



## TmTriangulate: A Tool for Phrase Table Triangulation

Duc Tam Hoang, Ondřej Bojar

Charles University in Prague, Faculty of Mathematics and Physics, Institute of Formal and Applied Linguistics

---

### Abstract

Over the past years, pivoting methods, i.e. machine translation via a third language, gained respectable attention. Various experiments with different approaches and datasets have been carried out but the lack of open-source tools makes it difficult to replicate the results of these experiments. This paper presents a new tool for pivoting for phrase-based statistical machine translation by so called phrase-table triangulation. Besides the tool description, this paper discusses the strong and weak points of various triangulation techniques implemented in the tool.

---

### 1. Introduction

Training algorithms for statistical machine translation (SMT) generally rely on a large parallel corpus between the source language and target language. This paradigm may suffer from serious problems for under-resourced language pairs, for which such bilingual data are insufficient. In fact, if we randomly pick two living human languages, the pair will likely be under-resourced. Hence, most of the language pairs cannot benefit from standard SMT algorithms.

To alleviate the problem of data scarcity, *pivoting* has been introduced. It involves the use of another language, called *pivot language*, *bridge language* or the *third language*, to include resources available for the pivot language in the system. Over the years, a number of pivoting methods have been proposed, including system cascades, synthetic corpus, phrase table translation and most recently, phrase table triangulation.

Figure 1 shows a schematic overview of the SMT process and the interaction with various pivoting methods.

“System cascades” basically consist of translating the input from the source language into the pivot language, e.g. English, and then translating the obtained hypotheses into the target language. In the synthetic corpus method, the pivot side of a

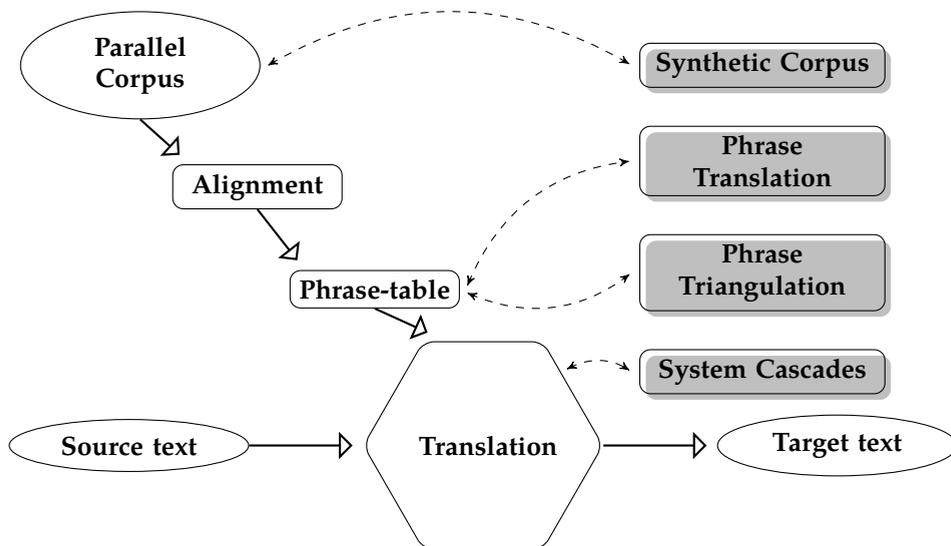


Figure 1: Pivoting methods

source-pivot or pivot-target parallel corpus is first translated to obtain a source-target corpus where one side is synthetic. A standard system is then trained from the obtained corpus. In the phrase table translation method, one side of an existing pivot-source or pivot-target phrase table is translated. And finally, the phrase table triangulation method (sometimes called simply *triangulation method*) combines two phrase tables, namely source-pivot and pivot-target, into an artificial source-target phrase table. Phrase table triangulation and translation thus manipulate directly with the internals of the SMT system, compared to system cascades, which uses source-to-pivot and pivot-to-target systems as black boxes, and synthetic corpus, which adjusts the training corpus.

