Return of the Linguist: Toward scalable parallel corpus generation for machine translation

Abstract

The large-scale generation of synthetic parallel corpora for data-driven machine translation is proposed in this demo paper. A rudimentary system for doing this is presented, with mechanisms for syntactic complexity in the spirit of classical rule-based machine translation and high-quality sentence pairs in the spirit of example-based machine translation. Preliminary experiments integrating these synthetic corpora into the IWSLT shared task are reported. In the process, it was incidentally found that correcting $\sim$4% of the shared task's training corpus for obvious, clerical-type errors resulted in a $\sim$6% relative increase in BLEU scores. Promising future directions are discussed, in particular, large-scale contributions from "amateur linguists" via crowdsourcing.

1 Motivation

As early as Jelinek's (in)famous quote during the 1980's that "*Every time I fire a linguist, my performance goes up*," both linguists and linguistics have been increasingly marginalized in natural language processing (NLP). More recently, for example, Collobert et al. (2011) mention the possibility "*[ideally]...to learn from letter sequences rather than words*," illustrating an AI-motivated desire to move away from even fundamental linguistic concepts. A specific instance of this overall trend is the gradual shift of machine translation (MT) away from classical linguistic rule-based approaches to contemporary machine learning approaches (Hutchins, 2005), which underlie statistical phrase-based systems like Moses (Koehn et al., 2007) and more recent neural network systems (Sutskever et al., 2014; Cho et al., 2014). But in the end, even the most cutting-edge machine learning MT system relies crucially on bilingual parallel corpora, typically taken from external sources like multilingual parliamentary proceedings. This results in what we perceive to be a disproportionate amount of effort being dedicated to the machine learning component of machine translation, with input corpora being treated almost as an afterthought.[1]

One of the major advantages of data-driven MT is that it removes the need for linguists to develop a new set of rules for an MT system to support a new language. In a sense, though, the rule-based MT linguist's job has really been "outsourced" to third-party translators, who

[1] Of course, these parallel corpora represent hundreds of translator-hours of work. However, such translations are optimized for human consumption, not machine learning, with no regard for such niceties as sentence alignment. This is not unlike an athlete having an Olympic-level training regimen, fueled by food scrounged off the streets.

An example of suboptimal parallel corpora being used for machine translation is given in Section 4.2. At the very least, it suggests a need for more careful post-editing of bitexts, particularly for development and test corpora. More ideal, though, would be the creation of parallel corpora for the express purpose of training, tuning, and evaluating data-driven MT systems.

implicitly encode linguistic information in parallel corpora. We believe there is still a significant "in-house" role for linguists to play in corpus-based machine translation, and that is to direct the crafting of high-quality, "designer" synthetic parallel corpora, leveraging technologies like crowdsourcing to maximize throughput. Such a division of labor would let linguists be linguists, with MT system development proceeding in parallel.

Large, high-quality synthetic parallel corpora could be useful as:

- Training sets for corpus-based MT systems, particularly for low-resource language pairs
- Benchmark tasks for MT evaluation, in the spirit of Facebook's bAbI project (Weston et al., 2015)
- Translation memories for computer-assisted translation
- "Language archives": New NLP methods are developed every few years; linguistic theories change every decade or so; parallel corpora live forever (like the Rosetta Stone).

We believe that natural language processing in general and machine translation in particular are particularly amenable to artifical data generation. Unlike many machine learning applications (image recognition, finance), NLP data sets are manmade to begin with; humans themselves *are* the primary natural producers and consumers of raw written language data. Translation is further privileged as perhaps the only NLP task for which millions of people (foreign-language learners) already undergo years of formal training, which has implications for scalability via crowdsourcing (Section 5).

Other practical considerations shine a favorable light on parallel corpus generation. As the "inverse" of translation, multilingual generation is also an easier task in some senses. For example, lexical ambiguity can be largely alleviated,[2] segmentation of languages like Chinese (Chang et al., 2008) becomes a non-issue, and there is the potential for generating annotated corpora with zero annotation error. In principle, high-quality alignments could be obtained at the word or phrase level for use in the statistical MT process, and these alignments could also be potentially useful in training methods for MT subtasks such as syntactic pre-reordering (Khapra et al., 2012). Parallel corpora also provide a natural avenue to interface contemporary MT systems with the countless linguist-years of expertise that have been invested into rule-based MT systems. And unlike the various different MT methods, which are by nature competitive, corpora are inherently collaborative.

This paper is organized as follows. Section 2 discusses previous related work, Section 3 describes our rudimentary parallel corpus generator, Section 4.1 gives experimental results, and Section 5 proposes future work.

2 Related Work

The idea of using synthetic parallel corpora in machine translation is hardly new. Previous attempts at creating synthetic parallel corpora have used rule-based MT (RBMT) systems (Hu et al., 2007; Dugast et al., 2008; Murakami et al., 2009; Rubino et al., 2014), but the input to the RBMT systems was again dictated by external corpora in the desire to produce immediate results. A longer-term approach that may pay dividends (Section 1) would be to run RBMT systems on specially chosen sentences for which they are known to produce high-quality translations, or perhaps eventually directly encoding their rules into carefully crafted parallel corpora.

