Syntactic Category Learning as Iterative Prototype-Driven Clustering

Jordan Kodner
University of Pennsylvania
Department of Linguistics
Department of Computer and Information Science
jkodner@sas.upenn.edu

Abstract

We lay out a model for minimally supervised syntactic category acquisition which combines psychologically plausible concepts from standard NLP part-of-speech tagging applications with simple cognitively motivated distributional statistics. The model assumes a small set of seed words (Haghighi and Klein, 2006), an approach with motivation in (Pinker, 1984)'s *semantic bootstrapping* hypothesis, and repeatedly constructs hierarchical agglomerative clusterings over a growing lexicon. Clustering is performed on the basis of word-adjacent syntactic frames alone (Mintz, 2003) with no reference to word-internal features, which has been shown to yield qualitatively coherent POS clusters (Redington et al., 1998). A prototype-driven labeling process based on tree-distance yields results comparable to unsupervised algorithms based on complex statistical optimization while maintaining its cognitive underpinnings.

1 Introduction

Supervised part-of-speech (POS) tagging is one of statistical NLP's classic problems (Meteer et al., 1991), but its reliance on large POS-annotated corpora makes it impractical to extend to new domains and unsuitable for low-resource languages without pre-existing annotation. Unsupervised tagging presents an interesting alternative because it does not require labelled training data, but it is a more difficult problem, and typical performance is substantially lower. The fundamental problem of training without examples aside, unsupervised tagging algorithms induce some number of clusters which must be mapped onto a desired tag set. This mapping need not be one-to-one, so error may be introduced going from algorithm output to final results.

Within NLP, unsupervised POS tagging is typically approached as a statistical optimization problem, though implementations vary widely. Brown et al. (1992) and Clark (2003) induce clusters via class-based n-gram models, the latter including morphological information. While these perform reasonably well, more recent approaches have instead been centered around HMMs. Goldwater and Griffiths (2007) and Johnson (2007) both place Dirichlet priors over the multinomial parameters of roughly typical HMM POS taggers. Berg-Kirkpatrick et al. (2010) define a feature-based HMM instead which allows the inclusion of discriminative orthographic information.[1] Haghighi and Klein (2006) implement a model based on Markov random fields (MRFs), an undirected generalization of HMMs.

Haghighi & Klein (H&K) also differs from the above models in that it is minimally supervised with *prototypes*. In their implementation, three words per tag are defined as prototypes or seeds and labelled with their correct tags, and the MRF sorts out which prototypes unknown words are most similar to. They choose the most frequent words per tag imposing the requirement that each seed must only support a single tag, so given the 45-tag Penn Treebank tag set, this yields 135 seeds, a tiny fraction of the tens of thousands of types in the Wall Street Journal corpus. Their MRF is trained on both distributional features and orthographic features follow-

[1] See Christodoulopoulos et al. (2010) for a fuller summary of such models.

Proceedings of the Society for Computation in Linguistics (SCiL) 2018, pages 44-54.
Salt Lake City, Utah, January 4-7, 2018

ing Smith and Eisner (2005). This achieves impressive results on English, but their reliance on orthographic features curtails the model's performance on Chinese.

H&K's and the other computational models above are engineered primarily with performance in mind rather than cognitive plausibility. The complex optimization models in particular are slow, taking tens of hours to complete, while the simpler n-gram based models run in tens of minutes (Christodoulopoulos et al., 2010). But fundamentally, they are all tools which take advantage of statistical, especially distributional information.

For well over half a century now, it has been understood that children make use of distributional cues in the process of *syntactic category* (roughly POS) assignment as well. For example, Brown (1957)'s classic study found that children recognize a nonce word "sib" as a noun when if it is introduced to them in a sentence like "This is a sib," but prefer to label it as a verb if it is introduced in "It is sibbing." Along similar lines, Shi and Melançon (2010) present nonce words to year-old children and watch whether they then tend to focus on an image depicting an object or an action. They find that children presented with "the mige" prefer to look at the object image over the action image, consistent with them understanding that the word should be a noun. These experiments demonstrate that learners make use of distributional information (e.g., "a/the ___") even with limited exposure. Studies of child-directed corpora suggest the presence of distributional cues as well in a more naturalistic setting (Maratsos, 1979; Redington et al., 1998).

