Data Issues in English-to-Hindi Machine Translation

Ondřej Bojar, Pavel Straňák, Daniel Zeman

Charles University in Prague
Univerzita Karlova v Praze, Ústav formální a aplikované lingvistiky
Malostranské náměstí 25, CZ-11800, Praha, Czechia
{bojar,stranak,zeman}@ufal.mff.cuni.cz

Abstract

Statistical machine translation to morphologically richer languages is a challenging task and more so if the source and target languages differ in word order. Current state-of-the-art MT systems thus deliver mediocre results. Adding more parallel data often helps improve the results; if it doesn't, it may be caused by various problems such as different domains, bad alignment or noise in the new data. In this paper we evaluate the English-to-Hindi MT task from this data perspective. We discuss several available parallel data sources and provide cross-evaluation results on their combinations using two freely available statistical MT systems. We demonstrate various problems encountered in the data and describe automatic methods of data cleaning and normalization. We also show that the contents of two independently distributed data sets can unexpectedly overlap, which negatively affects translation quality. Together with the error analysis, we also present a new tool for viewing aligned corpora, which makes it easier to detect difficult parts in the data even for a developer not speaking the target language.

1. Introduction

Machine translation is a challenging task and more so with significant differences in word order of the languages in question and with the target language explicitly marking more details in word forms than the source language does. Precisely this holds for the English-Hindi pair we study.

Bojar et al. (2008) tried to improve a statistical phrase-based baseline MT system by adding more data, standard lexicalized and heuristical rule-based reordering and an (unsupervised) explicit modeling of morphological coherence on the target side. Neither of the approaches helped much. Bojar et al. (2009) carried out a careful cleanup of the training data, experimented with several additional variants of morphological representation of the target side and evaluated the impact of a hiearchical instead of phrase-based translation model, achieving the best published BLEU score on the IIIT-TIDES test set.

Some of the published results however were counter-intuitive. The strangest result probably is that additional monolingual and parallel data did not significantly improve the performance, and in fact a renowned parallel corpus Emille caused a significant loss.

This paper examines the training and test sets in question as well as the most frequent errors of a few configurations of the phrase-based (e.g. Moses (Koehn et al., 2007)) and hierarchical (e.g. Joshua (Li et al., 2009)) models.

The structure of the paper is as follows: Section 3. describes the various datasets used in the experiments, including their specific problems and the normalization steps carried out. Section 4. examines the usefulness of various sections of the training data with respect to the test set, focusing mainly on the surprising detrimental effect of the Emille corpus. Section 5. analyzes the most frequent types of errors in MT output.

2. Phrase-Based Statistical Machine Translation Systems

In order to make the later sections accessible to a broader audience, we give a quick overview of a phrase-based SMT system. In general, phrase-based systems perform the following sequence of steps:

- Prepare the data. We need sentence-aligned parallel training data, development data and test data. In addition, we need monolingual (target language) data for training of the language model (might be much larger than just the target side of the parallel training data). Data preparation means substeps like tokenization, cleaning, sentence alignment and various preprocessing, if desired.
- Word-align the parallel training data. In all our experiments, we use mkcls and GIZA++[1] (Och and Ney, 2003) for this purpose, together with the *grow-diag-final-and* symmetrization heuristic (Koehn et al., 2003).
- Extract from the parallel training data a table of phrase pairs *(phrase table)* consistent with the symmetrized alignment. This is our **translation model (TM).** We experiment with two open-source translation systems, Moses (Koehn et al., 2007) and Joshua (Li et al., 2009). The most remarkable difference between the two is that Joshua represents the subset of *hierarchical* phrase-based systems (Chiang, 2007). These allow for phrases with gaps labeled by nonterminals; we thus speak of *grammars* instead of phrase tables.
- Train the target **language model (LM).** In all our experiments, we use the SRILM toolkit[2] (Stolcke, 2002) to create a trigram LM with the Kneser-Ney smoothing method (Kneser and Ney, 1995).