Deploying system cascades in practice requires the two black-box systems running. *Phrase table triangulation* removes this requirement, capturing the new knowledge in a standard static file. This phrase table can then be used with any other SMT technique, see e.g. Zhu et al. (2014). One of the common operations is to merge several phrase tables for one language pair into one. The Moses toolkit (Koehn et al., 2007) includes a number of methods or tools for this: *alternative decoding paths*, *phrase table interpolation* (*tmcombine*; Sennrich, 2012) and *phrase table fill-up* (*combine-ptables*; Nakov, 2008; Bisazza et al., 2011).

In the past few years, promising results have been reported using phrase table triangulation methods (Cohn and Lapata, 2007; Razmara and Sarkar, 2013; Zhu et al.,

2014), but without releasing any open-source tool. We decided to fill this gap and implement an easy-to-use tool for phrase table triangulation in its several variants.

## 2. Phrase Table Triangulation

In short, phrase table triangulation fuses together source-pivot and pivot-target phrase tables, generating an artificial source-target phrase table as the output. Since each of the phrase tables usually consists of millions of phrase pairs, phrase table triangulation is computationally demanding (but lends itself relatively easily to parallelization).

When constructing the source-target table, we need to provide the set of:

1. source and target phrases,  $\bar{s}-\bar{t}$ ,
2. word alignment  $a$  between them,<sup>1</sup>
3. scores (direct and reverse phrase and lexical translation probabilities).

Two techniques were examined for the last step, namely *pivoting probabilities* (see Section 2.3; Cohn and Lapata, 2007; Utiyama and Isahara, 2007; Wu and Wang, 2007) and *pivoting co-occurrence counts* (see Section 2.4; Zhu et al., 2014).

### 2.1. Linking Source and Target Phrases

For source and target phrases, we do the most straightforward thing. We connect  $\bar{s}$  and  $\bar{t}$  whenever there exists a pivot phrase  $\bar{p}$  such that  $\bar{s}-\bar{p}$  is listed in the source-pivot and  $\bar{p}-\bar{t}$  is listed in the pivot-target phrase table. This approach however potentially springs serious problems.

Firstly, we do not check any context or meaning of the phrases, so an ambiguous pivot phrase  $\bar{p}$  can connect source and target phrases with totally unrelated meaning. This issue is more likely for short and frequent phrases.

Secondly, errors and omissions caused by noisy word alignments, which are unavoidable, are encountered twice. This leads to much higher level of noise in the final source-target table.

Thirdly, the noise boosts the number of common or short phrase pairs and omits a great proportion of large or rare phrase pairs. As the method relies on identical pivot phrases to link source phrases and target phrases, the longer the phrase is, the smaller the probability that there will be a pair.

And finally, we create the Cartesian product of the phrases, so the resulting phrase table is much larger than the size of the source phrase tables.

---

<sup>1</sup>Strictly speaking, word alignment within phrases doesn't have to be provided, but the word alignment output of the final decoder run is useful in many applications. We also use the word alignment for pivoting lexical translation probabilities.

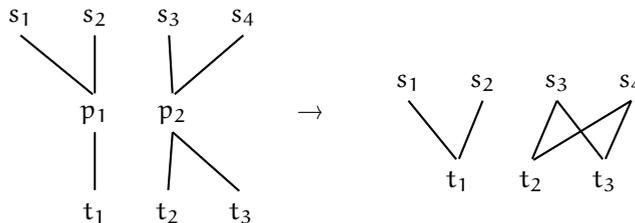


Figure 2: Constructing source-target alignment

### 2.2. Word Alignment for Linked Phrases

Given a triplet of source, pivot and target phrases  $(\bar{s}, \bar{p}, \bar{t})$  and the source-pivot ( $a_{sp}$ ) and pivot-target ( $a_{pt}$ ) word alignments, we need to construct the source-target alignment  $a$ . We do this simply by tracing the alignments from each source word  $s \in \bar{s}$  over any pivot word  $p \in \bar{p}$  to each target word  $t \in \bar{t}$  as illustrated in Figure 2. Formally:

$$(s, t) \in a \Leftrightarrow \exists p : (s, p) \in a_{sp} \ \& \ (p, t) \in a_{pt} \tag{1}$$

