



THE UNIVERSITY *of* EDINBURGH

Edinburgh Research Explorer

Integrating an Unsupervised Transliteration Model into Statistical Machine Translation

Citation for published version:

Durrani, N, Sajjad, H, Hoang, H & Koehn, P 2014, Integrating an Unsupervised Transliteration Model into Statistical Machine Translation. in *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden*. pp. 148-153. <<http://aclweb.org/anthology/E/E14/E14-4029.pdf>>

Link:

[Link to publication record in Edinburgh Research Explorer](#)

Document Version:

Peer reviewed version

Published In:

Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, EACL 2014, April 26-30, 2014, Gothenburg, Sweden

General rights

Copyright for the publications made accessible via the Edinburgh Research Explorer is retained by the author(s) and / or other copyright owners and it is a condition of accessing these publications that users recognise and abide by the legal requirements associated with these rights.

Take down policy

The University of Edinburgh has made every reasonable effort to ensure that Edinburgh Research Explorer content complies with UK legislation. If you believe that the public display of this file breaches copyright please contact openaccess@ed.ac.uk providing details, and we will remove access to the work immediately and investigate your claim.



Integrating an Unsupervised Transliteration Model into Statistical Machine Translation

Nadir Durrani

University of Edinburgh
dnadir@inf.ed.ac.uk

Hassan Sajjad

Qatar Computing Research Institute
hsajjad@qf.org.qa

Hieu Hoang Philipp Koehn

University of Edinburgh
hieu.hoang, pkoehn@inf.ed.ac.uk

Abstract

We investigate three methods for integrating an unsupervised transliteration model into an end-to-end SMT system. We induce a transliteration model from parallel data and use it to translate OOV words. Our approach is fully unsupervised and language independent. In the methods to integrate transliterations, we observed improvements from 0.23-0.75 (Δ 0.41) BLEU points across 7 language pairs. We also show that our mined transliteration corpora provide better rule coverage and translation quality compared to the gold standard transliteration corpora.

1 Introduction

All machine translation (MT) systems suffer from the existence of out-of-vocabulary (OOV) words, irrespective of the amount of data available for training. OOV words are mostly named entities, technical terms or foreign words that can be translated to the target language using transliteration.

Much work (Al-Onaizan and Knight, 2002; Zhao et al., 2007; Kashani et al., 2007; Habash, 2009) has been done on transliterating named entities and OOVs, and transliteration has been shown to improve MT quality. Transliteration has also shown to be useful for translating closely related language pairs (Durrani et al., 2010; Nakov and Tiedemann, 2012), and for disambiguation (Hermjakob et al., 2008; Azab et al., 2013). However, despite its utility, a transliteration module does not exist in the commonly used MT toolkits, such as Moses (Koehn et al., 2007). One of the main reasons is that the training data, a corpus of transliteration pairs, required to build a transliteration system, is not readily available for many language pairs. Even if such a training data is available, mechanisms to integrate transliterated words

into MT pipelines are unavailable in these toolkits. Generally, a supervised transliteration system is trained separately outside of an MT pipeline, and a naïve approach, to replace OOV words with their 1-best transliterations in the post/pre-processing step of decoding is commonly used.

In this work i) we use an unsupervised model based on Expectation Maximization (EM) to induce transliteration corpus from word aligned parallel data, which is then used to train a transliteration model, ii) we investigate three different methods for integrating transliteration during decoding, that we implemented within the Moses toolkit. To the best of our knowledge, our work is the foremost attempt to integrate unsupervised transliteration model into SMT.

This paper is organized as follows. Section 2 describes the unsupervised transliteration mining system, which automatically mines transliteration pairs from the same word-aligned parallel corpus as used for training the MT system. Section 3 describes the transliteration model that is trained using the automatically extracted pairs. Section 4 presents three methods for incorporating transliteration into the MT pipeline, namely: i) replacing OOVs with the 1-best transliteration in a post-decoding step, ii) selecting the best transliteration from the list of n-best transliterations using transliteration and language model features in a post-decoding step, iii) providing a transliteration phrase-table to the decoder on the fly where it can consider all features to select the best transliteration of OOV words. Section 5 presents results. Our integrations achieved an average improvement of 0.41 BLEU points over a competitive baseline across 7 language pairs (Arabic, Bengali, Farsi, Hindi, Russian, Telugu and Urdu-into-English). An additional experiment showed that our system provides better rule coverage as opposed to another built from gold standard transliteration corpus and produces better translations.