[1] http://fjoch.com/GIZA++.html

[2] http://www-speech.sri.com/projects/srilm/

- Optionally train other models, such as a reordering model.
- Each of the trained models (TM, LM and possibly others) is able to assign one or more scores *(features)* to any translation hypothesis. To combine these features into one overall score, we need a weight for every feature. Here is where the development data come into play. The core part of the system, called **decoder**, uses the models with some random initial weights to generate translation hypotheses for every source language sentence in the development set. The hypotheses are compared to reference translations and ranked according to an automatic translation quality measure (usually the BLEU score (Papineni et al., 2002)). The process is repeated and feature weights are updated between iterations so that locally optimal weights (w.r.t. translation quality on the development set) are converged to. This is called *minimum error rate training* **(MERT)** (Och, 2003).
- Finally, the decoder uses the locally optimal weights to produce translation hypotheses for the test data. These are evaluated against the target side of the test set (reference translations). Depending on setup, some post-processing steps may be needed before the actual evaluation.

3. Data

3.1. Tides

A dataset originally collected for the DARPA-TIDES surprise-language contest in 2002, later refined at IIIT Hyderabad and provided for the NLP Tools Contest at ICON 2008 (Venkatapathy, 2008): 50K sentence pairs for training, 1K development and 1K test data.

The corpus is a general domain dataset with news articles forming the greatest proportion. It is aligned on sentence level, and tokenized to some extent. We found the tokenization insufficient (e.g. *anglo-american* would be 1 token instead of 3, making the data sparseness problem more severe) and ran our own tokenizer on top of it.

There are over 2000 sentences containing Latin letters, none of which are used correctly (such as English citations within Hindi text would be). As far as we can tell, these are traces of encoding conversion failures. Most occurrences look like a misconverted punctuation, sometimes perhaps a lost Devanagari symbol. However, the most severe cases (more than 200) are sentences that start in Devanagari and suddenly switch into ASCII garbage. An example: प्रादेशिक - जनसंख्या बंगाली बंग्लादेश ह्यपूर्वी बंगालह् से आए अधिकांश विस्थापित दक्षिण अंडमान , नेल , हैवलाक , मध्य अंडमान , उ<*arI AMDmaana tqaa ilaiTla AMDmaana maoM basaae gae* .[3] Other noise, not betrayed by Latin characters, include ।भाष्; (misconversion of "|BAR;", standing for the sentence-terminating danda sign), or the mysterious sequence ऋ-ऊण्श्छ्ष्-[4].We apply automatic rules to get rid of all the problems mentioned, and a few more.

3.2. Daniel Pipes

A journalist Daniel Pipes' website:[5] limited-domain articles about the Middle East. The articles are originally written in English, many of them are translated to up to 25 other languages, including Hindi (322 articles, 6,761 sentence pairs).

3.3. Emille

EMILLE corpus (Baker et al., 2002) consists of three components: monolingual, parallel and annotated corpora. There are fourteen monolingual corpora, including both written and (for some languages) spoken data for fourteen South Asian languages: Assamese, Bengali, Gujarati, Hindi, Kannada, Kashmiri, Malayalam, Marathi, Oriya, Punjabi, Sinhala, Tamil, Telegu and Urdu. The EMILLE monolingual corpora contain approximately 92,799,000 words (including 2,627,000 words of transcribed spoken data for Bengali, Gujarati, Hindi, Punjabi and Urdu). The parallel corpus consists of 200,000 words of text in English and its accompanying translations in Hindi and other languages. Whenever we mention Emille, we mean the parallel English-Hindi section.

The original Emille had about 7000 sentences in each language; it turned out to be very badly aligned (many sentences without translation) and had spelling errors, so we worked with a manually cleaned and aligned subset of 3501 sentence pairs.[6]

3.4. Other Data

Various other small datasets:

ACL 2005: A subset of Emille, used in the shared task of the ACL 2005 workshop on "Building and Using Parallel Texts: Data Driven Machine Translation and Beyond".[7]

Wiki NEs: We have extracted English titles of Wikipedia entries, that are translations (or rather transcriptions) of Hindi, Marathi and Sanskrit named entities (names of persons, artifacts, places, etc.), and their translations in Devanagari. We used XML abstracts of English Wikipedia; the begining of an entry looks something like this, so it is easy to parse it with a regular expression:

> **Mumbai** (Marathi: मुंबई, *Mumbaī*…

We took not only Hindi, but also Marathi and Sanskrit entities on the assumption that these are also commonly used in Hindi texts.