Children's sensitivity to such distributional information has been operationalized through the notion of *frequent frames*, single-word contexts on either side of an item (Mintz, 2003). Experimental evidence demonstrates that children who are exposed to items within the same frame, (e.g, "the ___ is") treat those items as members of the same class. However, the large number classes induced from frequent frames do not provide a clean one-to-one mapping to syntactic categories (Chemla et al., 2009). Syntactic frames can be seen as a purely structural cue, but semantic information play a role as well. As described by the *semantic bootstrapping* hypothesis (Pinker, 1984), children have innate rules for mapping real-world semantic onto syntactic categories. For example, actions should be verbs, objects should be nouns, and so on. These then serve as anchors in the input to provide early distributional context. The validity of semantic bootstrapping's claims of innateness and the exact nature and number of syntactic classes are not critical for the present work, rather it is sufficient that bootstrapping guides children to some kind of categorization. Experimental work has long confirmed that semantic bootstrapping basic prediction holds and that children really do associate actions and concrete objects with verbs and nouns (Gleitman et al., 2005; Pinker, 1984; Rondal et al., 1987).

We develop a computational implementation for semantic bootstrapping in syntactic frames which draws inspiration from hierarchical clustering and from Haghighi and Klein (2006)'s prototype-driven model for POS tagging. Since we know that children do not wait patiently to build up large vocabularies replete with distributional information before attempting to assign syntactic categories, our algorithm runs iteratively on the lexicon as it grows, revising category assignments as more evidence comes in. Additionally, we discard all word-internal morphological and phonological information to test the distributional cues on their own. The rest of this paper is organized as follows: section 2 introduces the iterative prototype-driven clustering model for tagging along with the basic insights behind it. Section 3 discusses the problem of evaluation for unsupervised POS tagging, provides results for child-directed English under various conditions, comparative results across nine other languages, and a comparison with H&K on English and Chinese. Section 4 reviews the model's cognitive plausibility and discusses possible extensions to the algorithm.

2 The Model

The primary insight for this work comes from the observation that words may be grouped into rough part-of-speech clusters on the basis of their single-word contexts. This is an implicit assumption of n-gram tagging models (Clark, 2003; Brown et al., 1992) for POS tagging, and the feasibility of distributional contexts for distinguishing between syntactic categories has been explicitly studied in cognitive

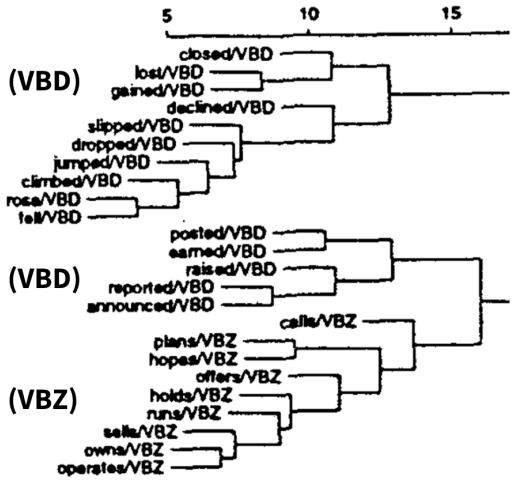

Figure 1: Example of the cutting problem from Parkes et al., (1998) with additional markup for readability

literature as well (Redington et al., 1998).

Parkes et al. (1998) provide a straightforward implementation for simple agglomerative clustering without orthographic input that serves to highlight the benefits and drawbacks of the approach. For each of the 400 most frequent words in the Wall Street Journal corpus, Parkes et al. tabulate the frequencies of left and right-context types. Using symmetricized KL-divergence ($\mathrm{KL}(p||q) + \mathrm{KL}(q||p)$) as a distance metric, they perform agglomerative clustering on those 400 types and note the qualitative purity of the resulting clusters. The immediately obvious problem is that there is no perfect mapping from clusters to tags. Figure 1 demonstrates this. Here, there are two pure VBD (past tense verb) clusters, but they cannot be joined without including a cluster of VBZ (present tense verb). Any simple cutting algorithm fails by either incorrectly postulating two VBD classes (VBD_1, VBD_2, VBZ), or postulating a mixed VBD/VBZ class. This is a general problem, independent of the choice of tag set, so while the WSJ VBD and VBZ should probably be collapsed into a single V class from a cognitive point of view, that alone does not solve it.

The simple counts over single-word contexts which Parkes et al. (1998) employ are equivalent to the syntactic frames described in the acquisition literature. So while the Parkes et al. study is neither presented as nor performs as a tagging algorithm, it holds promise from a cognitive perspective and provides clusters similar to those in the cognitive literature. The next two sections describe a prototype-driven labeling algorithm built on top of these clusters. The algorithm is described in two parts, first the inner loop basic algorithm which operates on a vocabulary of fixed size, then the outer loop, an iterative extension that operates on a growing lexicon.