2 Transliteration Mining

The main bottleneck in building a transliteration system is the lack of availability of transliteration training pairs. It is, however, fair to assume that any parallel data would contain a reasonable number of transliterated word pairs. Transliteration mining can be used to extract such word pairs from the parallel corpus. Most previous techniques on transliteration mining generally use supervised and semi-supervised methods (Sherif and Kondrak, 2007; Jiampojamarn et al., 2010; Darwish, 2010; Kahki et al., 2012). This constrains the mining solution to language pairs for which training data (seed data) is available. A few researchers proposed unsupervised approaches to mine transliterations (Lee and Choi, 1998; Sajjad et al., 2011; Lin et al., 2011). We adapted the work of Sajjad et al. (2012) as summarized below.

Model: The transliteration mining model is a mixture of two sub-models, namely: a transliteration and a non-transliteration sub-model. The idea is that the transliteration model would assign higher probabilities to transliteration pairs compared to the probabilities assigned by a non-transliteration model to the same pairs. Consider a word pair (e, f) , the **transliteration model** probability for the word pair is defined as follows:

$$p_{tr}(e, f) = \sum_{a \in Align(e, f)} \prod_{j=1}^{|a|} p(q_j)$$

where $Align(e, f)$ is the set of all possible sequences of character alignments, a is one alignment sequence and q_j is a character alignment.

The **non-transliteration model** deals with the word pairs that have no character relationship between them. It is modeled by multiplying source and target character unigram models:

$$p_{ntr}(e, f) = \prod_{i=1}^{|e|} p_E(e_i) \prod_{i=1}^{|f|} p_F(f_i)$$

The **transliteration mining model** is defined as an interpolation of the transliteration sub-model and the non-transliteration sub-model:

$$p(e, f) = (1 - \lambda)p_{tr}(e, f) + \lambda p_{ntr}(e, f)$$

λ is the prior probability of non-transliteration.

The non-transliteration model does not change during training. We compute it in a pre-processing step. The transliteration model learns character alignment using expectation maximization (EM). See Sajjad et al. (2012) for more details.

3 Transliteration Model

Now that we have transliteration word pairs, we can learn a transliteration model. We segment the training corpus into characters and learn a phrase-based system over character pairs. The transliteration model assumes that source and target characters are generated monotonically.¹ Therefore we do not use any reordering models. We use 4 basic phrase-translation features (direct, inverse phrase-translation, and lexical weighting features), language model feature (built from the target-side of mined transliteration corpus), and word and phrase penalties. The feature weights are tuned² on a dev-set of 1000 transliteration pairs.

4 Integration to Machine Translation

We experimented with three methods for integrating transliterations, described below:

Method 1: involves replacing OOVs in the output with the 1-best transliteration. The success of Method 1 is solely contingent on the accuracy of the transliteration model. Also, it ignores context which may lead to incorrect transliteration. For example, the Arabic word **بيل** transliterates to “Bill” when followed by “Clinton” and “Bell” if preceded by “Alexander Graham”.

Method 2: provides n-best transliterations to a monotonic decoder that uses a monolingual language model and a transliteration phrase-translation table to rescore transliterations. We carry forward the 4 translation model features used in the transliteration system to build a transliteration phrase-table. We additionally use an LM-OOV feature which counts the number of words in a hypothesis that are unknown to the language model. Smoothing methods such as Kneser-Ney assign significant probability mass to unseen events, which may cause the decoder to make incorrect transliteration selection. The LM-OOV feature acts as a prior to penalize such hypotheses.