Shabdanjali: A GPL-licensed collaborative English-Hindi dictionary, containing about 26,000 entries.

[3] We do not provide English translation of this example because its only purpose is to illustrate the broken script conversion.

[4] This repeatedly occurring sequence is not a valid Hindi word. In the so-called WX Romanization of Devanagari, it is rendered as "Q-UNSCR-" which is more likely to originate in a formatting control sequence.

[5] `http://www.danielpipes.org/`

[6] We are very grateful to Om Damani and his colleagues from IIT Mumbai for making their corrected subset of Emille available to us.

[7] Downloaded from the workshop website at `http://www.cse.unt.edu/~rada/wpt05/`. We used this dataset on the assumption that it might be better aligned, compared to the original Emille parallel corpus.

Available on the web in various formats and encodings. It is formatted similarly to printed bilingual dictionaries, so several filters had to be applied to refine a simple list of word pairs out of it. We test two variants of the dictionary. The full word list contains 32,159 word pairs (generated from the 26,000 entries – some words have more than one translation). Unfortunately, it also contains a lot of inherent noise and errors introduced by conversion steps carried out before we got the dictionary. Therefore we also tested a filtered version of only 1422 word pairs confirmed in a large monolingual corpus of Hindi. Later in the paper we refer to the full version as *dictfull* and to the filtered version as *dictfilt*.

Agriculture domain parallel corpus: English-Hindi-Marathi-UNL parallel corpus from Resource Center for Indian Language Technology Solutions.[8] It contains 17,105 English and 13,248 Hindi words in 527 parallel sentences.

3.5. Normalization

As always with statistical approaches, we want our training data to match the test data as closely as possible. There are style variations throughout our data that can and should be normalized automatically. For instance, the Tides corpus usually (except for some conversion errors) terminates a sentence with the period ("."). However, some of our additional data sets use the traditional Devanagari sign called *danda* ("।") instead. Our normalization procedure thus replaces all dandas by periods. A list of normalization steps follows. Note that some changes affect the English data as well!

- Convert the text to fully decomposed Unicode. For instance, any `DEVANAGARI LETTER FA` will be replaced by the sequence `DEVANAGARI LETTER PHA, DEVANAGARI SIGN NUKTA`. Note that both strings are identical in appearance.
- Replace Devanagari digits ("०१२३४५६७८९") by European digits ("0123456789").
- Replace danda ("।"), double danda ("॥") and the Devanagari abbreviation sign ("॰") by period (".").
- Replace *candrabindu* by *anusvar*, e.g. "पाँच" by "पांच".
- Remove all occurrences of *nukta*, effectively replacing "क़ख़ग़ज़ड़ढ़फ़" by "कखगजडढफ".
- Remove various control characters (yes, they occur in the data!), zero-width joiners and the like.
- Replace non-ASCII punctuation by their ASCII counterpart, e.g. "—" by "-".

Also common is the case where genuine English text has been (during some processing at the site of the data provider) interpreted as Devanagari encoded using Latin letters. Thus, ईन्ङोर्मटिओन् छोम्मिसिओनेर् *(īnṅormaṭion chommisioner)* should actually read (even in the Hindi text) *Information Commis(s)ioner*. If transcribed to Devanagari, it would probably yield something like इन्फ़ोर्मेशन कोमिशनेर *(informeśana komiśanera)*. Errors of this nature are not easy to detect automatically and we did not include their correction in the normalization procedure.

3.6. Features of Hindi Vocabulary

Hindi as the target language possesses some features that negatively influence MT performance: richer morphology than English and harder sparse-data problem due to vocabulary that combines words from various etymological sources.

One common cause of data sparseness is unstable orthography of English and other loanwords (or even transcribed citations), cf. the following counterparts of the English word "standards", all present in the data:

- स्टैंडर्डज *(sṭaiṁḍarḍaja)*
- स्टैंडर्डस *(sṭaiṁḍarḍasa)*
- स्टैंडड्र्स *(sṭaiṁḍarḍsa)*

Another source of data sparseness is the existence of many synonym pairs, in which one synonym is a Perso-Arabic loanword while the etymology of the other can be traced back to Sanskrit: see Table 1 and (Snell and Weightman, 2003), pp. 222–224.