2.1 Prototype-Driven labeling

Prototype-driven labeling operates over the trees created by hierarchical clustering and makes up the inner loop of the category labeling algorithm. Two vectors, one each of immediate left and right context type counts, are tabulated for the top k most frequent words in a corpus. Next, these k types are grouped by agglomerative clustering. This is slow ($O(k^2 \log(k))$ in a naïve implementation, but that can be mitigated with incremental clustering or other approaches. KL-divergence over the concatenated left and right context vectors serves as the distance metric. KL is not symmetric, so for each pair of vectors v and w, the sum of the KL-divergence between v and w and w and v is used as in Equation 1, where a and b are two words, and v and w are their corresponding context vectors.[2]

$$\begin{aligned} \mathrm{d}(a,b) &= \mathrm{KL}(v||w) + \mathrm{KL}(w||v) \\ &= \sum_i v_i \log \frac{v_i}{w_i} + w_i \log \frac{w_i}{v_i} \end{aligned} \quad (1)$$

In agglomerative clustering, it is necessary to define a *linkage criterion* that allows the constructed subtrees to be compared to each other along with individual words. The distance between the subtrees here is taken as the average distances between their members. This only performs slightly better than taking the minimum distance between any two members of the subtrees, and it can be updated on the fly as new members are added to the tree. Equation 2 gives a formal description of the average distance criterion. Clustering greedily joins whatever pair of words or subtrees has the smallest distance between them at each step. As a result, the final tree

[2]This is symmetric and is equivalent to Jenson-Shannon-divergence for the purpose of rank ordering, but it is marginally easier to compute.

can be thought of as a rank ordering of word similarities. There is no need to track the actual vector values of individual leaves once they have been joined into subtrees.

$$\mathrm{D}(A, B) = \frac{1}{|A||B|} \sum_{a \in A} \sum_{b \in B} \mathrm{d}(a, b) \qquad (2)$$

Before beginning the category labeling process, seed words are labelled in the tree to simulate semantic bootstrapping. This can be done by picking the three most frequent words per actual label as in H&K, or seeds can be chosen by contextual saliency. For a set of n tags, there is a *maximum* of $3n$ seeds in the tree. But if k is small, the actual number will usually be smaller. It is easy to see why when looking at the WSJ corpus and tag set, but the same is true of smaller cognitively-motivated tag sets as well. The three seeds with the FW "foreign word" tag are *perestroika*, *de*, and *kanji*.[3] None of these is in the top thousand most frequent words, so for $k = 1000$, there must be fewer than $3n$ seeds.

Once the seeds are assigned, the labeling can proceed for the remaining words. This can be equivalently implemented during the join steps of agglomerative clustering itself or on the completed agglomerative tree. Initially, the leaves of the tree are treated as labelled if they are seeds and unlabelled otherwise. Iterating over joins in the order that they occur from the leaves upward, if an unlabelled subtree is joined with a labelled one, each of its leaves is assigned the most common label from the labelled subtree. This is easiest to explain by example. In the simplest case, a seed is joined with an unlabelled word, and that word is assigned the same label as the seed (Figure 2 (left)). In the general case, when a labelled subtree (containing ≥ 1 seed) is joined with an unlabelled subtree, every word in the unlabelled subtree is assigned the most common label from the labelled subtree. For example, if a subtree that is 90% nouns and 10% verbs is joined with an unlabelled tree, every word in the unlabelled tree is tagged as a noun (Figure 2 (right)). This approach provides a solution to the cutting problem: it is not an issue if multiple clusters map to the same part-of-speech because what matters is the clusters' proximity to the seeds, not to each other. Referring back to the Parkes et al. example in Figure 1, the model will label the tree perfectly as long as each small cluster is centered around a seed.

[3]The frequent FW words effectively date this corpus.

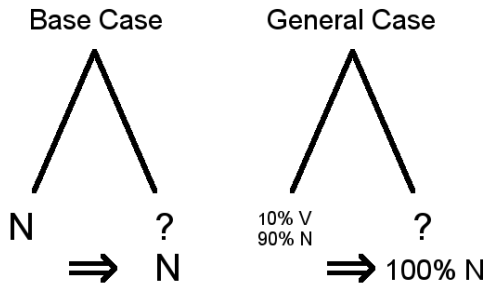

Figure 2: Visual depiction of assignment. (Left) leaves. (Right) subtrees.

2.2 Iterative Prototype-Driven labeling

The basic prototype-driven labeling algorithm operates on a hierarchical clustering of fixed size k. The labeling step itself is fast, but for large k, computing the pairwise distance function becomes becomes slow. It also runs afoul of experimental evidence on category learning. Children do not wait patiently to build a large vocabulary complete with distributional information before even attempting to assign syntactic categories. Rather, they assign what they can when they can, potentially revising their assignments as they go along (Pinker, 1984; Gleitman et al., 2005).

