A Class-Based Agreement Model for Generating Accurately Inflected Translations

Spence Green
Computer Science Department, Stanford University
spenceg@stanford.edu

John DeNero
Google
denero@google.com

Abstract

When automatically translating from a weakly inflected source language like English to a target language with richer grammatical features such as gender and dual number, the output commonly contains morpho-syntactic agreement errors. To address this issue, we present a target-side, class-based agreement model. Agreement is promoted by scoring a sequence of fine-grained morpho-syntactic classes that are predicted during decoding for each translation hypothesis. For English-to-Arabic translation, our model yields a +1.04 BLEU average improvement over a state-of-the-art baseline. The model does not require bitext or phrase table annotations and can be easily implemented as a feature in many phrase-based decoders.

1 Introduction

Languages vary in the degree to which surface forms reflect grammatical relations. English is a weakly inflected language: it has a narrow verbal paradigm, restricted nominal inflection (plurals), and only the vestiges of a case system. Consequently, translation *into* English—which accounts for much of the machine translation (MT) literature (Lopez, 2008)—often involves some amount of morpho-syntactic dimensionality reduction. Less attention has been paid to what happens during translation *from* English: richer grammatical features such as gender, dual number, and overt case are effectively latent variables that must be inferred during decoding. Consider the output of Google Translate for the simple English sentence in Fig. 1. The correct translation is a monotone mapping of the input. However, in Arabic, SVO word order requires both gender and number agreement between the subject السيارة 'the car' and verb يذهب 'go'. The MT system selects the correct verb stem, but with masculine inflection. Although the translation has the correct semantics, it is ultimately ungrammatical. This paper addresses the problem of generating text that conforms to morpho-syntactic agreement rules.

(1) السيارة يذهب بسرعة
the-car$_{\text{SG.DEF.FEM}}$ go$_{\text{SG.MASC}}$ with-speed$_{\text{SG.FEM}}$
The car goes quickly

Figure 1: Ungrammatical Arabic output of Google Translate for the English input *The car goes quickly*. The subject should agree with the verb in both gender and number, but the verb has masculine inflection. For clarity, the Arabic tokens are arranged left-to-right.

Agreement relations that cross statistical phrase boundaries are not explicitly modeled in most phrase-based MT systems (Avramidis and Koehn, 2008). We address this shortcoming with an agreement model that scores sequences of fine-grained morpho-syntactic *classes*. First, bound morphemes in translation hypotheses are segmented. Next, the segments are labeled with classes that encode both syntactic category information (i.e., parts of speech) and grammatical features such as number and gender. Finally, agreement is promoted by scoring the predicted class sequences with a generative Markov model.

Our model scores hypotheses *during decoding*. Unlike previous models for scoring syntactic relations, our model does not require bitext annotations, phrase table features, or decoder modifications. The model can be implemented using the feature APIs of popular phrase-based decoders such as Moses (Koehn et al., 2007) and Phrasal (Cer et al., 2010).

Intuition might suggest that the standard n-gram language model (LM) is sufficient to handle agreement phenomena. However, LM statistics are sparse, and they are made sparser by morphological variation. For English-to-Arabic translation, we achieve a +1.04 BLEU average improvement by tiling our model on top of a large LM.

Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics, pages 146–155,
Jeju, Republic of Korea, 8-14 July 2012.

It has also been suggested that this setting requires morphological generation because the bitext may not contain all inflected variants (Minkov et al., 2007; Toutanova et al., 2008; Fraser et al., 2012). However, using lexical coverage experiments, we show that there is ample room for translation quality improvements through better *selection* of forms that already exist in the translation model.

2 A Class-based Model of Agreement

2.1 Morpho-syntactic Agreement

Morpho-syntactic agreement refers to a relationship between two sentence elements a and b that must have at least one matching grammatical feature.[1] Agreement relations tend to be defined for particular syntactic configurations such as verb-subject, noun-adjective, and pronoun-antecedent. In some languages, agreement affects the surface forms of the words. For example, from the perspective of generative grammatical theory, the lexicon entry for the Arabic nominal السيارة 'the car' contains a feminine gender feature. When this nominal appears in the subject argument position, the verb-subject agreement relationship triggers feminine inflection of the verb.

Our model treats agreement as a sequence of scored, pairwise relations between adjacent words. Of course, this assumption excludes some agreement phenomena, but it is sufficient for many common cases. We focus on English-Arabic translation as an example of a translation direction that expresses substantially more morphological information in the target. These relations are best captured in a target-side model because they are mostly unobserved (from lexical clues) in the English source.