### 2.3. Pivoting Probabilities

Cohn and Lapata (2007) and Utiyama and Isahara (2007) considered triangulation as a generative probabilistic process which estimates new features based on features of the source-pivot and pivot-target phrase tables. This included an independence assumption of the conditional probabilities between  $\bar{s}$ ,  $\bar{t}$  and  $\bar{p}$ :

$$\begin{aligned} p(\bar{s}|\bar{t}) &= \sum_{\bar{p}} p(\bar{s}|\bar{p}, \bar{t}) p(\bar{p}|\bar{t}) \\ &\approx \sum_{\bar{p}} p(\bar{s}|\bar{p}) p(\bar{p}|\bar{t}) \end{aligned} \tag{2}$$

Equation 2 finds the conditional over the pair  $(\bar{s}, \bar{t})$  by going through all pivot phrases which are paired with both  $\bar{s}$  and  $\bar{t}$ . If we assume that each phrase  $\bar{p}$  represents a different sense, this can be viewed as including all phrase senses in the pivot language that  $\bar{s}$  and  $\bar{t}$  share.

The conditional probability can be simplified further by taking the maximum value instead of summing over all pivot phrases. This can potentially avoid the noise created

by alignment errors and corresponds to considering only the most prominent sense in the pivot language in our analogy. However, this may oversimplify the conditional probability and lead to information loss.

We apply the formula in Equation 2 to all four components of phrase pair scores: forward and direct phrase and lexically-weighted translation probabilities. Empirically, the resulting scores work reasonably well, but they are obviously not well defined probabilities.

### 2.4. Pivoting Co-Occurrence Counts

Zhu et al. (2014) introduced another approach to estimate the new features from the raw co-occurrence counts in the two source phrase tables.

Given the source-pivot co-occurrence count  $c(\bar{s}, \bar{p})$  and the pivot-target count  $c(\bar{p}, \bar{t})$ , we need to select a function  $f(\cdot, \cdot)$  that leads to a good estimate of the source-target count:

$$c(\bar{s}, \bar{t}) = \sum_p f(c(\bar{s}, \bar{p}), c(\bar{p}, \bar{t})) \tag{3}$$

There are four simple choices for  $f(\cdot, \cdot)$  in Equation 3: the minimum, maximum, arithmetic mean and geometric mean. Zhu et al. (2014) considers the minimum as the best option.

Once the co-occurrence count for the phrase pair  $(\bar{s}, \bar{t})$  in the synthetic source-target table is estimated, the direct and reverse phrase translation probabilities  $\phi$  and their lexically-weighted variants  $p_w$  can be calculated using the standard procedure (Koehn et al., 2003). The reverse probabilities are calculated using the following formulas, the direct ones are estimated similarly:

$$\begin{aligned} \phi(\bar{s}|\bar{t}) &= \frac{c(\bar{s}, \bar{t})}{\sum_s c(\bar{s}, \bar{t})} \\ p_w(\bar{s}|\bar{t}, \alpha) &= \prod_{i=1}^n \frac{1}{|j|(i, j) \in \alpha|} \sum_{(i, j) \in \alpha} w(s_i|t_j) \end{aligned} \tag{4}$$