Method 3: Method 2 can not benefit from all decoding features and phenomenon like reordering. It transliterates Urdu compound **بحيره عرب** (Arabian Sea) to “Sea Arabian”, if **عرب** is an unknown word. In method 3, we feed the transliteration phrase-table directly into the first-pass decoding which allows reordering of UNK words. We

¹Mining algorithm also makes this assumption.

²Tuning data is subtracted from the training corpus while tuning to avoid over-fitting. After the weights are tuned, we add it back, retrain GIZA, and estimate new models.

use the *decoding-graph-backoff* option in Moses, that allows multiple translation phrase tables and back-off models. As in method 2, we also use the LM-OOV feature in method 3.³

5 Evaluation

Data: We experimented with 7 language pairs, namely: Arabic, Bengali, Farsi, Hindi, Russian, Telugu and Urdu-into-English. For Arabic⁴ and Farsi, we used the TED talks data (Cettolo et al., 2012) made available for IWSLT-13, and we used the dev2010 set for tuning and the test2011 and test2012 sets for evaluation. For Indian languages we used the Indic multi-parallel corpus (Post et al., 2012), and we used the dev and test sets provided with the parallel corpus. For Russian, we used WMT-13 data (Bojar et al., 2013), and we used half of the news-test2012 for tuning and other half for testing. We also evaluated on the news-test2013 set. For all, we trained the language model using the monolingual WMT-13 data. See Table 1 for data statistics.

Lang	Train _{tm}	Train _{tr}	Dev	Test ₁	Test ₂
AR	152K	6795	887	1434	1704
BN	24K	1916	775	1000	
FA	79K	4039	852	1185	1116
HI	39K	4719	1000	1000	
RU	2M	302K	1501	1502	3000
TE	45K	4924	1000	1000	
UR	87K	9131	980	883	

Table 1: No. of sentences in Training Data and Mined Transliteration Corpus (Types) (**Train_{tr}**)

Baseline Settings: We trained a Moses system replicating the settings used in competition-grade systems (Durrani et al., 2013b; Birch et al., 2013): a maximum sentence length of 80, GDFA symmetrization of GIZA++ alignments (Och and Ney, 2003), an interpolated Kneser-Ney smoothed 5-gram language model with KenLM (Heafield, 2011) used at runtime, a 5-gram OSM (Durrani et al., 2013a), msd-bidirectional-fe lexical-

³Method 3 is desirable in cases where the decoder can translate or transliterate a word. For example Hindi word सीमा can be translated to “Border” and also transliterated to name “Seema”. Identifying such candidates that can be translated or transliterated is a challenge. Machine learning techniques (Goldwasser and Roth, 2008; Kirschenbaum and Wintner, 2009) and named entity recognizers (Klementiev and Roth, 2006; Hermjakob et al., 2008) have been used for this purpose. Though, we only focus on OOV words, method 3 can be used if such a classifier/NE tagger is available.

⁴Arabic and Urdu are segmented using MADA (Habash and Sadat, 2006) and UWS (Durrani and Hussain, 2010).

ized reordering, sparse lexical and domain features (Hasler et al., 2012), a distortion limit of 6, 100-best translation options, MBR decoding (Kumar and Byrne, 2004), Cube Pruning (Huang and Chiang, 2007), and the no-reordering-over-punctuation heuristic. We tuned with the k-best batch MIRA (Cherry and Foster, 2012).⁵

Transliteration Miner: The miner extracts transliterations from a word-aligned parallel corpus. We only used word pairs with 1-to-1 alignments.⁶ Before feeding the list into the miner, we cleaned it by removing digits, symbols, word pairs where source or target is composed from less than 3 characters, and words containing foreign characters that do not belong to this scripts. We ran the miner with 10 iterations of EM. The number of transliteration pairs (types) extracted for each language pair is shown in Table 1 (**Train_{tr}**).