4. Usefulness of Parallel Training Sets

Table 2, reproduced from (Bojar et al., 2009), documents the negative effect of adding Emille to the training set. The table summarizes automatic evaluation and manual quality judgments of 53 sentences drawn randomly from the Tides test set, with the baseline Moses setup (simple phrase-based translation, lexicalized reordering, language model trained on the full target side of the parallel data) trained on various subsets of the parallel data.[9] The starting point is the combination of Tides and Daniel Pipes (TIDP), adding Emille (EM), all other corpus sources (oth) and finally the two variants of the Shabdanjali dictionary: DICTFull and DICTFilt. Both the BLEU and manual scores indicate that Emille hurts the performance when tested on Tides test set: only 12 instead of 19 sentences were translated acceptably (labeled **). Adding further data reduced the detrimental effect of Emille (not observed in BLEU scores) but the best performance was achieved by Tides + Daniel Pipes only.

Of course, the destructive influence of Emille could be attributed to differences in domains of Emille vs. Tides (test data). If Emille just did not help, this would be an easy answer. However, it does not explain why Emille actually *hurts* so much.

Bojar et al. (2009) observed signs of overfitting to Tides development data when the MT system's training data contained Emille:

"MERT optimizes 5 feature weights of Joshua: the language model probability, the word penalty (preference for shorter translations), and three translation features: $P(e|f)$, $P_{lex}(f|e)$ and $P_{lex}(e|f)$. When Emille is involved, MERT always pushes the non-lexical translation probability extraordinarily high, and causes overfitting. While for other experiments we usually saw better BLEU scores on test data than on development data, the opposite was the case with Emille."

[8] `http://www.cfilt.iitb.ac.in/download/corpus/parallel/agriculture_domain_parallel_corpus.zip`

[9] The BLEU scores are computed on the full Tides test set, not just the 53 selected sentences.

English	Hindi/Persian	Hindi/Sanskrit
language	ज़बान *(zabāna)*	भाषा *(bhāṣā)*
book	किताब *(kitāba)*	पुस्तक *(pustaka)*
newspaper	अख़बार *(axbāra)*	समाचार-पत्र *(samācāra-patra)*
beautiful	ख़ूब्सूरत *(xūbsūrata)*	सुन्दर *(sundara)*
meat	गोश्त *(gośta)*	माँस *(mām̃sa)*
thank you	शुक्रिया *(śukriyā)*	धन्यवाद *(dhanyavāda)*

Table 1: Synonyms of different origin

System	0	*	**	BLEU
REF	0	8	45	
TIDP	20	14	19	11.89±0.76
TIDPEM	22	19	12	9.61±0.75
TIDPEMoth	17	25	11	10.97±0.79
TIDPEMothDICTFilt	23	17	13	10.96±0.75
TIDPEMothDICTFull	22	16	15	10.89±0.69

Table 2: The effect of additional (out-of-domain) parallel data in phrase-based translation.

In experiments evaluated on Emille, we split the 3501 sentence pairs of Emille to 3151 training, 175 development and 175 test pairs. In other cases (Emille used for training but not for MERT and testing), all 3501 pairs are used.

In our quest for the cause of the strange behavior of Emille with the original development set, we also ran a series of experiments on Tides with swapped development and test sets (identified as TideSwap).

Table 3 shows the results of the cross-evaluation of corpora. In all experiments there, word alignment was computed on automatic word "stems" defined as the first 4 characters of every word. The results were created by Joshua.

TM	LM	DT	DBleu	TBleu
Emille	Emille	Emille	9.33	10.16
Tides	Tides	Tides	11.45	12.08
Tides + DP	Tides	Tides	11.24	12.58
Tides + Emille	Tides	Tides	13.05	11.05
Tides + DP + Emille	Tides	Tides	12.98	11.32
Emille	Emille	Tides	9.03	1.75
Tides	Tides	TideSwap	12.78	10.66
Tides + DP	Tides	TideSwap	12.82	10.75
Tides + Emille	Tides	TideSwap	12.74	11.75
Tides + DP + Emille	Tides	TideSwap	12.64	11.68
Emille	Emille	TideSwap	2.26	7.38

Table 3: Cross-evaluation of the three main corpora with Joshua 1.1. DP = Daniel Pipes Corpus. DT = development and test data. DBLEU is the final MERT score on development set, TBLEU is the evaluation on the test data. For scores in the 10-12 range, the interval of statistical significance is about ±0.75.

