Unsupervised Approaches to Sequence Tagging, Morphology Induction, and Lexical Resource Acquisition

Reza Bosaghzadeh, Nathan Schneider

Language and Statistics II, Fall 2008

Contents

Abstract

We consider unsupervised approaches to three types of problems involving the prediction of natural language information at or below the level of words: sequence labeling (including part-of-speech tagging); decomposition (morphological analysis and segmentation); and lexical resource acquisition (building dictionaries to encode linguistic knowledge about words within and across languages). We highlight the strengths and weaknesses of these approaches, including the extent of labeled data/resources assumed as input, the robustness of modeling techniques to linguistic variation, and the semantic richness of the output relative to the input.

1 Introduction

In the last few years, many of the innovations in natural language processing research have been unsupervised techniques for predicting linguistic information from data. Given the abundance of unannotated text data, unsupervised approaches are very appealing for natural language processing. In this report we present unsupervised solutions to three tasks which achieve close to state-of-the-art results in domains previously dominated by fully supervised systems. **Supervision** is the extent to which a method for predicting the unobserved structure of new data relies upon having seen training data labeled with the same type of structure. Many unsupervised algorithms use data that has been annotated in some way, just not in the way that is to be predicted.

Linguistic structure comes in many forms, and is often quite costly to annotate manually. In this paper, we consider unsupervised problem areas corresponding to three categories of linguistic structure. The first problem area is **sequence labeling** (§2), for which we focus on part-of-speech tagging. We describe and compare two methods outlined in (Smith and Eisner, 2005) and (Haghighi and Klein, 2006). The second problem area is **morphological analysis** (§3; primarily segmentation of words or morphemes); the methods addressed are those outlined in (Monson et al., 2007, 2008a,b), (Snyder and Barzilay, 2008), and (Goldwater et al., 2006; Goldwater et al., in submission). For the third problem area, **lexical resource acquisition** (§4), we address the tasks of learning bilingual lexicons (Haghighi et al., 2008), labeling semantic roles for verbs (Grenager and Manning, 2006), and extracting narrative event chains (Chambers and Jurafsky, 2008) relating verbs. We conclude with a discussion of overall language variability along with required input, training procedures, and input and output complexity for each approach.

Not addressed here are a multitude of other tasks which use unsupervised methods: these include parsing and grammar induction (e.g. Klein, 2005; Smith, 2006; Liang et al., 2007); coreference resolution (e.g. Haghighi and Klein, 2007; Poon and Domingos, 2008); and many others.

2 Unsupervised Tagging of Sequences

Our first problem area concerns the prediction of labels for a sequence of observed outputs. The dominant task in this area has been part-of-speech (POS) tagging, though approaches to POS tagging can be applied to other sequence-based classification tasks, such as segmentation (e.g. *information field segmentation* (Grenager et al., 2005)—see §2.2 below). In the models we discuss, it is assumed that:

- at test time, the system is presented with a sequence of discrete observations (*emissions*);

- the task is to assign a single label to each emission in the sequence; and

- the model is given a finite, discrete set of possible labels.

Unsurprisingly, supervised methods outperform unsupervised methods for the problem of sequence tagging.

2.1 Contrastive Estimation: Preferring the Seen over the Unseen

2.1.1 Exploiting Implicit Negative Evidence

Smith and Eisner (2005) build unsupervised models for sequence labeling that exploit implicit negative evidence in a computationally efficient way. They describe an unsupervised parameter estimation method called *Contrastive Estimation* and apply it to a sequence labeling problem—POS tagging given a tagging dictionary and unlabeled text. They show that Contrastive Estimation outperforms EM when trained on the same features, and is somewhat robust to variation in the size of the tagging dictionary.

The authors point out that the motivation behind EM is to push probability mass toward the training examples. The key idea behind Contrastive Estimation (CE) is that it takes probability mass from *implicit negative examples* and gives the mass to the positive training examples. This allows CE to incorporate additional domain knowledge (that unseen word sequences are likely to be ungrammatical), which leads to improvements in accuracy.

A key contribution of this work is the hypothesis that each training example provides a set of implicit negative examples, which can be obtained by mutating a regular training example by way of deletions, transpositions, and other operations. By taking probability mass away from these negative examples and in turn favoring the original un-mutated training example, (Smith and Eisner, 2005) provide a compelling alternative to EM for the POS tagging task.

The availability of implicit negatives examples hinges on the assumption that local mutations of a correct example will likely create incorrect examples. This assumption is plausible and is experimentally tested. For example, the authors consider the sentence

"Natural language is a delicate thing."

Suppose one chooses one of its six words at random and removes it. For this sentence, there is 2/3 probability that the resulting sentence will be ungrammatical. Or, one could randomly choose two adjacent words and transpose them—none of these transformations produce valid conversational English. During training, Contrastive Estimation keeps in mind the generated negative examples as well as the positive training examples.

