A Word-Class Approach to Labeling PSCFG Rules for Machine Translation

Andreas Zollmann and **Stephan Vogel**
Language Technologies Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213, USA
{zollmann,vogel+}@cs.cmu.edu

Abstract

In this work we propose methods to label probabilistic synchronous context-free grammar (PSCFG) rules using only word tags, generated by either part-of-speech analysis or unsupervised word class induction. The proposals range from simple tag-combination schemes to a phrase clustering model that can incorporate an arbitrary number of features.

Our models improve translation quality over the single generic label approach of Chiang (2005) and perform on par with the syntactically motivated approach from Zollmann and Venugopal (2006) on the NIST large Chinese-to-English translation task. These results persist when using automatically learned word tags, suggesting broad applicability of our technique across diverse language pairs for which syntactic resources are not available.

1 Introduction

The Probabilistic Synchronous Context Free Grammar (PSCFG) formalism suggests an intuitive approach to model the long-distance and lexically sensitive reordering phenomena that often occur across language pairs considered for statistical machine translation. As in monolingual parsing, nonterminal symbols in translation rules are used to generalize beyond purely lexical operations. *Labels* on these nonterminal symbols are often used to enforce syntactic constraints in the generation of bilingual sentences and imply conditional independence assumptions in the translation model. Several techniques have been recently proposed to automatically identify and estimate parameters for PSCFGs (or related synchronous grammars) from parallel corpora (Galley et al., 2004; Chiang, 2005; Zollmann and Venugopal, 2006; Liu et al., 2006; Marcu et al., 2006).

While all of these techniques rely on word-alignments to suggest lexical relationships, they differ in the way in which they assign labels to nonterminal symbols of PSCFG rules. Chiang (2005) describes a procedure to extract PSCFG rules from word-aligned (Brown et al., 1993) corpora, where all nonterminals share the same generic label X. In Galley et al. (2004) and Marcu et al. (2006), target language parse trees are used to identify rules and label their nonterminal symbols, while Liu et al. (2006) use source language parse trees instead. Zollmann and Venugopal (2006) directly extend the rule extraction procedure from Chiang (2005) to heuristically label any phrase pair based on target language parse trees. Label-based approaches have resulted in improvements in translation quality over the single X label approach (Zollmann et al., 2008; Mi and Huang, 2008); however, all the works cited here rely on stochastic parsers that have been trained on manually created syntactic treebanks. These treebanks are difficult and expensive to produce and exist for a limited set of languages only.

In this work, we propose a labeling approach that is based merely on part-of-speech analysis of the source or target language (or even both). Towards the ultimate goal of building end-to-end machine translation systems without *any* human annotations, we also experiment with automatically inferred word classes using distributional clustering (Kneser and Ney, 1993). Since the number of classes is a parameter of the clustering method and the resulting nonterminal size of our grammar is a function of the number of word classes, the PSCFG grammar complexity can be adjusted to the specific translation task at hand.

Finally, we introduce a more flexible labeling approach based on K-means clustering, which allows

Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics, pages 1–11,
Portland, Oregon, June 19-24, 2011. ©2011 Association for Computational Linguistics

the incorporation of an arbitrary number of word-class based features, including phrasal contexts, can make use of multiple tagging schemes, and also allows non-class features such as phrase sizes.

2 PSCFG-based translation

In this work we experiment with PSCFGs that have been automatically learned from word-aligned parallel corpora. PSCFGs are defined by a source terminal set (source vocabulary) $\mathcal{T}_S$, a target terminal set (target vocabulary) $\mathcal{T}_T$, a shared nonterminal set $\mathcal{N}$ and rules of the form: $A \rightarrow \langle \gamma, \alpha, w \rangle$ where

- $A \in \mathcal{N}$ is a labeled nonterminal referred to as the left-hand-side of the rule,
- $\gamma \in (\mathcal{N} \cup \mathcal{T}_S)^*$ is the source side of the rule,
- $\alpha \in (\mathcal{N} \cup \mathcal{T}_T)^*$ is the target side of the rule,
- $w \in [0, \infty)$ is a non-negative real-valued weight assigned to the rule; in our model, w is the product of features ϕ_i raised to the power of weight λ_i.

Chiang (2005) learns a single-nonterminal PSCFG from a bilingual corpus by first identifying initial phrase pairs using the technique from Koehn et al. (2003), and then performing a generalization operation to generate phrase pairs with gaps, which can be viewed as PSCFG rules with generic ‘X’ nonterminal left-hand-sides and substitution sites. Bilingual features ϕ_i that judge the quality of each rule are estimated based on rule extraction frequency counts.

3 Hard rule labeling from word classes

We now describe a simple method of inducing a multi-nonterminal PSCFG from a parallel corpus with word-tagged target side sentences. The same procedure can straightforwardly be applied to a corpus with tagged source side sentences. We use the simple term ‘tag’ to stand for any kind of word-level analysis—a syntactic, statistical, or other means of grouping word types or tokens into classes, possibly based on their position and context in the sentence, POS tagging being the most obvious example.

