Developing language technology tools and resources for a resource-poor language: Sindhi

Raveesh Motlani
FC Kohli Center on Intelligent Systems (KCIS)
International Institute of Information Technology, Hyderabad
raveesh.motlani@gmail.com

Abstract

Sindhi, an Indo-Aryan language with more than 75 million native speakers[1] is a resource-poor language in terms of the availability of language technology tools and resources. In this thesis, we discuss the approaches taken to develop resources and tools for a resource-poor language with special focus on Sindhi. The major contributions of this work include raw and annotated datasets, a POS Tagger, a Morphological Analyser, a Transliteration and a Machine Translation System.

1 Introduction

Language technology tools are vital resources that ensure digital existence of a language for a long time. Such tools and resources are necessary for natural language processing and have aplenty applications in the digital era. For instance, cross-lingual technologies such as machine translation help people across the world communicate with each other using their native languages and access information present in a language they do not know. Similarly, automatic speech recognition helps people interact with machines using natural languages. There are many more such applications where a better understanding of natural languages by machines could be helpful in various ways. Language technology tools facilitate the understanding of natural languages by computers. A lot of popular languages in the world are equipped with such tools and applications but a larger set of languages in this world lack these basic tools. It is important to protect such languages from being digitally endangered.

Our work is based on one such resource-poor language, Sindhi. Our aim is to develop some basic resources, language processing tools and an application which will help Sindhi in its digital existence.

2 About the Sindhi language

Sindhi is an Indo-Aryan language spoken by more than 75 million speakers in the world. The majority of this population resides in India and Pakistan.[2] Historically, Sindhi was written using several writing systems (Landa, Khojki, Waranki, Khudawadi, Gurumukhi, Perso-Arabic and Devanagari), many of which are extinct now. Currently, Devanagari and Perso-Arabic scripts are primarily used to write in Sindhi. Both these scripts are official scripts of Sindhi in India, whereas only Perso-Arabic is the official script of Sindhi in Pakistan.

During the colonial rule, the British chose Perso-Arabic as the standard script, which led to creation of large amount of literature in this script. There are many news websites and blogs in Sindhi (Perso-Arabic) published from Pakistan.[3] This may be because Sindhi speakers are more in Pakistan than India and also have a geographical state called 'Sindh'. In contrast, literature in Sindhi (Devanagari) on the web is very small. In India, Sindhi is an official language but not of a particular geographical state and therefore it does not enjoy the support that other state-official languages do.

[1] https://en.wikipedia.org/wiki/Sindhi_language

[2] Sindhi is an official language in India and Pakistan.

[3] http://www.abyznewslinks.com/pakis.htm

Proceedings of NAACL-HLT 2016, pages 51–58,
San Diego, California, June 12-17, 2016.

3 Related Work

Sindhi is written using two writing forms, the Devanagari script and the Perso-Arabic script. Previously, some research on Sindhi has been done with Perso-Arabic as the preferred script. An account of this research is given below.

A rule-based POS Tagger was developed by Mahar et al. (2010) using a lexicon of 26,355 entries and 67 tags. Its accuracy was reported to be 96.28%. A finite-state machine for capturing noun inflections in Sindhi was developed by Rahman et al. (2010). Zeeshan et al. (2015) have worked on developing a spell checker. Unfortunately, the above described tools are not publicly available. Therefore we could not evaluate and compare them or use them for developing resources for Sindhi (Devanagari).

A computational grammar library for Sindhi (Perso-Arabic) in Grammatical Framework[4] (Ranta, 2009) was developed by Jherna Devi Oad (2012). This library has 44 categories, 190 functions and 360 lexical entries. It was referred to during the development of our Sindhi (Perso-Arabic) morphological analyser.

4 Developing Datasets

A dataset is the most important requirement for building language technology tools and resources for any language. The following section describes how we collected and developed the datasets for both the scripts of Sindhi. A summary of the datasets and tools developed by us or other researchers for both scripts of Sindhi is provided in Table 1.

4.1 Sindhi (Devanagari) Datasets

The amount of raw texts available on the web for Sindhi (Devanagari) is very small. Initially we contacted various publishers and news agencies to source raw data, but the problem was further compounded as many publishers on the web have not yet moved to Unicode standards.

