Generating English Determiners in Phrase-Based Translation with Synthetic Translation Options

Yulia Tsvetkov Chris Dyer Lori Levin Archna Bhatia
Language Technologies Institute
Carnegie Mellon University
Pittspurgh, PA, 15213, USA
{ytsvetko, cdyer, lsl, archna}@cs.cmu.edu

Abstract

We propose a technique for improving the quality of phrase-based translation systems by creating synthetic translation options—phrasal translations that are generated by auxiliary translation and post-editing processes—to augment the default phrase inventory learned from parallel data. We apply our technique to the problem of producing English determiners when translating from Russian and Czech, languages that lack definiteness morphemes. Our approach augments the English side of the phrase table using a classifier to predict where English articles might plausibly be added or removed, and then we decode as usual. Doing so, we obtain significant improvements in quality relative to a standard phrase-based baseline and to a to post-editing complete translations with the classifier.

1 Introduction

Phrase-based translation works as follows. A set of candidate translations for an input sentence is created by matching contiguous spans of the input against an inventory of phrasal translations, reordering them into a target-language appropriate order, and choosing the best one according to a discriminative model that combines features of the phrases used, reordering patterns, and target language model (Koehn et al., 2003). This relatively simple approach to translation can be remarkably effective, and, since its introduction, it has been the basis for further innovations, including developing better models for distinguishing the good translations from bad ones (Chiang, 2012; Gimpel and Smith, 2012; Cherry and Foster, 2012; Eidelman et al., 2013), improving the identification of phrase pairs in parallel data (DeNero et al., 2008; DeNero and Klein, 2010), and formal generalizations to gapped rules and rich nonterminal types (Chiang, 2007; Galley et al., 2006). This paper proposes a different mechanism for improving phrase-based translation: the use of **synthetic translation options** to supplement the standard phrasal inventory used in phrase-based translation systems.

In the following, we argue that phrase tables acquired in usual way will be expected to have gaps in their coverage in certain language pairs and that supplementing these with synthetic translation options is *a priori* preferable to alternative techniques, such as post processing, for generalizing beyond the translation pairs observable in training data (§2). As a case study, we consider the problem of producing English definite/indefinite articles (*the*, *a*, and *an*) when translating from Russian and Czech, two languages that lack overt definiteness morphemes (§3). We develop a classifier that predicts the presence and absence of English articles (§4). This classifier is used to generate synthetic translation options that are used to augment phrase tables used the usual way (§5). We evaluate their performance relative to post-processing approach and to a baseline phrase-based system, finding that synthetic translation options reliably outperform the other approaches (§6). We then discuss how our approach relates to previous work (§7) and conclude by discussing further applications of our technique (§8).

2 Why Synthetic Translation Options?

Before turning to the problem of generating English articles, we give arguments for why synthetic translation options are a useful extension of

Proceedings of the Eighth Workshop on Statistical Machine Translation, pages 271–280,
Sofia, Bulgaria, August 8-9, 2013

standard phrase-based translation approaches, and why this technique might be better than some alternative proposals that been made for generalizing beyond translation examples directly observable in the training data.

In language pairs that are typologically similar (i.e., when both languages lexicalize the same kinds of semantic and syntactic information), words and phrases map relatively directly from source to target languages, and the standard approach to learning phrase pairs is quite effective.[1] However, in language pairs in which individual source language words have many different possible translations (e.g., when the target language word could have many different inflections or could be surrounded by different function words that have no direct correspondence in the source language), we can expect the standard phrasal inventory to be incomplete, except when very large quantities of parallel data are available or for very frequent words. There simply will not be enough examples from which to learn the ideal set of translation options. Therefore, since phrase based translation can only generate input/output word pairs that were directly observed in the training corpus, the decoder's only hope for producing a good output is to find a fluent, meaning-preserving translation using incomplete translation lexicons. Synthetic translation option generation seeks to fill these gaps using secondary generation processes that produce possible phrase translation alternatives that are not directly extractable from the training data. We hypothesize that by filling in gaps in the translation options, discriminative translation models will be more effective (leading to better translation quality).

