The CMU Machine Translation Systems at WMT 2014

Austin Matthews **Waleed Ammar** **Archna Bhatia** **Weston Feely**
Greg Hanneman **Eva Schlinger** **Swabha Swayamdipta** **Yulia Tsvetkov**
Alon Lavie **Chris Dyer***
Language Technologies Institute
Carnegie Mellon University
Pittsburgh, PA 15213 USA
*Corresponding author: cdyer@cs.cmu.edu

Abstract

We describe the CMU systems submitted to the 2014 WMT shared translation task. We participated in two language pairs, German–English and Hindi–English. Our innovations include: a label coarsening scheme for syntactic tree-to-tree translation, a host of new discriminative features, several modules to create "synthetic translation options" that can generalize beyond what is directly observed in the training data, and a method of combining the output of multiple word aligners to uncover extra phrase pairs and grammar rules.

1 Introduction

The MT research group at Carnegie Mellon University's Language Technologies Institute participated in two language pairs for the 2014 Workshop on Machine Translation shared translation task: German–English and Hindi–English. Our systems showcase our multi-phase approach to translation, in which **synthetic translation options** supplement the default translation rule inventory that is extracted from word-aligned training data.

In the German–English system, we used our compound splitter (Dyer, 2009) to reduce data sparsity, and we allowed the translator to back off to translating lemmas when it detected case-inflected OOVs. We also demonstrate our group's syntactic system with coarsened nonterminal types (Hanneman and Lavie, 2011) as a contrastive German–English submission.

In both the German–English and Hindi–English systems, we used an array of supplemental ideas to enhance translation quality, ranging from lemmatization and synthesis of inflected phrase pairs to novel reordering and rule preference features.

2 Core System Components

The decoder infrastructure we used was cdec (Dyer et al., 2010). For our primary systems, all data was tokenized using cdec's tokenization tool. Only the constrained data resources provided for the shared task were used for training both the translation and language models. Word alignments were generated using both FastAlign (Dyer et al., 2013) and GIZA++ (Och and Ney, 2003). All our language models were estimated using KenLM (Heafield, 2011). Translation model parameters were chosen using MIRA (Eidelman, 2012) to optimize BLEU (Papineni et al., 2002) on a held-out development set.

Our data was filtered using qe-clean (Denkowski et al., 2012), with a cutoff of two standard deviations from the mean. All data was left in fully cased form, save the first letter of each segment, which was changed to whichever form the first token more commonly used throughout the data. As such, words like *The* were lowercased at the beginning of segments, while words like *Obama* remained capitalized.

Our primary German–English and Hindi–English systems were Hiero-based (Chiang, 2007), while our contrastive German–English system used cdec's tree-to-tree SCFG formalism.

Before submitting, we ran cdec's implementation of MBR on 500-best lists from each of our systems. For both language pairs, we used the Nelder–Mead method to optimize the MBR parameters. In the German–English system, we ran MBR on 500 hypotheses, combining the output of the Hiero and tree-to-tree systems.

The remainder of the paper will focus on our primary innovations in the two language pairs.

Proceedings of the Ninth Workshop on Statistical Machine Translation, pages 142–149,
Baltimore, Maryland USA, June 26–27, 2014.

3 Common System Improvements

A number of our techniques were used for both our German–English and Hindi–English primary submissions. These techniques each fall into one of three categories: those that create translation rules, those involving language models, or those that add translation features. A comparison of these techniques and their performance across the two language pairs can be found in Section 6.

3.1 Rule-Centric Enhancements

While many of our methods of enhancing the translation model with extra rules are language-specific, three were shared between language pairs.

First, we added sentence-boundary tokens <*s*> and </*s*> to the beginning and end of each line in the data, on both the source and target sides.

Second, we aligned all of our training data using both FastAlign and GIZA++ and simply concatenated two copies of the training corpus, one aligned with each aligner, and extracted rules from the resulting double corpus.

Third, we hand-wrote a list of rules that transform numbers, dates, times, and currencies into well-formed English equivalents, handling differences such as the month and day reversal in dates or conversion from 24-hour time to 12-hour time.

3.2 Employed Language Models

Each of our primary systems uses a total of three language models.

The first is a traditional 4-gram model generated by interoplating LMs built from each of the available monolingual corpora. Interpolation weights were calculated used the SRILM toolkit (Stolcke, 2002) and 1000 dev sentences from the Hindi–English system.

