Edinburgh's Syntax-Based Systems at WMT 2015

Philip Williams[1], Rico Sennrich[1], Maria Nadejde[1],
Matthias Huck[1], Philipp Koehn[1,2]
[1]School of Informatics, University of Edinburgh
[2]Center for Speech and Language Processing, The Johns Hopkins University

Abstract

This paper describes the syntax-based systems built at the University of Edinburgh for the WMT 2015 shared translation task. We developed systems for all language pairs except French-English. This year we focused on: translation out of English using tree-to-string models; continuing to improve our English-German system; and source-side morphological segmentation of Finnish using Morfessor.

1 Introduction

This year's WMT shared translation task featured five language pairs: English paired with Czech, Finnish, French, German, and Russian. We built syntax-based systems in both translation directions for all language pairs except English-French.

For English → German, we continued to develop our string-to-tree system, which has proven highly competitive in previous years. Additions this year included the use of a dependency language model, an alternative tuning metric, and soft source-syntactic constraints.

For translation from English into Czech, Finnish, and Russian, we built STSG-based tree-to-string systems. Support for this type of model is a recent addition to the Moses toolkit. In previous years, our systems have all used string-to-tree models and have only translated into English and German.

For Finnish → English, we experimented with unsupervised morphological segmentation using Morfessor 2.0 (Virpioja et al., 2013).

For the remaining systems (Czech → English, German → English, and Russian → English), our systems were essentially the same as last year's (Williams et al., 2014) except for the addition of this year's training data.

2 System Overview

2.1 Pre-processing

The training data was pre-processed using scripts from the Moses toolkit. We first normalized the data using the `normalize-punctuation.perl` script then performed tokenization, parsing, and truecasing. To parse the English data, we used the Berkeley parser (Petrov et al., 2006; Petrov and Klein, 2007). To parse the German data, we used the ParZu dependency parser (Sennrich et al., 2013).

2.2 Word Alignment

For word alignment we used either MGIZA++ (Gao and Vogel, 2008), a multi-threaded implementation of GIZA++ (Och and Ney, 2003), or `fast_align` (Dyer et al., 2013). In preliminary experiments, we found that the tree-to-string systems were particularly sensitive to the choice of word aligner, echoing a previous observation by Neubig and Duh (2014). See the individual tree-to-string system descriptions in Section 3.

2.3 Language Model

We used all available monolingual data to train one interpolated 5-gram language model for each system. Using either lmplz (Heafield et al., 2013) or the SRILM toolkit (Stolcke, 2002), we first trained an individual language model for each of the supplied monolingual training corpora. These models all used modified Kneser-Ney smoothing (Chen and Goodman, 1998). We then interpolated the individual models using SRILM, providing the target-side of the system's tuning set (Section 2.7) for perplexity-based weight optimization.

2.4 String-to-Tree Model

For English → German and the systems that translate into English, we used a string-to-tree model.

Proceedings of the Tenth Workshop on Statistical Machine Translation, pages 199–209,
Lisboa, Portugal, 17-18 September 2015.

2.4.1 Grammar

The string-to-tree translation model is based on a synchronous context-free grammar (SCFG) with linguistically-motivated labels on the target side.

SCFG rules were extracted from the word-aligned parallel data using the Moses implementation (Williams and Koehn, 2012) of the GHKM algorithm (Galley et al., 2004; Galley et al., 2006).

Minimal GHKM rules were composed into larger rules subject to restrictions on the size of the resulting tree fragment. We used the settings shown in Table 1, which were chosen empirically during the development of 2013's systems (Nadejde et al., 2013).

Parameter	Unbinarized	Binarized
Rule depth	5	7
Node count	20	30
Rule size	5	7

Table 1: Parameter settings for rule composition. The parameters were relaxed for systems that used binarization to allow for the increase in tree node density.

Further to the restrictions on rule composition, fully non-lexical unary rules were eliminated using the method described in Chung et al. (2011) and rules with scope greater than 3 (Hopkins and Langmead, 2010) were pruned from the translation grammar. Scope pruning makes parsing tractable without the need for grammar binarization.