The creation of synthetic translation options can be understood as a kind of translation or post-editing of phrasal units/translations. This raises a question: if we have the ability to post-edit a phrasal translation or retranslate a source phrase so as to fill in gaps in the phrasal inventory, we should be able to use the same technique to translate the sentence; why not do this? While the effectiveness of this approach will ultimately be assessed empirically, translation option generation is appealing because the translation option synthesizer need not produce only single-best guesses—

[1]When translating from a language with a richer lexical inventory to a simpler one, approximate matching or backing off to (e.g.) morphologically simpler forms likewise reliably produces good translations.

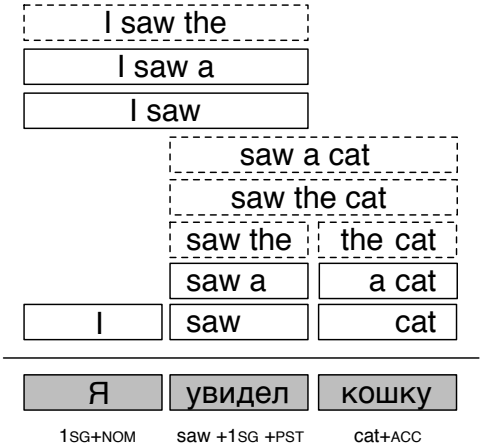

Figure 1: Russian-English phrase-based translation example. Since Russian lacks a definiteness morpheme the determiners *a*, *the* must be part of a translation option containing увидел or кошку in order to be present in the right place in the English output. Translation options that are in dashed boxes *should* exist but were not observed in the training data. This work seeks to produce such missing translation options *synthetically*.

if multiple possibilities appear to be equally good (say, multiple inflections of a translated lemma), then multiple translation options may be synthesized. Ultimately, of course, the global translation model must select one translation for every phrase it uses, but the decoder will have access to global information that it can use to pick better translation options.

3 Case Study: English Definite Articles

We now turn to a translation problem that we will use to assess the value of synthetic translation options: generating English in/definite articles when translating from Russian.

Definiteness is a semantic property of noun phrases that expresses information such as identifiability, specificity, familiarity and uniqueness (Lyons, 1999). In English, it is expressed through the use of article determiners and non-article determiners. Although languages may express definiteness through such morphemes, many languages use alternative mechanisms. For example they may use noncanonical word orders (Mohanan, 1994)[2] or different constructions such as existentials, differential object marking (Aissen, 2003), and the *ba* (吧) construction in Chinese

[2]See pp. 11–12 for an example in Hindi, a language without articles.

(Chen, 2004). While these languages lack articles, they may use demonstratives and the quantifier *one* to emphasize definiteness and indefiniteness, respectively.

Russian and Czech are examples of languages that use non-lexical means to express definiteness. As such, in Russian to English translation systems, we expect that most Russian nouns should have at least three translation options—the bare noun, the noun preceded by *the*, and the noun preceded *a/an*.

Fig. 1 illustrates how the definiteness mismatch between Russian and English can result in "gaps" in the phrasal inventory learned from a relatively large parallel corpus. The Russian input should translate (depending on context) as either *I saw a cat* or *I saw the cat*; however, the phrase table we learned is only able to generate the former.[3]

4 Predicting English Definite Articles

Although English articles express semantic content, their use is largely predictable in context, both for native English speakers and for automated systems (Knight and Chander, 1994).[4] In this section we describe a classifier that uses local contextual features to predict whether an article belongs in a particular position in a sequence of words, and if so, whether it is definite or indefinite (the form of the indefinite article is deterministic given the pronunciation of the following word).

4.1 Model

The classifier takes an English word sequence $\mathbf{w} = \langle w_1, w_2, \ldots, w_{|\mathbf{w}|} \rangle$ with missing articles and an index i and predicts whether no article, a definite article, or an indefinite article should appear before w_i. We parameterize the classifier as a multiclass logistic regression:

$$p(y \mid \mathbf{w}, i) \propto \exp \sum_j \lambda_j h_j(y, \mathbf{w}, i),$$

where $h_j(\cdot)$ are feature functions, λ_j are the corresponding weights, and $y \in \{\texttt{D}, \texttt{I}, \texttt{N}\}$ refer, respectively, to the outputs: definite article, indefinite article, and no article.[5]

4.2 Features

The English article system is extremely complex (as non-native English speakers will surely know!): in addition to a general placement rule that articles must precede a noun or its modifiers in an NP, multiple other factors can also affect article selection, including countability of the head noun, syntactic properties of an adjective modifying a noun (superlative, ordinal), discourse factors, general knowledge, etc. In this section, we define morphosyntactic features aimed at reflecting basic grammatical rules, we define statistical, semantic and shallow lexical features to capture additional regular and idiosyncratic usages of definite and indefinite articles in English. Below we provide brief details of the features and their motivation.