Raw Textual Data: We collected several short stories, books, articles, etc. and manually created data for Devanagari. Through this manual process, we were able to handle certain issues such as usage of correct Unicode encoding, normalization, script and grammar. Later, we developed a Unicode Converter[5] for legacy fonts, which helped us collect more data. We currently have a raw corpus of 326813 words, with average sentence length of 9.35 words and a vocabulary (unique words) of 22300 words.

Part-of-Speech Annotated Data: Since Sindhi did not have a POS Tagset, we adapted the BIS Tagset[6] which is comprehensive and designed to be extensible to any Indian Language. We annotated the data using this tagset and help from two annotators. We obtained a κ score (Cohen, 1960) of 0.96 when evaluated for Inter-Annotator Agreement on 793 words. Currently, we have tagged corpus of 44692 words. This data was subsequently used to build an automatic Part-of-Speech Tagger (discussed in Section 5.1).

4.2 Sindhi (Perso-Arabic) Datasets

As previously mentioned, large amount of content exists on the web for Sindhi in Perso-Arabic script, which can be used to source raw textual data.

Raw Textual Data: We collected textual data from Sindhi Wikipedia dump[7], news websites and blogs[8]. We currently have a corpus of about 1 million tokens.

Parallel Data: A sentence-aligned parallel corpora is an indispensable resource for any language pair. Many languages across the world are not fortunate enough to have such a parallel corpora available, including Sindhi. We have developed a small parallel corpus between Urdu and Sindhi, which are closely related languages. We initiated the development process by collecting some sentences from the Urdu Treebank (Bhat and Sharma, 2012), general conversations, news articles and essays and translating them to Sindhi manually. We now have a parallel corpus of 1400 sentences and it is being used for various purposes (Section 6), including automatic generation of more parallel data (see 6.3).

[4] http://www.grammaticalframework.org/lib/src/sindhi

[5] http://goo.gl/d5a8X2

[6] http://goo.gl/AZxk7x

[7] https://dumps.wikimedia.org/sdwiki/sdwiki-20150826-pages-articles.xml.bz2

[8] http://tinyurl.com/Sindhi-URLs

Data, Tools & Applications	Sindhi Devanagari	Sindhi Perso-Arabic
POS Annotated Data	**Yes**	Yes*
Chunk Annotated Data	No	No
Dependency Annotated Data	No	No
Parallel Data (Urdu-Sindhi)	No	**Yes**
POS Tagger	**Yes**	Yes*
Morphological Analyser	No	**Yes**
Spell-Checker	No	Yes*
Transliteration	**Yes**	**Yes**
Machine Translation (Urdu-Sindhi)	No	**Yes**

Table 1: The status of various resources developed for each script of Sindhi. * Resources developed by other researchers.

5 Developing Tools

After developing the datasets, we used them in creation of certain language technology tools which we describe below. Table 1 summarizes some tools developed for Sindhi by us and other researchers.

5.1 Part-of-Speech Tagger

Part-of-Speech (POS) tagging is the task of assigning an appropriate part-of-speech label to each word in a sentence, based on both its definition as well as its context. POS tagging is a fundamentally important task, as it gives information about words, their neighbors and the syntactic structure around them. This information is useful in syntactic parsing and other applications such as named-entity recognition, information retrieval, speech processing, etc.

The data that we annotated with POS tags was used to build an automatic POS Tagger[9] for Sindhi (Devanagari) (Motlani et al., 2015) using Conditional Random Fields[10] (Lafferty et al., 2001). We employed 10-fold cross validation to train and test the tagger. We experimented with several models by using various set of features, including linguistically motivated features such as *affixes* (which capture the morphological properties of the language) and *context* (which capture the syntactic structure of the language).

The current best performing model gives an average accuracy of 92.6% , which is 11% better than baseline[11] tagger. This tagger is being used to generate more POS annotated data through bootstrapping and post-editing methods.

5.2 Morphological Analyser

The morphological analysis of a word involves capturing its inherent properties such as gender, number, person, case, etc. Morphological features also help in improving the performance of other NLP tools such as, pos tagger, spell-checker, parsers, machine translation, etc. Thus, morphological analysis is a fundamental and crucial task.

We used Apertium's lttoolbox (Forcada et al., 2011) to develop a paradigm based finite-state morphological analyser for Sindhi (Perso-Arabic) (Motlani et al., 2016). This morphological analyser currently has about 3500 entries and a coverage of more than 81% on Sindhi Wikipedia dump consisting of 341.5k tokens. This analyser is publicly available on Apertium[12].