In Equation 4, the lexical translation probability  $w$  between source word  $s$  and target word  $t$  must be computed beforehand as follows:

$$w(s|t) = \frac{c(s, t)}{\sum_{s'} c(s', t)} \tag{5}$$

Since we no longer have access to the *word* co-occurrence counts or lexical probabilities (the files `.f2e` and `.e2f` in Moses training), we estimate them from the pivoted

phrase table, i.e. the set of phrase pairs  $(\bar{s}, \bar{t})$  that include the respective words  $s$  and  $t$  aligned:

$$c(s, t) = \sum_{\{(\bar{s}, \bar{t}) \mid s \in \bar{s} \& t \in \bar{t} \& (s, t) \in \alpha\}} c(\bar{s}, \bar{t}) \quad (6)$$

Pivoting co-occurrence counts is intuitively appealing because it leads to proper (maximum likelihood) probability estimates. On the other hand, it needs a good estimate for the co-occurrence counts in the first place. The approach works well if the parallel corpora are clean, of a similar size and distribution of words. Naturally, this is the case of multi-parallel corpora rather than two independent parallel corpora.

### 3. TmTriangulate

Our open-source tool for all the described variants of phrase table triangulation is designed to work with the Moses standard text format of phrase tables, making it compatible with other tools from the Moses toolkit, esp. tools for phrase table combination: *tmcombine* or *combine-ptables*.

As phrase table triangulation is a data-intensive operation, processing two huge files, it is not possible to keep the list of phrase pairs in memory. In fact, even the list of all phrase pairs associated with only one source phrase sometimes led to memory overload.

We therefore split triangulation into two steps: *triangulate* and *merge*. The first step, *triangulate*, is a *mergesort*-like process, handling phrase tables by travelling along the sorted pivot side of both input phrase tables. Once the same pivot phrase is spotted in both files, the source-target pair is established and emitted to a temporary output file with its (temporary) score values.

The second step sorts records of the temporary file and then merges values of all occurrences of the same source-target pair into one entry. Multi-threading is used in the second step for a better performance.

#### 3.1. TmTriangulate Parameters

TmTriangulate command-line options are simple:

**action** select whether is it probabilities (`features_based`) or co-occurrence counts (`counts_based`) that should be pivoted.

**weight combination (-w)** specifies handling for phrase pairs linked by more than one pivot phrase. The two accepted options `summation` and `maximization` correspond to summing over the pivot phrases or getting solely the maximum value for each score. If the value is not defined, `summation` option is chosen as default.

**co-occurrence counts computation (-co)** specifies the function  $f$  used to combine counts from the two input tables, see Section 2.4. Allowed values are: `min`, `max`, `a-mean` and `g-mean`.

Parallel Corpus	Sentences	Tokens		
		Czech	English	Vietnamese
Czech-Vietnamese	1.09M	6.71M	–	7.65M
Czech-English	14.83M	205.17M	235.67M	–
English-Vietnamese	1.35M	–	12.78M	12.49M

Table 1: Sizes of parallel corpora used in our experiments.

**mode (-m)** clarifies the direction of component phrase tables, i.e. *source*→*pivot* or *pivot*→*source*. Accepted values are *pspt*, *sppt*, *pstp* and *sptp*, where the first pair of characters describes the source-pivot table and the second pair describes the pivot-target table.

**source (-s) and target (-t)** specify source-pivot and pivot-target phrase table files or directories with a given structure.

**output phrase table (-o) and output lexical (-l)** specify the output files. If the output file is not defined, *tmtriangulate* writes the source-target phrase table to the standard output.

For example, the following command constructs a Czech-Vietnamese phrase table by pivoting probabilities from the English-Czech (*en2cs.ttable.gz*) and English-Vietnamese (*en2vi.ttable.gz*) files:

```
./tmtriangulate.py features_based -m pspt \
-s en2cs.ttable.gz -t en2vi.ttable.gz
```

A detailed description of all parameters is provided along with the source code.

## 4. Experiments with Czech and Vietnamese

To illustrate the utility of *tmtriangulate*, we carry out an experiment with translation between Czech (*cs*) and Vietnamese (*vi*). English (*en*) is chosen as the sole pivot language.