As in Chiang’s hierarchical system, we rely on an external phrase-extraction procedure such as the one of Koehn et al. (2003) to provide us with a set of phrase pairs for each sentence pair in the training corpus, annotated with their respective start and end positions in the source and target sentences. Let $\boldsymbol{f} = f_1 \cdots f_m$ be the current source sentence, $\boldsymbol{e} = e_1 \cdots e_n$ the current target sentence, and $\boldsymbol{t} = t_1 \cdots t_n$ its corresponding target tag sequence. We convert each extracted phrase pair, represented by its source span $\langle i, j \rangle$ and target span $\langle k, \ell \rangle$, into an initial rule

$$t_k\text{-}t_\ell \rightarrow f_i \cdots f_j \mid e_k \cdots e_\ell$$

by assigning it a nonterminal “t_k-t_ℓ” constructed by combining the tag of the target phrase’s left-most word with the tag of its right-most word.

The creation of complex rules based on all initial rules obtained from the current sentence now proceeds just as in Chiang’s model.

Consider the target-tagged example sentence pair:

Ich habe ihn gesehen | I/PRP saw/VBD him/PRP

Then (depending on the extracted phrase pairs), the resulting initial rules could be:

1: PRP-PRP → Ich | I
2: PRP-PRP → ihn | him
3: VBD-VBD → gesehen | saw
4: VBD-PRP → habe ihn gesehen | saw him
5: PRP-PRP → Ich habe ihn gesehen | I saw him

Now, by abstracting-out initial rule 2 from initial rule 4, we obtain the complex rule:

VBD-PRP → habe PRP-PRP$_1$ gesehen | saw PRP-PRP$_1$

Intuitively, the labeling of initial rules with tags marking the boundary of their target sides results in complex rules whose nonterminal occurrences impose weak syntactic constraints on the rules eligible for substitution in a PSCFG derivation: The left and right boundary word tags of the inserted rule’s target side have to match the respective boundary word tags of the phrase pair that was replaced by a nonterminal when the complex rule was created from a training sentence pair. Since consecutive words within a rule stem from consecutive words in the training corpus and thus are already consistent, the boundary word tags are more informative than tags of words between the boundaries for the task of combining different rules in a derivation, and are therefore a more appropriate choice for the creation of grammar labels than tags of inside words.

Accounting for phrase size A drawback of the current approach is that a single-word rule such as

PRP-PRP → Ich | I

can have the same left-hand-side nonterminal as a long rule with identical left and right boundary tags, such as (when using target-side tags):

PRP-PRP → Ich habe ihn gesehen | I saw him

We therefore introduce a means of distinguishing between one-word, two-word, and multiple-word phrases as follows: Each one-word phrase with tag T simply receives the label T, instead of T-T. Two-word phrases with tag sequence T_1T_2 are labeled T_1-T_2 as before. Phrases of length greater two with tag sequence $T_1 \cdots T_n$ are labeled $T_1..T_n$ to denote that tags were omitted from the phrase's tag sequence. The resulting number of grammar nonterminals based on a tag vocabulary of size t is thus given by $2t^2 + t$.

An alternative way of accounting for phrase size is presented by Chiang et al. (2008), who introduce *structural distortion features* into a hierarchical phrase-based model, aimed at modeling nonterminal reordering given source span length. Our approach instead uses distinct grammar rules and labels to discriminate phrase size, with the advantage of enabling all translation models to estimate distinct weights for distinct size classes and avoiding the need of additional models in the log-linear framework; however, the increase in the number of labels and thus grammar rules decreases the reliability of estimated models for rare events due to increased data sparseness.

Extension to a bilingually tagged corpus While the availability of syntactic annotations for both source *and* target language is unlikely in most translation scenarios, some form of word tags, be it part-of-speech tags or learned word clusters (cf. Section 3) might be available on both sides. In this case, our grammar extraction procedure can be easily extended to impose both source and target constraints on the eligible substitutions simultaneously.

Let N_f be the nonterminal label that would be assigned to a given initial rule when utilizing the source-side tag sequence, and N_e the assigned label according to the target-side tag sequence. Then our bilingual tag-based model assigns '$N_f + N_e$' to the initial rule. The extraction of complex rules proceeds as before. The number of nonterminals in this model, based on a source tag vocabulary of size s and a target tag vocabulary of size t, is thus given by s^2t^2 for the regular labeling method and $(2s^2 + s)(2t^2 + t)$ when accounting for phrase size.

Consider again our example sentence pair (now also annotated with source-side part-of-speech tags):

Ich/PRP habe/AUX ihn/PRP gesehen/VBN
I/PRP saw/VBD him/PRP

Given the same phrase extraction method as before, the resulting initial rules for our bilingual model, when also accounting for phrase size, are as follows:

1: PRP+PRP → Ich | I
2: PRP+PRP → ihn | him
3: VBN+VBD → gesehen | saw
4: AUX..VBN+VBD-PRP → habe ihn gesehen | saw him
5: PRP..VBN+PRP..PRP → Ich habe ihn gesehen | I saw him

Abstracting-out rule 2 from rule 4, for instance, leads to the complex rule:

AUX..VBN+VBD-PRP → habe PRP+PRP$_1$ gesehen | saw PRP+PRP$_1$

Unsupervised word class assignment by clustering As an alternative to POS tags, we experiment with unsupervised word clustering methods based on the exchange algorithm (Kneser and Ney, 1993). Its objective function is maximizing the likelihood

$$\prod_{i=1}^{n} P(w_i | w_1, \ldots, w_{i-1})$$

of the training data $w = w_1, \ldots, w_n$ given a partially class-based bigram model of the form

$$P(w_i | w_1, \ldots, w_{i-1}) \approx p(c(w_i) | w_{i-1}) \cdot p(w_i | c(w_i))$$

where $c : \mathcal{V} \rightarrow \{1, \ldots, N\}$ maps a word (type, not token) w to its class $c(w)$, $\mathcal{V}$ is the vocabulary, and N the fixed number of classes, which has to be chosen *a priori*. We use the publicly available implementation MKCLS (Och, 1999) to train this model. As training data we use the respective side of the parallel training data for the translation system.