The agreement model scores sequences of morpho-syntactic word classes, which express grammatical features relevant to agreement. The model has three components: a segmenter, a tagger, and a scorer.

2.2 Morphological Segmentation

Segmentation is a procedure for converting raw surface forms to component morphemes. In some languages, agreement relations exist between *bound morphemes*, which are syntactically independent yet phonologically dependent morphemes. For example,

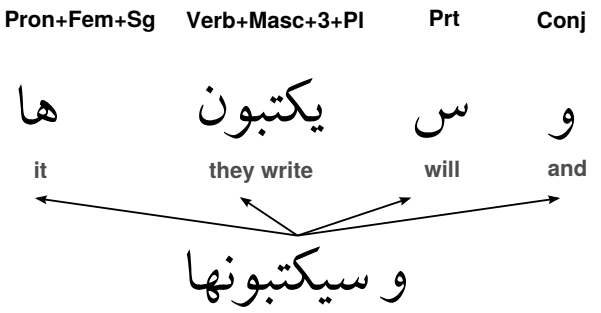

Figure 2: Segmentation and tagging of the Arabic token وسيكتبونها 'and they will write it'. This token has four segments with conflicting grammatical features. For example, the number feature is singular for the pronominal object and plural for the verb. Our model segments the raw token, tags each segment with a morpho-syntactic class (e.g., "Pron+Fem+Sg"), and then scores the class sequences.

the single raw token in Fig. 2 contains at least four grammatically independent morphemes. Because the morphemes bear conflicting grammatical features and basic parts of speech (POS), we need to segment the token before we can evaluate agreement relations.[2]

Segmentation is typically applied as a bitext preprocessing step, and there is a rich literature on the effect of different segmentation schemata on translation quality (Koehn and Knight, 2003; Habash and Sadat, 2006; El Kholy and Habash, 2012). Unlike previous work, we segment each translation hypothesis as it is generated (i.e., during decoding). This permits greater modeling flexibility. For example, it may be useful to count tokens with bound morphemes as a unit during phrase extraction, but to score segmented morphemes separately for agreement.

We treat segmentation as a character-level sequence modeling problem and train a linear-chain conditional random field (CRF) model (Lafferty et al., 2001). As a pre-processing step, we group contiguous non-native characters (e.g., Latin characters in Arabic text). The model assigns four labels:

- **I**: Continuation of a morpheme
- **O**: Outside morpheme (whitespace)
- **B**: Beginning of a morpheme
- **F**: Non-native character(s)

[1] We use *morpho-syntactic* and *grammatical* agreement interchangeably, as is common in the literature.

[2] Segmentation also improves translation of compounding languages such as German (Dyer, 2009) and Finnish (Macherey et al., 2011).

Translation Model	
e	Target sequence of I words
f	Source sequence of J words
a	Sequence of K phrase alignments for $\langle e, f \rangle$
Π	Permutation of the alignments for target word order e
h	Sequence of M feature functions
λ	Sequence of learned weights for the M features
H	A priority queue of hypotheses
Class-based Agreement Model	
$t \in T$	Set of morpho-syntactic classes
$s \in S$	Set of all word segments
θ_{seg}	Learned weights for the CRF-based segmenter
θ_{tag}	Learned weights for the CRF-based tagger
ϕ_o, ϕ_t	CRF potential functions (emission and transition)
τ	Sequence of I target-side predicted classes
π	T dimensional (log) prior distribution over classes
$\hat{s}$	Sequence of l word segments
σ	Model state: a tagged segment $\langle s, t \rangle$

Figure 3: Notation used in this paper. The convention e_i^I indicates a subsequence of a length I sequence.

The features are indicators for (character, position, label) triples for a five character window and bigram label transition indicators.

This formulation is inspired by the classic "IOB" text chunking model (Ramshaw and Marcus, 1995), which has been previously applied to Chinese segmentation (Peng et al., 2004). It can be learned from gold-segmented data, generally applies to languages with bound morphemes, and does not require a hand-compiled lexicon.[3] Moreover, it has only four labels, so Viterbi decoding is very fast. We learn the parameters θ_{seg} using a quasi-Newton (QN) procedure with l_1 (lasso) regularization (Andrew and Gao, 2007).

2.3 Morpho-syntactic Tagging

After segmentation, we tag each segment with a fine-grained morpho-syntactic class. For this task we also train a standard CRF model on full sentences with gold classes and segmentation. We use the same QN procedure as before to obtain θ_{tag}.

