lingo24, Edinburgh, UK
rejwanul.haque@lingo24.com

Abstract

This paper describes the DCU-Lingo24 submission to WMT 2014 for the Hindi-English translation task. We exploit miscellaneous methods in our system, including: Context-Informed PB-SMT, OOV Word Conversion (OWC), Multi-Alignment Combination (MAC), Operation Sequence Model (OSM), Stemming Align and Normal Phrase Extraction (SANPE), and Language Model Interpolation (LMI). We also describe various pre-processing steps we tried for Hindi in this task.

1 Introduction

This paper describes the DCU-Lingo24 submission to WMT 2014 for the Hindi-English translation task.

All our experiments on WMT 2014 are built upon the Moses phrase-based model (PB-SMT) (Koehn et al., 2007) and tuned with MERT (Och, 2003). Starting from this baseline system, we exploit various methods including Context-Informed PB-SMT (CIPBSMT), zero-shot learning (Palatucci et al., 2009) using neural network-based language modelling (Bengio et al., 2000; Mikolov et al., 2013) for OOV word conversion, various lexical reordering models (Axelrod et al., 2005; Galley and Manning, 2008), various Multiple Alignment Combination (MAC) (Tu et al., 2012), Operation Sequence Model (OSM) (Durrani et al., 2011) and Language Model Interpolation(LMI).

In the next section, the preprocessing steps are explained. In Section 3 a detailed explanation of the technique we exploit is provided. Then in Section 4, we provide our experimental results and resultant discussion.

2 Pre-processing Steps

We use all the training data provided for Hindi–English translation. Following Bojar et al. (2010), we apply a number of normalisation methods on the Hindi corpus. The HindEnCorp parallel corpus compiles several sources of parallel data. We observe that the source-side (Hindi) of the TIDES data source contains font-related noise, i.e. many Hindi sentences are a mixture of two different encodings: UTF-8[1] and WX[2] notations. We prepared a WX-to-UTF-8 font conversion script for Hindi which converts all WX encoded characters into UTF-8, thus removing all WX encoding appearing in the TIDES data.

We also observe that a portion of the English training corpus contained the following bracket-like sequences of characters: -LRB-, -LSB-, -LCB-, -RRB-, -RSB-, and -RCB-.[3] For consistency, those character sequences in the training data were replaced by the corresponding brackets.

For English – both monolingual and the target side of the bilingual data – we perform tokenization, normalization of punctuation, and truecasing. For parallel training data, we filter sentences pairs containing more than 80 tokens on either side and

[1] http://en.wikipedia.org/wiki/UTF-8

[2] http://en.wikipedia.org/wiki/WX_notation

[3] The acronyms stand for (Left | Right) (Round | Square | Curly) Bracket.

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 215–220,
Baltimore, Maryland USA, June 26–27, 2014. ©2014 Association for Computational Linguistics

sentence pairs with length difference larger than 3 times.

3 Techniques Deployed

3.1 Combination of Various Lexical Reordering Model (LRM)

Clearly, Hindi and English have quite different word orders, so we adopt three lexical reordering models to address this problem. They are word-based LRM and phrase-based LRM, which mainly focus on local reordering phenomena, and hierarchical phrase-based LRM, which mainly focuses on longer distance reordering (Galley and Manning, 2008).

3.2 Operation Sequence Model

The Operation Sequence Model (OSM) of Durrani et al. (2011) defines four translation operations: Generate(X,Y), Continue Source Concept, Generate Source Only (X) and Generate Identical, as well as three reordering operations: Insert Gap, Jump Back(W) and Jump Forward.

The probability of an operation sequence $O = (o_1 o_2 \cdots o_J)$ is calculated as in (1):

$$p(O) = \prod_{j=1}^{J} p(o_j | o_{j-n+1} \cdots o_{j-1}) \qquad (1)$$

where n indicates the number of previous operations used.

We employ a 9-order OSM in our framework.