Our parallel corpus generation system (Section 3) is very similar to the recursive equivalence class approach to example-based MT (Carl and Way, 2003) used by Brown (1999). Our

[2] Ideally the generator would "know" *a priori* whether it is writing a sentence about, say, (financial) banks or (river) banks.

generator has more of a rule-based flavor, using a somewhat more varied grammar that also admits the use of participial noun modifiers. But more importantly, we believe this line of work was ahead of its time and consider our main contribution to be bringing attention to its potential value to contemporary MT.

Equivalence classing is in fact an integral part of the standard pipeline of Moses (Koehn et al., 2007), the open-source statistical MT system. During training, the `mkcls` component adds "*an 'example-based' touch to the statistical approach*" by clustering words into automatically-generated equivalence classes to ameliorate data sparsity issues (Och, 1999). However, these clusters are trained only at the word level and not for phrases, and the clusters themselves are typically only used during alignment. Furthermore, since they are learned in an unsupervised manner from the parallel corpus, they can still benefit from the addition of more high-quality training examples, synthetic or otherwise.

In terms of freely available multilingual natural language generators, the most well-developed one we found was KPML (Bateman, 1997). But crafting sentences using KPML is a lengthy process,[3] involving a well-developed linguistic theory to explain the surface forms. In contrast, our primary goal is to generate large parallel corpora of surface forms quickly.

In the next section, we describe our own primitive attempt at building a multilingual generator from scratch. While we try to model straightforward linguistic phenomena compositionally, we also adopt the pragmatic attitude, in the tradition of example-based MT (Carl and Way, 2003), that sometimes the most effective way to represent a translation phenomenon is simply to write it down. Thus, with the long-term goal of scalability, we attempt to construct a system sufficiently flexible to write some sentence pairs in large quantities through syntactic variations and others with "hand-crafted" quality through example-based templates.

3 A Rudimentary Multilingual Generator

As an example of how our synthetic parallel corpus generation system works, consider the following English-Chinese template pair:

```
S V O .
S 把 O V 。
```

Our system then expands the subject `S` and object `O` as noun phrases, adding adjectival and participial modifiers as desired. Semantic constraints can be imposed from the verb `V` as well as externally. The generator can then vary the template lexically and syntactically,[4] allowing a large number of sentence pairs to be generated from a single template. We can also trade quantity for quality by adding more literals to the template, in the spirit of example-based MT.

3.1 Simplistic Linguistics

We adopt an extremely simplistic view of language as nouns, verbs, and their modifications and elaborations. Inspired by early work on transformations by Chomsky (1957), we focus on declarative indicative present active sentences in the hopes that they can be transformed into other forms in the long run. Similarly, although we enforce some semantic constraints, our system generally produces nonsensical sentences like Chomsky's celebrated "*Colorless green ideas sleep furiously.*" The fact that verb phrases can also serve as nouns (gerundives) and modifiers (participles) appears to be a major source of syntactic complexity, and we made it a priority to implement participial modifiers.

[3] For examples, see http://www.fb10.uni-bremen.de/anglistik/langpro/kpml/genbank/generation-bank.html

[4] Rewrite probabilities were set by hand; a more principled approach might use, say, a generative synchronous CFG.

3.2 Software Architecture

The bulk of our software, which we plan on eventually open-sourcing, along with its accompanying data, comprises four Python modules:[5]

- `data`: data abstraction layer
- `nodes`: internal tree representation
- `generator`: converts internal trees to surface forms (sentence pairs)
- `main`: outermost logic

The `data` module implements an abstraction layer that mediates between data resources on disk and the rest of the program. These data resources include bilingual (extensible to multilingual) lexicons for different parts of speech, monolingual morphology, templates, and a small taxonomy used to tag nouns. All of our data resources were written in YAML, a "human-readable" language particularly well-suited to manual editing. In particular, YAML has native Unicode support and does not even require quotation marks to enclose strings. An example of YAML code is shown in Figure 1.

```yaml
transitive:
  example: Alice|S kicks|V Bob|O .

  symbols:
    V:
      type: verb
      description: head verb
    S:
      type: NP
      description: subject
    O:
      type: NP
      description: object

  langs:
    en:
      template: S V O
      punctuation: '.'
      tags:
        S: subjective
        O: objective
      dependencies:
        V: S
    zh:
      template: S V O
      punctuation: 。
```

```yaml
action.creation:
  template: transitive
  tags:
    S: [person]
  verbsets:
  - en: build
    zh: 建立
  - en: [create, make]
    zh: [做, 创造, 制造]
  - en: design
    zh: 设计
  - en: develop
    zh: 发展
  - en: [generate, produce]
    zh: 产生
```

Figure 1: An example of YAML code, showing syntactic (left) and semantic (right) components of a clause template used for the "rule-based" mode of generation mentioned in the discussion of the `main` module.

[5] In particular, we used Python 3 for its improved Unicode support over Python 2.