2.4.2 Feature Functions

Our core set of string-to-tree feature functions is unchanged from previous years. It includes the n-gram language model's log probability for the target string, the target word count, the rule count, and various pre-computed rule-specific scores. For a grammar rule r of the form

$$C \rightarrow \langle \alpha, \beta, \sim \rangle$$

where C is a target-side non-terminal label, α is a string of source terminals and non-terminals, β is a string of target terminals and non-terminals, and $\sim$ is a one-to-one correspondence between source and target non-terminals, we score the rule according to (logarithms of) the following functions:

- $p\left(C, \beta \mid \alpha, \sim\right)$ and $p\left(\alpha \mid C, \beta, \sim\right)$, the direct and indirect translation probabilities.
- $p_{lex}\left(\beta \mid \alpha\right)$ and $p_{lex}\left(\alpha \mid \beta\right)$, the direct and indirect lexical weights (Koehn et al., 2003).
- $p_{pcfg}\left(\pi\right)$, the monolingual PCFG probability of the tree fragment π from which the rule was extracted.
- $\exp(-1/count(r))$, a rule rareness penalty.

2.5 Tree-to-String Model

For English → Czech, English → Finnish, and English → Russian, we used a tree-to-string model.

2.5.1 Grammar

In the tree-to-string model, the translation grammar is a synchronous tree-substitution grammar (Eisner, 2003) with parse tree fragments on the source-side and strings of terminals and non-terminals on the target-side.

As with the string-to-tree models, the grammar was extracted from the word-aligned parallel data using the Moses implementation of the GHKM algorithm. Minimal GHKM rules were composed into larger rules subject to the same size restrictions (Table 1). Unlike string-to-tree rule extraction, fully non-lexical unary rules were included in the grammar and scope pruning was not used.

2.5.2 Feature Functions

The tree-to-string feature functions are similar to those of the string-to-tree model. For a grammar rule r of the form

$$\langle \pi, \beta, \sim \rangle$$

where π is a source-side tree fragment, β is a string of target terminals and non-terminals, and $\sim$ is a one-to-one correspondence between source and target non-terminals, we score the rule according to (logarithms of) the following functions:

- $p\left(\beta \mid \pi, \sim\right)$ and $p\left(\pi \mid \beta, \sim\right)$, the direct and indirect translation probabilities.
- $p_{lex}\left(\beta \mid \pi\right)$ and $p_{lex}\left(\pi \mid \beta\right)$, the direct and indirect lexical weights (Koehn et al., 2003).
- $\exp(-1/count(r))$, a rule rareness penalty.

2.6 Decoding

Decoding for the string-to-tree models is based on Sennrich's (2014) recursive variant of the CYK+ parsing algorithm combined with LM integration via cube pruning (Chiang, 2007). Decoding for the tree-to-string models is based on the rule matching algorithm by Zhang et al. (2009) combined with LM integration via cube pruning.

2.7 Tuning

The feature weights were tuned using the Moses implementation of MERT (Och, 2003) for all systems except English-to-German, for which we used k-best MIRA (Cherry and Foster, 2012) due to the use of sparse features.

For the tree-to-string systems, we used all of the previous years' test sets as tuning data (except newstest2014, which was used as the development test set). For the string-to-tree systems, we used subsets of the test data to speed up decoding.

3 Individual Systems

In this section we describe individual systems and present experimental results. In many cases, the only difference from the generic setup of the previous section is that we perform right binarization of the training and test parse trees.

We also built hierarchical phrase-based systems (Chiang, 2007), which we refer to in tables as 'Hiero.' These systems were built using the Moses toolkit, with standard settings. They were not used in the submission and are included for comparison only.

For each system, we present results for both the development test set (newstest2014 in most cases) and for the test set (newstest2015) for which reference translations were provided after the system submission deadline. We refer to these as 'devtest' and 'test', respectively.

3.1 English to Czech

For English → Czech we built a tree-to-string system. We used `fast_align` for word alignment due to the large training data size and on the strength of its performance for English → Finnish and English → Russian. We used all test sets from 2008 to 2013 as tuning data. Table 2 gives the mean BLEU scores, averaged over three MERT runs. Our submitted system was the right binarized system that, out of the three runs, scored highest on devtest.

system	devtest	test
Hiero	20.2	16.8
Tree-to-string	19.0	15.7
+ right binarization	19.5	16.1

Table 2: English to Czech translation results (BLEU) on devtest (newstest2014) and test (newstest2015) sets.

3.2 English to Finnish

In preliminary English → Finnish experiments, we compared the use of MGIZA++ and `fast_align`. Since there was only one test set provided, in these initial experiments we split newsdev2015 into two halves, using the first half for tuning and the second half for testing. Table 3 gives the mean BLEU scores, averaged over three MERT runs.