Lexical. Because training data can be constructed inexpensively (from any unannotated English corpus), n-gram indicator features, such as $[\![w_{i-1} y w_i w_{i+1} = \texttt{with } y \texttt{ lot of}]\!]$, can be estimated reliably and capture construction-specific article use.

Morphosyntactic. We used part-of-speech (POS) tags produced by the Stanford POS tagger (Toutanova and Manning, 2000) to capture general article patterns. These are relevant features in the prediction of articles as we observe certain constraints regarding the use of articles in the neighborhood of certain POS tags. For example, we do not expect to predict an article following an adjective (JJ).

Semantic. We extract further information indicating whether a named entity, as identified by the Stanford NE Recognizer (Finkel et al., 2005) begins at w_i. These features are relevant as there

[3] The phrase table for this example was extracted from the WMT 2013 shared task training data consisting of 1.2M sentence pairs.

[4] An interesting contribution of this work is a discussion on lower and upper bounds that can be achieved by native English speakers in predicting determiners. 67% is a lower bound, obtained by guessing *the* for every instance. The upper bound was obtained experimentally, and was measured on noun phrases (NP) without context, in a context of 4 words (2 before and 2 after NP), and given full context. Human subjects achieved an accuracy of 94-96% given full context, 83-88% for NPs in a context of 4 words, and 79-80% for NPs without context. Since in the current state-of-the-art building an automated determiners prediction in a full context (representing meaning computationally) is not a feasible task, we view 83-88% accuracy as our goal, and 88% as an upper bound for our method.

[5] Realization of the classes D and N as lexical items is straightforward. To convert I into *a* or *an*, we use the CMU pronouncing dictionary (http://www.speech.cs.cmu.edu/cgi-bin/cmudict) and select *an* if w_i starts with a phonetic vowel.

is, in general, a constraint on the co-occurrence of articles with named entities which can help us predict the use of articles in such constructions. For example, proper nouns do not tend to co-occur with articles in English. Although there are some proper nouns that have an article included in them, such as *the Netherlands*, *the United States of America*, but these are fixed expressions and the model is easily able to capture such cases with lexical features.

Statistical. Statistical features capture probability of co-occurrences of a sample with each of the determiner classes, e.g., for $w_{i-1}yw_i$ we collect probabilities of $w_{i-1}\texttt{I}w_i$, $w_{i-1}\texttt{D}w_i$, and $w_{i-1}\texttt{N}w_i$.[6]

4.3 Training and evaluation

We employ the `creg` regression modeling framework to train a ternary logistic regression classifier.[7] All features were computed for the target-side of the Russian-English TED corpus (Cettolo et al., 2012); from 117,527 sentences we removed 5K sentences used as tuning and test sets in the MT system. We extract statistical features from monolingual English corpora released for WMT-11 (Callison-Burch et al., 2011).

In the training corpus there are 65,075 `I` instances, 114,571 `D` instances, and 2,435,287 `N` instances. To create a balanced training set we randomly sample 65K instances from each set of collected instances.[8] This training set of feature vectors has 142,604 features and 285,210 parameters. To minimize the number of free parameters in our model we use ℓ_1 regularization. We perform 10-fold cross validation experiments with various feature combinations, evaluating the classifier accuracy for all classes and for each class independently. The performance of the classifier on individual classes and consolidated results for all classes are listed in Table 1.

We observe that morphosyntactic and lexical features are highly significant, reducing the error rate of statistical features by 25%. A combination of morphosyntactic, lexical, and statistical features is also helpful, reducing 13% more errors. Semantic features do not contribute to the classifier accuracy (we believe, mainly due to the feature sparsity).

Feature combination	All	I	D	N
Statistical	0.80	0.76	0.79	0.87
Lexical	0.82	0.79	0.80	0.87
Morphosyntactic	0.75	0.71	0.64	0.86
Semantic	0.35	0.99	0.02	0.04
Statistical+Lexical	0.85	0.83	0.82	0.89
+ Morphosyntactic	**0.87**	0.86	0.83	0.92
+ Semantic	0.87	0.86	0.83	0.92

Table 1: 10-fold cross validation accuracy of the classifier over all and by class.