3.3 Language Model Interpolation (LMI)

We build a large language model by including data from the English Gigaword fifth edition, the English side of the UN corpus, the English side of the 10^9 French–English corpus and the English side of the Hindi–English parallel data provided by the organisers. We interpolate language models trained using each dataset, with the monolingual data provided split into three parts (news 2007-2013, Europarl (**?**) and news commentary) and the weights tuned to minimize perplexity on the target side of the devset.

The language models in our systems are trained with SRILM (Stolcke, 2002). We train a 5-gram model with Kneser-Ney discounting (Chen and Goodman, 1996).

3.4 Context-informed PB-SMT

Haque et al. (2011) express a context-dependent phrase translation as a multi-class classification problem, where a source phrase with given additional context information is classified into a distribution over possible target phrases. The size of this distribution needs to be limited, and would ideally omit irrelevant target phrase translations that the standard PB-SMT (Koehn et al., 2003) approach would normally include. Following Haque et al. (2011), we derive a context-informed feature $\hat{h}_{\text{mbl}}$ that is expressed as the conditional probability of the target phrase $\hat{e}_k$ given the source phrase $\hat{f}_k$ and its context information (CI), as in (2):

$$\hat{h}_{\text{mbl}} = \log \text{P}(\hat{e}_k | \hat{f}_k, \text{CI}(\hat{f}_k)) \qquad (2)$$

Here, CI may include any feature that can provide useful information to disambiguate the given source phrase. In our experiment, we use CCG supertag (Steedman, 2000) as a contextual features. CCG supertag expresses the specific syntactic behaviour of a word in terms of the arguments it takes, and more generally the syntactic environment in which it appears.

We consider the CCG supertags of the context words, as well as of the focus phrase itself. In our model, the supertag of a multi-word focus phrase is the concatenation of the supertags of the words composing that phrase. We generate a window of size $2l + 1$ features (we set l:=2), including the concatenated complex supertag of the focus phrase. Accordingly, the supertag-based contextual information (CI_{st}) is described as in (3):

$$\begin{aligned}\text{CI}_{\text{st}}(\hat{f}_k) = \{&\text{st}(f_{i_k-l}), ..., \text{st}(f_{i_k-1}), \text{st}(\hat{f}_k),\\ &\text{st}(f_{j_k+1}), ..., \text{st}(f_{j_k+l})\}\end{aligned} \qquad (3)$$

For the Hindi-to-English translation task, we use part-of-speech (PoS) tags[4] of the source phrase and the neighbouring words as the contextual feature, owing to the fact that supertaggers are readily available only for English.

We use a memory-based machine learning (MBL) classifier (TRIBL: (Daelemans, 2005))[5] that is able to estimate $\text{P}(\hat{e}_k | \hat{f}_k, \text{CI}(\hat{f}_k))$ by similarity-based reasoning over memorized nearest-neighbour examples of source–target phrase translations. Thus, we derive the feature $\hat{h}_{\text{mbl}}$ defined in Equation (2). In addition to $\hat{h}_{\text{mbl}}$,

[4] In order to obtain PoS tags of Hindi words, we used the LTRC shallow parser for Hindi from http://ltrc.iiit.ac.in/analyzer/hindi/shallow-parser-hin-4.0.fc8.tar.gz.

[5] An implementation of TRIBL is freely available as part of the TiMBL software package, which can be downloaded from http://ilk.uvt.nl/timbl.

we derive a simple two-valued feature $\hat{h}_{\text{best}}$, defined in Equation (4):

$$\hat{h}_{\text{best}} = \begin{cases} 1 & \text{if } \hat{e}_k \text{ maximizes } \mathrm{P}(\hat{e}_k|\hat{f}_k, \mathrm{CI}(\hat{f}_k)) \\ \approxeq 0 & \text{otherwise} \end{cases} \tag{4}$$

where $\hat{h}_{\text{best}}$ is set to 1 when $\hat{e}_k$ is one of the target phrases with highest probability according to $\mathrm{P}(\hat{e}_k|\hat{f}_k, \mathrm{CI}(\hat{f}_k))$ for each source phrase $\hat{f}_k$; otherwise $\hat{h}_{\text{best}}$ is set to 0.000001. We performed experiments by integrating these two features $\hat{h}_{\text{mbl}}$ and $\hat{h}_{\text{best}}$ directly into the log-linear model of Moses. Their weights are optimized using minimum error-rate training (MERT)(Och, 2003) on a held-out development set for each of the experiments.