### 4.1. Experiment Overview

The training data consist of three corpora: for *cs-en*, we use *CzEng 1.0* (Bojar et al., 2012) and for *cs-vi* and *en-vi*, we combine various sources including OPUS, TED talks and fragmented corpora published by previous works. Table 1 summarizes the sizes of our parallel data. Hence, the resources are unrelated and they are drastically different in size.

For completeness, our language model data are described in Table 2, we build standard 6-gram LMs with modified Kneser-Ney smoothing using KenLM (Heafield et al., 2013).

Monolingual Corpus	Sentences	Tokens
Czech	14.83M	205.17M
Vietnamese	1.81M	48.98M

Table 2: Sizes of monolingual corpora used for language models.

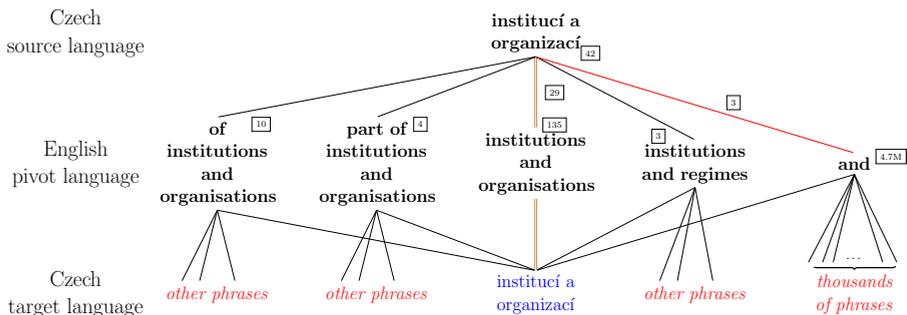


Figure 3: An example of triangulation with CzEng 1.0 corpus

Overall, the experiment is conducted with two directions:  $cs \rightarrow vi$  and  $vi \rightarrow cs$ . We use *tmtriangulate* to combine phrase tables of  $cs \rightarrow en$  and  $en \rightarrow vi$  into the  $cs \rightarrow vi$  and  $vi \rightarrow cs$  tables. We use several settings for the triangulation to highlight the differences between them. Finally, we combine the best pivoted model with the standard phrase-based model extracted from an OPUS and TED direct parallel corpus between Czech and Vietnamese.

All systems are evaluated on a golden test set, obtained by manually translating the WMT13 test set<sup>2</sup> into Vietnamese, so there is no overlap between the training, tuning and evaluation data.

## 4.2. Noise Gained through Pivoting

We start with a quick manual inspection of the pivoted phrase tables. Differences in the domains and sizes of the source corpora are generally considered as the reasons behind the poor performance of triangulated models. Our analysis shows that alignment errors generate an immense amount of noise, degrading phrase table quality. For illustration purposes, we use the same phrase table twice, pivoting “from Czech to Czech” via English. This is actually one of the standard approaches to data-driven paraphrasing (Bannard and Callison-Burch, 2005) and obviously there cannot be any

<sup>2</sup><http://www.statmt.org/wmt13/translation-task.html>

Approach	Option	vi→cs BLEU	cs→vi BLEU
Pivoting probabilities	summation	<b>7.44</b>	<b>10.28</b>
Pivoting probabilities	maximization	7.21	9.64
Pivoting co-occurrence counts	minimum	<b>7.24</b>	<b>9.86</b>
Pivoting co-occurrence counts	maximum	6.38	7.64
Pivoting co-occurrence counts	arithmetic-mean	6.25	6.95
Pivoting co-occurrence counts	geometric-mean	7.05	9.24
Direct system	-	7.62	10.59

Table 3: BLEU scores for phrase table triangulation for translation between Czech and Vietnamese via English.

discrepancies due to corpus size or domain. Yet, the pivoted phrase table contains many entries that distort the meaning. See Figure 3 for an example. The Czech phrase “institucí a organizací” by no doubt should be paired with a target phrase which has the sense: “institutions and organizations”. Indeed, the correct phrase pair has 29 co-occurrences, out of 135 appearances of “institutions and organizations” alone. The problem is that the single-word phrase “and” is listed as one of the possible translations and licenses a very large number of very distant phrases. It is just the 3 spurious co-occurrences with “and” that bring in the many bad phrases.