Sindhi is a morphologically rich language (Rahman and Bhatti, 2010). It uses suffixes for constructing derivational and inflectional morphemes. A major challenge for us is to incorporate the vast morphology. We currently have 72 paradigms in the analyser and are expanding them to cover all possible inflections. This, along with adding more entries to the lexicon, would help increase the coverage further. Another challenge is processing partially or fully diacritised input. The analyser can handle usual Sindhi texts which lack in diacritics but it tends to

[9] https://github.com/kindleton/sindhi_pos_tagger

[10] We used CRF++, an open source implementation of CRFs. https://taku910.github.io/crfpp/

[11] The baseline model assigns most frequent tag corresponding to a word, based on word-tag frequencies in training data.

[12] http://tinyurl.com/SindhiMorphAnalyser

make errors for other kinds of input because it is difficult to lookup in the lexicon and disambiguate.

5.3 Transliteration System

A transliteration system is a much needed tool to bridge the gap between content in Perso-Arabic and Devanagari scripts. Moreover, such a system could also facilitate sharing of resources developed in either scripts. Although a transliteration system would be very useful but there are various challenges that we face. Some of them are :

1. **Unavailability of Transliteration Pairs** : Transliteration pairs is a key resource for learning a transliteration model. In cases where a seed set is not available, transliteration pairs can be easily mined from a parallel data between the source and target language pair. We do not have any parallel data between Sindhi (Perso-Arabic) and Sindhi (Devanagari).

2. **Missing Diacritics** : Many Perso-Arabic script based languages do not use diacritics marks in their texts. This further leads to semantic and syntactic ambiguities, because a word can have multiple interpretations. An example: 'چپ' *cp* can be either *capa* 'lips' or *cupa* 'silent'.

3. **Differences in Character-Sets** : The alphabets in Sindhi (Perso-Arabic) are a variant of the Persian script. It is composed of 52 letters, including Persian letters, digraphs and eighteen other letters (illustrated in Table 2) to capture the sounds particular to Sindhi and other Indo-Aryan languages. The alphabets in Sindhi Devanagari are composed of 65 characters, including, short-vowels and 4 special characters representing Sindhi implosives. A one-to-one mapping cannot be developed between them.

5.3.1 Unsupervised Transliteration Pairs Mining

There is a lot of literature on automatic extraction of transliteration pairs using seed data and parallel corpora (Sajjad et al., 2012; Durrani et al., 2014; Jiampojamarn et al., 2010; Kumaran et al., 2010). Since our scenario is resource-poor, we designed and used an unsupervised approach for transliteration pair mining that prescinds from prior knowledge of seed corpus and parallel data.

ڱ	[ŋ]	ڃ	[ɲ]	ٻ	[ɓ]
ڳ	[ɠ]	ڄ	[ʄ]	ڀ	[bʱ]
ک	[k]	ڇ	[cʰ]	ڌ	[dʱ]
ڻ	[ɳ]	ٺ	[ʈʰ]	ڏ	[ɗ]
ڦ	[pʰ]	ٽ	[ʈ]	ڊ	[ɖ]
ڙ	[ɽ]	ٿ	[tʰ]	ڍ	[ɖʱ]

Table 2: The characters found in the alphabet of Sindhi (Perso-Arabic) script which are not present in the Persian alphabet and their phonetic representation.

In this approach, a discriminative transliteration detection engine takes three inputs: a limited character mapping[13] and unique word-list in source and target language.

These lists are iterated over to obtain a list of candidate word pairs. These candidate word pairs are then discriminated based on orthographic similarity. The orthographic similarity is measured by converting the characters of source and target word into an intermediate representation using the character mapping and calculating the edit-distance between them normalized by their word-length. The candidate pairs with larger edit-distance are pruned out and the remaining are treated as possible transliteration pairs.

5.3.2 Preliminary Results

The transliteration problem can be posed as a phrase-based SMT problem, where sentences are analogous to words and words are analogous to characters. We used the MOSES (Koehn et al., 2003) toolkit to train transliteration models by treating each transliteration pair (space separated sequence of characters) as the parallel data.

We had mined 112434 possible transliteration pairs from our raw datasets and trained a transliteration model. We evaluated it on a short story of 3247 words and obtained the following results shown in Table 3. We have also demonstrated an example in

[13] In our experiments we used a mapping of only those consonants and vowels which can be mapped one-to-one or many-to-one. Diacritics, most vowels and other ambiguous characters were not mapped. The bash command 'uconv' can be used to develop a mapping between various scripts.