The `nodes` module implements the data structures used internally by the program. To capture recursive structures like participles, we adopt a rudimentary tree representation that is something intermediate between constituency and dependency trees, with constituent-like modifiers that also maintain dependency-like links to their targets. Following our highly simplified picture of linguistics in Section 3.1, the basic backbone of the tree is constituency-like, as shown in Figure 2.

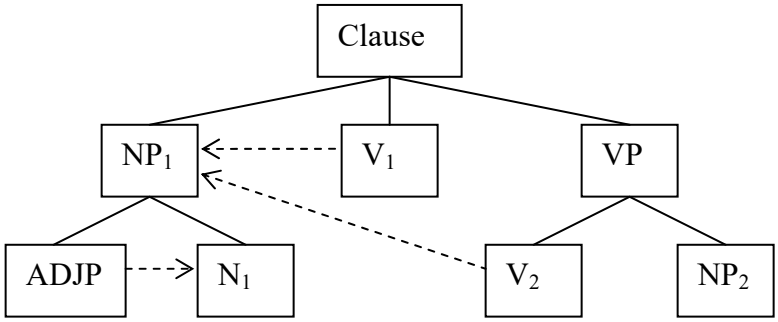

Figure 2: An example tree made of nodes from the `nodes` module. Solid lines indicate the "constituent" backbone that dictates the order of traversal in the `generator` module, while dashed lines indicate longer-range syntactic or semantic dependencies that are stored as links between nodes.

The `generator` module traverses the backbone of the "interlingual" tree from the `nodes` module and recursively builds up parallel surface forms (bilingual sentence pairs). Language-specific rules (e.g., word order) are implemented as ad hoc bits of code in this module,[6] reading long-range dependency information from the tree as needed. All language-dependent code is confined to this module, although this is coupled with language-dependent information from the YAML data (e.g., the tags and punctuation in Figure 1). For example, it is left to the generator to order prepositional phrases after their targets in English but before their targets in Chinese.

Finally, the outermost `main` module is responsible for tree instantiation and modification. To generate a sentence pair in "rule-based mode," a syntactic clause template is randomly selected, followed by a semantic verb category that uses that syntactic template, as in Figure 1. Modifier nodes are then randomly attached, recursively so for constructs like participles and prepositions, in principle admitting arbitrarily long sentences. The probabilities for these random processes are chosen heuristically as mentioned in Footnote 4. The system also allows for semantic constraints by adding semantic tags to nodes and querying the taxonomy through the `data` module for eligible lexical choices. Building a comprehensive taxonomy and labeling the lexical database exhaustively is a labor-intensive task; results in Section 4 below were obtained with relatively minimal semantic constraints.

[6]Formally, the "rule-based" component of our system can be considered an extremely rudimentary synchronous (bilingual) context-free grammar of sorts (Saers and Wu, 2013) that also admits long-range dependencies and semantic constraints. But it would be a disservice to those other well-developed methods to refer to our system as anything other than a makeshift way to generate multilingual sentences.

One of the things we wanted to do differently from KPML (Bateman, 1997) was to implement an "example-based mode" in which sentence pairs that would be difficult to generate in "rule-based mode" could simply be written down and then amplified as proposed by Brown (1999). This would trade the quantity available through arbitrarily complex sentences from "rule-based" mode for the quality of sentence pairs with a larger degree of hand-translation. Our software achieves this by allowing the `template` entries as on the left side of Figure 1 to contain literal strings in addition to symbols. These "example-based" templates are currently selected by hand for the experiments described in Section 4, with all amplification resulting from varying the lexical choices (i.e., without adding). Whether it helps to add modifier nodes in this mode is a potential area for future work. A helper script also supports the use of "meta-templates" to facilitate hand-construction of templates, allowing a minimal annotation of a sentence pair to produce a full YAML template similar to Figure 1.

3.3 Implemented Linguistic Phenomena

Finally, to conclude this section, we briefly list linguistic constructs and their implementations in terms of the tree representation from Section 3.2. Note that since we are interested primarily in the generated surface forms, the linguistic soundness of how they are implemented in our tree representation is of relatively secondary importance. Aside from manageability, there are a few other reasons for focusing on just the most basic linguistic phenomena for the "rule-based" mode. We strongly believe that for the purposes of generating large and high-quality parallel corpora, it is most effective to treat difficult, non-compositional phenomena by simply writing them down, in tradition of example-based MT. Also, minimizing the learning curve makes it potentially easier to scale up to admit contributions from non-linguists.

Clauses	indicative present active; intransitive, transitive, "meta" (clause subnode), "modalish" (e.g., "Alice wants to kick Bob" - with "Alice" linked to "kick" as in Figure 2 for subject-verb semantics)
NP	noun phrases headed by nouns, names, or pronouns; optional modifiers as child nodes; pronoun antecedents via links
ADJP	adjectives as NP modifiers, with links to head nouns (e.g., this vs. these)
ADVP	adverbs as Clause or ADJP modifiers
VP	transformations of clauses into participles or modal(ish) complements; convert subject child node into a linked node, allowing reuse of semantic constraints
PP	prepositional phrases (relations) between two nodes (noun/noun or noun/verb), with relation-dependent semantic requirements on both nodes; indirect objects as "null preposition" nodes

Table 1: Linguistic constructs implemented for the "rule-based" mode of our generator.