5 Experimental Setup

Our experimental workflow includes the following steps. First, we select a phrase table PT_{source} from which we generate synthetic phrases. For each phrase pair $\langle f, e\rangle$ in PT_{source} we generate n synthetic variants of the target side phrase e which we then append to $PT_{baseline}$. We annotate both the original and synthetic phrases with additional translation features in $PT_{baseline}$.

For this language pair, we have several options for how to construct PT_{source}. The most straightforward way is to extract the phrasal inventory as usual; a second option is to extract phrases from training data from which definite articles have been removed (since we will rely on the classifier to reinsert them where they belong).

To synthesize phrases, we employ two different techniques: LM-based and classifier-based. We use a LM for one- or two-word phrases or an auxiliary classifier for longer phrases and create a new phrase in which we insert, remove or substitute an article between each adjacent pair of words in the original phrase. Such distinction between short and longer phrases has clear motivation: phrases without context may allow alternative, equally plausible options for article selection, therefore we can just rely on a LM, trained on large monolingual corpora, to identify phrases unobserved in MT training corpus. Longer context restricts determiners usage and statistical model decisions are less prone to generating ungrammatical synthetic phrases.

LM-based method is applied to phrases shorter than three words. These phrases are numerous, roughly 20% of a phrase table, and extracted from

[6] Although statistical features are *de rigeur* in NLP, they are arguably justified for this problem on linguistic grounds since human subjects use frequency-based in addition to their grammatical knowledge. For example, we say *He is at school* rather than *He is at the school*, but Americans say *He is in the hospital* while UK English speakers might prefer *He is in hospital*.

[7] https://github.com/redpony/creg

[8] Preliminary experiments indicated that the excess of `N` labels resulted in poor performance.

many sites in the training data. For each short (target) phrase we add all possible alternative entries observed in the LM and not observed in the original translation model. For example, for a short target phrase *a cat* we extract *the cat*.

We apply an auxiliary classifier to longer phrases, containing three or more words. Based on the classifier prediction, we use the maximally probable class to insert, remove or substitute an article between each adjacent pair of words in the original phrase. Synthetic phrases are generated by linguistically-informed features and can introduce alternative grammatically-correct translations of source phrases by adding or removing existing articles (since the English article selection in a local context is often ambiguous and not categorical). We add a synthetic phrase only if the phrase pair not observed in the original model.

We compare two possible applications of a classifier: one-pass and iterative prediction. With one-pass prediction we decide on the prediction for each position independently of other decisions. With iterative update we adopt the best first (greedy) strategy, selecting in each iteration the update-location in which the classifier obtains highest confidence score. In each iteration we incorporate a prediction in a target phrase, and in the next iteration the best first decision is made on an updated phrase. Iterative prediction stops when no updates are introduced.

Synthetic phrases are added to a phrase table with the five standard phrasal translation features that were found in the source phrase, and with several new features. First, we add a boolean feature indicating the origin of a phrase: synthetic or original. Second, we experiment with a posterior probability of a classifier averaged over all locations where it could be extracted from the training data. The next feature is derived from this score: it is a boolean feature indicating a confidence of the classifier: the feature value is 1 iff the average classifier score is higher than some threshold.

Consider again a phrase *I saw a cat* discussed in Section 1. Synthetic entry generation from the original phrase table entry is illustrated in Figure 2.

6 Translation Results

We now review the results of experiments using synthetic translation options in a machine translation system. We use the Moses toolkit (Koehn et al., 2007) to train a baseline phrase-based SMT system. Each configuration we compare has a different phrase table, with synthetic phrases generated with best-first or iterative strategies, from a phrase table with- or without-determiners, with variable number of translation features. To verify that system improvement is consistent, and is not a result of optimizer instability (Clark et al., 2011), we replicate each experimental setup three times, and then estimate the translation quality of the median MT system using the MultEval toolkit.[9]

The corpus is the same as in Section 4.3: the training part contains 112,527 sentences from Russian-English TED corpus, randomly sampled 3K sentences are used for tuning and a disjoint set of 2K sentences is used for test. We lowercase both sides, and use Stanford CoreNLP[10] tools to tokenize the corpora. We employ SRILM toolkit (Stolcke, 2002) to linearly interpolate the target side of the training corpus with the WMT English corpus, optimizing towards the MT tuning set. This LM is used in all experiments.