3.5 Morphological Segmentation

Haque et al. (2012) applied a morphological suffix separation process in a Bengali-to-English translation task and showed that suffix separation significantly reduces data sparseness in the Bengali corpus. They also showed an SMT model trained on the suffix-stripped training data significantly outperforms the state-of-the-art PB-SMT baseline. Like Bengali, Hindi is a morphologically very rich and highly inflected Indian language. As done previously for Bengali-to-English (Haque et al., 2012), we employ a suffix-stripping method for lemmatizing inflected Hindi words in the WMT Hindi-to-English translation task. Following Dasgupta and Ng (2006), we developed an unsupervised morphological segmentation method for Hindi. We also used a Hindi lightweight stemmer (Ramanathan and Rao, 2003) in order to prepare a training corpus with only Hindi stems. We prepared Hindi-to-English SMT systems on the both types of training data (i.e. suffix-stripped and stemmed).[6]

3.6 Multi-Alignment Combination (MAC)

Word alignment is a critical component of MT systems. Various methods for word alignment have been proposed, and different models can produce signicantly different outputs. For example, Tu et al. (2012) demonstrates that the alignment agreement between the two best-known alignment tools, namely Giza++(Och and Ney, 2003) and the Berkeley aligner[7] (Liang et al., 2006), is below 70%. Taking into consideration the small size of the the corpus, in order to extract more effective phrase tables, we concatenate three alignments: Giza++ with grow-diag-final-and, Giza++ with intersection, and that derived from the Berkeley aligner.

3.7 Stemming Alignment and Normal Phrase Extraction (SANPE)

The rich morphology of Hindi will cause word alignment sparsity, which results in poor alignment quality. Furthermore, word stemming on the Hindi side usually results in too many English words being aligned to one stemmed Hindi word, i.e. we encounter the problem of phrase over-extraction. Therefore, we conduct word alignment with the stemmed version of Hindi, and then at the phrase extraction step, we replace the stemmed form with the original Hindi form.

3.8 OOV Word Conversion Method

Our algorithm for OOV word conversion uses the recently developed zero-shot learning (Palatucci et al., 2009) using neural network language modelling (Bengio et al., 2000; Mikolov et al., 2013). The same technique is used in (Okita et al., 2014). This method requires neither parallel nor comparable corpora, but rather two monolingual corpora. In our context, we prepare two monolingual corpora on both sides, which are neither parallel nor comparable, and a small amount of already known correspondences between words on the source and target sides (henceforth, we refer to this as the 'dictionary'). Then, we train both sides with the neural network language model, and use a continuous space representation to project words to each other on the basis of a small amount of correspondences in the dictionary. The following algorithm shows the steps involved:

1. Prepare the monolingual source and target sentences.

2. Prepare the dictionary which consists of U entries of source and target sentences comprising non-stop-words.

3. Train the neural network language model on the source side and obtain the real vectors of X dimensions for each word.

[6] Suffixes were separated and completely removed from the training data.

[7] http://code.google.com/p/berkeleyaligner/

4. Train the neural network language model on the target side and obtain the real vectors of X dimensions for each word.

5. Using the real vectors obtained in the above steps, obtain the linear mapping between the dictionary items in two continuous spaces using canonical component analysis (CCA).

In our experiments we use U the same as the entries of Wiki corpus, which is provided among WMT14 corpora, and X as 50. The resulted projection by this algorithm can be used as the OOV word conversion which projects from the source language which among OOV words into the target language. The overall algorithm which uses the projection which we build in the above step is shown in the following.