A translation derivation is a tuple $\langle e, f, a \rangle$ where e is the target, f is the source, and a is an alignment between the two. The CRF tagging model predicts a target-side class sequence τ^*

$$\tau^* = \arg\max_{\tau} \sum_{i=1}^{I} \theta_{tag} \cdot \{\phi_o(\tau_i, i, e) + \phi_t(\tau_i, \tau_{i-1})\}$$

where further notation is defined in Fig. 3.

[3]Mada, the standard tool for Arabic segmentation (Habash and Rambow, 2005), relies on a manually compiled lexicon.

Set of Classes The tagger assigns morpho-syntactic classes, which are coarse POS categories refined with grammatical features such as gender and definiteness. The coarse categories are the universal POS tag set described by Petrov et al. (2012). More than 25 treebanks (in 22 languages) can be automatically mapped to this tag set, which includes "Noun" (nominals), "Verb" (verbs), "Adj" (adjectives), and "ADP" (pre- and post-positions). Many of these treebanks also contain per-token morphological annotations. It is easy to combine the coarse categories with selected grammatical annotations.

For Arabic, we used the coarse POS tags plus definiteness and the so-called phi features (gender, number, and person).[4] For example, السيارة 'the car' would be tagged "Noun+Def+Sg+Fem". We restricted the set of classes to observed combinations in the training data, so the model implicitly disallows incoherent classes like "Verb+Def".

Features The tagging CRF includes emission features ϕ_o that indicate a class τ_i appearing with various orthographic characteristics of the word sequence being tagged. In typical CRF inference, the entire observation sequence is available throughout inference, so these features can be scored on observed words in an arbitrary neighborhood around the current position i. However, we conduct CRF inference in tandem with the translation decoding procedure (§3), creating an environment in which subsequent words of the observation are not available; the MT system has yet to generate the rest of the translation when the tagging features for a position are scored. Therefore, we only define emission features on the observed words at the current and previous positions of a class: $\phi_o(\tau_i, e_i, e_{i-1})$.

The emission features are word types, prefixes and suffixes of up to three characters, and indicators for digits and punctuation. None of these features are language specific.

Bigram transition features ϕ_t encode local agreement relations. For example, the model learns that the Arabic class "Noun+Fem" is followed by "Adj+Fem" and not "Adj+Masc" (noun-adjective gender agreement).

[4]Case is also relevant to agreement in Arabic, but it is mostly indicated by diacritics, which are absent in unvocalized text.

2.4 Word Class Sequence Scoring

The CRF tagger model defines a conditional distribution $p(\tau|e;\theta_{tag})$ for a class sequence τ given a sentence e and model parameters θ_{tag}. That is, the sample space is over class—not word—sequences. However, in MT, we seek a measure of sentence quality $q(e)$ that is comparable *across* different hypotheses on the beam (much like the n-gram language model score). Discriminative model scores have been used as MT features (Galley and Manning, 2009), but we obtained better results by scoring the 1-best class sequences with a generative model. We trained a simple add-1 smoothed bigram language model over gold class sequences in the same treebank training data:

$$q(e) = p(\tau) = \prod_{i=1}^{I} p(\tau_i|\tau_{i-1})$$

We chose a bigram model due to the aggressive recombination strategy in our phrase-based decoder. For contexts in which the LM is guaranteed to back off (for instance, after an unseen bigram), our decoder maintains only the minimal state needed (perhaps only a single word). In less restrictive decoders, higher order scoring models could be used to score longer-distance agreement relations.

We integrate the segmentation, tagging, and scoring models into a self-contained component in the translation decoder.

3 Inference during Translation Decoding

Scoring the agreement model as part of translation decoding requires a novel inference procedure. Crucially, the inference procedure does not measurably affect total MT decoding time.

3.1 Phrase-based Translation Decoding

We consider the standard phrase-based approach to MT (Och and Ney, 2004). The distribution $p(e|f)$ is modeled directly using a log-linear model, yielding the following decision rule:

$$e^* = \arg\max_{e,a,\Pi} \left\{ \sum_{m=1}^{M} \lambda_m h_m(e, f, a, \Pi) \right\} \quad (1)$$

This decoding problem is NP-hard, thus a beam search is often used (Fig. 4). The beam search relies on three operations, two of which affect the agreement model:

Input: implicitly defined search space
  generate initial hypotheses and add to H
  set H_{final} to $\emptyset$
  while H is not empty:
    set H_{ext} to $\emptyset$
    for each hypothesis η in H:
      if η is a goal hypothesis:
        add η to H_{final}
      else Extend η and add to H_{ext} ▶Score agreement
    Recombine and Prune H_{ext}
    set H to H_{ext}
Output: argmax of H_{final}

Figure 4: Breadth-first beam search algorithm of Och and Ney (2004). Typically, a hypothesis stack H is maintained for each unique source coverage set.