Transliteration System: Before evaluating our integrations into the SMT system, we performed an intrinsic evaluation of the transliteration system that we built from the mined pairs. We formed test data for Arabic–English (1799 pairs), Hindi–English (2394 pairs) and Russian–English (1859 pairs) by concatenating the seed data and gold standard transliteration pairs both provided for the Shared Task on Transliteration mining (Kumaran et al., 2010). Table 2 shows precision and recall of the mined transliteration system (MTS).

	AR	HI	RU
Precision (1-best Accuracy)	20.0%	25.3%	46.1%
Recall (100-best Accuracy)	80.2%	79.3%	87.5%

Table 2: Precision and Recall of MTS

The precision (1-best accuracy) of the transliteration model is quite low. This is because the transliteration corpus is noisy and contains imperfect transliteration pairs. For example, the miner extracted the pair (استراليا, Australasia), while the correct transliteration is “Australia”. We can improve the precision by tightening the mining threshold probability. However, our end goal is to improve end-to-end MT and not the transliteration system. We observed that recall is more important than precision for overall MT quality. We provide an empirical justification for this when discussing the final experiments.

⁵Retuning the transliteration features was not helpful, default weights are used.

⁶M-N/1-N alignments are less likely to be transliterations.

MT Experiments: Table 3 gives a comprehensive evaluation of the three methods of integration discussed in Section 4 along with the number⁷ of OOV words (types) in different tests. We report BLEU gains (Papineni et al., 2002) obtained by each method. Method 1 (M_1), that replaces OOV words with 1-best transliteration gave an average improvement of +0.13. This result can be attributed to the low precision of the transliteration system (Table 2). Method 2 (M_2), that transliterates OOVs in second pass monotonic decoding, gave an average improvement of +0.39. Slightly higher gains were obtained using Method 3 (M_3), that integrates transliteration phrase-table inside decoder on the fly. However, the efficacy of M_3 in comparison to M_2 is not as apparent, as M_2 produced better results than M_3 in half of the cases.

Lang	Test	B_0	M_1	M_2	M_3	OOV
AR	iwslt ₁₁	26.75	+0.12	+0.36	+0.25	587
	iwslt ₁₂	29.03	+0.10	+0.30	+0.27	682
BN	jhu ₁₂	16.29	+0.12	+0.42	+0.46	1239
FA	iwslt ₁₁	20.85	+0.10	+0.40	+0.31	559
	iwslt ₁₂	16.26	+0.04	+0.20	+0.26	400
HI	jhu ₁₂	15.64	+0.21	+0.35	+0.47	1629
RU	wmt ₁₂	33.95	+0.24	+0.55	+0.49	434
	wmt ₁₃	25.98	+0.25	+0.40	+0.23	799
TE	jhu ₁₂	11.04	-0.09	+0.40	+0.75	2343
UR	jhu ₁₂	23.25	+0.24	+0.54	+0.60	827
Avg		21.9	+0.13	+0.39	+0.41	950

Table 3: End-to-End MT Evaluation – B_0 = Baseline, M_1 = Method₁, M_2 = Method₂, M_3 = Method₃, BLEU gains shown for each method

In an effort to test whether improving transliteration precision would improve end-to-end SMT results, we carried out another experiment. Instead of building a transliteration system from mined corpus, we built it using the gold standard corpus (for Arabic, Hindi and Russian), that we also used previously to do an intrinsic evaluation. We then replaced our mined transliteration systems with the gold standard transliteration systems, in the best performing SMT systems for these languages. Table 4 shows a comparison of performances. Although the differences are small, systems using mined transliteration system (MTS) outperformed its counterpart that uses gold standard transliteration system (GTS), except in Hindi–English where

⁷Note that not all OOVs can be transliterated. This number is therefore an upper bound what can be transliterated.

both systems were equal.