1. Collect unknown words in the translation outputs.

2. Do Hindi named-entity recognition (NER) to detect noun phrases.

3. If they are noun phrases, do the projection from each unknown word in the source side into the target words (We use the projection prepared in the above steps). If they are not noun phrases, run the transliteration to convert each of them.

We perform Hindi NER by training CRF++ (Kudo et al., 2004) using the Hindi named entity corpus, and use the Hindi shallow parser (Begum et al., 2008) for preprocessing of the inputs.

4 Results and Discussion

4.1 Data

We conduct our experiments on the standard datasets released in the WMT14 shared translation task. We use HindEnCorp[8] (Bojar et al., 2014) parallel corpus for MT system building. We also used the CommonCrawl Hindi monolingual corpus (Bojar et al., 2014) in order to build an additional language model for Hindi.

For the Hindi-to-English direction, we also employed monolingual English data used in the other translation tasks for building the English language model.

4.2 Moses Baseline

We employ a standard Moses PB-SMT model as our baseline. The Hindi side is preprocessed but unstemmed. We use Giza++ to perform word alignment, the phrase table is extracted via the grow-diag-final-and heuristic and the max-phrase-length is set to 7.

4.3 Automatic Evaluation

Experiments	BLEU
Moses Baseline	8.7
Context-Based	9.4
Context-Based + CommonCrawl LM	11.4

Table 1: BLEU scores of the English-to-Hindi MT Systems on the WMT test set.

Experiments	BLEU
Moses Baseline	10.1
Context-Based	10.1
Suffix-Stripped	10.0
OWC	11.2
OSM	10.3
Three LRMs	10.5
MAC	10.7
SANPE	10.6
LMI	10.9
LMI+SANPE+MAC+ThreeLRMs+OSM	11.7

Table 2: BLEU scores of the Hindi-to-English MT Systems on the WMT test set.

We prepared a number of MT systems for both English-to-Hindi and Hindi-to-English, and submitted their runs in the WMT 2014 Evaluation Matrix. The BLEU scores of the different English-to-Hindi MT systems (Moses Baseline, Context-Based (CCG) MT system, and Context-Based (CCG) MT system with an additional LM built on the CommonCrawl Hindi monolingual corpus (Bojar et al., 2014)) on the WMT 2014 English-to-Hindi test set are reported in Table 1. As can be seen from Table 1, Context-Based (CCG) MT system produces 0.7 BLEU points improvement (8.04% relative) over the Moses Baseline. When we add an additional large LM built on the CommonCrawl data to the Context-Based (CCG) MT system, we achieved a 2 BLEU-point improvement (21.3% relative) (cf. last row in Table 1) over

[8] http://ufallab.ms.mff.cuni.cz/ bojar/hindencorp/

the Context-Based (CCG) MT system.[9]

The BLEU scores of the different Hindi-to-English MT systems on the WMT 2014 Hindi-to-English test set are reported in Table 2. The first row of Table 2 shows the BLEU score for the Baseline MT system. We note that the performance of the Context-Based (PoS) MT system obtains identical performance to the Moses baseline (10.1 BLEU points) on the WMT 2014 Hindi-to-English test set.

We employed a source language (Hindi) normalisation technique, namely suffix separation, but unfortunately this did not bring about any improvement for the Hindi-to-English translation task. The improvement gained by individually employing OSM, three lexical reordering models, Multi-alignment Combination, Stem-align and normal Phrase Extraction and Language Model Interpolation can be seen in Table 2. Our best system is achieved by combining OSM, Three LMR, MAC, SANPE and LMI, which results in a 1.6 BLEU point improvement over the Baseline.

5 Acknowledgments

This research is supported by the Science Foundation Ireland (Grant 12/CE/I2267) as part of the CNGL Centre for Global Intelligent Content (www.cngl.ie) at Dublin City University.