4 Experimental Results

4.1 Training Set Amplification

We now consider training set amplification as a first application, using the Chinese to English MT task of IWSLT 2015 (Cettolo et al., 2015), mentioned briefly in Section 4.2. This task uses the WIT^3 corpus (Cettolo et al., 2012), which consists of multilingual subtitles of TED talks covering a wide variety of subjects. The freely available MultiUN corpus (Eisele and Chen, 2010) from the OPUS project (Tiedemann, 2012) was also used for comparison.

To investigate the relative effect of our synthetic parallel corpora, we chose a simple baseline system and fixed all other variables. Namely, we used the baseline configuration of Moses

(Koehn et al., 2007) as described in (Cettolo et al., 2015), with the exception of changing the target language model to KenLM (Heafield et al., 2013), which we found to be faster and stabler at decoding time than IRSTLM (Federico et al., 2008). Since monolingual corpora are typically available at scale, target language models were obtained using the same in-domain monolingual corpus for all runs. For the remaining unspecified model settings, we used the `grow-diag` alignment heuristic recommended by Chang et al. (2008) and the `mslr-fe` configuration for lexicalized reordering. Chinese text was preprocessed using the latest version of the Stanford Segmenter (Chang et al., 2008). Phrase and lexicalized reordering tables were interpolated using the `tmcombine` script included with Moses (Sennrich, 2012), with the interpolation optimized on the `dev2010` data set. Following Cettolo et al. (2015), the model was tuned on the `tst2010` data set using the MERT procedure included in Moses. For Table 2, BLEU scores were computed as described by Cettolo et al. (2015) using the IWSLT 2015 progress data set (`tst2014`).[7]

	BLEU		BLEU
25% corpus	9.95	Official	11.43
+ 75%	**11.47**	Our baseline	11.54
+ MultiUN	10.76	+ MultiUN	**11.67**
+ Generated	9.91	+ Generated	11.46

Table 2: (Left) Chinese-to-English BLEU scores for our baseline Moses configuration with phrase and reordering tables trained on a random 25% subset of the IWSLT15 training corpus and interpolated (+) with tables for the remaining 75% (∼150K) of the in-domain corpus, the MultiUN corpus (a ∼1M line subset), and a synthetic corpus from our generator (∼5M lines). (Right) Same as the left, but starting from tables trained on the entire IWSLT15 corpus. Official published scores of the IWSLT15 baseline system are also given for comparison. Qualitatively similar trends for NIST and TER scores were also observed (omitted for clarity).

The results of these experiments are shown in Table 2. For the table on the left, note that adding in-domain data improves performance significantly more than a much larger amount of out-of-domain data (MultiUN), a commonly reported phenomenon (Sennrich, 2012). TED talks, which are prepared lectures, fall somewhere between the conversational language of movie subtitles and the formal written language of UN resolutions. As such, the OpenSubtitles[8] 2016 corpus from the OPUS project was also out-of-domain; results for this corpus were similar to MultiUN and are not shown. For the table on the right, note the diminishing returns of out-of-domain data as the amount of in-domain data increases. These observations underscore the importance of domain overlap and hint at the potential value to be found in the amplification of in-domain corpora.

Our generated corpora do not yet make a significant impact on the final BLEU score,[9] indicating that our parallel corpus generator has a long way to go. Development on our generator has so far focused on "rule-based" syntactic diversity and growing a lexical database tailored to the training set. Given the resulting scores, a change of direction seems in order; work on harvesting more "example-based" templates from the IWSLT15 training corpus is currently in progress. As a first step in this direction, we examine the gold corpora qualitatively.

[7] We chose the `tst2014` test set to facilitate comparison with published results from both IWSLT 2014 and 2015 in future work.

[8] http://www.opensubtitles.org/

[9] The sub-baseline scores shown are for our most recently generated synthetic corpus; scores slightly higher than our baseline have also been obtained on older generated corpora. Both of these outcomes seem to be within experimental error bars.

4.2 Qualitative Analysis of Corpora

Consider the official Chinese to English baseline translation of the `tst2010` data set[10]. As a crude way of assessing translation quality, we used the official evaluation software MultEval (Clark et al., 2011) to compute scores for each translated line individually. Upon manual inspection, the worst scoring individual lines (especially above 150%) were almost all found to correspond to poor human translations on the Chinese side of the gold corpus. This is particularly unsettling because the `tst2010` data set has been used as the official development/tuning set for the past several IWSLT evaluation campaigns. The fact that these errors have persisted this long (`tst2010` has been distributed with only minor changes for every IWSLT Chinese-English evaluation campaign since 2012) is a testament to how insidious these errors can be, even for a prominent international workshop.

Gold zh	安德森：是六万美元。
Gold en	60,000.
Moses output	CA : That is 60,000 dollars .