We also experiment with the extension of this model by Clark (2003), who incorporated morphological information by imposing a Bayesian prior on the class mapping c, based on N individual distributions over strings, one for each word class. Each such distribution is a character-based hidden Markov model, thus encouraging the grouping of morphologically similar words into the same class.

4 Clustering phrase pairs directly using the K-means algorithm

Even though we have only made use of the first and last words' classes in the labeling methods described so far, the number of resulting grammar nonterminals quickly explodes. Using a scheme based on source and target phrases with accounting for phrase size, with 36 word classes (the size of the Penn English POS tag set) for both languages, yields a grammar with $(36+2*36^2)^2 = 6.9$m nonterminal labels.

Quite plausibly, phrase labeling should be informed by more than just the classes of the first and last words of the phrase. Taking phrase context into account, for example, can aid the learning of syntactic properties: a phrase beginning with a determiner and ending with a noun, with a verb as right context, is more likely to be a noun phrase than the same phrase with another noun as right context. In the current scheme, there is no way of distinguishing between these two cases. Similarly, it is conceivable that using non-boundary words inside the phrase might aid the labeling process.

When relying on unsupervised learning of the word classes, we are forced to chose a fixed number of classes. A smaller number of word clusters will result in smaller number of grammar nonterminals, and thus more reliable feature estimation, while a larger number has the potential to discover more subtle syntactic properties. Using multiple word clusterings simultaneously, each based on a different number of classes, could turn this global, hard trade-off into a local, soft one, informed by the number of phrase pair instances available for a given granularity.

Lastly, our method of accounting for phrase size is somewhat displeasing: While there is a hard partitioning of one-word and two-word phrases, no distinction is made between phrases of length greater than two. Marking phrase sizes greater than two explicitly by length, however, would create many sparse, low-frequency rules, and one of the strengths of PSCFG-based translation is the ability to substitute flexible-length spans into nonterminals of a derivation. A partitioning where phrase size is instead merely a feature informing the labeling process seems more desirable.

We thus propose to represent each phrase pair instance (including its bilingual one-word contexts) as feature vectors, i.e., points of a vector space. We then use these data points to partition the space into clusters, and subsequently assign each phrase pair instance the cluster of its corresponding feature vector as label.

The feature mapping Consider the phrase pair instance

$$(f_0)f_1 \cdots f_m(f_{m+1}) \mid (e_0)e_1 \cdots e_n(e_{n+1})$$

(where $f_0, f_{m+1}, e_0, e_{n+1}$ are the left and right, source and target side contexts, respectively). We begin with the case of only a single, target-side word class scheme (either a tagger or an unsupervised word clustering/POS induction method). Let $C = \{c_1, \ldots, c_N\}$ be its set of word classes. Further, let c_0 be a short-hand for the result of looking up the class of a word that is out of bounds (e.g., the left context of the first word of a sentence, or the second word of a one-word phrase). We now map our phrase pair instance to the real-valued vector (where $\mathbb{1}_{[P]}$ is the indicator function defined as 1 if property P is true, and 0 otherwise):

$$\Big\langle \mathbb{1}_{[e_1=c_0]}, \ldots, \mathbb{1}_{[e_1=c_N]}, \mathbb{1}_{[e_n=c_0]}, \ldots, \mathbb{1}_{[e_n=c_N]},$$
$$\alpha_{\text{sec}}\mathbb{1}_{[e_2=c_0]}, \ldots, \alpha_{\text{sec}}\mathbb{1}_{[e_2=c_N]},$$
$$\alpha_{\text{sec}}\mathbb{1}_{[e_{n-1}=c_0]}, \ldots, \alpha_{\text{sec}}\mathbb{1}_{[e_{n-1}=c_N]},$$
$$\frac{\alpha_{\text{ins}}\sum_{i=1}^{n}\mathbb{1}_{[e_i=c_0]}}{n}, \ldots, \frac{\alpha_{\text{ins}}\sum_{i=1}^{n}\mathbb{1}_{[e_i=c_N]}}{n},$$
$$\alpha_{\text{cntxt}}\mathbb{1}_{[e_0=c_0]}, \ldots, \alpha_{\text{cntxt}}\mathbb{1}_{[e_0=c_N]},$$
$$\alpha_{\text{cntxt}}\mathbb{1}_{[e_{n+1}=c_0]}, \ldots, \alpha_{\text{cntxt}}\mathbb{1}_{[e_{n+1}=c_N]},$$
$$\alpha_{\text{phrsize}}\sqrt{N+1}\,\log_{10}(n)\Big\rangle$$

The α parameters determine the influence of the different types of information. The elements in the first line represent the phrase boundary word classes, the next two lines the classes of the second and penultimate word, followed by a line representing the accumulated contents of the whole phrase, followed by two lines pertaining to the context word classes. The final element of the vector is proportional to the logarithm of the phrase length.[1] We chose the logarithm assuming that length deviation of syntactic phrasal units is not constant, but proportional to the average length. Thus, all other features being equal, the distance between a two-word and a four-word phrase is

[1] The $\sqrt{N+1}$ factor serves to make the feature's influence independent of the number of word classes by yielding the same distance (under L_2) as $N+1$ identical copies of the feature.

the same as the distance between a four-word and an eight-word phrase.