The second is a model trained on word clusters instead of surface forms. For this we mapped the LM vocabulary into 600 clusters based on the algorithm of Brown et al. (1992) and then constructed a 7-gram LM over the resulting clusters, allowing us to capture more context than our traditional surface-form language model.

The third is a bigram model over the *source* side of each language's respective bitext. However, at run time this LM operates on the target-side output of the translator, just like the other two. The intuition here is that if a source-side LM likes our output, then we are probably passing through more than we ought to.

Both source and target surface-form LM used modified Kneser-Ney smoothing (Kneser and Ney, 1995), while the model over Brown clusters (Brown et al., 1992) used subtract-0.5 smoothing.

3.3 New Translation Features

In addition to the standard array of features, we added four new indicator feature templates, leading to a total of nearly 150,000 total features.

The first set consists of target-side n-gram features. For each bigram of Brown clusters in the output string generated by our translator, we fire an indicator feature. For example, if we have the sentence, *Nato will ihren Einfluss im Osten stärken* translating as *NATO intends to strengthen its influence in the East*, we will fire an indicator features NGF_C367_C128=1, NGF_C128_C31=1, etc.

The second set is source-language n-gram features. Similar to the previous feature set, we fire an indicator feature for each $n$gram of Brown clusters in the output. Here, however, we use $n = 1$, and we use the map of *source* language words to Brown clusters, rather than the target language's, despite the fact that this is examining target language output. The intuition here is to allow this feature to penalize passthroughs differently depending on their source language Brown cluster. For example, passing through the German word *zeitung* ("newspaper") is probably a bad idea, but passing through the German word *Obama* probably should not be punished as severely.

The third type of feature is source path features. We can imagine translation as a two-step process in which we first permute the source words into some order, then translate them phrase by phrase. This set of features examines that intermediate string in which the source words have been permuted. Again, we fire an indicator feature for each bigram in this intermediate string, this time using surface lexical forms directly instead of first mapping them to Brown clusters.

Lastly, we create a new type of rule shape feature. Traditionally, rule shape features have indicated, for each rule, the sequence of terminal and non-terminal items on the right-hand side. For example, the rule [X] → der [X] :: the [X] might have an indicator feature Shape_TN_TN, where T represents a terminal and N represents a non-terminal. One can also imagine lexicalizing such rules by replacing each T with its surface form. We believe such features would be too sparse, so instead of replacing each terminal by its surface form, we instead replace it with its Brown cluster,

creating a feature like Shape_C37_N_C271_N.

4 Hindi–English Specific Improvements

In addition to the enhancements common to the two primary systems, our Hindi–English system includes improved data cleaning of development data, a sophisticated linguistically-informed tokenization scheme, a transliteration module, a synthetic phrase generator that improves handling of function words, and a synthetic phrase generator that leverages source-side paraphrases. We will discuss each of these five in turn.

4.1 Development Data Cleaning

Due to a scarcity of clean development data, we augmented the 520 segments provided with 480 segments randomly drawn from the training data to form our development set, and drew another random 1000 segments to serve as a dev test set.

After observing large discrepencies between the types of segments in our development data and the well-formed news domain sentences we expected to be tested on, we made the decision to prune our tuning set by removing any segment that did not appear to be a full sentence on both the Hindi and English sides. While this reduced our tuning set from 1000 segments back down to 572 segments, we believe it to be the single largest contributor to our success on the Hindi–English translation task.

4.2 Nominal Normalization

Another facet of our system was normalization of Hindi nominals. The Hindi nominal system shows much more morphological variation than English. There are two genders (masculine and feminine) and at least six noun stem endings in pronunciation and 10 in writing.

The pronominal system also is much richer than English with many variants depending on whether pronouns appear with case markers or other postpositions.

Before normalizing the nouns and pronouns, we first split these case markers / postpositions from the nouns / pronouns to result in two words instead of the original combined form. If the case marker was ने (*ne*), the ergative case marker in Hindi, we deleted it as it did not have any translation in English. All the other postpositions were left intact while splitting from and normalizing the nouns and pronouns.

These changes in stem forms contribute to the sparsity in data; hence, to reduce this sparsity, we construct for each input segment an input lattice that allows the decoder to use the split or original forms of all nouns or pronouns, as well as allowing it to keep or delete the case marker *ne*.