	MGIZA++	`fast_align`
Hiero	11.7	11.6
Tree-to-string	11.5	12.3
+ right binarization	11.9	12.8

Table 3: Comparison of word alignment tools for English to Finnish. BLEU on subset of newsdev2015.

For our final system, we used `fast_align` for word alignment and we used the full newsdev2015 test set as tuning data. Table 4 gives the mean BLEU scores for this setup. Our submitted system was the right binarized system that, out of the three MERT runs, scored highest on devtest.

system	dev	test
Hiero	11.4	11.5
Tree-to-string	11.9	11.8
+ right binarization	12.2	12.3

Table 4: Final English to Finnish translation results (BLEU) on dev (newsdev2015) and test (newstest2015) sets.

3.3 English to German

We experiment with the following additions to last year's submission system: a relational dependency language model (RDLM) (Sennrich, 2015); tuning on the syntactic metric HWCM (Liu and Gildea, 2005; Sennrich, 2015); soft source-syntactic constraints (Huck et al., 2014); a large-scale n-gram Neural Network language model (NPLM) (Vaswani et al., 2013); treebank binarization (Sennrich and Haddow, 2015); particle verb restructuring (Sennrich and Haddow, 2015). We do not include syntactic constraints in this year's baseline. Our string-to-tree baseline uses a dependency representation of compounds, as described in (Sennrich and Haddow, 2015).

RDLM is a relational dependency language model which predicts the dependency relations

system	BLEU	2+ SUBJ
original trees	20.1	0
+ RDLM	21.0	0
+ RDLM (bidir.)	21.2	0
right binarization	20.4	272
head binarization	20.5	152
+ RDLM	21.3	43
+ RDLM (bidir.)	21.5	32

Table 5: English to German translation results (on newstest2013) with different binarizations and language models. *2+ SUBJ*: number of finite clauses with more than one subject.

and words in the translation hypotheses based on the dependency relations and words of the ancestor and sibling nodes in the dependency tree. Our model contains several extensions over the original paper (Sennrich, 2015). Like the original paper, we use an ancestor context size of 2, but we increase the sibling context size from 1 to 3, and allow bidirectional context, using the 3 closest siblings to both the left and right of the current node. The original model predicts a virtual stop node as the last child of each tree, which models the probability that a node has no more children. This is mirrored by a virtual start node in the bidirectional model.

We binarize the treebanks before rule extraction. We note that treebank binarization allows the extraction of rules that overgeneralize, e.g. allowing structures with zero, or multiple, preterminals per node, effectively allowing verb clauses without verb and similar. We use *head binarization* (Sennrich and Haddow, 2015), which ensures that each constituent contains exactly one head. During decoding, the generated target trees are unbinarized to allow scoring with RDLM. Table 5 shows that both right binarization and head binarization overgeneralize, exemplified by the fact that they allow finite clauses to have multiple subjects[1]. The RDLM reduces this problem, and the bidirectional RDLM slightly outperforms the unidirectional variant, both in terms of BLEU and the number of overgeneralizations.

For the soft source-syntactic constraints, we annotate the source text with the Stanford Neural Network dependency parser (Chen and Manning, 2014), along with heuristic projectivization (Nivre and Nilsson, 2005).

[1]Compound subjects are represented as a single node.

system	devtest	test
Hiero	19.2	21.0
String-to-tree baseline	19.8	21.4
+ $\frac{\text{HWCM}+\text{BLEU}}{2}$ tuning	20.1	21.6
+ head binarization	20.5	22.3
+ RDLM (bidirectional)	21.5	23.3
+ source-syntactic constraints	21.6	23.8
+ 5-gram NPLM	22.0	24.1
+ less pruning (**submission**)	22.0	24.0
+ particle verb restructuring	22.0	24.4

Table 6: English to German translation results (BLEU) on devtest (newstest2013) and test (newstest2015) sets.

The NPLM is a 5-gram feed-forward neural language model, and for both RDLM and NPLM we use a single hidden layer of size 750, a 150-dimensional input embedding layer with a vocabulary size of 500000, noise-contrastive estimation with 100 noise samples, and 2 iterations over the monolingual training set. Estimating LM probabilities for OOV words is a well-known problem, and we avoid this by filtering the translation model according to the vocabulary of the neural models.