We will mainly use the Euclidean (L_2) distance to compare points for clustering purposes. Our feature space is thus the Euclidean vector space $\mathbb{R}^{7N+8}$.

To additionally make use of source-side word classes, we append elements analogous to the ones above to the vector, all further multiplied by a parameter $\alpha_{\rm src}$ that allows trading off the relevance of source-side and target-side information. In the same fashion, we can incorporate multiple tagging schemes (e.g., word clusterings of different granularities) into the same feature vector. As finer-grained schemes have more elements in the feature vector than coarser-grained ones, and thus exert more influence, we set the α parameter for each scheme to $1/N$ (where N is the number of word classes of the scheme).

The K-means algorithm To create the clusters, we chose the K-means algorithm (Steinhaus, 1956; MacQueen, 1967) for both its computational efficiency and ease of implementation and parallelization. Given an initial mapping from the data points to K clusters, the procedure alternates between (i) computing the centroid of each cluster and (ii) reallocating each data point to the closest cluster centroid, until convergence.

We implemented two commonly used initialization methods: Forgy and Random Partition. The Forgy method randomly chooses K observations from the data set and uses these as the initial means. The Random Partition method first randomly assigns a cluster to each observation and then proceeds straight to step (ii). Forgy tends to spread the initial means out, while Random Partition places all of them close to the center of the data set. As the resulting clusters looked similar, and Random Partition sometimes led to a high rate of empty clusters, we settled for Forgy.

5 Experiments

We evaluate our approach by comparing translation quality, as evaluated by the IBM-BLEU (Papineni et al., 2002) metric on the NIST Chinese-to-English translation task using MT04 as development set to train the model parameters λ, and MT05, MT06 and MT08 as test sets. Even though a key advantage of our method is its applicability to resource-poor languages, we used a language pair for which linguistic resources are available in order to determine how close translation performance can get to a fully syntax-based system. Accordingly, we use Chiang's hierarchical phrase based translation model (Chiang, 2007) as a base line, and the syntax-augmented MT model (Zollmann and Venugopal, 2006) as a 'target line', a model that would not be applicable for language pairs without linguistic resources.

We perform PSCFG rule extraction and decoding using the open-source "SAMT" system (Venugopal and Zollmann, 2009), using the provided implementations for the hierarchical and syntax-augmented grammars. Apart from the language model, the lexical, phrasal, and (for the syntax grammar) label-conditioned features, and the rule, target word, and glue operation counters, Venugopal and Zollmann (2009) also provide both the hierarchical and syntax-augmented grammars with a rareness penalty $1/\operatorname{cnt}(r)$, where $\operatorname{cnt}(r)$ is the occurrence count of rule r in the training corpus, allowing the system to learn penalization of low-frequency rules, as well as three indicator features firing if the rule has one, two unswapped, and two swapped nonterminal pairs, respectively.[2] Further, to mitigate badly estimated PSCFG derivations based on low-frequency rules of the much sparser syntax model, the syntax grammar also contains the hierarchical grammar as a backbone (cf. Zollmann and Vogel (2010) for details and empirical analysis).

We implemented our rule labeling approach within the SAMT rule extraction pipeline, resulting in comparable features across all systems. For all systems, we use the bottom-up chart parsing decoder implemented in the SAMT toolkit with a reordering limit of 15 source words, and correspondingly extract rules from initial phrase pairs of maximum source length 15. All rules have at most two nonterminal symbols, which must be non-consecutive on the source side, and rules must contain at least one source-side terminal symbol. The beam settings for the hierarchical system are 600 items per 'X' (generic rule) cell, and 600 per 'S' (glue) cell.[3] Due to memory limitations, the multi-nonterminal grammars have to be pruned more harshly: We al-

[2] Penalization or reward of purely-lexical rules can be indirectly learned by trading off these features with the rule counter feature.

[3] For comparison, Chiang (2007) uses 30 and 15, respectively, and further prunes items that deviate too much in score from the best item. He extracts initial phrases of maximum length 10.

low 100 ‘S’ items, and a total of 500 non-‘S’ items, but maximally 40 items per nonterminal. For all systems, we further discard non-initial rules occurring only once.[4] For the multi-nonterminal systems, we generally further discard all non-generic non-initial rules occurring less than 6 times, but we additionally give results for a ‘slow’ version of the Syntax target-line system and our best word class based systems, where only single-occurrences were removed.

For parameter tuning, we use the L_0-regularized minimum-error-rate training tool provided by the SAMT toolkit. Each system is trained separately to adapt the parameters to its specific properties (size of nonterminal set, grammar complexity, features sparseness, reliance on the language model, etc.).