Top-k	k=1	k=5	k=10	k=25
Precision (%)	60.14	83.27	87.12	90.08

Table 3: Results of preliminary experiments on transliteration. Top-k refers to first k candidates output by the system.

Table 4, where words of a source sentence in Sindhi (Perso-Arabic) are transliterated to Sindhi (Devanagari).

5.3.3 Context Aware Transliteration

The systems developed using previous approach can produce transliteration candidates for a given input word (as shown in Table 4), but there are various challenges in case of Sindhi (described in Section 5.3) because of which the precision of best output (top-1) is low. We believe this system can be improved using context in selecting the correct transliteration from candidate transliterations (top-k) of an input word. Currently, we are experimenting with context-based transliteration using Viterbi decoding and Language Model re-ranking.[14]

6 Statistical Machine Translation for Sindhi

Development of fundamental tools and resources discussed in the previous sections are important for larger NLP applications on Sindhi. An important application that can be developed without using these tools is an SMT system. Although phrase-based SMT requires only parallel data, rule-based and factored based machine translation systems depend on these fundamental tools.

In this section we shall discuss our ongoing work on developing a Sindhi-Urdu SMT system.

6.1 The Language Pair: Urdu and Sindhi

Sindhi and Urdu are spoken by a large number of native speakers (75 million and 159 million[15] around the world). These languages belong to Indo-Aryan language family and have evolved in the same geographical region for many years. Thus, they have many syntactic and semantic similarities. For instance, they share vocabulary, typography and sentence structures (for example, both follow subject-object-verb word-order). These similarities are major reasons behind choosing this language pair for the purpose of developing parallel data (Section 4.2) and subsequently a SMT system.

Source Word	Translit. (Top-3)	Ref. Word	Pos.
سندس	संदसि संदस संदनि	संदसि	1
جيب	जी जीब जीबु	जेब	4
مان	मां मानु में	मां	1
ڪو	को कयो कवि	को	1
موبائيل	मोबाइल मोबाइलु मोबाईल	मोबाईल	3
ڪڍي	कढी कढ़ी कढीय	कढी	1
ويو	वयो वियो वयोसि	व्यो	6
هئس	हुयस हिकु हास	हुउसि	None

Table 4: This table shows words of source sentence, their top-3 transliteration outputs given by the system, the reference word and the position at which an output matches the reference. This sentence is taken from test data, in English it means '*Someone had taken out mobile (phone) from his pocket (and left)*'.

In our opinion, Sindhi would benefit a lot from Sindhi-Urdu parallel data, as Urdu is comparatively resource rich language and techniques like projection (Yarowsky et al., 2001; Das and Petrov, 2011; Täckström et al., 2012) can be employed to leverage several resources for Sindhi.

6.2 Development

When we started working on SMT for Sindhi-Urdu, we only had about 1200 parallel sentences, a baseline

[14] Re-ranking the top-k transliteration candidates for ambiguous words in a particular context window

[15] https://en.wikipedia.org/wiki/Urdu

SMT system[16] was created using them.

This baseline system was evaluated on 70 held-out test sentences. The output sentences were given to a translator evaluation by rating each sentence on a scale of 1-5 (where, 1-very bad and 5-very good). The average rating obtained was 2.65 points. We also calculated other popular metrics for evaluation of MT system. BLEU (Papineni et al., 2002) score was 38.62, METEOR (Banerjee and Lavie, 2005) score was 77.97 and TER (Translation Error Rate) (Snover et al., 2006) was 38.28 . Note that, BLEU and METEOR scores are high due to small size of training data and vocabulary. Results of TER and human-evaluation are a better reflection of baseline system's performance.

6.3 Improvement

We manually analysed the errors made by the baseline SMT system and found that too many out-of-vocabulary (OOV) words. Other than those, words which were incorrectly translated were either due to presence in very low frequency in training data or due to ambiguity created by multiple possible translations.

Thus, we need to employ various techniques in order to significantly improve over baseline performance and develop a reasonably good translation system. One such technique is bootstrapping more parallel data using the baseline SMT system. Although, creating parallel data is faster in this process but it is still a time consuming and laborious task. Therefore, we also need to use certain automatic techniques. Some of them are described below

6.3.1 Bootstrapping more Parallel Data

The performance of a SMT system depends largely on the amount of parallel data used for training the system, which is very less in our case. Therefore, we are trying to generate more parallel data by using the baseline SMT system to bootstrap more parallel corpus. We source new sentences from the web (news articles, essays, short stories, etc.), translate it and then provide it to translators for post-editing.