The impact of all experimental components is shown in Table 6. Each system in Tables 5 and 6 was tuned separately with MIRA. For our submission system, we increased the Moses parameters *cube-pruning-pop-limit* from 1000 to 4000, and *rule-limit* from 100 to 400, but this had little effect on devtest, and gave even slightly lower BLEU on test. Particle verb restructuring, which was done after the submission deadline, increases BLEU on test. In total, we observe substantial improvements over our baseline, which roughly corresponds to last year's submission systems: 2.2 BLEU on devtest, and 3.0 BLEU on test.

3.4 English to Russian

For English → Russian we built a tree-to-string system. During preliminary experiments we found that `fast_align` gave consistent gains over MGIZA++ (albeit smaller than Finnish → English at around 0.3 BLEU). In final experiments we used `fast_align` for word alignment and we used the 2012 and 2013 test sets as tuning data. Table 7 gives the mean BLEU scores, averaged over three MERT runs. Our submitted system was the right binarized system that, out of the three runs, scored highest on devtest.

system	devtest	test
Hiero	29.8	23.8
Tree-to-string	27.5	22.1
+ right binarization	28.3	23.0

Table 7: English to Russian translation results (BLEU) on devtest (newstest2014) and test (newstest2015) sets.

3.5 Czech to English

For Czech → English we built a string-to-tree system. We used all test sets from 2008 to 2013 as tuning data. Table 8 gives the mean BLEU scores, which are averaged over three MERT runs. Our submitted system was the right binarized system that, out of the three runs, scored highest on devtest.

system	devtest	test
Hiero	28.5	24.9
String-to-tree	27.8	24.4
+ right binarization	27.8	24.5

Table 8: Czech to English translation results (BLEU) on devtest (newstest2014) and test (newstest2015) sets.

3.6 Finnish to English

In preliminary Finnish → English experiments, we tried using Morfessor to segment Finnish words into morphemes. We used Morfessor 2.0 (with default settings) to learn an unsupervised segmentation model from all of the available Finnish data, which was then used to segment all words in the source-side training and test data. We compared systems with and without segmentation and using a system combination of the two — an approach that has been shown to improve translation quality for this language pair (de Gispert et al., 2009).

As with English → Finnish, we split newsdev2015 into two halves, using the first half for tuning and the second half for testing. Table 9 shows the results: the column headed 'word' gives BLEU scores for the unsegmented systems; the column headed 'morph' gives scores for systems trained on segmented data; and the column headed 'syscomb' gives results for a system combination using MEMT (Heafield and Lavie, 2010).

For our final system, we used morphological segmentation but not system combination. We used the full newsdev2015 test as tuning data. Table 10 gives mean BLEU scores for this setup, averaged over three MERT runs. Our submitted system was the right binarized system that, out of the three, scored highest on newsdev2015.

	word	morph	syscomb
Hiero	17.8	19.1	19.2
String-to-tree	17.6	18.5	18.7
+ right binarization	17.8	18.9	18.9

Table 9: Finnish to English experiments with morphological segmentation.

system	dev	test
Hiero	18.6	17.5
String-to-tree	18.3	17.2
+ right binarization	18.5	17.7

Table 10: Finnish to English translation results (BLEU) on dev (newsdev2015) and test (newstest2015) sets.

3.7 German to English

For German → English we built a tree-to-string system with similar setup as last year's (Williams et al., 2014). Our submitted system was right binarized with the following extraction parameters: *Rule Depth = 7, Node Count = 100, Rule Size = 7*. At decoding time we used the following non-default parameter value: *max-chart-span = 25*. This limits sub derivations to a maximum span of 25 source words. For the Hiero baseline system we used *max-chart-span = 15*. For tuning we used a random subset of 2000 sentences drawn from the full tuning set.

We performed some preliminary experiments with neural bilingual language models, our re-implementation of the "joint" model of (Devlin et al., 2014). The bilingual language models are trained with the NPLM toolkit (Vaswani et al., 2013). We used 250-dimensional input embedding and hidden layers, and input and output vocabulary sizes of 500000 and 250000 respectively. One bilingual language model was a 5-gram model with an additional context of 9 source words, the affiliated source word and a window of 4 words on either side. A second model was a 1-gram model with an additional context of 13 source words. The language models were trained on the available parallel corpora.