The experiment with test data from Emille does not support suspicion that Emille is a bad corpus per se (note however that the development and test sets are very small in this case). The overfitting effect in the experiment trained on Tides + DP + Emille, tested on Tides, might suggest that there is some special relation between the Emille corpus and Tides development set. The TideSwap experiment without Emille gives a baseline for cross-evaluation of Emille in this environment. (Note that Tides test is easier to translate than Tides development, as evidenced by experiments with both Tides and TideSwap.) Finally, adding Emille to the TideSwap environment shows no sign of overfitting, and it actually significantly improves the test BLEU score. Altogether, the experiments suggest that training on Emille is extraordinarily suitable for Tides dev set but not for test set (the overfitting was due to that, too: MERT realized the value of Emille and pushed the weight of the translation model unrealistically high).

Finally, we took a closer look at the contents of Tides and Emille. To our surprise, it turned out that these two independently acquired resources overlap significantly:

- 2320 English sentences from Tides training set have been found in Emille (5% of the 50000 Tides sentences)
- 107 English sentences from Tides development set have been found in Emille (11% of the 1000 Tides sentences). Thus, adding Emille to the training data meant we were tuning the weights partially on our training data.
- No overlap of Emille and Tides test set has been detected.
- From the point of view of Emille, 69% of its 3501 English sentences have been found somewhere in Tides.
- Note that only English sentences were searched for. There is no guarantee that their Hindi equivalents are the same in both Tides and Emille. The careful manual alignment and cleaning performed on our version of Emille should mean that in case of any differences, Emille is the better source.

5. Error Analysis

Manual evaluation is an important complement of BLEU score; however, native Hindi speakers are not available for the full duration of the project and cannot carry full data-error analysis. We attempt to overcome this handicap by a number of data consistency checks, out-of-vocabulary statistics, and by developing tools for viewing interesting parts of the parallel corpora.

5.1. Out-of-Vocabulary Rates

Table 4 documents the coverage of various training sets we have tried with respect to both tides test sets: developement and evaluation. Out-of-vocaburary (OOV) rate is a ratio of words and their occurrences (i.e. types and tokens) from a test set that were not found in a training dataset. That means that a statistical translation system is required to translate these words, but could not have learned their translations during the training.

- We can see that all the data other then Tides train cover approx. 90% of tokens and 60% of types in Tides test set on the English side. On the Hindi side, the coverage in terms of types is quite a bit worse, but in terms of tokens it is quite similar.
- We can see that the numbers of types and tokens are always very similar. That means, that vast majority of OOV words have only one occurrence.
- The impact of the DP data is small, but still larger than the Shabdanjali dictionary.
- Hindi side of the data is always a bit worse. The reasons might be the richer morphology, but also some noise in the data. For instance there is an underscore that occurs cca 140 times in the Tides test, but never in the Tides train.

5.2. Dissecting Word Alignment

As part of our error analysis efforts, we developed a tool that can visualize alignments, quickly find word examples and summarize their alignments. This alignment viewer helps us better understand the corpora the SMT system uses for training. It also provides a great perspective for the translation hypotheses on the test data.

We assume that the corpus is word-aligned, i.e. that a file describing symmetrized word alignments is available. For the training part of the corpus, we must have created such alignment during training of the translation model. For the test data, we can obtain similar alignments if we append test source + reference translations and test source + system hypotheses to the training data, then run Giza++ again.

The tool consists of two parts: an indexer and a viewer. The indexer first reads the aligned corpus and remembers all occurrences and alignments of all words. It writes this information to index files. The viewer uses the index to find and access example sentence pairs quickly.