Our preliminary observations suggest that, after adding the pivot-target phrase table and estimating pivoted co-occurrence counts, the differences between good pairs and bad pairs get blurred. Estimating the new scores from source tables’ probabilities seems to keep the gap between good pairs and bad pairs wider. A more thorough analysis is nevertheless desirable.

### 4.3. Results of Pivoted Models Alone

Table 3 shows our first experimental results based on pivoted phrase tables.

The high level of noise leads to very large pivoted phrase tables with many bad phrases. The pivoted systems thus achieve relatively bad scores despite the large size of their phrase tables, many times larger than the size of the component phrase tables. Of the six triangulation options, the best one achieves results similar to the direct system, which is based on parallel cs-vi data.

The overall differences between the various triangulation approaches are not very big, especially concerning the high level of noise. We nevertheless see that for this set of languages and corpora, pivoting probabilities leads to better results than pivoting co-occurrence counts.

Method	Table Size	vi→cs BLEU	cs→vi BLEU
Direct System	8.8M	7.62	10.59
Best Pivoted System	61.5M	7.44	10.28
Linear Interpolation ( <i>tmcombine</i> )	69.3M	8.33	11.98
Alternative Decoding Paths	8.8M/61.5M	8.34	11.85

Table 4: Combining direct and pivoted phrase tables.

#### 4.4. Combination with the Baseline Phrase Table

While the triangulation results did not improve over the baseline in the previous section, triangulation has reportedly brought gains *in combination* with the direct phrase table. Since the direct and the pivoted phrase tables have the same format, it is very easy to merge them.

We examine two options to combine the direct phrase table with the best pivoted phrase table: alternative decoding paths and phrase table interpolation. Alternative decoding paths in Moses use both tables at once and the standard MERT is used to optimize the (twice as big) set of weights, estimating the relative importance of the tables. Phrase table interpolation is implemented in *tmcombine* (among others) and merges the two tables with uniform weights before Moses is launched.

Table 4 confirms the reported results: the combined systems are significantly better than each of their components. We do not see much difference between alternative decoding paths and phrase table interpolation.

## 5. Conclusion

We discussed several options of pivoting, using a third language in machine translation. We focussed on phrase table triangulation and implemented a tool for several variants of the method. The tool, *tmtriangulate*, is freely available here:

<https://github.com/tamhd/MultiMT>

In our first experiment, phrase tables constructed by triangulation lead to results comparable but not better with the direct baseline translation. An improvement was achieved when we merged the direct and pivoted phrase tables with tools readily available in the Moses toolkit. It is however important to realize that different sets of languages, domains and corpora may show different behaviour patterns.

## Acknowledgments

This project has received funding from the European Union’s Horizon 2020 research and innovation programme under grant agreements n° 645452 (QT21) and

n° 644402 (HimL). The project was also supported by the grant SVV 260 104, and it is using language resources hosted by the LINDAT/CLARIN project LM2010013 of the Ministry of Education, Youth and Sports.