4.3 Transliteration

We used the 12,000 Hindi–English transliteration pairs from the ACL 2012 NEWS workshop on transliteration to train a linear-chained CRF tagger[1] that labels each character in the Hindi token with a sequence of zero or more English characters (Ammar et al., 2012). At decoding, unseen Hindi tokens are fed to the transliterator, which produces the 100 most probable transliterations. We add a synthetic translation option for each candidate transliteration.

In addition to this sophisticated transliteration scheme, we also employ a rule-based transliterator that specifically targets acronyms. In Hindi, many acronyms are spelled out phonetically, such as *NSA* being rendered as एनएसए (*en.es.e*). We detected such words in the input segments and generated synthetic translation options both with and without periods (e.g. *N.S.A.* and *NSA*).

4.4 Synthetic Handling of Function Words

In different language pairs, individual source words may have many different possible translations, e.g., when the target language word has many different morphological inflections or is surrounded by different function words that have no direct counterpart in the source language. Therefore, when very large quantities of parallel data are not available, we can expect our phrasal inventory to be incomplete. Synthetic translation option generation seeks to fill these gaps using secondary generation processes that exploit existing phrase pairs to produce plausible phrase translation alternatives that are not directly extractable from the training data (Tsvetkov et al., 2013; Chahuneau et al., 2013).

To generate synthetic phrases, we first remove function words from the source and target sides of existing non-gappy phrase pairs. We manually constructed English and Hindi lists of common function words, including articles, auxiliaries, pronouns, and adpositions. We then employ the SRILM hidden-ngram utility (Stolcke, 2002) to restore missing function words according to an n-gram language model probability, and add the resulting synthetic phrases to our phrase table.

[1] https://github.com/wammar/transliterator

4.5 Paraphrase-Based Synthetic Phrases

We used a graph-based method to obtain translation distributions for source phrases that are not present in the phrase table extracted from the parallel corpus. Monolingual data is used to construct separate similarity graphs over phrases (word sequences or n-grams), using distributional features extracted from the corpora. The source similarity graph consists of phrase nodes representing sequences of words in the source language. In our instance, we restricted the phrases to bigrams, and the bigrams come from both the phrase table (the *labeled* phrases) and from the evaluation set but not present in the phrase table (unlabeled phrases).

The labels for these source phrases, namely the target phrasal inventory, can also be represented in a graph form, where the distributional features can also be computed from the target monolingual data. Translation information is then propagated from the labeled phrases to the unlabeled phrases in the source graph, proportional to how similar the phrases are to each other on the source side, as well as how similar the translation candidates are to each other on the target side. The newly acquired translation distributions for the unlabeled phrases are written out to a secondary phrase table. For more information, see Saluja et al. (2014).

5 German–English Specific Improvements

Our German–English system also had its own suite of tricks, including the use of "pseudo-references" and special handling of morphologically inflected OOVs.

5.1 Pseudo-References

The development sets provided have only a single reference, which is known to be sub-optimal for tuning of discriminative models. As such, we use the output of one or more of last year's top performing systems as pseudo-references during tuning. We experimented with using just one pseudo-reference, taken from last year's Spanish–English winner (Durrani et al., 2013), and with using four pseudo-references, including the output of last year's winning Czech–English, French–English, and Russian–English systems (Pino et al., 2013).

5.2 Morphological OOVs

Examination of the output of our baseline systems lead us to conclude that the majority of our system's OOVs were due to morphologically inflected nouns in the input data, particularly those in genitive case. As such, for each OOV in the input, we attempt to remove the German genitive case marker *-s* or *-es*. We then run the resulting form f through our baseline translator to obtain a translation e of the lemma. Finally, we add two translation rules to our translation table: $f \rightarrow e$, and $f \rightarrow e$'s.

6 Results

As we added each feature to our systems, we first ran a one-off experiment comparing our baseline system with and without each individual feature. The results of that set of experiments are shown in Table 1 for Hindi–English and Table 2 for German–English. Features marked with a * were not included in our final system submission.

The most surprising result is the strength of our Hindi–English baseline system. With no extra bells or whistles, it is already half a BLEU point ahead of the second best system submitted to this shared task. We believe this is due to our filtering of the tuning set, which allowed our system to generate translations more similar in length to the final test set.