The viewer is powered by a Perl script that dynamically generates HTML (Web) content; the viewer itself is any web browser. Together with a web server[10] the scripts create an application that displays the contents of the corpus and quickly navigates through the examples. The web technology provides an easy and cheap approach to get a reasonably working text-oriented GUI portable to multiple platforms. It would be easier for the users not to be forced to use a web server; however, the contents must be dynamic, otherwise we would have to generate millions of static web pages that nobody would ever look at. The dynamic approach makes for easy access to all occurrences of all words in the corpus. All words are clickable, so the space of translations of translations (of ...) can be explored to any depth.

Another feature of the indexer is that it creates overall statistics of alignment counterparts of any word. Such summaries make it easy to discover ambiguous translations such as those of *book* (Figure 2).

Useful as the tool is, one has to bear in mind that it is based solely on the word alignment, i.e. on the input that the SMT training module works with. It cannot reveal directly what happens *inside* the SMT system, and thus *why* did it output what it did. However, training corpora are often the key to the success of the SMT systems, so the alignment viewer can at least shed light on a substantial part of the potential error sources.[11]

6. Conclusion

We have described a number of English-Hindi corpora and other parallel resources available for machine translation research. We have presented methods for data cleaning and pointed out many normalization issues specific for Hindi.

In the experimental part, we have presented a detailed discussion of BLEU scores obtained by training and cross-testing on various datasets. Both new and previously published results have been compared. The most important outcome of this paper is the overlap detected in two substantial corpora, Tides and Emille. Although it may seem trivial, due to close to zero documentation of the available datasets, the overlap is not obvious for new users – but it significantly affects the research results.

Finally, in the error analysis part, we presented a new tool for browsing through aligned corpora, which helps even a non-speaker of Hindi to find alternative translations of the same word, match test data examples to training data and much more.

7. Acknowledgements

The research has been supported by the grants MSM0021620838 (Czech Ministry of Education) and EuromatrixPlus (FP7-ICT-2007-3-231720 of the EU and 7E09003 of the Czech Republic).

8. References

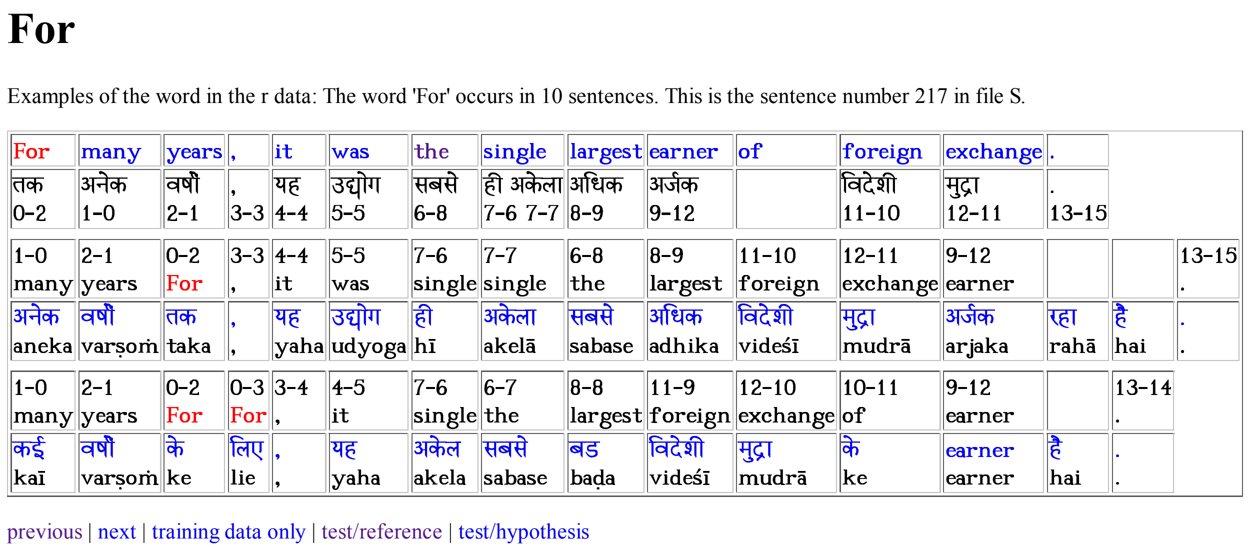
Paul Baker, Andrew Hardie, Tony McEnery, Hamish Cunningham, and Rob Gaizauskas. 2002. EMILLE, A 67-Million Word Corpus of Indic Languages: Data Collection, Mark-up and Harmonisation. In *Proceedings of LREC 2002*, pages 819–827, Lancaster. Lancaster University.