	AR		HI	RU	
	iwslt ₁₁	iwslt ₁₂	jhu ₁₂	wmt ₁₂	iwslt ₁₃
MTS	27.11	29.33	16.11	34.50	26.38
GST	26.99	29.20	16.11	34.33	26.22

Table 4: Comparing Gold Standard Transliteration (GST) and Mined Transliteration Systems

In the error analysis we found that the GST system suffered from sparsity and did not provide enough coverage of rules to produce right transliterations. For example, Arabic drops the determiner ال (al), but such additions were not observed in gold transliteration pairs. Arabic word الغيغابكسل (Gigapixel) is therefore transliterated to “algegabksl”. Similarly the GST system learned no transliteration pairs to account for the rule “b → p” and therefore erroneously transliterated سبرلوك (Spurlock) to “Sbrlok”. Similar observations were true for the case of Russian–English. The rules “a → u” and “y → e” were not observed in the gold set, and hence харрикейнз (hurricane) was transliterated to “herrricane” and талботу (Talbot) to “Talboty”. This shows that better recall obtained from the mined pairs led to overall improvement.

6 Conclusion

We incorporated unsupervised transliteration mining model into standard MT pipeline to automatically transliterate OOV words without needing additional resources. We evaluated three methods for integrating transliterations on 7 language pairs and showed improvements ranging from 0.23-0.75 (Δ 0.41) BLEU points. We also showed that our mined transliteration corpus provide better recall and overall translation quality compared to the gold standard transliteration corpus. The unsupervised transliteration miner and its integration to SMT has been made available to the research community via the Moses toolkit.

Acknowledgments

We wish to thank the anonymous reviewers and Kareem Darwish for their valuable feedback on an earlier draft of this paper. The research leading to these results has received funding from the European Union Seventh Framework Programme (FP7/2007-2013) under grant agreement n^o 287658. This publication only reflects the authors’ views.

References

- Al-Onaizan, Y. and Knight, K. (2002). Translating named entities using monolingual and bilingual resources. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*.
- Azab, M., Bouamor, H., Mohit, B., and Oflazer, K. (2013). Dudley north visits north london: Learning when to transliterate to arabic. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 439–444, Atlanta, Georgia. Association for Computational Linguistics.
- Birch, A., Durrani, N., and Koehn, P. (2013). Edinburgh SLT and MT system description for the IWSLT 2013 evaluation. In *Proceedings of the 10th International Workshop on Spoken Language Translation*, pages 40–48, Heidelberg, Germany.
- Bojar, O., Buck, C., Callison-Burch, C., Federmann, C., Haddow, B., Koehn, P., Monz, C., Post, M., Soricut, R., and Specia, L. (2013). Findings of the 2013 Workshop on Statistical Machine Translation. In *Eighth Workshop on Statistical Machine Translation, WMT-2013*, pages 1–44, Sofia, Bulgaria.
- Cettolo, M., Girardi, C., and Federico, M. (2012). Wit³: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.
- Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.
- Darwish, K. (2010). Transliteration mining with phonetic conflation and iterative training. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.
- Durrani, N., Fraser, A., Schmid, H., Hoang, H., and Koehn, P. (2013a). Can Markov Models Over Minimal Translation Units Help Phrase-Based SMT? In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, Sofia, Bulgaria. Association for Computational Linguistics.
- Durrani, N., Haddow, B., Heafield, K., and Koehn, P. (2013b). Edinburgh’s machine translation systems for European language pairs. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, Sofia, Bulgaria. Association for Computational Linguistics.
- Durrani, N. and Hussain, S. (2010). Urdu word segmentation. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 528–536, Los Angeles, California. Association for Computational Linguistics.
- Durrani, N., Sajjad, H., Fraser, A., and Schmid, H. (2010). Hindi-to-urdu machine translation through transliteration. In *Proceedings of the 48th Annual Conference of the Association for Computational Linguistics*, Uppsala, Sweden.
- Goldwasser, D. and Roth, D. (2008). Active sample selection for named entity transliteration. In *Proceedings of ACL-08: HLT, Short Papers*, pages 53–56, Columbus, Ohio. Association for Computational Linguistics.
- Habash, N. (2009). Remoov: A tool for online handling of out-of-vocabulary words in machine translation. In *Proceedings of the Second International Conference on Arabic Language Resources and Tools*, Cairo, Egypt. The MEDAR Consortium.
- Habash, N. and Sadat, F. (2006). Arabic preprocessing schemes for statistical machine translation. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, pages 49–52, New York City, USA. Association for Computational Linguistics.
- Hasler, E., Haddow, B., and Koehn, P. (2012). Sparse lexicalised features and topic adaptation for smt. In *Proceedings of the seventh International Workshop on Spoken Language Translation (IWSLT)*, pages 268–275.
- Heafield, K. (2011). Kenlm: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland, United Kingdom.