The rest of this section is organized as follows. First, we compare two approaches to the determiners classifier application. Then, we provide detailed description of experiments with synthetic phrases. We evaluate various aspects of synthetic phrases generation and summarize all the results in Table 3. In Table 5 we show examples of improved translations.

Classifier application: one-pass vs. iterative. First, as an intrinsic evaluation of the prediction strategy we remove definite and indefinite articles from the reference translations (2K test sentences) and then employ the determiners classifier to reproduce the original sentences. In Table 2 we report on the word error rate (WER) derived from the Levenshtein distance between the original sentences and the sentences (1) without articles, (2) with articles recovered using one-pass prediction, and (3) articles recovered using iterative prediction. The WER is averaged over all test sentences. Both one-pass and iterative approaches are effective in the task of determiners prediction, reducing the number of errors by 44%. The iterative approach yields slightly lower WER, hence we employ the iterative prediction in the future experiments with synthetic phrases.

[9] https://github.com/jhclark/multeval

[10] http://nlp.stanford.edu/software/corenlp.shtml

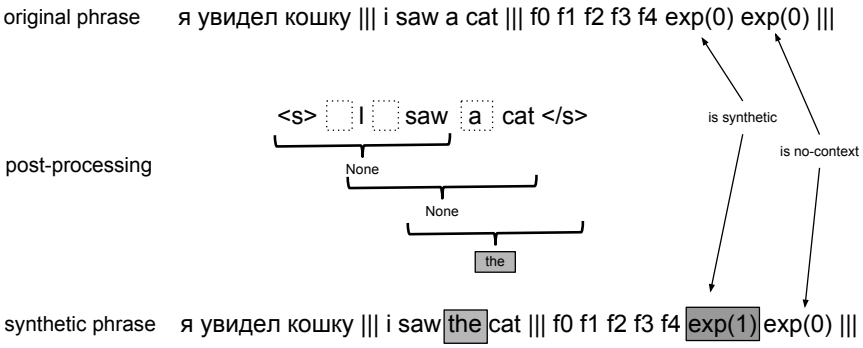

Figure 2: Synthetic entry generation example. The original parallel phrase has two additional boolean features (set to false) indicating that this is not a synthetic phrase and not a short phrase. We apply our determiners classifier to predict an article at each location marked with a dashed box. Based on a classifier prediction we derive a new phrase *I saw the cat*. Since corresponding parallel entry is not in the original phrase table, we set the synthetic indicator feature to 1.

Post-processing	**WER**
None	5.6%
One-pass	3.2%
Iterative	3.1%

Table 2: WER (lower is better) of reference translations without articles and of post-processed reference translations. Both one-pass and iterative approaches are effective in the task of determiners prediction.

MT output post-processing. We then evaluate the post-processing strategy directly on the MT output. We experiment with one-pass and iterative post-processing of two variants of the baseline system outputs: original output and the output without articles (we remove the articles prior to post-processing). The results are listed in Table 3. Interestingly, we do not obtain any improvements applying the determiners classifier in a conventional way of a MT output post-processing. It is the combination of linguistically-motivated features with synthetic phrases that contribute to the best performance.

LM-based synthetic phrases. As discussed above, LM-based (short) phrases are shorter than 3 tokens and their synthetic variants contain same words with articles inserted or deleted between each adjacent pair of words. The phrase table of the baseline system contains 2,441,678 phrase pairs. There are 518,453 original short phrases, and our technique yields 842,252 new synthetic entries which we append to the baseline phrase table. Table 3 shows the evaluation of the median SMT system (derived from three systems) with short phrases. In these systems the five phrasal translation features are the same as in the baseline systems. Improvement in the BLEU score (Papineni et al., 2002) is statistically significant ($p < .05$), compared to the baseline system

Classifier-generated synthetic phrases We apply classifier with the iterative prediction directly on the baseline phrase table entries and synthesize 944,145 new parallel phrases, increasing the phrase table size by 38%. The phrasal translation features in each synthetic phrase are the same as in the phrase it was derived from. The BLEU score of the median SMT system with synthetic phrases is $22.9 \pm .1$, the improvement is statistically significant ($p < .01$). Post-processing of a phrase table created from corpora without articles and adding synthetic phrases to the baseline phrase table yielded similar results.