Table 3: A "poorly translated" sentence pair from `tst2010`, with a TER score of 250%.

An example is given in Table 3. The baseline Moses system tries valiantly, but it ultimately scores poorly for producing a good machine translation of a liberal human translation. Alternatively, one can interpret these errors as an additional source of noise in the human encoding of English into Chinese. It is not immediately obvious how the MT process is affected by this kind of noise.

The most effective automated ways we found to identify lines in dire need of post-editing were:

- Duplicated Chinese segments on the same line. This was presumably caused by having the same sentence displayed at multiple time points in the video subtitles. Often, this would also indicate passages that were poorly aligned at the sentence level. Such translations might suffice for viewing translated videos monolingually, but they are far from ideal for training a machine translation system.

- Missing segments (usually English) when compared with the original lecture transcripts online.[11] This issue seems to have been introduced in the IWSLT15 Chinese-English corpus, and was also indicative of misaligned sentences.

4.3 Experiments on Post-Edited Corpora

Table 4 below shows the scores obtained by "lightly" post-editing the MERT tuning set `tst2010` as follows. On the Chinese side, duplicated segments, applause lines, and translator notes were removed. Speaker annotations were normalized on both sides, as well as sentence alignments wherever possible. As a result, small but noticeable differences in scores are observed. (For comparison, the difference between the best and worst scoring Chinese-to-English systems in IWSLT14 (Cettolo et al., 2014) was 6 BLEU points and 7 TER points.) In this case, the post-edited tuning set seems to guide the MERT optimization procedure of Moses to a slightly different local optimum.

For the rest of this section, we show the results of applying post-editing to corpora used for multiple purposes. To preserve the integrity of `tst2014`, evaluation is now performed using MultEval (Clark et al., 2011) on the `tst2010` data set itself, with the MERT procedure tuned

[10] https://wit3.fbk.eu/score.php?release=2014-01

[11] http://www.ted.com/

	BLEU	NIST	TER
(a) Official IWSLT15	11.43	**4.67**	72.65
(b) Our baseline	11.54	4.43	**72.43**
(c) Post-edited `tst2010`	**11.91**	4.33	75.34

Table 4: Chinese-to-English translation scores evaluated on the `tst2014` data set, with experimental parameters as described in Section 4.1. (a) Official published IWSLT15 baseline scores (Cettolo et al., 2015), for comparison. (b) Our baseline Moses configuration, which uses KenLM instead of IRSTLM, as described previously. (c) Same as (b), but with a post-edited tuning set `tst2010`.

on the `dev2010` data set instead. Our baseline model is also trained on the IWSLT corpus Chinese-English training corpus from 2014 instead of 2015. These data sets and this evaluation software were used by the official WIT[3] benchmark linked in footnote 10. MultEval has the added benefit of providing bootstrapped error bar estimates.

In tables with MultEval scores below, metric scores for all systems were generated using MultEval 0.5.1 (Clark et al., 2011): jBLEU V0.1.1 (an exact reimplementation of NIST's mteval-v13.pl without tokenization); Translation Error Rate (TER) V0.8.0. The p-values are relative to our baseline and indicate whether a difference of this magnitude (between the baseline and the system on that line) is likely to be generated again by some random process (a randomized optimizer). Metric scores are averages over three runs per system. s_{sel} indicates the variance due to test set selection and has nothing to do with optimizer instability.[12]

Metric	**System**	**Avg**	$\bar{s}_{\text{sel}}$	s_{Test}	p**-value**
BLEU ↑	Official IWSLT14 baseline	11.2	-	-	-
	Our baseline	11.4	0.3	0.2	-
	Tuning: `dev2010` (post-edited)	10.8	0.3	0.3	0.00
	Training: 2015 corpus	11.5	0.3	0.3	0.30
	Training: 2015 corpus (post-edited)	**12.1**	0.3	0.1	0.00
	+ generated	11.9	0.3	0.0	0.00
TER ↓	Official IWSLT14 baseline	77.0	-	-	-
	Our baseline	76.2	0.8	0.3	-
	Tuning: `dev2010` (post-edited)	83.4	1.0	0.5	0.00
	Training: 2015 corpus	76.4	0.8	0.5	0.26
	Training: 2015 corpus (post-edited)	**74.8**	0.7	0.1	0.00
	+ generated	74.9	0.7	0.0	0.00

Table 5: MultEval results for systems evaluated on the `tst2010` data set, tuned on the `dev2010` data set, and trained on the 2014 training corpus unless indicated otherwise. Official baseline results (with IRSTLM instead of KenLM as discussed previously) are also given for comparison. Post-editing of the 2015 training corpus involved correcting ∼7500 lines (3.6% of the corpus) as described in Section 4.2, as well as adding ∼750 sentence pairs obtained by splitting lines that were longer than 100 tokens and hence removed from the training procedure by the `clean-corpus-n.perl` script in the Moses toolchain.

Table 5 gives results evaluated on the `tst2010` data set. The IWSLT 2015 training corpus did not produce a significant improvement over the IWSLT 2014 baseline, even though the

[12]This paragraph and the tables with MultEval results are adopted from LaTeX code automatically generated by MultEval.