The parallel training data comprises of 9.6M sentence pairs (206M Chinese and 228M English words). The source and target language parses for the syntax-augmented grammar, as well as the POS tags for our POS-based grammars were generated by the Stanford parser (Klein and Manning, 2003).

The results are given in Table 1. Results for the Syntax system are consistent with previous results (Zollmann et al., 2008), indicating improvements over the hierarchical system. Our approach, using target POS tags (‘POS-tgt (no phr. s.)’), outperforms the hierarchical system on all three tests sets, and gains further improvements when accounting for phrase size (‘POS-tgt’). The latter approach is roughly on par with the corresponding Syntax system, slightly outperforming it on average, but not consistently across all test sets. The same is true for the ‘slow’ version (‘POS-tgt-slow’).

The model based on bilingually tagged training instances (‘POS-src&tgt’) does not gain further improvements over the merely target-based one, but actually performs worse. We assume this is due to the huge number of nonterminals of ‘POS-src&tgt’ ($(2 * 33^2 + 33)(2 * 36^2 + 36) = 5.8$M in principle) compared to ‘POS-tgt’ ($2 * 36^2 + 36 = 2628$), increasing the sparseness of the grammar and thus leading to less reliable statistical estimates.

We also experimented with a source-tag based model (‘POS-src’). In line with previous findings for syntax-augmented grammars (Zollmann and Vogel, 2010), the source-side-based grammar does not reach the translation quality of its target-based counterpart; however, the model still outperforms the hierarchical system on all test sets. Further, decoding is much faster than for ‘POS-ext-tgt’ and even slightly faster than ‘Hierarchical’. This is due to the fact that for the source-tag based approach, a given chart cell in the CYK decoder, represented by a start and end position in the source sentence, almost uniquely determines the nonterminal any hypothesis in this cell can have: Disregarding part-of-speech tag ambiguity and phrase size accounting, that nonterminal will be the composition of the tags of the start and end source words spanned by that cell. At the same time, this demonstrates that there is hence less of a role for the nonterminal labels to resolve translational ambiguity in the source based model than in the target based model.

Performance of the word-clustering based models To empirically validate the unsupervised clustering approaches, we first need to decide how to determine the number of word classes, N. A straightforward approach is to run experiments and report test set results for many different N. While this would allow us to reliably conclude the optimal number N, a comparison of that best-performing clustering method to the hierarchical, syntax, and POS systems would be tainted by the fact that N was effectively tuned on the test sets. We therefore choose N merely based on development set performance. Unfortunately, variance in development set BLEU scores tends to be higher than test set scores, despite of SAMT MERT’s inbuilt algorithms to overcome local optima, such as random restarts and zeroing-out. We have noticed that using an L_0-penalized BLEU score[5] as MERT’s objective on the merged n-best lists over all iterations is more stable and will therefore use this score to determine N.

Figure 1 (left) shows the performance of the distributional clustering model (‘Clust’) and its morphology-sensitive extension (‘Clust-morph’) according to this score for varying values of $N = 1, \ldots, 36$ (the number Penn treebank POS tags, used for the ‘POS’ models, is 36).[6] For ‘Clust’, we see a comfortably wide plateau of nearly-identical scores from $N = 7, \ldots, 15$. Scores for ‘Clust-morph’ are lower throughout, and peak at $N = 7$.

Looking back at Table 1, we now compare the clustering models chosen by the procedure above—

[4] As shown in Zollmann et al. (2008), the impact of these rules on translation quality is negligible.

[5] Given by: BLEU $-\beta \times |\{i \in \{1, \ldots, K\} | \lambda_i \neq 0\}|$, where $\lambda_1, \ldots, \lambda_K$ are the feature weights and the constant β (which we set to 0.00001) is the regularization penalty.

[6] All these models account for phrase size.

	Dev (MT04)	MT05	MT06	MT08	**TestAvg**	Time
Hierarchical	38.63	36.51	33.26	25.77	**31.85**	14.3
Syntax	39.39	37.09	34.01	26.53	**32.54**	18.1
Syntax-slow	39.69	37.56	34.66	26.93	**33.05**	34.6
POS-tgt (no phr. s.)	39.31	37.29	33.79	26.13	**32.40**	27.7
POS-tgt	39.14	37.29	33.97	26.77	**32.68**	19.2
POS-src	38.74	36.75	33.85	26.76	**32.45**	12.2
POS-src&tgt	38.78	36.71	33.65	26.52	**32.29**	18.8
POS-tgt-slow	39.86	37.78	34.37	27.14	**33.10**	44.6
Clust-7-tgt	39.24	36.74	34.00	26.93	**32.56**	24.3
Clust-7-morph-tgt	39.08	36.57	33.81	26.40	**32.26**	23.6
Clust-7-src	38.68	36.17	33.23	26.55	**31.98**	11.1
Clust-7-src&tgt	38.71	36.49	33.65	26.33	**32.16**	15.8
Clust-7-tgt-slow	39.48	37.70	34.31	27.24	**33.08**	45.2
kmeans-POS-src&tgt	39.11	37.23	33.92	26.80	**32.65**	18.5
kmeans-POS-src&tgt-L_1	39.33	36.92	33.81	26.59	**32.44**	17.6
kmeans-POS-src&tgt-cosine	39.15	37.07	33.98	26.68	**32.58**	17.7
kmeans-POS-src&tgt ($\alpha_{\text{ins}} = .5$)	39.07	36.88	33.71	26.26	**32.28**	16.5
kmeans-Clust-7-src&tgt	39.19	36.96	34.26	26.97	**32.73**	19.3
kmeans-Clust-7..36-src&tgt	39.09	36.93	34.24	26.92	**32.70**	17.3
kmeans-POS-src&tgt-slow	39.28	37.16	34.38	27.11	**32.88**	36.3
kmeans-Clust-7..36-s&t-slow	39.18	37.12	34.13	27.35	**32.87**	34.3