An iterative extension to the labeling algorithm solves both the practical and empirical problems. Instead of running once on a fixed large k, the algorithm is run multiple times on a sequence of lexicons with monotonically increasing sizes $K = (k_0, k_1, \ldots, k_m)$ where each lexicon contains the top k_i most frequent types in the corpus of study. This is meant to approximate which words a learner is statistically most likely to have heard early in development. The algorithm begins by labeling a small k_i, which is always quick to compute. Then it has working hypotheses for the first k_i words while it re-evaluates those words and learns new words up to k_{i+1}. While not purely online (Hewitt, 2017), this provides the sort of incremental advancement seen in the literature and is amenable to online implementation (Guedalia et al., 1999). Additionally,

once a partial tree has been built, it becomes unnecessary to compute pairwise distance between every new vocabulary item and existing item. Once it is discovered which seeds are close to a new word, it is sufficient to only compute distances between the new words and those words which are close to its nearest seeds to achieve a kind of clustering. This dovetails nicely with some efficient agglomerative clustering implementations, e.g., (McCallum et al., 2000)'s canopies.

The algorithm attempts classification on frequent words multiple times, so it has the opportunity to use evidence from early classification to inform later attempts before picking its best guess in the end. To accomplish this, a *confidence* value is set for each subtree assignment operation. This is defined as the purity of the assigning tree, so a subtree that is 100% adjectives assigns members of an unlabelled subtree with a confidence of 1.0, while a 40% adjective, 30% noun, 30% verb subtree only assigns with a confidence of 0.4. Intuitively, a pure subtree is likely to represent an actual category cluster, while an impure tree is either a higher-level clustering of category clusters or garbage. Then at the end of each iteration, all words assigned with a confidence above some fixed threshold are added to the seed set and become prototypes for the next iteration. All other words are reassigned in subsequent iterations until they become prototypes themselves or the experiment finishes. At the end, all words which were never added to the prototype set are assigned the highest confidence label that they encountered during any iteration.

This iterative extension greatly improves performance because it expands the seed set as the size of the trees increase. This guarantees that the ratio of seeds to non-seeds is always relatively high, which improves assignment performance as long as the augmented seed set remains high accuracy. The ratio of seeds to non-seeds is always high for small k, but without this process, the ratio is prohibitively low at large k. Cognitively, this corresponds to the fact that children build up a lexicon of known words as they are available to them as evidence when categorizing words learned later.

3 Evaluation and Results

In this section, the iterative prototype-driven labeling model is evaluated on corpora from a wide range of languages. First, we discuss results on the English Brown (Harvard) corpus within the CHILDES corpus of child-directed speech (MacWhinney, 2000), since this serves as an in-domain application. Performance is compared on a range of tag sets, seed set sizes, and seed selection procedures. Next, we test on a range of languages from the Universal Dependency Treebank (McDonald et al., 2013) in order to gain insight into how language specific factors influence results while keeping extra-linguistic factors constant to the extent possible. Finally, we run on the WSJ portion of the Penn Treebank (Marcus et al., 1993) and on the Penn Chinese Treebank (Xue et al., 2005) in order to compare against H&K's prototype-driven model and ground this work in the wider field.

3.1 Evaluation Metrics

Evaluation for supervised POS tagging is straightforward. Sentences from a test corpus are labelled, then for each token, the proposed label is compared to the human annotated label. The simplest of such metrics is *one-to-one token accuracy*. On the other hand, the problem of evaluation is more complicated for unsupervised algorithms. Results are broadly not comparable across experiments because a wide of evaluation metrics are employed.[4]

Token-based metrics are an excellent option when the task is to automatically annotate running text with part-of-speech tags, but they have undesirable traits when applied to syntactic category learning. If syntactic category learning is analogous to labeling types in a dictionary or lexicon, then labeling sequences of text just obfuscates results. Type frequencies are not uniform across a corpus, so token-based metrics weight assignments to frequent types higher than assignments to infrequent types. For example, an algorithm which labels the pronoun "you" incorrectly will be punished more severely than one which labels the pronoun "whomever" incorrectly simply because "you" is more common than "whomever" in most corpora. This is particularly problematic because word frequencies follow

[4] See (Christodoulopoulos et al., 2010).

a distribution where just a few types are hundreds of times more frequent than most others. Mislabeling any pronoun is hundreds of times more damaging than labeling almost any noun from our daily lives (like "table" or "bug") incorrectly. Mislabeling any common noun is much more damaging than mislabeling a rare noun, and noun frequencies can be highly corpus specific. This makes it difficult to gauge the relative performance of different models.

A *one-to-many type accuracy* is a better choice for scoring syntactic category learning because each item is weighted the same for scoring. In natural language, types can potentially support multiple labels, so a one-to-many metric is needed to account for this. For example, "bat" could be a noun or a verb, so an algorithm which classifies it as either should be correct. The *many-to-one* accuracy used in some unsupervised models does not make sense here because the model does not output clusters which need to be mapped. The algorithm never demarcates clusters and instead assigns labels to individual items.