2015 corpus includes an additional ∼30,000 lines of in-domain data, increasing the corpus size by ∼17%. Post-editing the 2015 corpus to correct for the missing fragments and resulting misaligned sentences mentioned in Section 4.2 resulted in a modest but statistically significant improvement in scores,[13] demonstrating the importance of corpus quality, not just quantity. Finally, unlike Table 4, post-editing the MERT tuning development set `dev2010` surprisingly makes things strictly worse. The effect of corrective post-editing tuning corpora appears to be relatively unpredictable, guiding the MERT optimizer to different local optima with varying out-of-sample performance.

Finally, the "+ generated" entry shows results obtained by concatenating sentence pairs from the quality-oriented "example-based" mode of our generator to the post-edited 2015 corpus. The starting point for writing template pairs was non-compositional translations from the 2015 training corpus with high TER scores, which were only practical to hand-inspect after post-editing to remove the clerical-like errors described in Section 4.2. The basic rationale was that poorly scoring training pairs are ones that the model has not fully "learned" yet. These pairs were amplified by replacing one or more words in the templates with items from a lexical database, as proposed by Brown (1999). Unamplified alternative translations to some liberal (but not incorrect) translations were also included. As there were relatively few of them compared with Table 2, the generated sentences here were simply concatenated to the end of the training corpus. Not unlike the "rule-based" generated corpora from Table 2, these "example-based" generated corpora do not make a significant impact on system performance. These atypical translations may have wound up being overamplified and thus also "teaching" spurious phrases to the statistical MT system (which can be thought of as variable-length example-based MT of sorts). Alternatively, this can be interpreted as indicating that a crucial component of in-domain data is syntactic diversity.

Unlike its use as a tuning set in Section 4.2, post-editing `tst2010` as the test set had a more predictable effect. Namely, translation corrections improved TER scores by ∼5 points compared with Table 5; it was easier for the MT system to reproduce lines that have not been grossly mistranslated. Unlike TER scores, the BLEU scores did not change appreciably. Interestingly, in Table tab:tst2010, BLEU scores also changed much less than TER scores as a result of post-editing `tst2010`, even though it was being used as a tuning set there. Scores evaluated on post-edited `tst2010` are not shown, for the sake of clarity, and since the basic qualitative trends between the baseline, post-edited `dev2010`, etc. were observed to be the same as in Table 5.

4.4 Data Snooping

Warning: This subsection is dedicated to "data snooping" experiments that included information from the test set during training in one way or another. The scores given here must not be considered indicative of actual out-of-sample system performance.

Our reason for using the taboo practice of snooping is to try to isolate the most effective characteristics of in-domain data, whose power was seen in Section 4.1. It is hoped that these characteristics can guide future development of our generator from Section 3, in particular, whether future efforts are better spent on scaling up the lexical database or the template database. The snooping is performed on `tst2010`, to keep `tst2014` relatively pristine for fu-

[13] By comparison, the best scoring systems from the IWSLT 2014 evaluation campaign (Cettolo et al., 2014) scored ∼4 BLEU points and ∼3 TER points better than the baseline. Also, this might look like a proportional improvement in BLEU, but gains were found to come in discrete steps during the editing process. This is thought to be due to the non-parametric nature of statistical MT, with changes in scores only occurring when there are appreciable changes to the parts of the probability tables that are relevant to the test set. It remains to be seen what effect post-editing of training, tuning, and test corpora would have on parametric systems like recent deep-learning based MT methods (Sutskever et al., 2014; Cho et al., 2014), in which all training examples contribute to the same shared connection weight matrices.

ture work. Since they were so much smaller ($\sim$1500 lines vs. $\sim$180000 lines), snooping data sets were simply concatenated to the end of the IWSLT 2014 training corpus instead of using `tmcombine`, with the rationale being that unlike our large, "rule-based" generated corpora from Section 4.1, these snooping additions should only perturb the resulting translation tables slightly.

Metric	**Snooping Type**	**Avg**	$\overline{s}_{\textbf{sel}}$	$s_{\textbf{Test}}$	p**-value**
BLEU ↑	Our baseline	11.4	0.3	0.2	-
	Full snooping	**36.8**	0.6	0.0	0.00
	Syntactic snooping	17.6	0.4	1.0	0.00
	Lexical snooping	11.2	0.3	0.1	0.02
TER ↓	Our baseline	76.2	0.8	0.3	-
	Full snooping	**48.3**	0.7	0.0	0.00
	Syntactic snooping	69.8	0.7	1.3	0.00
	Lexical snooping	76.3	0.7	0.1	0.48

Table 6: Data snooping results for systems trained on information from the `tst2010` data set concatenated to the end of the IWSLT 2014 training corpus and also evaluated with MultEval on `tst2010`. The different types of snooping are defined in the text.