Another interesting result is that only one feature set, namely our rule shape features based on Brown clusters, helped on the test set in both language pairs. No feature hurt the BLEU score on the test set in both language pairs, meaning the majority of features helped in one language and hurt in the other.

If we compare results on the tuning sets, however, some clearer patterns arise. Brown cluster language models, n-gram features, and our new rule shape features all helped.

Furthermore, there were a few features, such as the Brown cluster language model and tuning to Meteor (Denkowski and Lavie, 2011), that helped substantially in one language pair while just barely hurting the other. In particular, the fact that tuning to Meteor instead of BLEU can actually help both BLEU and Meteor scores was rather unexpected.

7 German–English Syntax System

In addition to our primary German–English system, we also submitted a contrastive German–English system showcasing our group's tree-to-tree syntax-based translation formalism.

System	Test (2014) BLEU	Met	TER	Dev Test (2012) BLEU	Met	TER
Baseline	15.7	25.3	68.0	11.4	22.9	70.3
*Meteor Tuning	15.2	25.8	71.3	12.8	23.7	71.3
Sentence Boundaries	15.2	25.4	69.1	12.1	23.4	70.0
Double Aligners	16.1	25.5	66.6	11.9	23.1	69.2
Manual Number Rules	15.7	25.4	68.5	11.6	23.0	70.3
Brown Cluster LM	15.6	25.1	67.3	11.5	22.7	69.8
*Source LM	14.2	25.1	72.1	11.3	23.0	72.3
N-Gram Features	15.6	25.2	67.9	12.2	23.2	69.2
Src N-Gram Features	15.3	25.2	68.9	12.0	23.4	69.5
Src Path Features	15.8	25.6	68.8	11.9	23.3	70.4
Brown Rule Shape	15.9	25.4	67.2	11.8	22.9	69.6
Lattice Input	15.2	25.8	71.3	11.4	22.9	70.3
CRF Transliterator	15.7	25.7	69.4	12.1	23.5	70.1
Acronym Translit.	15.8	25.8	68.8	12.4	23.4	70.2
Synth. Func. Words	15.7	25.3	67.8	11.4	22.8	70.4
Source Paraphrases	15.6	25.2	67.7	11.5	22.7	69.9
Final Submission	16.7					

Table 1: BLEU, Meteor, and TER results for one-off experiments conducted on the primary Hiero Hindi–English system. Each line is the baseline plus that one feature, non-cumulatively. Lines marked with a * were not included in our final WMT submission.

System	Test (2014) BLEU	Met	TER	Dev Test (2012) BLEU	Met	TER
Baseline	25.3	30.4	52.6	26.2	31.3	53.6
*Meteor Tuning	26.2	31.3	53.1	26.9	32.2	54.4
Sentence Boundaries	25.4	30.5	52.2	26.1	31.4	53.3
Double Aligners	25.2	30.4	52.5	26.0	31.3	53.6
Manual Number Rules	25.3	30.3	52.5	26.1	31.4	53.4
Brown Cluster LM	26.4	31.0	51.9	27.0	31.8	53.2
*Source LM	25.8	30.6	52.4	26.4	31.5	53.4
N-Gram Features	25.4	30.4	52.6	26.7	31.6	53.0
Src N-Gram Features	25.3	30.5	52.5	26.2	31.5	53.4
Src Path Features	25.0	30.1	52.6	26.0	31.2	53.3
Brown Rule Shape	25.5	30.5	52.4	26.3	31.5	53.2
One Pseudo Ref	25.5	30.4	52.6	34.4	32.7	49.3
*Four Psuedo Refs	22.6	29.2	52.6	49.8	35.0	46.1
OOV Morphology	25.5	30.5	52.4	26.3	31.5	53.3
Final Submission	27.1					

Table 2: BLEU, Meteor, and TER results for one-off experiments conducted on the primary Hiero German–English system. Each line is the baseline plus that one feature, non-cumulatively.