The difference between token and type-based metrics becomes clear when calculating the baselines for our model. Scoring by type accuracy, the baseline is arrived at by scoring the initial seeds as correct and marking everything else wrong. This typically yields a score of under 10%, often under 1%, and corresponds directly to the proportion of types which are selected as seeds. However, this tends to correspond to a $> 50\%$ token accuracy because seeds are drawn from the most frequent types. A final type score of, say, 70% is more meaningful than a final token score of 70% because the improvement over the baseline is greater.

3.2 Experiments on English CHILDES

The CHILDES corpus contains transcriptions of naturalistic child-directed speech. The English subset studied here consists of all caregiver text extracted from Adam, Eve, and Sarah of the Harvard (Brown) corpus, yielding 8,307 types and 588,888 tokens. The tag set used in the Brown corpus consists of 55 idiosyncratic tags, and this is tested along with a mapping that reduces these to an 8-tag (+SKIP[5]) set (DT, IN, JJ, NN, PRP, RB, VB, SKIP). Table 1 reports the impact of k on performance and indicates the best achieved type accuracy results training on the full Brown tag set then mapping to the reduced set (Brown-to-Reduced) for scoring for final $k = 100$, $k = 1000$, and $k = 8307$.[6] As expected, percent correct decreases for higher k because infrequent words which occur only once or twice in the corpus provide weaker distributional information than frequent words do. The number of seeds used never reaches 55*3 because some tags do not appear in the top k words. This is the same as the WSJ corpus FW-problem described earlier.

k	# Seeds	Baseline %	Type Acc.
100	58	58.0	94.0
1000	100	10.0	81.2
8307	130	1.6	62.8

Table 1: CHILDES type accuracy by tree size. Baseline indicates the contribution from the seeds alone. Accuracy presented as percents

To determine what impact the choice of tag set and number of seeds have on the results, the experiment was run and evaluated directly on the Brown tag set as well as the reduced tag set. Reduced tag set experiments were run with 3 seeds per tag and 11 per tag. The Brown-to-reduced tests smooth out the difference between related tags when computing accuracy scores. The difference between B-to-R and straight Brown in Table 2 implies that the model struggles to differentiate some of the Brown tags. This could be because some of the more eccentric Brown tags do not actually distinguish distributionally coherent classes. The difference between the 3-seed and 11-seed results indicates that performance largely depends on the number of seeds. Note, however, that the baseline is still quite low even with 11 seeds per tag.

The point of syntactic frames is that they are available as primary evidence early on, and nobody who works on them would argue that they are the only source of evidence, so it is unsurprising that performance declines using frames alone for large k. If syntactic frames are indeed useful, then we would expect their application on early vocabulary to carry benefits downstream, and this this is what Table 3

[5]Tags which do not correspond to any of the 8 are mapped to SKIP and not scored.

[6]k sequence $K = (100, 500, 900, 1000, 2000, 4000, 6000, 8307)$ was used.

k	Tag Set	# Seeds	Baseline	Type Acc.
1000	B-to-R	100	10.0	**81.2**
1000	Brown	100	10.0	70.3
1000	Reduced	24	2.4	51.8
1000	Reduced	85	8.5	80.6
8307	B-to-R	130	1.6	**62.8**
8307	Brown	130	1.6	44.0
8307	Reduced	24	0.3	25.3
8307	Reduced	85	1.0	53.3

Table 2: CHILDES type accuracy by tag set and seed set size

Tag Set	# Seeds	Basic Acc.	Iter. Acc.
B-to-R	130	44.2	**62.8**
Brown	130	27.7	**44.0**
Reduced	24	9.3	**25.3**
Reduced	85	44.0	**53.3**

Table 3: Comparison between CHILDES basic and iterative prototype-driven labeling performance

shows. It compares the results of iterative application from $k = 100$ to $k = 8307$ used for the previous experiments to a single non-iterative application $k = 8307$. The iterative results are 9 to 18 points higher, demonstrating that syntactic frames applied to early vocabulary set up better performance later. The iterative application of the algorithm which expands the seed set as the lexicon grows is critical to model performance.

Up to this point, seeds have been selected post hoc by corpus frequency, which cannot possibly be how children use semantic bootstrapping. To correct for this, a set of 82 lower frequency but high saliency seed words was selected for comparison based on studies of salience in child-directed speech (Carlson et al., 2014). Tables 5 lays out the result achieved with salient seeds on the reduced tag set, which can be compared to the 11 frequent seed results from Table 2. The type accuracy is lower and roughly what is to be expected given the size of the seed set even though the seeds themsleves are of lower frequency on average.