For full snooping, `tst2010` was directly concatenated to the training, establishing an "upper bound" of sorts on how much performance boost is obtainable by simply adding more data. For syntactic snooping, before concatenation, rare tokens in `tst2010` were replaced by their part of speech (POS) from the Stanford POS tagger (Toutanova et al., 2003). Non-rare tokens were arbitrarily defined for each language as the most common 80% in the IWSLT 2014 corpus: 670 tokens for English and 1406 tokens for Chinese, with the latter tokenized using the latest version of the Stanford Segmenter (Chang et al., 2008). This achieves a much smaller but still substantial increase in performance,[14] even though this essentially performs zero snooping on out-of-vocabulary (OOV) words that the baseline system failed to translate. Finally, for lexical snooping, we hand-selected $\sim$100 mistranslated or OOV words from the baseline translation and used the rule-based mode of our generator to generate $\sim$30000 sentences with OOV words for concatenation with the training corpus. This method of lexical snooping in fact makes performance slightly worse (although the TER score is within error bars). This is thought to be because any gains from the translated OOV words are offset by the noise introduced into the phrase table by the nonsensical artificial sentences, which are otherwise composed of very common words. More generally, we speculate that although nonsensical sentences like Chomsky's "colorless green ideas sleep furiously" are fully licensed by English grammar, they can be detrimental in practical NLP applications like MT that operate on naturally occurring sentences at test time. Even when using an in-domain lexicon, their syntax may be sufficiently out-of-domain to result in a net loss of performance.

Taken together, these snooping results reinforce the sentiment from the end of Section 4.1 that future scaled-up work on our generator should eschew complex "rule-based" recursive modifications in favor of focus on the "example-based" approach. Alternatively, much more work on semantic constraints in the "rule-based" component is warranted, before trying to model more complex syntactic linguistic phenomena. Also, the observed relative unimportance of OOV words suggests it is less of a priority for the lexical database to be exhaustive.

[14] The performance gain of 5 BLEU points or 6 TER points over the baseline is comparable with the gains obtained by the best scoring (and of course non-snooping) systems from IWSLT 2014 (Cettolo et al., 2014).

5 Proposed Future Work

Many linguistic constructions have yet to be implemented in our rudimentary/primitive generator (for example, past tense is only partially supported). Future development may be expedited by adapting work from existing rule-based MT or natural language generation systems. Dynamic, on-the-fly corpus generation may also have a role to play in online or curriculum learning settings.

Deep learning methods like RNN encoder-decoders (Sutskever et al., 2014; Cho et al., 2014) are but the latest in a succession of corpus-based approaches to MT, following in the footsteps of example-based and statistical machine translation; in this era of Big Data it is unlikely that they will be the last. But all such methods share a reliance on parallel corpora, and we believe parallel corpus generation will remain a relevant pursuit as even newer methods emerge.

As alluded to in Section 1, parallel corpora have the innate potential for open collaboration; not only can they be expert-sourced to linguists, but they can also crowdsourced to "amateur linguists." One particularly promising application domain is conversational language, for which syntactic and lexical diversity is expected to be relatively low compared with more formal domains. Multilingual lexical databases for slang terms, in particular, would be particularly suitable for crowdsourcing. At the very least, crowdsourcing should be strongly considered for post-editing small corpora used as potentially noise-sensitive components of shared tasks, like development or test sets. For shared evaluation tasks with many language pairs, crowdsourcing may also be the only practical way to do this.

We believe the most promising path to large-scale parallel corpus creation is through educational crowdsourcing. As with the success story of reCAPTCHA (von Ahn et al., 2008), we believe this will scale best when there is a natural harmony between data supply and demand. Unlike other staple NLP tasks like parsing or part-of-speech tagging, MT is particularly well-suited to this, with English being taught as a foreign language in classrooms around the world. The data format (English itself) is widely agreed upon, and there is even an existing market, a vast untapped resource of student problem sets that is yet to be harnessed by machine learning. It is also worth noting that in a sense, this could regarded as a "massively-online" version of elicitation-based techniques from fieldwork linguistics (Probst et al., 2001) for low-resource language pairs, with an emphasis on acquiring unprocessed parallel sentences, and doing so at scale.

Large quantities of supervised data underlie state-of-the-art performance in many NLP tasks. While current paid approaches like Amazon's Mechanical Turk are already a viable way of obtaining crowdsourced data, we believe compelling, widely adopted educational apps like Duolingo[15] to be much more cost-effective yet presently untapped platforms from which to harvest supervised natural language data from its natural source. Compared with current paid approaches, the cost per datum could be significantly reduced or perhaps even become negative (a profitable app), although the value of the harvested data itself might be worth subsidizing. We consider the effective exploitation of such potentially game-changing sources of supervised NLP data to be an open challenge to the NLP community in general.

Acknowledgments

We are very grateful to the anonymous EMNLP 2016 reviewers for many helpful comments on an earlier version of this paper.

[15] https://schools.duolingo.com/

References

Bateman, J. A. (1997). Enabling technology for multilingual natural language generation: The kpml development environment. *Nat. Lang. Eng.*, 3(1):15–55.