We also added a 7-gram class-based language model, with 50 word classes trained using `mkcls`

system	devtest	test
Hiero	27.7	28.0
String-to-tree	28.7	28.7
+ bilingual LMs	28.6	28.7
+ bilingual & class LMs	28.3	28.7

Table 11: German to English translation results (BLEU) on devtest (newstest2014) and test (newstest2015) sets.

(Och, 1999). The language model was trained on all available monolingual corpora, filtering out singletons.

Table 11 shows the results. As the preliminary results were not encouraging, we did not include the bilingual LMs and class LMs in our submitted system.

3.8 Russian to English

For Russian → English we built a string-to-tree system, using the 2012 and 2013 test sets as tuning data. Table 12 gives the mean BLEU scores, averaged over three MERT runs. Our submitted system was the right binarized system that, out of the three runs, scored highest on devtest.

system	devtest	test
Hiero	31.2	27.1
String-to-tree	30.5	25.9
+ right binarization	30.6	26.2

Table 12: Russian to English translation results (BLEU) on devtest (newstest2014) and test (newstest2015) sets.

4 Manual Error Analysis

Our syntax-based systems for the German–English language pairs have greatly improved over the last years and outperformed traditional phrase-based statistical machine translation systems. Translating between German and English is a challenge for those systems, since extensive long distance reordering and long distance agreement constraints do not fit that approach. Are our syntax-based systems tackling these problems better? And what are the main remaining problems?

For both German–English and English–German, we analyzed 100 sentences, we carried out an error analysis using linguistic error categories that roughly match other efforts in this area (Vilar et al., 2006; Toral et al., 2013; Herrmann et al., 2014; Lommel et al., 2014; Aranberri, 2015). We used the following error annotation protocol:

1. A bilingual speaker corrects the machine translation output with minimal necessary edits to render an acceptable translation. This is done in view of the human reference translation, but typically a much more literal translation was obtained.
2. Each edit is noted in a list in the form "old string → new string", where either old or new string may also be empty or discontinuous.
3. In a second pass, all edits are classified with error categories.

Such an error analysis is subjective. There are many ways to correct errors (step 1), many ways to split corrections into units (step 2), and many ways to classify the errors (step 3). Moreover, analyzing only 100 sentences does not lead to strong statistically significant findings. With this in mind, the following analysis is broadly indicative of the main error types in our syntax-based systems.

Occasionally, parts of a machine translation are just too muddled that a sequence of edits could be established. This happened in 8 German–English sentences, and 7 English–German sentences.

4.1 German–English

16 sentences have no error, while 18 sentences have only one error. These are of course typically the shorter ones. The longest sentence without error is:

- Source: *Der Oppositionspolitiker Imran Khan wirft Premier Sharif vor, bei der Parlamentswahl im Mai vergangenen Jahres betrogen zu haben.*
- MT: *The opposition politician Imran Khan accuses Premier Sharif of having cheated in the parliamentary election in May of last year.*

This is not a trivial sentence, since it requires the translation of the complex subclause construction *accuses ... of having cheated*, which is rendered quite differently in German as *wirft ... vor ... betrogen zu haben*.

An overview of the major error categories is shown is Figure 13. On average, 2.85 errors per sentence were identified. This gives us guidance on the major problems we should be working on in the future.

Count	Category	Count	Category
29	Wrong content word - noun	6	Wrong content word - phrasal verb
25	Wrong content word - verb	6	Added function word - determiner
22	Wrong function word - preposition	5	Unknown word - noun
21	Inflection - verb	5	Missing content word - adverb
14	Reordering: verb	5	Missing content word - noun
13	Reordering: adjunct	5	Inflection - noun
12	Missing function word - preposition	4	Reordering: NP
10	Missing content word - verb	3	Missing content word - adjective
9	Wrong function word - other	3	Inflection - wrong POS
9	Wrong content word - wrong POS	3	Casing
9	Added punctuation	2	Unknown word - verb
8	Muddle	2	Reordering: punctuation
8	Missing function word - connective	2	Reordering: noun
8	Added function word - preposition	2	Reordering: adverb
7	Missing punctuation	2	Missing function word - determiner
7	Wrong content word - adverb	2	Inflection - adverb

Table 13: Main error types in German–English system (count in 100 sentences).

Lexical choice The biggest group of error types concern translation of basic concepts. On average, such errors occur 0.76 times per sentence. Given the vast number of content words that need to be translated, the actual performance on the task of lexical translation is pretty high, but it is by no means solved.