Tag Set	# Seeds	Baseline	Token Acc.
B-to-R	130	50.2	82.7
Brown	130	48.0	73.4
Reduced	24	28.8	60.0
Reduced	85	52.3	**83.5**

Table 4: CHILDES one-to-one token accuracy performance

k	# Seeds	Baseline	Type Acc.
1000	74	7.4	73.4
8307	82	1.0	49.5

Table 5: CHILDES type accuracy with salient seeds

Language	# Seeds	k=1,000	k=10,000
French	28	77.92	62.07
German	30	79.04	26.52
Indonesian	30	75.84	65.21
Italian	26	54.26	37.08
Japanese	24	47.78	48.31
Korean	26	33.47	39.19
Portuguese	28	65.40	49.44
Spanish	29	63.41	46.14
Swedish	37	51.10	33.96

Table 6: UTB type accuracy by language

3.3 Experiments on Other Languages

The CHILDES results suggest that information from syntactic frames is useful for assigning syntactic categories from English child-directed speech. In order to compare performance across other languages as well, we apply the algorithm to nine languages from the Universal Dependency Treebank: German, Spanish, French, Indonesian, Italian, Japanese, Korean, Brazilian Portuguese, and Swedish. These corpora share a 10-tag tag set (excluding punctuation and non-word tags) and were compiled for the same task, which means that different performances reflect differences in the languages themselves to the extent possible. In order to more closely align these experiments to syntactic category learning, no seeds were provided for the punctuation (_, MAD, MID, PAD) or non-word (X) tags, and punctuation and non-words were not scored.[7]

Performance varies substantially across languages. At $k = 1000$, type accuracy ranges from the low 30s (Korean), to the high 70s (German, French, Indonesian), and at $k = 10000$ from the mid 20s (German) to the 60s (French, Indonesian). Much of this variation can be explained linguistically with the important caveat that extra-linguistic factors in the corpora must still be at play.

Languages with complicated inflection are at a

[7]Each language was tested with $confidence = 0.3$, 0.5, 0.7, 0.9, and 1.0. Only the best confidence for each language is reported. The same k sequence $K = (100, 500, 900, 1000, 2000, 5000, 9000, 10000)$ was used each time.

disadvantage because the distributional context information of their roots is spread across inflectional forms, and without reference to word-internal features, it is impossible to group these forms by root to pool that information. For example, while all the distributional information for English BLUE is collected in the contexts for the word "blue," German spreads its distributional information across the contexts for its six inflected forms, "blau," "blauer," "blauen," "blauem," "blaue," and "blaues," and our algorithm has no way to combine them.

Another way to think about this is to compare token/type ratios between languages, which are equivalent to how many syntactic frames on average contribute to the context vectors for each type. A low ratio indicates that the typical context vector is sparser because it accounts for information from fewer frames. Languages with complex inflection naturally have more types and a lower token/type ratio.

The particularly poor performance for Korean and Japanese is at least partially due to the UTB corpus tokenization which attaches phrase final syntactic clitics to the preceding word as opposed to standalone tokens. This is in line with the traditional conception of "word" boundaries in these languages (called *bunsetsu* in Japanese and *eojeol* in Korean), but is not an obviously correct choice from a cognitive standpoint, and would be equivalent to not segmenting the possessive *'s* clitic in English. As shown in Table 7, *bunsetsu* tokenization creates multiple words for APPLE and PEAR and two example clitics, each with disjoint right contexts, while standalone tokenization would yield one word each for APPLE and PEAR with both clitics as right contexts. This technical choice effectively renders the clitics as inflections like in German rather than as useful syntactic context information. Also, it is unclear why Korean and Japanese perform better at high k than low k at the reported confidences.[8]

Languages with freer word order are also at a disadvantage because the entire premise of syntactic frames assumes that the syntax forces categories to appear adjacent to certain other categories. A freer word order means more violations of this assumption relative to the number of frames attested, which manifests as more uniform and less discriminable context vectors.

Tokenization	Text Strings	Right Frames
Bunsetsu	ringo-ga X	ringo-ga: {X}
	ringo-wo Y	ringo-wo: {Y}
	nashi-ga Z	nashi-ga: {Z}
	nashi-wo W	nashi-wo: {W}
Standalone	ringo ga X	ringo: {ga, wo}
	ringo wo Y	nashi: {ga, wo}
	nashi ga Z	ga: {X, Z}
	nashi wo W	wo: {Y, W}

Table 7: Right frames for *apple-NOM X*, *apple-ACC Y*, *pear-NOM Z*, and *pear-ACC W*

3.4 Comparison with Haghighi & Klein

In order to ground the performance of this algorithm in the broader research context, we compare results with Haghighi and Klein (2006) since it is similarly semi-supervised. The H&K PROTO model represents the fairest comparison because they were calculated on atypically small datasets, fractions of the English Wall Street Journal and Mandarin Chinese Treebank. They report *one-to-one token* accuracy, so we follow suit.