Brown, R. D. (1999). Adding linguistic knowledge to a lexical example-based translation system. In *Proceedings of the Eighth International Conference on Theoretical and Methodological Issues in Machine Translation (TMI-99)*, pages 22–32.

Carl, M. and Way, A., editors (2003). *Recent Advances in Example-Based Machine Translation*. Springer Netherlands, Dordrecht.

Cettolo, M., Girardi, C., and Federico, M. (2012). Wit^3: Web inventory of transcribed and translated talks. In *Proceedings of the 16th Conference of the European Association for Machine Translation (EAMT)*, pages 261–268, Trento, Italy.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., Cattoni, R., and Federico, M. (2015). The iwslt 2015 evaluation campaign. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 2–14, Da Nang, Vietnam.

Cettolo, M., Niehues, J., Stüker, S., Bentivogli, L., and Federico, M. (2014). Report on the 11th iwslt evaluation campaign, iwslt 2014. In *Proceedings of the International Workshop on Spoken Language Translation*, pages 2–17, Lake Tahoe, United States.

Chang, P.-C., Galley, M., and Manning, C. D. (2008). Optimizing chinese word segmentation for machine translation performance. In *Proceedings of the Third Workshop on Statistical Machine Translation*, StatMT '08, pages 224–232, Stroudsburg, PA, USA. Association for Computational Linguistics.

Cho, K., Van Merriënboer, B., Gülçehre, Ç., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (2014). Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Chomsky, N. (1957). *Syntactic Structures*, volume 1 of *Janua linguarum. Series minor 4*. Mouton.

Clark, J. H., Dyer, C., Lavie, A., and Smith, N. A. (2011). Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *Proceedings of the Association for Computational Linguistics*.

Collobert, R., Weston, J., Bottou, L., Karlen, M., Kavukcuoglu, K., and Kuksa, P. P. (2011). Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12:2493–2537.

Dugast, L., Senellart, J., and Koehn, P. (2008). Can we relearn an rbmt system? In *Proceedings of the Third Workshop on Statistical Machine Translation*, pages 175–178.

Eisele, A. and Chen, Y. (2010). Multiun: A multilingual corpus from united nation documents. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC 2010)*, pages 2868–2872.

Federico, M., Bertoldi, N., and Cettolo, M. (2008). Irstlm: an open source toolkit for handling large scale language models. In *Proceedings of the International Conference on Spoken Language Processing (ICSLP)*, pages 1618–1621, Brisbane, Australia.

Heafield, K., Pouzyrevsky, I., Clark, J. H., and Koehn, P. (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.

Hu, X., Wang, H., and Wu, H. (2007). Using rbmt systems to produce bilingual corpus for smt. In *Proc. of EMNLP-CoNLL 2007*, pages 287–295.

Hutchins, J. (2005). The history of machine translation in a nutshell.

Khapra, M. M., Ramanathan, A., and Visweswariah, K. (2012). Report of the shared task on learning reordering from word alignments at rsmt 2012. In *Proceedings of the Workshop on Reordering for Statistical Machine Translation*, pages 9–16, Mumbai, India. The COLING 2012 Organizing Committee.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Murakami, J., Tokuhisa, M., and Ikehara, S. (2009). Statistical machine translation adding pattern-based machine translation in chinese-english translation. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT 2009)*, pages 107–112.

Och, F. J. (1999). An efficient method for determining bilingual word classes. In *Ninth Conference of the European Chapter of the Association for Computational Linguistics (EACL'99)*, pages 71–76, Bergen, Norway.

Probst, K., Brown, R., Carbonell, J., Lavie, A., Levin, L., and Peterson, E. (2001). Design and implementation of controlled elicitation for machine translation of low-density languages. In *Proceedings of the MT-2010 Workshop Machine Translation Summit VIII*, pages 189–192, Santiago de Compostela, Spain.

Rubino, R., Toral, A., Ljubešić, N., and Ramírez-Sánchez, G. (2014). Quality estimation for synthetic parallel data generation. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, Reykjavik, Iceland. European Language Resources Association (ELRA).

Saers, M. and Wu, D. (2013). Unsupervised learning of bilingual categories in inversion transduction grammar induction. In *Proceedings of the International Workshop on Spoken Language Translation (IWSLT)*.

Sennrich, R. (2012). Perplexity minimization for translation model domain adaptation in statistical machine translation. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL '12, pages 539–549, Stroudsburg, PA, USA. Association for Computational Linguistics.

Sutskever, I., Vinyals, O., and Le, Q. V. (2014). Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3104–3112.

Tiedemann, J. (2012). Parallel data, tools and interfaces in opus. In *Proceedings of the Eight International Conference on Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Toutanova, K., Klein, D., Manning, C., and Singer, Y. (2003). Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, pages 252–259.

von Ahn, L., Maurer, B., McMillen, C., Abraham, D., and Blum, M. (2008). recaptcha: Human-based character recognition via web security measures. *Science*, 321(5895):1465–1468.

Weston, J., Bordes, A., Chopra, S., and Mikolov, T. (2015). Towards ai-complete question answering: A set of prerequisite toy tasks. *CoRR*, abs/1502.05698.