Count	Category
29	Wrong content word - noun
25	Wrong content word - verb
9	Wrong content word - wrong POS
7	Wrong content word - adverb
6	Wrong content word - phrasal verb

Prepositions We were surprised by the large number of errors revolving prepositions. Prepositions are frequent, but not as frequent as content words, so the performance on the preposition translation task is not as good. Prepositions mostly mark relationships of adjuncts, which involve quite complex considerations — the adjunct, the modified verb or noun phrase, identifying the relationship between them in the source sentence, and the fuzzy meaning of prepositions.

Count	Category
22	Wrong function word - preposition
12	Missing function word - preposition
8	Added function word - preposition

Reordering We were also surprised by the low number of reordering errors. The different word order between German and English has hampered translation quality for this language pair historically. While we cannot declare complete success, our syntax-based systems constitute great progress in this area.

Count	Category
14	Reordering: verb
13	Reordering: adjunct
4	Reordering: NP
2	Reordering: noun
2	Reordering: adverb

Other issues with verbs Reordering errors involving verbs top the list in the previous group of error types, but there are also other problems with verbs: their inflection and the unacceptable frequency of dropping verbs. The latter has its roots in faulty word alignment which are based on IBM Models which often fail to align the out-of-English-order German verb, thus enabling the translation model to drop them, which the language model often prefers. Inflection is here to be understood broadly, including the need for the right function words to form a grammatical correct verb complex (e.g., *will have been resolved*).

Count	Category
21	Inflection - verb
10	Missing content word - verb

Overall, the main thrust of future research should be focused on lexical choice, selecting correct prepositions, and producing the correct verb.

Count	Category	Count	Category
41	Wrong content word - verb	9	Compound merging
37	Wrong content word - noun	8	Added function word - preposition
33	Reordering - verb	7	Punctuation - inserted
30	Inflection - verb	7	Muddle
22	Missing function word - preposition	7	Missing function word - clausal connective
17	Inflection - np	7	Added function word - determiner
14	Wrong function word - preposition	5	Punctuation - missing
12	Wrong content word - phrasal verb	5	Missing content word - verb
12	Wrong content word - wrong POS	4	Reordering - adverb
12	Wrong function word - clausal connective	4	Wrong content word - adverb
11	Reordering - pp	3	Missing content word - adjective
11	Inflection - noun	2	Reordering - pronoun
10	Wrong function word - pronoun	2	Wrong content word - name
10	Missing function word - pronoun	2	Missing content word - adverb
10	Missing function word - determiner	2	Wrong content word - adjective
9	Reordering - noun	2	Added function word - pronoun

Table 14: Main error types in English–German system (count in 100 sentences).

4.2 English–German

12 Sentences had no error, 13 sentences only one error. Less than German–English, which supports the general contention that translating into German is harder. On average, a total of 3.8 errors per sentence were marked, one error per sentence more than German–English. An overview of the major error categories is shown is Figure 14.

The longest sentence with no error is:

- Source: *Congressmen Keith Ellison and John Lewis have proposed legislation to protect union organizing as a civil right.*
- Target: *Die Kongressabgeordneten Keith Ellison und John Lewis haben Gesetze zum Schutz der gewerkschaftlichen Organisation als Bürgerrecht vorgeschlagen.*

In terms of word order, this is not a complicated sentence (besides the verb movement *proposed→vorgeschlagen*), but it does involve switching of part-of-speech for two content words: *protect→Schutz* (verb→noun), *union→gewerkschaftlichen* (noun→adjective).

Lexical choice As with German–English, this is biggest group of error types, with 1.08 errors per sentence. Verb sense errors tend to be more subtle, such that a media outlet does not *sagt* (*says*) but *berichtet* (*reports*) a news item. For nouns, there were several stark errors, such the mis-translation of *patient* as *Geduld* (*patience*) in a medical context. In general, there is no reason to believe that models that more strongly draw on a wider context could not resolve many of these cases.

Count	Category
41	Wrong content word - verb
37	Wrong content word - noun
12	Wrong content word - phrasal verb
12	Wrong content word - wrong POS
4	Wrong content word - adverb
2	Wrong content word - adjective

Role and order of adjuncts and arguments While the overall sentence structure is mostly correct, there are often problems with the handling of adjunct and argument phrases. Their role is identified in German by a preposition or the case of a noun phrase (the main cause of inflection errors). Their position in the sentence is less strict, but mistakes can be and are made.