Input: $(e_1^I,\ n,$ is_goal)
  run segmenter on attachment e_{n+1}^I to get $\hat{s}_1^L$
  get model state $\sigma = \langle s, t \rangle$ for translation prefix e_1^n
  initialize π to $-\infty$
  set $\pi(t) = 0$
  compute τ^* from parameters $\langle s,\ \hat{s}_1^L,\ \pi,$ is_goal$\rangle$
  compute $q(e_{n+1}^I) = p(\tau^*)$ under the generative LM
  set model state $\sigma_{new} = \langle \hat{s}_L, \tau_L^* \rangle$ for prefix e_1^I
Output: $q(e_{n+1}^I)$

Figure 5: Procedure for scoring agreement for each hypothesis generated during the search algorithm of Fig. 4. In the extended hypothesis e_1^I, the index $n+1$ indicates the start of the new attachment.

- Extend a hypothesis with a new phrase pair
- Recombine hypotheses with identical states

We assume familiarity with these operations, which are described in detail in (Och and Ney, 2004).

3.2 Agreement Model Inference

The class-based agreement model is implemented as a feature function h_m in Eq. (1). Specifically, when Extend generates a new hypothesis, we run the algorithm shown in Fig. 5. The inputs are a translation hypothesis e_1^I, an index n distinguishing the *prefix* from the *attachment*, and a flag indicating if their concatenation is a goal hypothesis.

The beam search maintains state for each derivation, the score of which is a linear combination of the feature values. States in this program depend on some amount of lexical history. With a trigram language model, the state might be the last two words of the translation prefix. Recombine can be applied to any two hypotheses with equivalent states. As a

result, two hypotheses with different full prefixes—and thus potentially different sequences of agreement relations—can be recombined.

Incremental Greedy Decoding Decoding with the CRF-based tagger model in this setting requires some slight modifications to the Viterbi algorithm. We make a greedy approximation that permits recombination and works well in practice. The agreement model state is the *last tagged segment* $\langle s, t \rangle$ of the concatenated hypothesis. We tag a new attachment by assuming a prior distribution π over the starting position such that $\pi(t) = 0$ and $-\infty$ for all other classes, a deterministic distribution in the tropical semiring. This forces the Viterbi path to go through t. We only tag the final boundary symbol for goal hypotheses.

To accelerate tagger decoding in our experiments, we also used tagging dictionaries for frequently observed word types. For each word type observed more than 100 times in the training data, we restricted the set of possible classes to the set of observed classes.

3.3 Translation Model Features

The agreement model score is one decoder feature function. The output of the procedure in Fig. 5 is the log probability of the class sequence of each attachment. Summed over all attachments, this gives the log probability of the whole class sequence.

We also add a new length penalty feature. To discriminate between hypotheses that might have the same number of raw tokens, but different underlying segmentations, we add a penalty equal to the length difference between the segmented and unsegmented attachments $|\hat{s}_1^L| - |e_{n+1}^I|$.

4 Related Work

We compare our class-based model to previous approaches to scoring syntactic relations in MT.

Unification-based Formalisms Agreement rules impose syntactic and semantic constraints on the structure of sentences. A principled way to model these constraints is with a unification-based grammar (UBG). Johnson (2003) presented algorithms for learning and parsing with stochastic UBGs. However, training data for these formalisms remains extremely limited, and it is unclear how to learn such knowledge-rich representations from unlabeled data. One partial solution is to manually extract unification rules from phrase-structure trees. Williams and Koehn (2011) annotated German trees, and extracted translation rules from them. They then specified manual unification rules, and applied a penalty according to the number of unification failures in a hypothesis. In contrast, our class-based model does not require any manual rules and scores similar agreement phenomena as probabilistic sequences.

Factored Translation Models Factored translation models (Koehn and Hoang, 2007) facilitate a more data-oriented approach to agreement modeling. Words are represented as a vector of features such as lemma and POS. The bitext is annotated with separate models, and the annotations are saved during phrase extraction. Hassan et al. (2007) noticed that the target-side POS sequences could be scored, much as we do in this work. They used a target-side LM over Combinatorial Categorial Grammar (CCG) supertags, along with a penalty for the number of operator violations, and also modified the phrase probabilities based on the tags. However, Birch et al. (2007) showed that this approach captures the same re-ordering phenomena as lexicalized re-ordering models, which were not included in the baseline. Birch et al. (2007) then investigated source-side CCG supertag features, but did not show an improvement for Dutch-English.