Table 1: Translation quality in % case-insensitive IBM-BLEU (i.e., brevity penalty based on closest reference length) for Chinese-English NIST-large translation tasks, comparing baseline Hierarchical and Syntax systems with POS and clustering based approaches proposed in this work. 'TestAvg' shows the average score over the three test sets. 'Time' is the average decoding time per sentence in seconds on one CPU.

resulting in $N = 7$ for the morphology-unaware model ('Clust-7-tgt') as well as the morphology-aware model ('Clust-7-morph-tgt')—to the other systems. 'Clust-7-tgt' improves over the hierarchical base line on all three test sets and is on par with the corresponding Syntax and POS target lines. The same holds for the 'Clust-7-tgt-slow' version. We also experimented with a model variant based on seven source and seven target language clusters ('Clust-7-src&tgt') and a source-only labeled model ('Clust-7-src')—both performing worse.

Surprisingly, the morphology-sensitive clustering model ('Clust-7-morph-tgt'), while still improving over the hierarchical system, performs worse than the morphology-unaware model. An inspection of the trained word clusters showed that the model, while far superior to the morphology-unaware model in e.g. mapping all numbers to the same class, is overzealous in discovering morphological regularities (such as the '-ed' suffix) to partition functionally only slightly dissimilar words (such present-tense and past-tense verbs) into different classes. While these subtle distinctions make for good partitionings when the number of clusters is large, they appear to lead to inferior results for our task that relies on coarse-grained partitionings of the vocabulary. Note that there are no 'src' or 'src&tgt' systems for 'Clust-morph', as Chinese, being a monosyllabic writing system, does not lend itself to morphology-sensitive clustering.

K-means clustering based models To establish suitable values for the α parameters and investigate the impact of the number of clusters, we looked at the development performance over various parameter combinations for a K-means model based on source and/or target part-of-speech tags.[7] As can be seen from Figure 1 (right), our method reaches its peak performance at around 50 clusters and then levels off slightly. Encouragingly, in contrast to the hard labeling procedure, K-means actually improves when adding source-side information. The optimal ratio of weighting source and target classes is 0.5:1, corresponding to $\alpha_{\text{src}} = .5$. Incorporating context information also helps, and does best for $\alpha_{\text{cntxt}} = 0.25$, i.e. when giving contexts 1/4 the influence of the phrase boundary words.

[7] We set $\alpha_{\text{sec}} = .25$, $\alpha_{\text{ins}} = 0$, and $\alpha_{\text{phrsize}} = .5$ throughout.

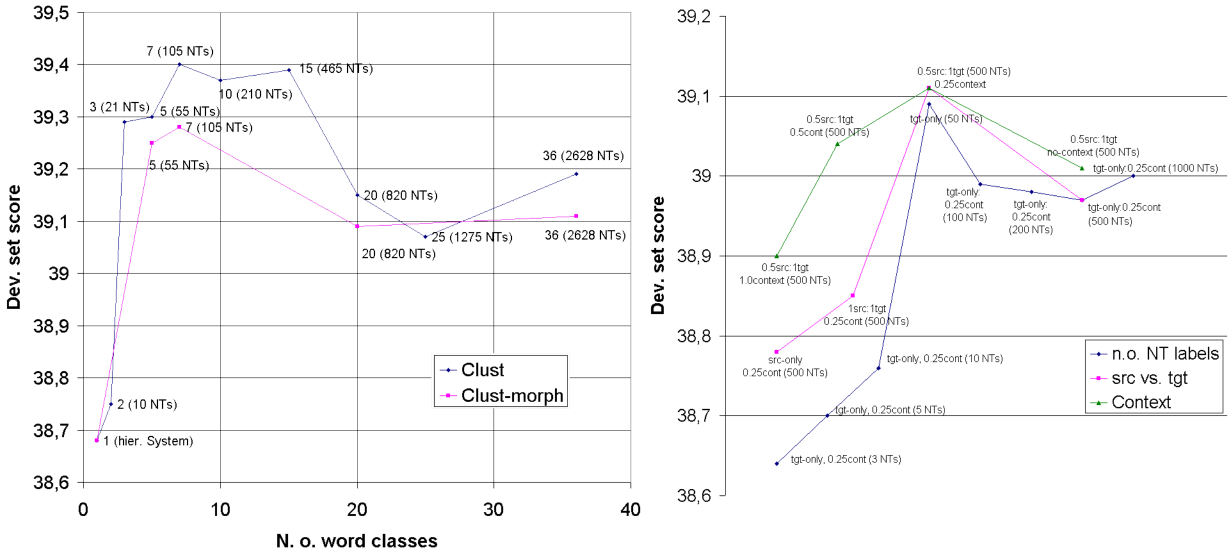

Figure 1: ***Left***: Performance of the distributional clustering model 'Clust' and its morphology-sensitive extension 'Clust-morph' according to L_0-penalized development set BLEU score for varying numbers N of word classes. For each data point N, its corresponding n.o. nonterminals of the induced grammar is stated in parentheses.
Right: Dev. set performance of K-means for various n.o. labels and values of α_{src} and α_{cntxt}.