The WSJ dataset contains 16,839 types across 193,000 tokens[9], capitalization removed.[10] The model takes 3 seeds for each of the 45 tags in the Penn Treebank tag set. Table 8 presents WSJ type accuracy at $k = 1000$ and $k = 10000$. The 1000-type model takes about five minutes to train, while the 10,000-type model takes just over eight hours under a naïve clustering implementation. The performance is overall worse than for CHILDES, likely because the type diversity is much higher than what would be expected in child-directed speech. CHILDES has a token/type ratio of 71.4, and the WSJ's ratio is almost three times lower at 27.8.

The CTB dataset contains 8,842 types across only 60,000 tokens and is annotated with a 33-tag set similar to the Penn Treebank tag set. Table 8 compares the Chinese type results to English. The WSJ model outperforms the CTB model, which, among factors like corpus size, is probably related to the number of seeds present and the token/type ratio.

[8] This only happens for Japanese and Korean, and only at $c = 0.7, 0.9$ for both.

[9] Starting from section 2. H&K report 18,423 types.

[10] H&K retain capitalization as a model feature.

Corpus	k	# Seeds	Baseline	Type Acc.
WSJ	1000	95	9.5	57.9
WSJ	10000	95	1.0	30.2
CTB	1000	74	7.4	50.4
CTB	8842	74	0.8	27.5

Table 8: Wall Street Journal and Chinese Treebank Type Accuracy

Model	# Seeds	Base	Top k	All	All+
k=1000	95	40.5	74.3	54.7	60.2
k=10000	95	40.5	63.2	60.9	61.4
H&K06	135	41.3	-	**68.8**	-

Table 9: Wall Street Journal token accuracy and comparison

The CTB model was trained on all types, but the WSJ model was only trained up to $k = 10000$ (accounting for 96.46% of tokens) because of time considerations, which makes token accuracy less straightforward to compute. Three numbers are provided: *Top k* only scores words appearing in the top k by frequency, *All* scores the top k as before and marks all other tokens incorrect, and *All+* instead assigns tokens outside the top k the type of their nearest seed by symmetricized KL-distance. Table 9 compares WSJ results with H&K. Overall performance is lower but still above 60.

Table 10 compares against H&K's CTB results. Here, the iterative prototype-driven labeling model is clearly superior. Even the $k = 1000$ *All* score is 8 points higher than H&K's MRF model, and the best $k = 10000$ *All+* score is over 15 points higher. That large discrepancy is due to H&K's reliance on orthographic features. Since suffix n-grams are meaningless in Chinese, they were forced to discard them from their model. It seems that the reason why H&K outperformed our model by over 7 points on English was that their model made reference to the word-internal features which ours discards. Chinese represents a more level playing field which demonstrates how our model makes good use of sparse distributional information.

Model	# Seeds	Base	Top k	All	All+
k=1000	74	29.5	62.80	46.9	50.4
k=8841	74	29.5	-	**54.1**	-
H&K06	99	34.4	-	39.0	-

Table 10: Chinese Treebank token accuracy and comparison

4 Discussion and Future Work

The minimally supervised iterative prototype-driven labeling algorithm laid out in this paper leverages only simple distributional statistics over adjacent word-forms to perform syntactic category labeling. Nevertheless it achieves English and Chinese results comparable to and surpassing the more complex H&K model respectively, and similar performance on French and Indonesian, though it struggles on other languages.

The most glaring deficiency in the model is that it lacks any notion of word-internal features. Throwing away this critical information prevents it from identifying and utilizing affixes as cues for syntactic categories and from pooling evidence from word-forms that share common roots. One promising avenue of research therefore, is determining how to cleanly incorporate morphological information into the clustering algorithm. This will be helpful for languages like Japanese under *bunsetsu* tokenization or German, and absolutely critical for language families with complex agglutinative or polysynthetic morphology like Turkic, Eskimo-Aleut, or Bantu.

Perhaps less obviously, the model is missing out on an important generalization by only training on lexical syntactic frames. Children are not only sensitive to the specific lexical items surrounding a word, but also the syntactic category of those items (Reeder et al., 2013). For example, a word preceded by a determiner and followed by a noun (DT ___ N) is almost certainly an adjective, regardless of which determiner and which noun it is, so even though the adjectives in "the shiny ball" and "a scary cube" have different lexical contexts, they share the same category context. Relevant to our algorithm, category contexts reveal distributional similarities that are hidden by lexical contexts alone.

Acknowledgments

We would like to thank Charles Yang and Mitch Marcus for their helpful insights along with the rest of the Penn LORELEI team. This work was funded by the DARPA LORELEI program under Agreement No. HR0011-15-2-0023 and by the U.S. Army Research Office (ARO) through an NDSEG fellowship awarded to the author.