## Bibliography

- Bannard, Colin and Chris Callison-Burch. Paraphrasing with bilingual parallel corpora. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics, ACL '05*, pages 597–604, Stroudsburg, PA, USA, 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219914. URL <http://dx.doi.org/10.3115/1219840.1219914>.
- Bisazza, Arianna, Nick Ruiz, and Marcello Federico. Fill-up versus interpolation methods for phrase-based SMT adaptation. In *2011 International Workshop on Spoken Language Translation, IWSLT 2011, San Francisco, CA, USA, December 8-9, 2011*, pages 136–143, 2011. URL [http://www.isca-speech.org/archive/iwslt\\_11/sltb\\_136.html](http://www.isca-speech.org/archive/iwslt_11/sltb_136.html).
- Bojar, Ondřej, Zdeněk Žabokrtský, Ondřej Dušek, Petra Galuščáková, Martin Majliš, David Mareček, Jiří Maršík, Michal Novák, Martin Popel, and Aleš Tamchyna. The Joy of Parallelism with CzEng 1.0. In *Proceedings of LREC2012, Istanbul, Turkey, 2012*. ELRA, European Language Resources Association.
- Cohn, Trevor and Mirella Lapata. Machine Translation by Triangulation: Making Effective Use of Multi-Parallel Corpora. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic, 2007*. URL <http://aclweb.org/anthology-new/P/P07/P07-1092.pdf>.
- Heafield, Kenneth, Ivan Pouzyrevsky, Jonathan H. Clark, and Philipp Koehn. Scalable Modified Kneser-Ney Language Model Estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics, Sofia, Bulgaria, 8 2013*. URL [http://khefield.com/professional/edinburgh/estimate\\_paper.pdf](http://khefield.com/professional/edinburgh/estimate_paper.pdf).
- Koehn, Philipp, Franz Josef Och, and Daniel Marcu. Statistical Phrase-Based Translation. In *HLT-NAACL, 2003*. URL <http://acl.ldc.upenn.edu/N/N03/N03-1017.pdf>.
- Koehn, Philipp, 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. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic, 2007*. URL <http://aclweb.org/anthology-new/P/P07/P07-2045.pdf>.
- Nakov, Preslav. Improving English-Spanish Statistical Machine Translation: Experiments in Domain Adaptation, Sentence Paraphrasing, Tokenization, and Recasing. In *Proceedings of the Third Workshop on Statistical Machine Translation, StatMT '08*, pages 147–150, Stroudsburg, PA, USA, 2008. Association for Computational Linguistics.
- Razmara, Majid and Anoop Sarkar. Ensemble Triangulation for Statistical Machine Translation. In *Sixth International Joint Conference on Natural Language Processing, IJCNLP 2013, Nagoya, Japan, October 14-18, 2013*, pages 252–260, 2013. URL <http://aclweb.org/anthology/I/I13/I13-1029.pdf>.

- Sennrich, Rico. Perplexity Minimization for Translation Model Domain Adaptation in Statistical Machine Translation. In *EACL 2012, 13th Conference of the European Chapter of the Association for Computational Linguistics, Avignon, France, April 23-27, 2012*, pages 539–549, 2012. URL <http://aclweb.org/anthology-new/E/E12/E12-1055.pdf>.
- Utiyama, Masao and Hitoshi Isahara. A Comparison of Pivot Methods for Phrase-Based Statistical Machine Translation. In *Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics, Proceedings, April 22-27, 2007, Rochester, New York, USA*, pages 484–491, 2007. URL <http://www.aclweb.org/anthology/N07-1061>.
- Wu, Hua and Haifeng Wang. Pivot Language Approach for Phrase-Based Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*, 2007. URL <http://aclweb.org/anthology-new/P/P07/P07-1108.pdf>.
- Zhu, Xiaoning, Zhongjun He, Hua Wu, Conghui Zhu, Haifeng Wang, and Tiejun Zhao. Improving Pivot-Based Statistical Machine Translation by Pivoting the Co-occurrence Count of Phrase Pairs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL*, pages 1665–1675, 2014. URL <http://aclweb.org/anthology/D/D14/D14-1174.pdf>.

**Address for correspondence:**

Ondřej Bojar  
bojar@ufal.mff.cuni.cz  
Institute of Formal and Applied Linguistics  
Faculty of Mathematics and Physics,  
Charles University in Prague  
Malostranské náměstí 25  
118 00 Praha 1, Czech Republic