Entry 'kmeans-POS-src&tgt' in Table 1 shows the test set results for the development-set best K-means configuration (i.e., $\alpha_{\text{src}} = .5$, $\alpha_{\text{cntxt}} = 0.25$, and using 500 clusters). While beating the hierarchical baseline, it is only minimally better than the much simpler target-based hard labeling method 'POS-tgt'. We also tried K-means variants in which the Euclidean distance metric is replaced by the city block distance L_1 and the cosine dissimilarity, respectively, with slightly worse outcomes. Configuration 'kmeans-POS-src&tgt ($\alpha_{\text{ins}} = .5$)' investigates the incorporation of non-boundary word tags inside the phrase. Unfortunately, these features appear to deteriorate performance, presumably because given a fixed number of clusters, accounting for contents inside the phrase comes at the cost of neglect of boundary words, which are more relevant to producing correctly reordered translations.

The two completely unsupervised systems 'kmeans-Clust-7-src&tgt' (based on 7-class MKCLS distributional word clustering) and 'kmeans-Clust-7..36-src&tgt' (using six different word clustering models simultaneously: all the MKCLS models from Figure 1 (left) except for the two-, three- and five-class models) have the best results, outperforming the other K-means models as well as 'Syntax' and 'POS-tgt' on average, but not on all test sets.

Lastly, we give results for 'slow' K-means configurations ('kmeans-POS-src&tgt-slow' and 'kmeans-Clust-7..36-s&t-slow'). Unfortunately (or fortunately, from a pragmatic viewpoint), the models are outperformed by the much simpler 'POS-tgt-slow' and 'Clust-7-tgt-slow' models.

6 Related work

Hassan et al. (2007) improve the statistical phrase-based MT model by injecting *supertags*, lexical information such as the POS tag of the word and its subcategorization information, into the phrase table, resulting in generalized phrases with placeholders in them. The supertags are also injected into the language model. Our approach also generates phrase labels and placeholders based on word tags (albeit in a different manner and without the use of subcategorization information), but produces PSCFG rules for use in a parsing-based decoding system.

Unsupervised *synchronous* grammar induction, apart from the contribution of Chiang (2005) discussed earlier, has been proposed by Wu (1997) for inversion transduction grammars, but as Chiang's model only uses a single generic nonterminal label. Blunsom et al. (2009) present a nonparametric PSCFG translation model that directly induces a grammar from parallel sentences without the use of or constraints from a word-alignment model, and

Cohn and Blunsom (2009) achieve the same for tree-to-string grammars, with encouraging results on small data. Our more humble approach treats the training sentences' word alignments and phrase pairs, obtained from external modules, as ground truth and employs a straight-forward generalization of Chiang's popular rule extraction approach to labeled phrase pairs, resulting in a PSCFG with multiple nonterminal labels.

Our phrase pair clustering approach is similar in spirit to the work of Lin and Wu (2009), who use K-means to cluster (monolingual) phrases and use the resulting clusters as features in discriminative classifiers for a named-entity-recognition and a query classification task. Phrases are represented in terms of their contexts, which can be more than one word long; words within the phrase are not considered. Further, each context contributes one dimension per vocabulary word (not per word class as in our approach) to the feature space, allowing for the discovery of subtle semantic similarities in the phrases, but at much greater computational expense. Another distinction is that Lin and Wu (2009) work with phrase types instead of phrase instances, obtaining a phrase type's contexts by averaging the contexts of all its phrase instances.

Nagata et al. (2006) present a reordering model for machine translation, and make use of clustered phrase pairs to cope with data sparseness in the model. They achieve the clustering by reducing phrases to their head words and then applying the MKCLS tool to these pseudo-words.

Kuhn et al. (2010) cluster the phrase pairs of an SMT phrase table based on their co-occurrence counts and edit distances in order to arrive at semantically similar phrases for the purpose of phrase table smoothing. The clustering proceeds in a bottom-up fashion, gradually merging similar phrases while alternating back and forth between the two languages.

7 Conclusion and discussion

In this work we proposed methods of labeling phrase pairs to create automatically learned PSCFG rules for machine translation. Crucially, our methods only rely on "shallow" lexical tags, either generated by POS taggers or by automatic clustering of words into classes. Evaluated on a Chinese-to-English translation task, our approach improves translation quality over a popular PSCFG baseline—the hierarchical model of Chiang (2005) —and performs on par with the model of Zollmann and Venugopal (2006), using heuristically generated labels from parse trees. Using automatically obtained word clusters instead of POS tags yields essentially the same results, thus making our methods applicable to all languages pairs with parallel corpora, whether syntactic resources are available for them or not.