6.3.2 Bilingual Lexicon Extraction from Comparable Corpora

Bilingual lexicon extraction is an automatic way to extract parallel data from non-parallel texts. Research in this area has been active for past several years and various approaches with promising results have been proposed (Rapp, 1995; Haghighi et al., 2008; Laroche and Langlais, 2010; Vulić et al., 2011; Irvine and Callison-Burch, 2013). The process involves finding possible translation candidates for a source word in target data using several features like orthographic, temporal, topical and contextual similarity. Presence of seed lexicon further benefits this process. Since Urdu and Sindhi are closely related languages and we have small parallel data, we can compute these features to induce lexicon of Urdu in Sindhi and obtain possible translation pairs.

We are exploring these different techniques on comparable data sourced from Wikipedia pages inter-lingually linked between Sindhi and Urdu and some news articles[17] published in these languages. The extracted parallel data will be supplemented to the phrase-table learned by Moses (Klementiev et al., 2012). This parallel data shall improve the coverage of the SMT system, although its impact on the SMT system's performance is yet to be evaluated.

6.3.3 Rule-Based Transliteration

The Sindhi (Perso-Arabic) and Urdu alphabets share many characters with very few differences. This typographic similarity can also be used to reduce OOV errors, specially for named entities. Therefore, we are developing a rule-based transliteration system by mapping the different characters in their scripts.

7 Conclusion

My thesis aims at developing some fundamental tools and resources and an application for a resource-poor and multi-script language, Sindhi. The main contribution of my work includes collection and creation of raw and annotated datasets, constructing NLP tools such as POS tagger, morphological analyser, building a transliteration system without parallel data in an unsupervised fashion and developing

[16] The baseline is a phrase-based SMT system, trained using Moses toolkit (Koehn et al., 2003) with word-alignments extracted from GIZA++ (Och and Ney, 2000) and using 3-gram language models created using KenLm (Heafield et al., 2013)

[17] Sindhi : http://www.onlineindusnews.com/
Urdu : http://www.onlineindusnews.net/

a SMT system for Sindhi-Urdu and improving it using various techniques. While my work shall supplement development of NLP applications for Sindhi, it shall also motivate research on languages surviving in similar resource-poor setting.

Acknowledgments

The research presented in this paper was done in collaboration with my advisors, Prof. Dipti M. Sharma and Dr. Manish Shrivastava. The Part-of-Speech annotation was done in collaboration with Dr. Pinkey Nainwani and Harsh Lalwani. I would like to acknowledge the people who have helped me various tasks like collecting the data, translations and understanding Sindhi better such as, Mehtab Ahmed Solangi, Mr. Bhagwan Babani and Mr. Chunnilaal Wadhwani. I would like to thank Nehal J. Wani, Arnav Sharma, Himanshu Sharma and Dr. Francis M. Tyers for their constant motivation and support. Lastly, I am thankful to the anonymous reviewers for their invaluable feedback and suggestions. This publication reflects the authors views only.

References

Satanjeev Banerjee and Alon Lavie. 2005. Meteor: An automatic metric for mt evaluation with improved correlation with human judgments. In *Proceedings of the acl workshop on intrinsic and extrinsic evaluation measures for machine translation and/or summarization*, volume 29, pages 65–72.

Riyaz Ahmad Bhat and Dipti Misra Sharma. 2012. A dependency treebank of urdu and its evaluation. In *Proceedings of the Sixth Linguistic Annotation Workshop*, pages 157–165. Association for Computational Linguistics.

Zeeshan Bhatti, Imdad Ali Ismaili, Waseem Javid Soomro, et al. 2015. Phonetic-based sindhi spellchecker system using a hybrid model. *Digital Scholarship in the Humanities*.

Jacob Cohen. 1960. A coefficient of agreement for nominal scales. *Educational and Psychological Measurement*.

Dipanjan Das and Slav Petrov. 2011. Unsupervised part-of-speech tagging with bilingual graph-based projections. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL '11)*, page Best Paper Award.

Nadir Durrani, Hassan Sajjad, Hieu Hoang, and Philipp Koehn. 2014. Integrating an unsupervised transliteration model into statistical machine translation. In *EACL*, pages 148–153.