Translation features for synthetic phrases In the following experiments we aim to establish the optimal set of translation features that should be used with synthetic phrases. We train several SMT systems, each containing synthetic phrases derived from the original phrase table by iterative classification, and with LM-based short phrases. Each synthetic phrase has five translation features as an original phrase it was derived from. The additional features that we evaluate are:

1. Boolean feature for LM-based synthetic phrases

MT System	BLEU
Baseline	$22.6 \pm .1$
MT output post-processing	
one-pass, MT output with articles	20.8
one-pass, MT output without articles	19.7
iterative, MT output with articles	22.6
iterative, MT output without articles	21.8
With synthetic phrases	
LM-based phrases	$22.9 \pm .1$
+ classifier-generated phrases	$22.9 \pm .1$
+ features 1,2	**23.0** $\pm .1$
+ features 1,2,3	$22.8 \pm .1$
+ features 1,2,3,4	$22.8 \pm .1$
+ feature 5	$22.9 \pm .1$

Table 3: Summary of experiments with MT output post-processing and with synthetic translation options in a phrase table. Post-processing of the MT output do not improve translations. Best performing system with synthetic phrases has five original phrase translation features and two additional boolean features indicating if the phrase is LM-based or not, is classifier-generated or not. All the synthetic systems are significantly better than the baseline system.

2. Boolean feature for classifier-generated synthetic phrases
3. Classifier confidence: posterior probability of the classifier averaged over all samples in a target phrase.
4. Boolean feature indicating a confidence of the classifier: the feature value is 1 iff the Feature 3 scores higher than some threshold. The threshold was set to 0.8, we did not experiment with other values.
5. Boolean feature for a synthetic phrase of any type: LM-based or classifier-generated

Table 3 details the change in the BLEU score of each experimental setup. The best performing system has five original phrase translation features and two additional boolean features indicating if the phrase is LM-based or not, is classifier-generated or not. Note that all the synthetic systems are significantly better than the baseline.

Czech-English. Our technique was developed using Russian-English system in the TED domain, so we want to see how our method generalizes to a different domain when translating from a different language. We therefore applied our most successful configuration to a Czech-English news translation task.[11] For training, we use the WMT Czech-English parallel corpus CzEng0.7; we tune using the WMT2011 test set and test on the WMT2012 test set. The LM is trained on the target side of the training corpus. Determiners classifier, re-trained on the English side of this corpus, with statistical, lexical, morphosyntactic and dependency features obtained an accuracy of 88%.

In Table 4, we report the results of evaluating the performance of the Russian-to-English and Czech-to-English MT systems with synthetic phrases. The results of both systems show a statistically significant ($p < .01$) improvement in terms of BLEU score.

	Russian	Czech
Baseline	$22.6 \pm .1$	$16.0 \pm .05$
Synthetic	**23.0** $\pm .1$	**16.2** $\pm .03$

Table 4: BLEU score of Russian-to-English and Czech-to-English MT systems with synthetic phrases and features 1 and 2 show a significant improvement.

Qualitative analysis. Table 5 shows some examples from the output of our Russian-to-English systems. Although both systems produce comprehensible translations, the system augmented with determiner classifier is more fluent. The first example represents a case where a singular count noun (*piece*) is present which requires an article. The baseline is not able to identify this requirement and hence does not insert the article *an* before the phrase *extraordinary engineering piece*. Our system, however, correctly identifies the construction requiring an article and thus provides an appropriate form of the article (an- Indefinite article for lexical items beginning with a vowel). Thus we see that our system is able to capture the linguistic requirement of the singular count nouns to co-occur with an article. In the second row, the lexical item *poor* is used as an adjective. The baseline has inserted an article in front of it, changing it to a noun. Our system, however, is able to maintain the status of *poor* as an adjective since it has the option not to insert an article. Thus we see that besides fluency, our system also does better in maintaining the grammatical category of a lexical item. In the third row, the phrase *three*

[11] Like Russian, Czech is a Slavic language that does not have definite or indefinite articles.

Source:	но тем не менее , это выдающееся произведение инженерного искусства .
Reference:	but nonetheless , it 's an extraordinary piece of engineering .
Baseline:	but nevertheless , it 's extraordinary engineering piece of art .
Ours:	but nevertheless , it 's an extraordinary piece of engineering art .
Source:	и по многим дефинициям она уже не бедная .
Reference:	and by many definitions she is no longer poor .
Baseline:	and in a lot definitions , it 's not a poor .
Ours:	and in a lot definitions she 's not poor .
Source:	нам нужно накормить три миллиарда городских жителей .
Reference:	we must feed three billion people in cities .
Baseline:	we need to feed the three billion urban hundreds of them .
Ours:	we need to feed three billion people in the city .