Count	Category
22	Missing function word - preposition
17	Inflection - np
14	Wrong function word - preposition
11	Reordering - pp
11	Inflection - noun
8	Added function word - preposition

Verbs Reordering errors of verbs mainly occur in complex subclause constructions. German verbs are more strongly inflected for count and person, and often a few function words are needed

in just the right order and placement for a correct verb complex.

33	Reordering - verb
30	Inflection - verb
5	Missing content word - verb

Pronouns Due to grammatical gender of nouns in German, translating *it* and *they* is a complex undertaking. German verbs also require more frequently reflexive pronouns.

Count	Category
10	Wrong function word - pronoun
10	Missing function word - pronoun
2	Added function word - pronoun

Clausal connectives A specific problem of English–German translations are clausal connectives. In English, the relationship of the sub clause is often not explicitly marked (e.g., *Police say the rider*), while German requires a function word.

Count	Category
12	Wrong function word - clausal connective
7	Missing function word - clausal connective

Overall, while there are more structural problems than for German–English, often the remaining challenge is the disambiguation of lexical choices and the correct labelling of syntactic relationships.

5 Conclusion

This year we submitted syntax-based systems for all language pairs except English-French. Our English → German system included significant improvements over last year's and we intend to continue developing this system. We presented the first results using Moses' STSG-based tree-to-string model.

Acknowledgements

This project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreements 645452 (QT21) and 644402 (HimL), and from the Swiss National Science Foundation under grant P2ZHP1_148717.

References

Nora Aranberri. 2015. Smt error analysis and mapping to syntactic, semantic and structural fixes. In *Proceedings of the Ninth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 30–38, Denver, Colorado, USA, June. Association for Computational Linguistics.

Stanley F. Chen and Joshua Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report, Harvard University.

Danqi Chen and Christopher Manning. 2014. A Fast and Accurate Dependency Parser using Neural Networks. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 740–750, Doha, Qatar.

Colin Cherry and George Foster. 2012. Batch Tuning Strategies for Statistical Machine Translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada, June.

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

Tagyoung Chung, Licheng Fang, and Daniel Gildea. 2011. Issues concerning decoding with synchronous context-free grammar. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 413–417, Portland, Oregon, USA, June.

Adrià de Gispert, Sami Virpioja, Mikko Kurimo, and William Byrne. 2009. Minimum bayes risk combination of translation hypotheses from alternative morphological decompositions. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics, Companion Volume: Short Papers*, pages 73–76, Boulder, Colorado, June.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and Robust Neural Network Joint Models for Statistical Machine Translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, MD, USA, June.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A Simple, Fast, and Effective Reparameterization of IBM Model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, GA, USA, June.

Jason Eisner. 2003. Learning non-isomorphic tree mappings for machine translation. In *The Companion Volume to the Proceedings of 41st Annual Meeting of the Association for Computational Linguistics*, pages 205–208, Sapporo, Japan, July. Association for Computational Linguistics.

Michel Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a Translation Rule? In *HLT-NAACL '04*.

Michel Galley, Jonathan Graehl, Kevin Knight, Daniel Marcu, Steve DeNeefe, Wei Wang, and Ignacio Thayer. 2006. Scalable inference and training of context-rich syntactic translation models. In *ACL-44: Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 961–968, Morristown, NJ, USA.

Qin Gao and Stephan Vogel. 2008. Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, SETQA-NLP '08, pages 49–57, Stroudsburg, PA, USA.

Kenneth Heafield and Alon Lavie. 2010. Combining Machine Translation Output with Open Source The Carnegie Mellon Multi-Engine Machine Translation Scheme. *The Prague Bulletin of Mathematical Linguistics*, 93:27–36.

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.

Teresa Herrmann, Jan Niehues, and Alex Waibel. 2014. Manual analysis of structurally informed reordering in german-english machine translation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC-2014)*. European Language Resources Association (ELRA).

Mark Hopkins and Greg Langmead. 2010. SCFG decoding without binarization. In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 646–655, Cambridge, MA, October.

Matthias Huck, Hieu Hoang, and Philipp Koehn. 2014. Preference Grammars and Soft Syntactic Constraints for GHKM Syntax-based Statistical Machine Translation. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 148–156, Doha, Qatar.

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.