Subotin (2011) recently extended factored translation models to hierarchical phrase-based translation and developed a discriminative model for predicting target-side morphology in English-Czech. His model benefited from gold morphological annotations on the target-side of the 8M sentence bitext.

In contrast to these methods, our model does not affect phrase extraction and does not require annotated translation rules.

Class-based LMs Class-based LMs (Brown et al., 1992) reduce lexical sparsity by placing words in equivalence classes. They have been widely used for speech recognition, but not for MT. Och (1999) showed a method for inducing bilingual word classes that placed each phrase pair into a two-dimensional equivalence class. To our knowledge, Uszkoreit and Brants (2008) are the only recent authors to show an improvement in a state-of-the-art MT system using class-based LMs. They used a classical exchange algorithm for clustering, and learned 512 classes from

a large monolingual corpus. Then they mixed the classes into a word-based LM. However, both Och (1999) and Uszkoreit and Brants (2008) relied on automatically induced classes. It is unclear if their classes captured agreement information.

Monz (2011) recently investigated parameter estimation for POS-based language models, but his classes did not include inflectional features.

Target-Side Syntactic LMs Our agreement model is a form of syntactic LM, of which there is a long history of research, especially in speech processing.[5] Syntactic LMs have traditionally been too slow for scoring during MT decoding. One exception was the quadratic-time dependency language model presented by Galley and Manning (2009). They applied a quadratic time dependency parser to every hypothesis during decoding. However, to achieve quadratic running time, they permitted ill-formed trees (e.g., parses with multiple roots). More recently, Schwartz et al. (2011) integrated a right-corner, incremental parser into Moses. They showed a large improvement for Urdu-English, but decoding slowed by three orders of magnitude.[6] In contrast, our class-based model encodes shallow syntactic information without a noticeable effect on decoding time.

Our model can be viewed as a way to score local syntactic relations without extensive decoder modifications. For long-distance relations, Shen et al. (2010) proposed a new decoder that generates target-side dependency trees. The target-side structure enables scoring hypotheses with a trigram dependency LM.

5 Experiments

We first evaluate the Arabic segmenter and tagger components independently, then provide English-Arabic translation quality results.

5.1 Intrinsic Evaluation of Components

Experimental Setup All experiments use the Penn Arabic Treebank (ATB) (Maamouri et al., 2004) parts 1–3 divided into training/dev/test sections according to the canonical split (Rambow et al., 2005).[7]

[5] See (Zhang, 2009) for a comprehensive survey.

[6] In principle, their parser should run in linear time. An implementation issue may account for the decoding slowdown. (*p.c.*)

[7] LDC catalog numbers: LDC2008E61 (ATBp1v4), LDC2008E62 (ATBp2v3), and LDC2008E22 (ATBp3v3.1). The data was pre-processed with packages from the Stanford Arabic parser (Green and Manning, 2010). The corpus split is available at http://nlp.stanford.edu/projects/arabic.shtml.

	FULL (%)	INCREMENTAL (%)
Segmenter	98.6	–
Tagger	96.3	96.2

Table 1: Intrinsic evaluation accuracy [%] (development set) for Arabic segmentation and tagging.

The ATB contains clitic-segmented text with *per-segment* morphological analyses (in addition to phrase-structure trees, which we discard). For training the segmenter, we used markers in the vocalized section to construct the IOB character sequences. For training the tagger, we automatically converted the ATB morphological analyses to the fine-grained class set. This procedure resulted in 89 classes.

For the segmentation evaluation, we report *per-character* labeling accuracy.[8] For the tagger, we report *per-token* accuracy.

Results Tbl. 1 shows development set accuracy for two settings. FULL is a standard evaluation in which features may be defined over the whole sentence. This includes next-character segmenter features and next-word tagger features. INCREMENTAL emulates the MT setting in which the models are restricted to current and previous observation features. Since the segmenter operates at the character level, we can use the same feature set. However, next-observation features must be removed from the tagger. Nonetheless, tagging accuracy only decreases by 0.1%.

5.2 Translation Quality

Experimental Setup Our decoder is based on the phrase-based approach to translation (Och and Ney, 2004) and contains various feature functions including phrase relative frequency, word-level alignment statistics, and lexicalized re-ordering models (Tillmann, 2004; Och et al., 2004). We tuned the feature weights on a development set using lattice-based minimum error rate training (MERT) (Macherey et al.,

[8] We ignore orthographic re-normalization performed by the annotators. For example, they converted the contraction 'لل' *ll* back to 'ل ال' *l Al*. As a result, we can report accuracy since the guess and gold segmentations have equal numbers of non-whitespace characters.