Ondřej Bojar, Pavel Straňák, and Daniel Zeman. 2008. English-Hindi Translation in 21 Days. In *Proceedings of ICON 2008 NLP Tools Contest*, Pune, India.

Ondřej Bojar, Pavel Straňák, Daniel Zeman, Gaurav Jain, Michal Hrušecký, Michal Richter, and Jan Hajič. 2009. English-Hindi Translation—Obtaining Mediocre Results with Bad Data and Fancy Models. In *Proceedings of ICON 2009*, Hyderabad, India.

[10] Such as the Apache web server, which is freely available for several platforms and can be installed locally even on a Windows laptop.

[11] The viewer is still under development. There is some documentation at `https://wiki.ufal.ms.mff.cuni.cz/user:zeman:addicter`, and the code can be obtained from `https://failfinder.googlecode.com/svn/trunk/`



For

Examples of the word in the r data: The word 'For' occurs in 10 sentences. This is the sentence number 217 in file S.

For	many	years	,	it	was	the	single	largest	earner	of	foreign	exchange	.		
तक	अनेक	वर्षों	,	यह	उद्योग	सबसे	ही अकेला	अधिक	अर्जक		विदेशी	मुद्रा	.		
0-2	1-0	2-1	3-3	4-4	5-5	6-8	7-6 7-7	8-9	9-12		11-10	12-11	13-15		
1-0	2-1	0-2	3-3	4-4	5-5	7-6	7-7	6-8	8-9	11-10	12-11	9-12			13-15
many	years	For	,	it	was	single	single	the	largest	foreign	exchange	earner			.
अनेक	वर्षों	तक	,	यह	उद्योग	ही	अकेला	सबसे	अधिक	विदेशी	मुद्रा	अर्जक	रहा	है	.
aneka	varṣoṁ	taka	,	yaha	udyoga	hī	akelā	sabase	adhika	videśī	mudrā	arjaka	rahā	hai	.
1-0	2-1	0-2	0-3	3-4	4-5	7-6	6-7	8-8	11-9	12-10	10-11	9-12		13-14	
many	years	For	For	,	it	single	the	largest	foreign	exchange	of	earner		.	
कई	वर्षों	के	लिए	,	यह	अकेल	सबसे	बड	विदेशी	मुद्रा	के	earner	है	.	
kaī	varṣoṁ	ke	lie	,	yaha	akela	sabase	baḍa	videśī	mudrā	ke	earner	hai	.	

previous | next | training data only | test/reference | test/hypothesis

Figure 1: A screenshot of the alignment viewer showing example sentence for the English word *For*. This example is from the test data set, hence there are three versions of the sentence (top-down): the English source sentence, the Hindi reference translation, and the translation hypothesized by the SMT system. The alignment counterparts of the source words are identified directly beneath every source word. Besides the alignment indices from the alignment file, the corresponding target word (from reference translation) is copied over here, too. Similarly, source counterparts of target words are described immediately above the target words. The hyperlinks make browsing other corpus occurrences of the word really fast and easy. Also, all words in the example sentence are clickable and provide direct links to their own sets of examples.

David Chiang. 2007. Hierarchical Phrase-Based Translation. *Computational Linguistics*, 33(2):201–228.

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 181–184, Los Alamitos, California, USA. IEEE Computer Society Press.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 48–54, Morristown, NJ, USA. 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 ACL Demo and Poster Sessions*, pages 177–180, Praha, Czechia.

Zhifei Li, Chris Callison-Burch, Sanjeev Khudanpur, and Wren Thornton. 2009. Decoding in Joshua: Open Source, Parsing-Based Machine Translation. *The Prague Bulletin of Mathematical Linguistics*, 91:47–56, 1.