- Hermjakob, U., Knight, K., and Daumé III, H. (2008). Name translation in statistical machine translation - learning when to transliterate. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, Columbus, Ohio.
- Huang, L. and Chiang, D. (2007). Forest rescoring: Faster decoding with integrated language models. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 144–151, Prague, Czech Republic. Association for Computational Linguistics.
- Jiampojarn, S., Dwyer, K., Bergsma, S., Bhargava, A., Dou, Q., Kim, M.-Y., and Kondrak, G. (2010). Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, Uppsala, Sweden.
- Kahki, A. E., Darwish, K., El Din, A. S., and El-Wahab, M. A. (2012). Transliteration mining using large training and test sets. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12.
- Kashani, M. M., Joanis, E., Kuhn, R., Foster, G., and Popowich, F. (2007). Integration of an arabic transliteration module into a statistical machine translation system. In *Proceedings of the Second Workshop on Statistical Machine Translation*, Prague, Czech Republic.
- Kirschenbaum, A. and Wintner, S. (2009). Lightly supervised transliteration for machine translation. In *Proceedings of the 12th Conference of the European Chapter of the ACL (EACL 2009)*, pages 433–441, Athens, Greece. Association for Computational Linguistics.
- Klementiev, A. and Roth, D. (2006). Named entity transliteration and discovery from multilingual comparable corpora. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 82–88, New York City, USA. Association for Computational Linguistics.
- 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 the 45th Annual Meeting of the Association for Computational Linguistics, Demonstration Program*, Prague, Czech Republic.
- Kumar, S. and Byrne, W. J. (2004). Minimum bayes-risk decoding for statistical machine translation. In *HLT-NAACL*, pages 169–176.
- Kumaran, A., M. Khapra, M., and Li, H. (2010). Whitepaper of news 2010 shared task on transliteration mining. In *Proceedings of the 2010 Named Entities Workshop*, pages 29–38, Uppsala, Sweden. Association for Computational Linguistics.
- Lee, J.-S. and Choi, K.-S. (1998). English to korean statistical transliteration for information retrieval. *Computer Processing of Oriental Languages*, 12(1):17–37.
- Lin, W.-P., Snover, M., and Ji, H. (2011). Unsupervised language-independent name translation mining from wikipedia infoboxes. In *Proceedings of the First workshop on Unsupervised Learning in NLP*, pages 43–52, Edinburgh, Scotland. Association for Computational Linguistics.
- Nakov, P. and Tiedemann, J. (2012). Combining word-level and character-level models for machine translation between closely-related languages. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 301–305, Jeju Island, Korea. Association for Computational Linguistics.
- Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1).
- Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, ACL '02, pages 311–318, Morristown, NJ, USA.
- Post, M., Callison-Burch, C., and Osborne, M. (2012). Constructing parallel corpora for six indian languages via crowdsourcing. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 401–409, Montréal, Canada. Association for Computational Linguistics.

- Sajjad, H., Fraser, A., and Schmid, H. (2011). An algorithm for unsupervised transliteration mining with an application to word alignment. In *Proceedings of the 49th Annual Conference of the Association for Computational Linguistics*, Portland, USA.
- Sajjad, H., Fraser, A., and Schmid, H. (2012). A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Conference of the Association for Computational Linguistics*, Jeju, Korea.
- Sherif, T. and Kondrak, G. (2007). Bootstrapping a stochastic transducer for arabic-english transliteration extraction. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics*, Prague, Czech Republic.
- Zhao, B., Bach, N., Lane, I., and Vogel, S. (2007). A log-linear block transliteration model based on bi-stream hmms. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics*, Rochester, New York.