M. L. Forcada, M. Ginestí-Rosell, J. Nordfalk, J. O'Regan, S. Ortiz-Rojas, J. A. Pérez-Ortiz, F. Sánchez-Martínez, G. Ramírez-Sánchez, and F. M. Tyers. 2011. Apertium: a free/open-source platform for rule-based machine translation. *Machine Translation*, 25(2):127–144.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. 2008. Learning bilingual lexicons from monolingual corpora.

Kenneth Heafield, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. 2013. Scalable modified Kneser-Ney language model estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 690–696, Sofia, Bulgaria, August.

Ann Irvine and Chris Callison-Burch. 2013. Combining bilingual and comparable corpora for low resource machine translation. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 262–270.

Sittichai Jiampojamarn, Kenneth Dwyer, Shane Bergsma, Aditya Bhargava, Qing Dou, Mi-Young Kim, and Grzegorz Kondrak. 2010. Transliteration generation and mining with limited training resources. In *Proceedings of the 2010 Named Entities Workshop*, pages 39–47. Association for Computational Linguistics.

Alexandre Klementiev, Ann Irvine, Chris Callison-Burch, and David Yarowsky. 2012. Toward statistical machine translation without parallel corpora. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 130–140. Association for Computational Linguistics.

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.

A Kumaran, Mitesh M Khapra, and Haizhou Li. 2010. Report of news 2010 transliteration mining shared task. In *Proceedings of the 2010 Named Entities Workshop*, pages 21–28. Association for Computational Linguistics.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Audrey Laroche and Philippe Langlais. 2010. Revisiting context-based projection methods for term-translation spotting in comparable corpora. In *Proceedings of the 23rd international conference on computational linguistics*, pages 617–625. Association for Computational Linguistics.

Javed Ahmed Mahar and Ghulam Qadir Memon. 2010. Rule based part of speech tagging of sindhi language. In *Proceedings of the 2010 International Conference on Signal Acquisition and Processing*, ICSAP '10, pages 101–106, Washington, DC, USA. IEEE Computer Society.

Raveesh Motlani, Harsh Lalwani, Manish Shrivastava, and Dipti Misra Sharma. 2015. Developing part-of-speech tagger for a resource poor language: Sindhi. In *Proceedings of the 7th Language and Technology Conference (LTC 2015), Poznan, Poland*.

Raveesh Motlani, Francis M. Tyers, and Dipti M. Sharma. 2016. A finite-state morphological analyser for sindhi. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*. European Language Resources Association (ELRA), May.

Jherna Devi Oad. 2012. *Implementing GF Resource Grammar for Sindhi language*. Msc. thesis, Chalmers University of Technology, Gothenburg, Sweden.

Franz Josef Och and Hermann Ney. 2000. Improved statistical alignment models. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 440–447. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Mutee U Rahman and Mohammad Iqbal Bhatti. 2010. Finite state morphology and Sindhi noun inflections. *In Proceedings of the 24th Pacific Asia Conference on Language, Information and Computation, PACLIC 24, Tohoku University, Japan*, 134:669–676.

Aarne Ranta. 2009. Gf: A multilingual grammar formalism. *Language and Linguistics Compass*, 3.

Reinhard Rapp. 1995. Identifying word translations in non-parallel texts. In *Proceedings of the 33rd Annual Meeting on Association for Computational Linguistics*, ACL '95, pages 320–322, Stroudsburg, PA, USA. Association for Computational Linguistics.

Hassan Sajjad, Alexander Fraser, and Helmut Schmid. 2012. A statistical model for unsupervised and semi-supervised transliteration mining. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics: Long Papers-Volume 1*, pages 469–477. Association for Computational Linguistics.

Matthew Snover, Bonnie Dorr, Richard Schwartz, Linnea Micciulla, and John Makhoul. 2006. A study of translation edit rate with targeted human annotation. In *Proceedings of association for machine translation in the Americas*, pages 223–231.

Oscar Täckström, Ryan McDonald, and Jakob Uszkoreit. 2012. Cross-lingual word clusters for direct transfer of linguistic structure. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, NAACL HLT '12, pages 477–487, Stroudsburg, PA, USA. Association for Computational Linguistics.

Ivan Vulić, Wim De Smet, and Marie-Francine Moens. 2011. Identifying word translations from comparable corpora using latent topic models. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers - Volume 2*, HLT '11, pages 479–484, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Yarowsky, Grace Ngai, and Richard Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of the First International Conference on Human Language Technology Research*, HLT '01, pages 1–8, Stroudsburg, PA, USA. Association for Computational Linguistics.