Ding Liu and Daniel Gildea. 2005. Syntactic Features for Evaluation of Machine Translation. In *Proceedings of the ACL Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization*, pages 25–32, Ann Arbor, Michigan.

Arle Lommel, Aljoscha Burchardt, Maja Popović, Kim Harris, Eleftherios Avramidis, and Hans Uszkoreit. 2014. Using a new analytic measure for the annotation and analysis of mt errors on real data. In *Proceedings of 17th Annual conference of the European Association for Machine Translation*, pages 165–172.

Maria Nadejde, Philip Williams, and Philipp Koehn. 2013. Edinburgh's Syntax-Based Machine Translation Systems. In *Proceedings of the Eighth Workshop on Statistical Machine Translation*, pages 170–176, Sofia, Bulgaria, August.

Graham Neubig and Kevin Duh. 2014. On the elements of an accurate tree-to-string machine translation system. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 143–149, Baltimore, Maryland, June.

Joakim Nivre and Jens Nilsson. 2005. Pseudo-Projective Dependency Parsing. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 99–106, Ann Arbor, Michigan.

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

Franz Josef Och. 1999. An Efficient Method for Determining Bilingual Word Classes. In *Proceedings of the 9th Conference of the European Chapter of the Association for Computational Linguistics (EACL)*, pages 71–76.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics - Volume 1*, ACL '03, pages 160–167, Morristown, NJ, USA.

Slav Petrov and Dan Klein. 2007. Improved Inference for Unlexicalized Parsing. In *Human Language Technologies 2007: The Conference of the North American Chapter of the Association for Computational Linguistics; Proceedings of the Main Conference*, pages 404–411, Rochester, New York, April.

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*, ACL-44, pages 433–440.

Rico Sennrich and Barry Haddow. 2015. A Joint Dependency Model of Morphological and Syntactic Structure for Statistical Machine Translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Lisbon, Portugal. Association for Computational Linguistics.

Rico Sennrich, Martin Volk, and Gerold Schneider. 2013. Exploiting Synergies Between Open Resources for German Dependency Parsing, POS-tagging, and Morphological Analysis. In *Proceedings of the International Conference Recent Advances in Natural Language Processing 2013*, pages 601–609, Hissar, Bulgaria.

Rico Sennrich. 2014. A cyk+ variant for scfg decoding without a dot chart. In *Proceedings of SSST-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*, pages 94–102, Doha, Qatar, October.

Rico Sennrich. 2015. Modelling and Optimizing on Syntactic N-Grams for Statistical Machine Translation. *Transactions of the Association for Computational Linguistics*, 3:169–182.

Andreas Stolcke. 2002. SRILM – an Extensible Language Modeling Toolkit. In *Proc. of the Int. Conf. on Spoken Language Processing (ICSLP)*, volume 3, Denver, CO, USA, September.

Antonio Toral, Sudip Kumar Naskar, Joris Vreeke, Federico Gaspari, and Declan Groves. 2013. A web application for the diagnostic evaluation of machine translation over specific linguistic phenomena. In *Proceedings of the 2013 NAACL HLT Demonstration Session*, pages 20–23, Atlanta, Georgia, June. Association for Computational Linguistics.

Ashish Vaswani, Yinggong Zhao, Victoria Fossum, and David Chiang. 2013. Decoding with Large-Scale Neural Language Models Improves Translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1387–1392, Seattle, WA, USA.

David Vilar, Jia Xu, Luis Fernando D'Haro, and Hermann Ney. 2006. Error analysis of statistical machine translation output. In *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC 06)*, pages 697–702.

Sami Virpioja, Peter Smit, Stig-Arne Grönroos, and Mikko Kurimo. 2013. Morfessor 2.0: Python implementation and extensions for morfessor baseline. Technical report, Aalto University, Helsinki.

Philip Williams and Philipp Koehn. 2012. GHKM Rule Extraction and Scope-3 Parsing in Moses. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 388–394, Montréal, Canada, June.

Philip Williams, Rico Sennrich, Maria Nadejde, Matthias Huck, Eva Hasler, and Philipp Koehn. 2014. Edinburgh's Syntax-Based Systems at WMT 2014. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 207–214, Baltimore, Maryland, USA, June.

Hui Zhang, Min Zhang, Haizhou Li, and Chew Lim Tan. 2009. Fast translation rule matching for syntax-based statistical machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1037–1045, Singapore, August.