We also propose a more flexible way of obtaining the phrase labels from word classes using K-means clustering. While currently the simple hard-labeling methods perform just as well, we hope that the ease of incorporating new features into the K-means labeling method will spur interesting future research.

When considering the constraints and independence relationships implied by each labeling approach, we can distinguish between approaches that label rules differently within the context of the sentence that they were extracted from, and those that do not. The Syntax system from Zollmann and Venugopal (2006) is at one end of this extreme. A given target span might be labeled differently depending on the syntactic analysis of the sentence that it is a part of. On the other extreme, the clustering based approach labels phrases based on the contained words alone.[8] The POS grammar represents an intermediate point on this spectrum, since POS tags can change based on surrounding words in the sentence; and the position of the K-means model depends on the influence of the phrase contexts on the clustering process. Context *insensitive* labeling has the advantage that there are less alternative left-hand-side labels for initial rules, producing grammars with less rules, whose weights can be more accurately estimated. This could explain the strong performance of the word-clustering based labeling approach.

All source code underlying this work is available under the GNU Lesser General Public License as part of the Hadoop-based 'SAMT' system at: `www.cs.cmu.edu/~zollmann/samt`

Acknowledgments

We thank Jakob Uszkoreit and Ashish Venugopal for helpful comments and suggestions and Yahoo! for the access to the M45 supercomputing cluster.

[8] Note, however, that the creation of clusters itself did take the context of the clustered words into account.

References

Phil Blunsom, Trevor Cohn, Chris Dyer, and Miles Osborne. 2009. A Gibbs sampler for phrasal synchronous grammar induction. In *Proceedings of ACL*, Singapore, August.

Peter F. Brown, Vincent J. Della Pietra, Stephen A. Della Pietra, and Robert L. Mercer. 1993. The mathematics of statistical machine translation: parameter estimation. *Computational Linguistics*, 19(2).

David Chiang, Yuval Marton, and Philip Resnik. 2008. Online large-margin training of syntactic and structural translation features. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, Honolulu, Hawaii, October.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

David Chiang. 2007. Hierarchical phrase based translation. *Computational Linguistics*, 33(2).

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the European chapter of the Association for Computational Linguistics (EACL)*, pages 59–66.

Trevor Cohn and Phil Blunsom. 2009. A Bayesian model of syntax-directed tree to string grammar induction. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Singapore.

Michael Galley, Mark Hopkins, Kevin Knight, and Daniel Marcu. 2004. What's in a translation rule? In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*.

Hany Hassan, Khalil Sima'an, and Andy Way. 2007. Supertagged phrase-based statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, Prague, Czech Republic, June.

Dan Klein and Christoper Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Reinhard Kneser and Hermann Ney. 1993. Improved clustering techniques for class-based statistical language modelling. In *Proceedings of the 3rd European Conference on Speech Communication and Technology*, pages 973–976, Berlin, Germany.

Philipp Koehn, Franz J. Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics Conference (HLT/NAACL)*.

Roland Kuhn, Boxing Chen, George Foster, and Evan Stratford. 2010. Phrase clustering for smoothing TM probabilities - or, how to extract paraphrases from phrase tables. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 608–616, Beijing, China, August.

Dekang Lin and Xiaoyun Wu. 2009. Phrase clustering for discriminative learning. In *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics (ACL)*.

Yang Liu, Qun Liu, and Shouxun Lin. 2006. Tree-to-string alignment template for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*.

J. B. MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In L. M. Le Cam and J. Neyman, editors, *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, volume 1, pages 281–297. University of California Press.

Daniel Marcu, Wei Wang, Abdessamad Echihabi, and Kevin Knight. 2006. SPMT: Statistical machine translation with syntactified target language phrases. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Sydney, Australia.

Haitao Mi and Liang Huang. 2008. Forest-based translation rule extraction. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Masaaki Nagata, Kuniko Saito, Kazuhide Yamamoto, and Kazuteru Ohashi. 2006. A clustered global phrase reordering model for statistical machine translation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, ACL-44, pages 713–720.

Franz Josef Och. 1999. An efficient method for determining bilingual word classes. In *Proceedings of the European chapter of the Association for Computational Linguistics (EACL)*, pages 71–76.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*.

Hugo Steinhaus. 1956. Sur la division des corps matériels en parties. *Bull. Acad. Polon. Sci. Cl. III. 4*, pages 801–804.

Ashish Venugopal and Andreas Zollmann. 2009. Grammar based statistical MT on Hadoop: An end-to-end toolkit for large scale PSCFG based MT. *The Prague Bulletin of Mathematical Linguistics*, 91:67–78.

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics*, 23(3).

Andreas Zollmann and Ashish Venugopal. 2006. Syntax augmented machine translation via chart parsing. In *Proceedings of the Workshop on Statistical Machine Translation, HLT/NAACL.*

Andreas Zollmann and Stephan Vogel. 2010. New parameterizations and features for PSCFG-based machine translation. In *Proceedings of the 4th Workshop on Syntax and Structure in Statistical Translation (SSST)*, Beijing, China.

Andreas Zollmann, Ashish Venugopal, Franz J. Och, and Jay Ponte. 2008. A systematic comparison of phrase-based, hierarchical and syntax-augmented statistical MT. In *Proceedings of the Conference on Computational Linguistics (COLING).*