Table 5: Examples of translations with improved articles handling.

billion people refers to a nonidentifiable referent. The baseline inserts the definite article *the*. If a human subject reads this translation, it would mislead him/her to interpret the object *three billion people* as referring to a specific identifiable set. Our system, on the other hand, correctly selects the determiner class N and hence does not insert an article. Thus we see that our system does not just add fluency but it also captures a semantic distinction, namely **identifiability**, that a human subject makes when producing or interpreting a phrase.

7 Related Work

Automated determiner prediction has been found beneficial in a variety of applications, including postediting of MT output (Knight and Chander, 1994), text generation (Elhadad, 1993; Minnen et al., 2000), and more recently identification and correction of ESL errors (Han et al., 2006; De Felice and Pulman, 2008; Gamon et al., 2009; Rozovskaya and Roth, 2010). Our work on determiners extends previous studies in several dimensions. While all previous approaches were tested only on NP constructions, we evaluate our classifier on any sequence of tokens.

To the best of our knowledge, the only studies that directly address generation of synthetic phrase table entries was conducted by Chen et al. (2011) and Koehn and Hoang (2007). The former find semantically similar source phrases and produce "fabricated" translations by combining these source phrases with a set of their target phrases; however, they do not observe improvements. The later work integrates the synthesis of translation options into the decoder. While related in spirit, their method only supports a limited set of generative processes for producing the candidate set (lacking, for instance, the simple and effective phrase post-editing process we have used), and their implementation has been plagued by computational challenges.

Post-processing techniques have been extremely popular. These can be understood as using a translation model to generate a translation skeleton (or k-best skeletons) and then post-editing these in various ways. These have been applied to translation into morphologically rich languages, such as Japanese, German, Turkish, and Finnish (de Gispert et al., 2005; Suzuki and Toutanova, 2006; Suzuki and Toutanova, 2007; Fraser et al., 2012; Clifton and Sarkar, 2011; Oflazer and Durgar El-Kahlout, 2007).

8 Conclusions and future work

The contribution of this work is twofold. First, we propose a new supervised method to predict definite and indefinite articles. Our log-linear model trained on a linguistically-motivated set of features outperforms previously reported results, and obtains an upper bound of an accuracy achieved by human subjects given a context of four words. However, more important result of this work is the experimentally verified idea of improving phrase-based SMT via synthetic phrases. While we have focused on a limited problem in this paper, there are numerous alternative applications including translation into morphologically rich languages, as a method for incorporating (source) contextual information in making local translation decisions, enriching the target language lexicon using lexical translation resources, and many others.

Acknowledgments

We are grateful to Shuly Wintner for insightful suggestions and support. This 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.

References

J. Aissen. 2003. Differential object marking: Iconicity vs. economy. *Natural Language and Linguistic Theory*, 21(3):435–483.

C. Callison-Burch, P. Koehn, C. Monz, and O. Zaidan. 2011. Findings of the 2011 workshop on statistical machine translation. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 22–64, Edinburgh, Scotland, July. Association for Computational Linguistics.

M. Cettolo, C. Girardi, and M. Federico. 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, May.

B. Chen, R. Kuhn, and G. Foster. 2011. Semantic smoothing and fabrication of phrase pairs for SMT. In *Proceedings of the International Workshop on Spoken Lanuage Translation (IWSLT-2011)*.

P. Chen. 2004. Identifiability and definiteness in chinese. *Linguistics*, 42(6):1129–1184.

C. Cherry and G. Foster. 2012. Batch tuning strategies for statistical machine translation. In *Proceedings of HLT-NAACL 2012*, volume 12, pages 34–35.

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

D. Chiang. 2012. Hope and fear for discriminative training of statistical translation models. *The Journal of Machine Learning Research*, 98888:1159–1187.

J. H. Clark, C. Dyer, A. Lavie, and N. A. Smith. 2011. Better hypothesis testing for statistical machine translation: Controlling for optimizer instability. In *In Proc. of ACL*.

A. Clifton and A. Sarkar. 2011. Combining morpheme-based machine translation with post-processing morpheme prediction. In *Proceedings of ACL*.