	MT04 (tune)		MT02		MT03		MT05		Avg
Baseline	18.14		23.87		18.88		22.60		
+POS	18.11	−0.03	23.65	−0.22	18.99	+0.11	22.29	−0.31	−0.17
+POS+Agr	18.86	+**0.72**	24.84	+**0.97**	20.26	+**1.38**	23.48	+**0.88**	+1.04
genres	nw		nw		nw		nw		
#sentences	*1353*		*728*		*663*		*1056*		*2447*

Table 2: Translation quality results (BLEU-4 [%]) for newswire (nw) sets. *Avg* is the weighted averaged (by number of sentences) of the individual test set gains. All improvements are statistically significant at $p \leq 0.01$.

	MT06		MT08		Avg
Baseline	14.68		14.30		
+POS	14.57	−0.11	14.30	+0.0	−0.06
+POS+Agr	15.04	+**0.36**	14.49	+**0.19**	+0.29
genres	nw,bn,ng		nw,ng,wb		
#sentences	*1797*		*1360*		*3157*

Table 3: Mixed genre test set results (BLEU-4 [%]). The MT06 result is statistically significant at $p \leq 0.01$; MT08 is significant at $p \leq 0.02$. The genres are: nw, broadcast news (bn), newsgroups (ng), and weblog (wb).

2008). For each set of results, we initialized MERT with uniform feature weights.

We trained the translation model on 502 million words of parallel text collected from a variety of sources, including the Web. Word alignments were induced using a hidden Markov model based alignment model (Vogel et al., 1996) initialized with bilexical parameters from IBM Model 1 (Brown et al., 1993). Both alignment models were trained using two iterations of the expectation maximization algorithm. Our distributed 4-gram language model was trained on 600 million words of Arabic text, also collected from many sources including the Web (Brants et al., 2007).

For development and evaluation, we used the NIST Arabic-English data sets, each of which contains one set of Arabic sentences and multiple English references. To reverse the translation direction for each data set, we chose the first English reference as the source and the Arabic as the reference.

The NIST sets come in two varieties: newswire (MT02-05) and mixed genre (MT06,08). Newswire contains primarily Modern Standard Arabic (MSA), while the mixed genre data sets also contain transcribed speech and web text. Since the ATB contains MSA, and significant lexical and syntactic differences may exist between MSA and the mixed genres, we achieved best results by tuning on MT04, the largest newswire set.

We evaluated translation quality with BLEU-4 (Papineni et al., 2002) and computed statistical significance with the approximate randomization method of Riezler and Maxwell (2005).[9]

6 Discussion of Translation Results

Tbl. 2 shows translation quality results on newswire, while Tbl. 3 contains results for mixed genres. The baseline is our standard system feature set. For comparison, +POS indicates our class-based model trained on the 11 coarse POS tags only (e.g., "Noun"). Finally, +POS+Agr shows the class-based model with the fine-grained classes (e.g., "Noun+Fem+Sg").

The best result—a +1.04 BLEU average gain—was achieved when the class-based model training data, MT tuning set, and MT evaluation set contained the same genre. We realized smaller, yet statistically significant, gains on the mixed genre data sets. We tried tuning on both MT06 and MT08, but obtained insignificant gains. In the next section, we investigate this issue further.

Tuning with a Treebank-Trained Feature The class-based model is trained on the ATB, which is predominantly MSA text. This data set is syntactically regular, meaning that it does not have highly dialectal content, foreign scripts, disfluencies, etc. Conversely, the mixed genre data sets contain more irregularities. For example, 57.4% of MT06 comes from non-newswire genres. Of the 764 newsgroup sentences, 112 contain some Latin script tokens, while others contain very little morphology:

[9] With the implementation of Clark et al. (2011), available at: http://github.com/jhclark/multeval.

(2) اخلطي 1/2 كوب خل تفاح
mix 1/2 cup vinegar apple
Mix 1/2 cup apple vinegar

(3) بدأ برنامج ماتش ميوزك MusicMatch
start program *miozik maatsh* MusicMatch
Start the program music match (MusicMatch)

In these imperatives, there are no lexically marked agreement relations to score. Ex. (2) is an excerpt from a recipe that appears in full in MT06. Ex. (3) is part of usage instructions for the MusicMatch software. The ATB contains few examples like these, so our class-based model probably does not effectively discriminate between alternative hypotheses for these types of sentences.