When estimating parameters for CE, the authors take an approach similar to EM but also take into consideration the implicit negatives examples. They let $\overrightarrow{x} = \langle x_1, x_2, \ldots \rangle$ be the observed example sentences, where each $x_i \in \mathcal{X}$, and let $y_i^* \in \mathcal{Y}$ be the unobserved correct hidden structure for x_i (e.g. a POS sequence). Then we seek a model parameterized by $\overrightarrow{\theta}$ such that the unknown correct analysis y_i^*, is the best analysis for x_i. To find $\overrightarrow{\theta}$, one typically uses the EM algorithm to maximize

$$\prod_i p(X = x_i | \overrightarrow{\theta}) = \prod_i \sum_{y \in \mathcal{Y}} p(X = x_i, Y = y | \overrightarrow{\theta})$$

where X is a random variable over sentences and Y is a random variable over analyses.

Contrastive estimation takes a new approach and instead maximizes

$$\prod_i p(X_i = x_i | X_i \in \mathcal{N}(x_i), \vec{\theta})$$

the "neighborhood" function $\mathcal{N}(x_i) \subseteq \mathcal{X}$ returns a set of generated implicit negative examples, in addition to the example x_i itself, which is hopefully the only positive example in the neighborhood. A listing of considered perturbations is in Figure 1:

neighborhood	size	lattice arcs	perturbations
DEL1WORD	n+1	$O(n)$	delete up to 1 word
TRANS1	n	$O(n)$	transpose any bigram
DELORTRANS1	$O(n)$	$O(n)$	DEL1WORD ∪ TRANS1
DEL1SUBSEQUENCE	$O(n^2)$	$O(n^2)$	delete any contiguous subsequence
Σ* (EM)	∞	-	replace each word with anything

Figure 1. Reproduced from (Smith and Eisner, 2005) slides. Neighborhood functions used for CE training.

The authors investigate each of the neighborhood functions in Figure 1. The only neighborhood not in Figure 1 is LENGTH, the neighborhood of all sentences which have the same length as the sentence being perturbed.

The authors experiment with trigrams using only unlabeled data, assuming a complete tagging dictionary, with 96K words from *The Wall Street Journal*. Initial training parameters were set to be uniform (i.e. completely uninformed). Smoothing was varied and the best result was reported in Figure 2.
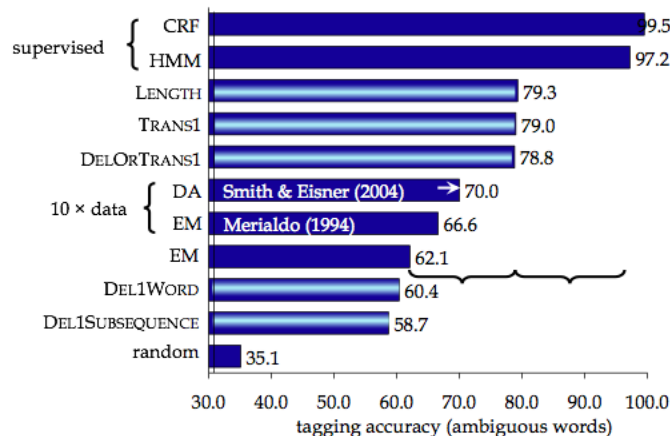

Figure 2. Reproduced from (Smith and Eisner, 2005) slides. POS tagging task performance.

Popular supervised approaches to tagging sequences include HMMs and CRFs. For POS tagging, a comparison is provided in Figure 2. Unsurprisingly, supervised methods outperform unsupervised methods for

the problem of sequence tagging. Note that (Toutanova and Johnson, 2008) also provide a competitive unsupervised POS tagger; we outline their results in §2.3.

LENGTH, TRANS1, DELORTRANS1 consistently outperform EM, while DEL1WORD and DEL1SUBSEQUENCE perform poorly. An important implication of this result is that "neighborhoods do not succeed by virtue of *approximating* log-linear EM; if that were so, one would expect larger neighborhoods such as DEL1SUB-SEQ[UENCE] to out-perform smaller ones (like TRANS1)—but this is not so." DEL1SUBSEQ[UENCE] and DEL1WORD, Smith and Eisner argue, perform poorly because their neighborhoods do not often enough give classes of negative evidence: deleting a word or subsequence often doesn't do enough damage, whereas TRANS1 is more dangerous.

2.1.2 Language Variation

The evaluations presented in (Smith and Eisner, 2005) are limited to English for the POS tagging task (though Contrastive Estimation was used successfully for grammar induction in multiple languages (Smith, 2006)). We thus speculate as to how well this approach would extend to part-of-speech tagging for other languages.