System	Dev (2013) BLEU	Met	TER	Dev Test (2012) BLEU	Met	TER
Baseline	20.98	29.81	58.47	18.65	28.72	61.80
+ Label coarsening	23.07	30.71	56.46	20.43	29.34	60.16
+ Meteor tuning	23.48	30.90	56.18	20.96	29.60	59.87
+ Brown LM + Lattice + Synthetic	24.46	31.41	56.66	21.50	30.28	60.51
+ Span limit 15	24.20	31.25	55.48	21.75	29.97	59.18
+ Pseudo-references	24.55	31.30	56.22	22.10	30.12	59.73

Table 3: BLEU, Meteor, and TER results for experiments conducted in the tree-to-tree German–English system. The system in the bottom line was submitted to WMT as a contrastive entry.

7.1 Basic System Construction

Since all training data for the tree-to-tree system must be parsed in addition to being word-aligned, we prepared separate copies of the training, tuning, and testing data that are more suitable for input into constituency parsing. Importantly, we left the data in its original mixed-case format. We used the Stanford tokenizer to replicate Penn Treebank tokenization on the English side. On the German side, we developed new in-house normalization and tokenization script.

We filtered tokenized training sentences by sen-

tence length, token length, and sentence length ratio. The final corpus for parsing and word alignment contained 3,897,805 lines, or approximately 86 percent of the total training resources released under the WMT constrained track. Word alignment was carried out using FastAlign (Dyer et al., 2013), while for parsing we used the Berkeley parser (Petrov et al., 2006).

Given the parsed and aligned corpus, we extracted synchronous context-free grammar rules using the method of Hanneman et al. (2011).

In addition to aligning subtrees that natively exist in the input trees, our grammar extractor also introduces "virtual nodes." These are new and possibly overlapping constituents that subdivide regions of flat structure by combining two adjacent sibling nodes into a single nonterminal for the purposes of rule extraction. Virtual nodes are similar in spirit to the "A+B" extended categories of SAMT (Zollmann and Venugopal, 2006), and their nonterminal labels are constructed in the same way, but with the added restriction that they do not violate any existing syntactic structure in the parse tree.

7.2 Improvements

Nonterminals in our tree-to-tree grammar are made up of pairs of symbols: one from the source side and one from the target side. With virtual nodes included, this led to an initial German–English grammar containing 153,219 distinct nonterminals — a far larger set than is used in SAMT, tree-to-string, string-to-tree, or Hiero systems. To combat the sparsity introduce by this large nonterminal set, we coarsened the label set with an agglomerative label-clustering technique(Hanneman and Lavie, 2011; Hanneman and Lavie, 2013). The stopping point was somewhat arbitrarily chosen to be a grammar of 916 labels.

Table 3 shows a significant improvement in translation quality due to coarsening the label set: approximately +1.8 BLEU, +0.6 Meteor, and –1.6 TER on our dev test set, newtest2012.[2]

In the MERT runs, however, we noticed that the length of the MT output can be highly variable, ranging on the tuning set from a low of 92.8% of the reference length to a high of 99.1% in another. We were able to limit this instability by tuning to Meteor instead of BLEU. Aside from a modest score improvement, we note that the variability in length ratio is reduced from 6.3% to 2.8%.

Specific difficulties of the German–English language pair led to three additional system components to try to combat them.

First, we introduced a second language model trained on Brown clusters instead of surface forms.

Next we attempted to overcome the sparsity of German input by making use of cdec's lattice input functionality introduce compound-split versions of dev and test sentences.

Finally, we attempted to improve our grammar's coverage of new German words by introducing synthetic rules for otherwise out-of-vocabulary items. Each token in a test sentence that the grammar cannot translate generates a synthetic rule allowing the token to be translated as itself. The left-hand-side label is decided heuristically: a (coarsened) "noun" label if the German OOV starts with a capital letter, a "number" label if the OOV contains only digits and select punctuation characters, an "adjective" label if the OOV otherwise starts with a lowercase letter or a number, or a "symbol" label for anything left over.

The effect of all three of these improvements combined is shown in the fourth row of Table 3.

By default our previous experiments were performed with a span limit of 12 tokens. Increasing this limit to 15 has a mixed effect on metric scores, as shown in the fifth row of Table 3. Since two out of three metrics report improvement, we left the longer span limit in effect in our final system.

Our final improvement was to augment our tuning set with the same set of pseudo-references as our Hiero systems. We found that using one pseudo-reference versus four pseudo-references had negligible effect on the (single-reference) tuning scores, but four produced a better improvement on the test set.

The best MERT run of this final system (bottom line of Table 3) was submitted to the WMT 2014 evaluation as a contrastive entry.