Phrase Table Coverage In a standard phrase-based system, effective translation into a highly inflected target language requires that the phrase table contain the inflected word forms necessary to construct an output with correct agreement. If the requisite words are not present in the search space of the decoder, then no feature function would be sufficient to enforce morpho-syntactic agreement.

During development, we observed that the phrase table of our large-scale English-Arabic system did often contain the inflected forms that we desired the system to select. In fact, correctly agreeing alternatives often appeared in n-best translation lists. To verify this observation, we computed the lexical coverage of the MT05 reference sentences in the decoder search space. The statistics below report the token-level recall of reference unigrams:[10]

- Baseline system translation output: 44.6%
- Phrase pairs matching source n-grams: 67.8%

The bottom category includes all lexical items that the decoder could produce in a translation of the source. This large gap between the unigram recall of the actual translation output (top) and the lexical coverage of the phrase-based model (bottom) indicates that translation performance can be improved dramatically by altering the translation model through features such as ours, without expanding the search space of the decoder.

[10]To focus on possibly inflected word forms, we excluded numbers and punctuation from this analysis.

Human Evaluation We also manually evaluated the MT05 output for improvements in agreement.[11] Our system produced different output from the baseline for 785 (74.3%) sentences. We randomly sampled 100 of these sentences and counted agreement errors of all types. The baseline contained 78 errors, while our system produced 66 errors, a statistically significant 15.4% error reduction at $p \leq 0.01$ according to a paired t-test.

In our output, a frequent source of remaining errors was the case of so-called "deflected agreement": inanimate plural nouns require feminine singular agreement with modifiers. On the other hand, animate plural nouns require the sound plural, which is indicated by an appropriate masculine or feminine suffix. For example, the inanimate plural الولايات 'states' requires the singular feminine adjective المتحدة 'united', not the sound plural المتحدات. The ATB does not contain animacy annotations, so our agreement model cannot discriminate between these two cases. However, Alkuhlani and Habash (2011) have recently started annotating the ATB for animacy, and our model could benefit as more data is released.

7 Conclusion and Outlook

Our class-based agreement model improves translation quality by promoting local agreement, but with a minimal increase in decoding time and no additional storage requirements for the phrase table. The model can be implemented with a standard CRF package, trained on existing treebanks for many languages, and integrated easily with many MT feature APIs. We achieved best results when the model training data, MT tuning set, and MT evaluation set contained roughly the same genre. Nevertheless, we also showed an improvement, albeit less significant, on mixed genre evaluation sets.

In principle, our class-based model should be more robust to unseen word types and other phenomena that make non-newswire genres challenging. However, our analysis has shown that for Arabic, these genres typically contain more Latin script and transliterated words, and thus there is less morphology to score. One potential avenue of future work would be to adapt our component models to new genres by self-training them on the target side of a large bitext.

[11]The annotator was the first author.

Acknowledgments We thank Zhifei Li and Chris Manning for helpful discussions, and Klaus Macherey, Wolfgang Macherey, Daisy Stanton, and Richard Zens for engineering support. This work was conducted while the first author was an intern at Google. At Stanford, the first author is supported by a National Science Foundation Graduate Research Fellowship.

References

S. Alkuhlani and N. Habash. 2011. A corpus for modeling morpho-syntactic agreement in Arabic: Gender, number and rationality. In *ACL-HLT*.

G. Andrew and J. Gao. 2007. Scalable training of l_1-regularized log-linear models. In *ICML*.

E. Avramidis and P. Koehn. 2008. Enriching morphologically poor languages for statistical machine translation. In *ACL*.

A. Birch, M. Osborne, and P. Koehn. 2007. CCG supertags in factored statistical machine translation. In *WMT*.

T. Brants, A. C. Popat, P. Xu, F. J. Och, and J. Dean. 2007. Large language models in machine translation. In *EMNLP-CoNLL*.

P. F. Brown, P. V. deSouza, R. L. Mercer, V. J. Della Pietra, and J. C. Lai. 1992. Class-based n-gram models of natural language. *Computational Linguistics*, 18:467–479.

P. F. Brown, S. A. Della Pietra, V. J. Della Pietra, and R. L. Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–313.

D. Cer, M. Galley, D. Jurafsky, and C. D. Manning. 2010. Phrasal: A statistical machine translation toolkit for exploring new model features. In *HLT-NAACL, Demonstration Session*.

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

C. Dyer. 2009. Using a maximum entropy model to build segmentation lattices for MT. In *NAACL*.

A. El Kholy and N. Habash. 2012. Orthographic and morphological processing for English-Arabic statistical machine translation. *Machine Translation*, 26(1-2):25–45.