References

Amittai Axelrod, Ra Birch Mayne, Chris Callison-burch, Miles Osborne, and David Talbot. 2005. Edinburgh system description for the 2005 iwslt speech translation evaluation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

Rafiya Begum, Samar Husain, Arun Dhwaj, Dipti Misra Sharma, Lakshmi Bai, and Rajeev Sangal. 2008. Dependency annotation scheme for indian languages. *In Proceedings of The Third International Joint Conference on Natural Language Processing (IJCNLP)*.

Yoshua Bengio, Rejean Ducharme, and Pascal Vincent. 2000. A neural probabilistic language model. *In Proceedings of Neural Information Systems*.

Ond Bojar, Pavel Stranak, and Daniel Zeman. 2010. Data issues in english-to-hindi machine translation. In *LREC*.

Ondrej Bojar, V. Diatka, Rychly P., Pavel Stranak, A. Tamchyna, and Daniel Zeman. 2014. Hindi-english and hindi-only corpus for machine translation. In *LREC*.

Stanley F. Chen and Joshua Goodman. 1996. An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting on Association for Computational Linguistics*, ACL '96, pages 310–318, Stroudsburg, PA, USA. Association for Computational Linguistics.

Walter Daelemans. 2005. *Memory-based language processing*. Cambridge University Press.

Sajib Dasgupta and Vincent Ng. 2006. Unsupervised morphological parsing of bengali. *Language Resources and Evaluation*, 40(3-4):311–330.

Nadir Durrani, Helmut Schmid, and Alexander Fraser. 2011. A joint sequence translation model with integrated reordering. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies - Volume 1*, HLT '11, pages 1045–1054, Stroudsburg, PA, USA. Association for Computational Linguistics.

Michel Galley and Christopher D. Manning. 2008. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, Honolulu, Hawaii, October. Association for Computational Linguistics.

Rejwanul Haque, Sudip Kumar Naskar, Antal van den Bosch, and Andy Way. 2011. Integrating source-language context into phrase-based statistical machine translation. *Machine translation*, 25(3):239–285.

Rejwanul Haque, Sergio Penkale, Jie Jiang, and Andy Way. 2012. Source-side suffix stripping for bengali-to-english smt. In *Asian Language Processing (IALP), 2012 International Conference on*, pages 193–196. IEEE.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology-Volume 1*, pages 48–54. Association for Computational Linguistics.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 177–180, Stroudsburg, PA, USA. Association for Computational Linguistics.

[9] Please note that this is an unconstrained submission.

Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. 2004. Appliying conditional random fields to japanese morphological analysis. *In Proceedings of EMNLP*.

Percy Liang, Ben Taskar, and Dan Klein. 2006. Alignment by agreement. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 104–111. Association for Computational Linguistics.

Tomas Mikolov, Quoc V. Le, and Ilya Sutskever. 2013. Exploiting similarities among languages for machine translation. *ArXiv*.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Stroudsburg, PA, USA. Association for Computational Linguistics.

Tsuyoshi Okita, Ali Hosseinzadeh Vahid, Andy Way, and Qun Liu. 2014. Dcu terminology translation system for medical query subtask at wmt14.

Mark Palatucci, Dean Pomerleau, Geoffrey Hinton, and Tom Mitchell. 2009. Zero-shot learning with semantic output codes. In *Neural Information Processing Systems (NIPS)*, December.

Ananthakrishnan Ramanathan and Durgesh D Rao. 2003. A lightweight stemmer for hindi. In *the Proceedings of EACL*.

Mark Steedman. 2000. *The syntactic process*, volume 35. MIT Press.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of the International Conference Spoken Language Processing*, pages 901–904, Denver, CO.

Zhaopeng Tu, Yang Liu, Yifan He, Josef van Genabith, Qun Liu, and Shouxun Lin. 2012. Combining multiple alignments to improve machine translation. In *COLING (Posters)*, pages 1249–1260.