When using Contrastive Estimation for languages other than English, one has to consider the effectiveness of the neighborhood functions with respect to the language. Even though the LENGTH neighborhood performed best for English, it may perform differently in other languages. For example, it may be that in some languages deleting a word is much more likely to make a sentence ungrammatical, thus making the DELETE neighborhood favorable. Other neighborhood functions not considered in the paper might be appropriate for other languages or tasks. For instance, a neighborhood function whereby suffix strings are modified so as to yield other words in the vocabulary might facilitate the learning of morpheme segmentation or subject-verb agreement in a morphologically rich language.

As the authors mention, "the assumption that the tagging dictionary is completely known is difficult to justify. While a POS lexicon might be available for a language, certainly it will not give exhaustive information about all word types in a corpus." Smith and Eisner experimented with removing knowledge from the tagging dictionary, to see how well various objective functions could recover. The authors attempt to recover the lost information by adding features to the models.

The authors compared the performance of the best neighborhoods (LENGTH, DELORTRANS1, and TRANS1) from the first experiment, plus EM, using three *diluted* dictionaries and the original one. They produced diluted dictionaries by adding redundant tags to the set of possible tags for each word such that for rare words all tags are possible, effectively "simulating zero prior knowledge" about the word.

In the absence of spelling features, all models perform worse, as the task has become strictly harder. However, the improvement from adding spelling features is striking: DELORTRANS1 and TRANS1 "recover nearly completely (ignoring the model selection problem) from the diluted dictionaries". Unfortunately, it is untested how well spelling features would fare in languages other than English.

Though Smith and Eisner show that Contrastive Estimation is robust to degradation of the dictionary, their experiments nevertheless assume dictionaries will contain thousands of entries. Even though the dictionary is diluted, there is still a large amount of supervision for the common word types in the corpus. Requiring fewer examples is more useful when tagging sentences from a rare language. We next consider an approach which requires only a few prespecified associations between tags and word types.

2.2 Categorizing with Prototypes

2.2.1 An Exemplar-Based Model

Haghighi and Klein (2006) accomplish unsupervised learning of sequence models given just a few output (emission) types that exemplify each label. These exemplars are called *prototypes*. The intuition is explained as follows: "In providing a prototype [...] we may intend that words which are in some sense similar to a prototype generally be given the same label(s) as that prototype." Put another way, if a category corresponding to a label is thought of as a cluster under some similarity metric, the prototypes are the data points nearest to the center of the cluster.

A *prototype list* contains a set of prototype words for each possible label. For POS tagging, this means a few example words corresponding to each part of speech. The authors build a list with three prototypes per POS: for instance, *years*, *shares*, and *companies* are designated prototypes of plural common nouns (NNS) and *it*, *he*, and *they* exemplify personal pronouns (PRP). Prototype lists can be very small (the experiments use three prototypes per POS); as the authors note, this is an advantage over previous methods (Merialdo, 1994; Smith and Eisner, 2005) which used lists of all possible tags for each word. For the experiments, the three prototypes per tag were selected automatically from labeled data (the three most frequent word types which did not appear more frequently with another POS tag); in principle, however, the prototype list could be constructed by hand. Note that prototypes need not be exclusive to their most frequent POS; for instance, *shares* could be a verb, but was more likely to be a noun in the corpus used to select the prototypes.

Once the prototype list was created, it was used in the following way: feature vectors were created for each word type, with frequent nearby words as features.[1] Based on these features, SVD was used to compute the distributional similarity of all pairs of word types. Using this information, non-prototype words were associated with "similar" prototypes (i.e. prototypes with similarity scores exceeding a threshold). For each prototype z in this set, a feature PROTO $= z$ was added to the feature vectors of every instance of this word, to be used in estimating the model parameters. Additionally, all instances of the prototype words were fixed to their respective labels from the prototype list.[2]

These prototype features (along with baseline features such as suffixes) were incorporated into a linear-chain Markov Random Field (MRF) model. A type of log-linear model, linear-chain MRFs are the undirected equivalent of HMMs; they resemble CRFs, but model the joint distribution of labels and emissions, not the labels conditioned on the emissions. Like any other undirected graphical model, this joint distribution decomposes into a product of local distributions corresponding to cliques in the graph. Due to the Markov assumption present in linear-chain MRFs (also present in HMMs and CRFs), there are two types of cliques: transitions (pairs of adjacent labels) and emissions (label-observation pairs). To approximate the maximum likelihood estimate, the authors use the forward-backward algorithm for lattices of a particular length—i.e. the LENGTH neighborhood function of (Smith and Eisner, 2005).