References

Taylor Berg-Kirkpatrick, Alexandre Bouchard-Côté, John DeNero, and Dan Klein. 2010. Painless unsupervised learning with features. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 582–590. Association for Computational Linguistics.

Peter F Brown, Peter V Desouza, Robert L Mercer, Vincent J Della Pietra, and Jenifer C Lai. 1992. Class-based n-gram models of natural language. *Computational linguistics*, 18(4):467–479.

Roger W Brown. 1957. Linguistic determinism and the part of speech. *The Journal of Abnormal and Social Psychology*, 55(1):1.

Matthew T Carlson, Morgan Sonderegger, and Max Bane. 2014. How children explore the phonological network in child-directed speech: A survival analysis of childrens first word productions. *Journal of memory and language*, 75:159–180.

Emmanuel Chemla, Toben H Mintz, Savita Bernal, and Anne Christophe. 2009. Categorizing words using frequent frames: what cross-linguistic analyses reveal about distributional acquisition strategies. *Developmental science*, 12(3):396–406.

Christos Christodoulopoulos, Sharon Goldwater, and Mark Steedman. 2010. Two decades of unsupervised pos induction: How far have we come? In *Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing*, pages 575–584. Association for Computational Linguistics.

Alexander Clark. 2003. Combining distributional and morphological information for part of speech induction. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics-Volume 1*, pages 59–66. Association for Computational Linguistics.

Lila R Gleitman, Kimberly Cassidy, Rebecca Nappa, Anna Papafragou, and John C Trueswell. 2005. Hard words. *Language Learning and Development*, 1(1):23–64.

Sharon Goldwater and Tom Griffiths. 2007. A fully bayesian approach to unsupervised part-of-speech tagging. In *Annual meeting-association for computational linguistics*, volume 45, page 744.

Isaac David Guedalia, Mickey London, and Michael Werman. 1999. An on-line agglomerative clustering method for nonstationary data. *Neural computation*, 11(2):521–540.

Aria Haghighi and Dan Klein. 2006. Prototype-driven learning for sequence models. In *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, pages 320–327. Association for Computational Linguistics.

Yang Charles Hewitt, John. 2017. Bootstrapping for syntactic categories. In *Cognitive Science Society*, London, UK.

Mark Johnson. 2007. Why doesn't em find good hmm pos-taggers? In *EMNLP-CoNLL*, pages 296–305.

Brian MacWhinney. 2000. *The CHILDES project: The database*, volume 2. Psychology Press.

Michael Maratsos. 1979. How to get from words to sentences. *Perspectives in Psycholinguistics*.

Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330.

Andrew McCallum, Kamal Nigam, and Lyle H Ungar. 2000. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178. ACM.

Ryan T McDonald, Joakim Nivre, Yvonne Quirmbach-Brundage, Yoav Goldberg, Dipanjan Das, Kuzman Ganchev, Keith B Hall, Slav Petrov, Hao Zhang, Oscar Täckström, et al. 2013. Universal dependency annotation for multilingual parsing. In *ACL (2)*, pages 92–97.

Marie Meteer, Richard Schwartz, and Ralph Weischedel. 1991. Studies in part of speech labelling. In *Proceedings of the workshop on Speech and Natural Language*, pages 331–336. Association for Computational Linguistics.

Toben H Mintz. 2003. Frequent frames as a cue for grammatical categories in child directed speech. *Cognition*, 90(1):91–117.

Cornelia Parkes, Alexander Malek, Er M Malek, and Mitchell Marcus. 1998. Towards unsupervised extraction of verb paradigms from large corpora.

Steven Pinker. 1984. Language learnability and language development.

Martin Redington, Nick Chater, and Steven Finch. 1998. Distributional information: A powerful cue for acquiring syntactic categories. *Cognitive science*, 22(4):425–469.

Patricia A Reeder, Elissa L Newport, and Richard N Aslin. 2013. From shared contexts to syntactic categories: The role of distributional information in learning linguistic form-classes. *Cognitive psychology*, 66(1):30–54.

Jean A Rondal, Martine Ghiotto, Serge Brédart, and Jean-François Bachelet. 1987. Age-relation, reliability and grammatical validity of measures of utterance length. *Journal of child language*, 14(3):433–446.

Rushen Shi and Andréane Melançon. 2010. Syntactic categorization in french-learning infants. *Infancy*, 15(5):517–533.

Noah A Smith and Jason Eisner. 2005. Contrastive estimation: Training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 354–362. Association for Computational Linguistics.

Naiwen Xue, Fei Xia, Fu-Dong Chiou, and Marta Palmer. 2005. The penn chinese treebank: Phrase structure annotation of a large corpus. *Natural language engineering*, 11(2):207–238.