R. De Felice and S. G. Pulman. 2008. A classifier-based approach to preposition and determiner error correction in L2 English. In *Proceedings of the 22nd International Conference on Computational Linguistics-Volume 1*, pages 169–176. Association for Computational Linguistics.

A. de Gispert, J. B. Mariño, and J. M. Crego. 2005. Improving statistical machine translation by classifying and generalizing inflected verb forms. In *Proceedings of InterSpeech*.

J. DeNero and D. Klein. 2010. Discriminative modeling of extraction sets for machine translation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 1453–1463. Association for Computational Linguistics.

J. DeNero, A. Bouchard-Côté, and D. Klein. 2008. Sampling alignment structure under a Bayesian translation model. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 314–323. Association for Computational Linguistics.

V. Eidelman, Y. Marton, and P. Resnik. 2013. Online relative margin maximization for statistical machine translation. In *Proceedings of ACL*.

M. Elhadad. 1993. Generating argumentative judgment determiners. In *AAAI*, pages 344–349.

J. R. Finkel, T. Grenager, and C. Manning. 2005. Incorporating non-local information into information extraction systems by gibbs sampling. In *ACL '05: Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 363–370, Morristown, NJ, USA. Association for Computational Linguistics.

A. Fraser, M. Weller, A. Cahill, and F. Cap. 2012. Modeling inflection and word-formation in SMT. In *Proceedings of EACL*.

M. Galley, J. Graehl, K. Knight, D. Marcu, S. DeNeefe, W. Wang, and I. 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. Association for Computational Linguistics.

M. Gamon, J. Gao, C. Brockett, A. Klementiev, W. B. Dolan, D. Belenko, and L. Vanderwende. 2009. Using contextual speller techniques and language modeling for ESL error correction. *Urbana*, 51:61801.

K. Gimpel and N. A. Smith. 2012. Structured ramp loss minimization for machine translation. In *Proceedings of 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies HLT-NAACL 2012, Montreal, Canada*.

N.-R. Han, M. Chodorow, and C. Leacock. 2006. Detecting errors in English article usage by non-native speakers.

K. Knight and I. Chander. 1994. Automated postediting of documents. In *Proceedings of the National Conference on Artificial Intelligence*, pages 779–779, Seattle, WA.

P. Koehn and H. Hoang. 2007. Factored translation models. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pages 868–876, Prague, Czech Republic, June. Association for Computational Linguistics.

P. Koehn, F. J. Och, and D. 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. Association for Computational Linguistics.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, B. Cowan, W. Shen, C. Moran, R. Zens, C. Dyer, O. Bojar, A. Constantin, and E. Herbst. 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, June. Association for Computational Linguistics.

C. Lyons. 1999. *Definiteness*. Cambridge University Press.

G. Minnen, F. Bond, and A. Copestake. 2000. Memory-based learning for article generation. In *Proceedings of the 2nd workshop on Learning language in logic and the 4th conference on Computational natural language learning-Volume 7*, pages 43–48. Association for Computational Linguistics.

T. Mohanan. 1994. *Argument Structure in Hindi*. CSLI Publications.

K. Oflazer and I. Durgar El-Kahlout. 2007. Exploring different representational units in English-to-Turkish statistical machine translation. In *Proceedings of the Second Workshop on Statistical Machine Translation*, pages 25–32, Prague, Czech Republic, June. Association for Computational Linguistics.

K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL '02: Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, pages 311–318, Morristown, NJ, USA. Association for Computational Linguistics.

A. Rozovskaya and D. Roth. 2010. Training paradigms for correcting errors in grammar and usage. *Urbana*, 51:61801.

A. Stolcke. 2002. SRILM—an extensible language modeling toolkit. In *Procedings of International Conference on Spoken Language Processing*, pages 901–904.

H. Suzuki and K. Toutanova. 2006. Learning to predict case markers in Japanese. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 1049–1056. Association for Computational Linguistics.

H. Suzuki and K. Toutanova. 2007. Generating case markers in machine translation. In *Proceedings of HLT-NAACL 2007*, pages 49–56.

K. Toutanova and C. D. Manning. 2000. Enriching the knowledge sources used in a maximum entropy part-of-speech tagger. In *Proceedings of the 2000 Joint SIGDAT conference on Empirical methods in natural language processing and very large corpora*, pages 63–70, Morristown, NJ, USA. Association for Computational Linguistics.