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

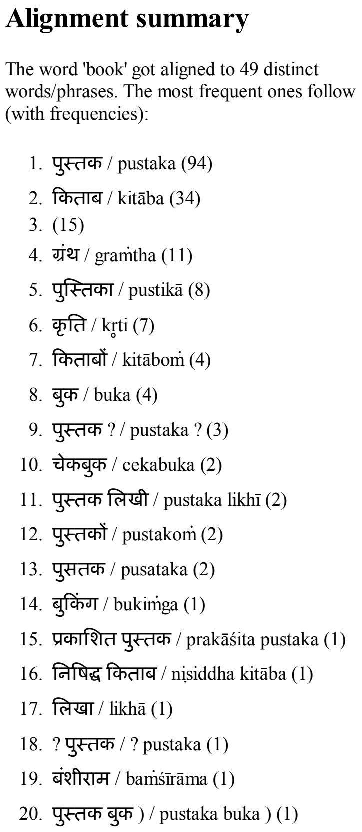
Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *ACL '03: Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, pages 160–167, Morristown, NJ, USA. Association for Computational Linguistics.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

Rupert Snell and Simon Weightman. 2003. *Teach Yourself Hindi*. Hodder Education, London, UK.

Andreas Stolcke. 2002. Srilm – an extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*, Denver, Colorado, USA.

Sriram Venkatapathy. 2008. NLP Tools Contest – 2008: Summary. In *Proceedings of ICON 2008 NLP Tools Contest*, Pune, India.



Alignment summary

The word 'book' got aligned to 49 distinct words/phrases. The most frequent ones follow (with frequencies):

1. पुस्तक / pustaka (94)
2. किताब / kitāba (34)
3. (15)
4. ग्रंथ / gramtha (11)
5. पुस्तिका / pustikā (8)
6. कृति / kr̥ti (7)
7. किताबों / kitābom̐ (4)
8. बुक / buka (4)
9. पुस्तक ? / pustaka ? (3)
10. चेकबुक / cekabuka (2)
11. पुस्तक लिखी / pustaka likhī (2)
12. पुस्तकों / pustakom̐ (2)
13. पुसतक / pusataka (2)
14. बुकिंग / bukiṁga (1)
15. प्रकाशित पुस्तक / prakāśita pustaka (1)
16. निषिद्ध किताब / niṣiddha kitāba (1)
17. लिखा / likhā (1)
18. ? पुस्तक / ? pustaka (1)
19. बंशीराम / baṁśīrāma (1)
20. पुस्तक बुक) / pustaka buka) (1)

Figure 2: Summary of the most frequent alignment counterparts, as provided by the alignment viewer for every source and target word. Here, the counterparts of the English word *book* are summarized. The most frequent translations are the two etymologically different Hindi synonyms from Table 1: Sanskrit-originating पुस्तक and Perso-Arabic किताब. Note also the transliterated English *(buka)* (8). Other remarkable translations are the plural oblique forms (7, 12), transliteration of the English *booking* (14) and phrases such as प्रकाशित पुस्तक *(prakāśita pustaka)* “published book” (15). The line no. 3 indicates that in 15 cases, *book* lacked any aligned counterpart.

	Coverage									
	tokens unseen in train					**types unseen in train**				
	Tides	*Tides+DP*	*Tides+dict*	*Tides+DP+dict*	*All-Tides*	*Tides*	*Tides+DP*	*Tides+dict*	*Tides+DP+dict*	*All-Tides*
Tides-test-en	369	348	363 (1.336%)	343 (1.262%)	2429 (8.940%)	363	343	357 (6.011%)	338 (5.691%)	1901 (32.009%)
Tides-test-hi	839	830	836 (2.926%)	828 (2.898%)	3310 (11.584%)	642	633	639 (10.882%)	631 (10.746%)	2465 (41.979%)
Tides-dev-en	464	421	462 (2.055%)	419 (1.863%)	1873 (8.330%)	459	418	457 (8.167%)	416 (7.434%)	1608 (28.735%)
Tides-dev-hi	619	607	618 (2.537%)	606 (2.487%)	2661 (10.922%)	580	568	579 (10.262%)	567 (10.050%)	2129 (37.735%)

Table 4: Out-of-vocabulary rates for both English and Hindi and various combinations of training data: Tides, Tides plus DanielPipes, Tides with the Shabdanjali dictionary filtered as described in Section 3.4. and everything except Tides.