Acknowledgments

We sincerely thank the organizers of the workshop for their hard work, year after year, and the reviewers for their careful reading of the submitted draft of this paper. This research work was supported in part by the U. S. Army Research Laboratory and the U. S. Army Research Office under contract/grant number W911NF-10-1-0533, by the National Science Foundation under grant

[2]We follow the advice of Clark et al. (2011) and evaluate our tree-to-tree experiments over multiple independent MERT runs. All scores in Table 3 are averages of two or three runs, depending on the row.

IIS-0915327, by a NPRP grant (NPRP 09-1140-1-177) from the Qatar National Research Fund (a member of the Qatar Foundation), and by computing resources provided by the NSF-sponsored XSEDE program under grant TG-CCR110017. The statements made herein are solely the responsibility of the authors.

References

Waleed Ammar, Chris Dyer, and Noah A. Smith. 2012. Transliteration by sequence labeling with lattice encodings and reranking. In *NEWS workshop at ACL*.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Victor Chahuneau, Eva Schlinger, Noah A. Smith, and Chris Dyer. 2013. Translating into morphologically rich languages with synthetic phrases. In *Proceedings of EMNLP*.

David Chiang. 2007. Hierarchical phrase-based translation. *Computational Linguistics*, 33(2):201–228.

Jonathan H. Clark, Chris Dyer, Alon Lavie, and Noah A. Smith. 2011. Better hypothesis testing for statistical machine translation: Crontrolling for optimizer instability. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Short Papers*, pages 176–181, Portland, Oregon, USA, June.

Michael Denkowski and Alon Lavie. 2011. Meteor 1.3: Automatic metric for reliable optimization and evaluation of machine translation systems. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 85–91, Edinburgh, Scotland, UK, July.

Michael Denkowski, Greg Hanneman, and Alon Lavie. 2012. The cmu-avenue french-english translation system. In *Proceedings of the NAACL 2012 Workshop on Statistical Machine Translation*.

Nadir Durrani, Barry Haddow, Kenneth Heafield, and Philipp Koehn. 2013. Edinburgh's machine translation systems for european language pairs.

Chris Dyer, Adam Lopez, Juri Ganitkevitch, Johnathan Weese, Ferhan Ture, Phil Blunsom, Hendra Setiawan, Vladimir Eidelman, and Philip Resnik. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proc. of ACL*.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM Model 2. In *Proc. of NAACL*.

Chris Dyer. 2009. Using a maximum entropy model to build segmentation lattices for mt. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 406–414. Association for Computational Linguistics.

Vladimir Eidelman. 2012. Optimization strategies for online large-margin learning in machine translation. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*.

Greg Hanneman and Alon Lavie. 2011. Automatic category label coarsening for syntax-based machine translation. In *Proceedings of SSST-5: Fifth Workshop on Syntax, Semantics, and Structure in Statistical Translation*, pages 98–106, Portland, Oregon, USA, June.

Greg Hanneman and Alon Lavie. 2013. Improving syntax-augmented machine translation by coarsening the label set. In *Proceedings of NAACL-HLT 2013*, pages 288–297, Atlanta, Georgia, USA, June.

Greg Hanneman, Michelle Burroughs, and Alon Lavie. 2011. A general-purpose rule extractor for SCFG-based machine translation. In *Proceedings of SSST-5: Fifth Workshop on Syntax, Semantics, and Structure in Statistical Translation*, pages 135–144, Portland, Oregon, USA, June.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, Scotland, UK, July.

R. Kneser and H. Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of IEEE Internation Conference on Acoustics, Speech, and Signal Processing*, pages 181–184.

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

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: A method for automatic evalution of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA, July.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440. Association for Computational Linguistics.

Juan Pino, Aurelien Waite, Tong Xiao, Adrià de Gispert, Federico Flego, and William Byrne. 2013. The university of cambridge russian-english system at wmt13.

Avneesh Saluja, Hany Hassan, Kristina Toutanova, and Chris Quirk. 2014. Graph-based semi-supervised learning of translation models from monolingual data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*, Baltimore, Maryland, June.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *INTERSPEECH*.

Yulia Tsvetkov, Chris Dyer, Lori Levin, and Archna Batia. 2013. Generating English determiners in phrase-based translation with synthetic translation options. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*.

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation*, pages 138–141, New York, New York, USA, June.