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

M. Galley and C. D. Manning. 2009. Quadratic-time dependency parsing for machine translation. In *ACL-IJCNLP*.

S. Green and C. D. Manning. 2010. Better Arabic parsing: baselines, evaluations, and analysis. In *COLING*.

N. Habash and O. Rambow. 2005. Arabic tokenization, part-of-speech tagging and morphological disambiguation in one fell swoop. In *ACL*.

N. Habash and F. Sadat. 2006. Arabic preprocessing schemes for statistical machine translation. In *NAACL*.

H. Hassan, K. Sima'an, and A. Way. 2007. Supertagged phrase-based statistical machine translation. In *ACL*.

M. Johnson. 2003. Learning and parsing stochastic unification-based grammars. In *COLT*.

P. Koehn and H. Hoang. 2007. Factored translation models. In *EMNLP-CoNLL*.

P. Koehn and K. Knight. 2003. Empirical methods for compound splitting. In *EACL*.

P. Koehn, H. Hoang, A. Birch, C. Callison-Burch, M. Federico, N. Bertoldi, et al. 2007. Moses: Open source toolkit for statistical machine translation. In *ACL, Demonstration Session*.

J. Lafferty, A. McCallum, and F. Pereira. 2001. Conditional random fields: Probablistic models for segmenting and labeling sequence data. In *ICML*.

A. Lopez. 2008. Statistical machine translation. *ACM Computing Surveys*, 40(8):1–49.

M. Maamouri, A. Bies, T. Buckwalter, and W. Mekki. 2004. The Penn Arabic Treebank: Building a large-scale annotated Arabic corpus. In *NEMLAR*.

W. Macherey, F. Och, I. Thayer, and J. Uszkoreit. 2008. Lattice-based minimum error rate training for statistical machine translation. In *EMNLP*.

K. Macherey, A. Dai, D. Talbot, A. Popat, and F. Och. 2011. Language-independent compound splitting with morphological operations. In *ACL*.

E. Minkov, K. Toutanova, and H. Suzuki. 2007. Generating complex morphology for machine translation. In *ACL*.

C. Monz. 2011. Statistical machine translation with local language models. In *EMNLP*.

F. J. Och and H. Ney. 2004. The alignment template approach to statistical machine translation. *Computational Linguistics*, 30(4):417–449.

F. J. Och, D. Gildea, S. Khudanpur, A. Sarkar, K. Yamada, A. Fraser, et al. 2004. A smorgasbord of features for statistical machine translation. In *HLT-NAACL*.

F. J. Och. 1999. An efficient method for determining bilingual word classes. In *EACL*.

K. Papineni, S. Roukos, T. Ward, and W. Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL*.

F. Peng, F. Feng, and A. McCallum. 2004. Chinese segmentation and new word detection using conditional random fields. In *COLING*.

S. Petrov, D. Das, and R. McDonald. 2012. A universal part-of-speech tagset. In *LREC*.

O. Rambow, D. Chiang, M. Diab, N. Habash, R. Hwa, et al. 2005. Parsing Arabic dialects. Technical report, Johns Hopkins University.

L. A. Ramshaw and M. Marcus. 1995. Text chunking using transformation-based learning. In *Proc. of the Third Workshop on Very Large Corpora*.

S. Riezler and J. T. Maxwell. 2005. On some pitfalls in automatic evaluation and significance testing in MT. In *ACL-05 Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization (MTSE)*.

L. Schwartz, C. Callison-Burch, W. Schuler, and S. Wu. 2011. Incremental syntactic language models for phrase-based translation. In *ACL-HLT*.

L. Shen, J. Xu, and R. Weischedel. 2010. String-to-dependency statistical machine translation. *Computational Linguistics*, 36(4):649–671.

M. Subotin. 2011. An exponential translation model for target language morphology. In *ACL-HLT*.

C. Tillmann. 2004. A unigram orientation model for statistical machine translation. In *NAACL*.

K. Toutanova, H. Suzuki, and A. Ruopp. 2008. Applying morphology generation models to machine translation. In *ACL-HLT*.

J. Uszkoreit and T. Brants. 2008. Distributed word clustering for large scale class-based language modeling in machine translation. In *ACL-HLT*.

S. Vogel, H. Ney, and C. Tillmann. 1996. HMM-based word alignment in statistical translation. In *COLING*.

P. Williams and P. Koehn. 2011. Agreement constraints for statistical machine translation into German. In *WMT*.

Y. Zhang. 2009. *Structured Language Models for Statistical Machine Translation*. Ph.D. thesis, Carnegie Mellon University.