2.2.2 Experiments

The effectiveness of a prototype-based approach was tested experimentally on two tasks. First, the prototype-based model achieved near state-of-the-art performance on unsupervised POS tagging for English and Chinese, despite the fact that state-of-the-art systems use large dictionaries of possible tags for each word. For Chinese POS tagging, their baseline was 34.4% where as POS tagging with prototypes achieved 57.4%. For English POS tagging, the best prototype-based result is somewhat worse than that of Smith and Eisner's

---

[1]For the POS tagging model, word contexts were represented in terms of relative position; for information field segmentation, on the other hand, only distance was used—whether the context word appeared to the right or to the left was irrelevant.

[2]The paper also lists results demonstrating that the similarity measure for non-prototype words improves accuracy over simply fixing the prototype word labels. The prototype-augmented results listed below include the similarity technique.

2005 best unsupervised result using a full tagging dictionary, but better than Contrastive Estimation with a diluted dictionary. More recently, Toutanova and Johnson (2008) reported an unsupervised result surpassing these two methods. A comparison of the three unsupervised techniques for English POS tagging is given in Figure 3.

Second, a prototype-based model was constructed for the *information field segmentation* task of (Grenager et al., 2005). Given a data set of classified ads, the task is to segment each ad and label parts according to attributes/fields: the 11 possible labels include ROOMMATES, NEIGHBORHOOD, and UTILITIES, which are common attributes of apartment listings. Approaching the segmentation and labeling problem with a sequence over words (where field boundaries are implicitly defined as consecutive words with different labels), the authors construct a prototype list similar to a POS list: for instance, *close*, *near*, and *shopping* exemplify the NEIGHBORHOOD field. The prototype list was extracted from labeled data using the same criteria as for the POS tagging experiment. Unsupervised performance on this task surpasses that of previous work. The results for English are in the table below.

Information Field Segmentation (Classifieds data set)

Haghighi and Klein (2006)	
Baseline	46.4%
Prototype-augmented	74.1%
Grenager et al. (2005)	
Best unsupervised	72.4%
Supervised	74.4%

2.2.3   Discussion

This formulation of the problem seems apt for sequences with multiple—but still reasonably few—target (label) categories, provided that each category is semantically (and distributionally) coherent. The prototype-based technique could be highly beneficial for languages without enough data for a tagging dictionary of the size expected by other methods. Ideally, for a resource-poor language, it would be possible for a linguist to provide a few exemplars for each tag without consulting a large corpus. It is not entirely clear to what extent the choice of prototypes affects accuracy; in their experiments, Haghighi and Klein used a labeled corpus to choose frequently-occurring words that were good exemplars of the POS category.

Taking the problem one step further, it is perhaps a weakness that all categories need to be listed and assigned prototypes. What would happen if the state space were much larger or more structured? It might be useful to have a model that could learn to propose new categories (or subcategories) for words which don't pattern much like any of the known prototypes. This paper uses labeled data to identify prototypes. Unsupervised learning of prototype-based categories using a clustering framework would be an interesting related research direction.

We note that the role of prototypes in mental categories has been a focal point in psychology and cognitive linguistics. Evidence for prototypes has been found in perceptual categories such as color (Berlin and Kay, 1969); in semantic categories (Rosch, 1978); and even in linguistic categories (Lakoff, 1987). Given that degrees of prototypicality are indeed a psychological reality, this could explain the successful use of prototypes in NLP applications such as POS tagging.

However, it's difficult to see how prototype features could be used for certain formulations of problems with very few categories, where words on their own do not provide much information about the category. The prototype-based approach could be generalized to take feature sets as prototypes, and use a similarity function to find the most similar feature vector prototype. In effect, this would be a form of dimensionality reduction.

Finally, we wonder if a different technique for automatic prototype selection from labeled data would have been more effective—for instance, choosing frequent words with very low entropy over labels. Their technique doesn't remove the possibility that a word which occurs frequently with two different labels would be selected as a prototype for the more frequent of the two.

2.3   Summary of Sequences

We detailed two papers that address the POS tagging problem in an unsupvervised way. Smith and Eisner (2005) use a tag dictionary along with a novel modification of EM to label sequences with POS tags. Haghighi and Klein (2006) use a few prototypical examples of each tag to do the same. Both methods are minimally supervised in that they require at least some examples of each tag. This may hinder their practicality when used to label sequences of a rare language.

Sequences	Smith & Eisner	Haghighi & Klein
Word-Level		
Word tokens in sequence	●	●
Discrete set of labels/tags	●	●
Per-type possible tags	●	
Per-label prototypes		●
Per-token labels	○	○
Word context counts		●
Sub–Word-Level		
Character n-grams	(●)	●

●= input, ○= output

In addition to these papers, we'd like to point to (Toutanova and Johnson, 2008), which provides a very competitive solution for this task. As shown in Figure 3, over past years there has been a healthy increase in accuracy for unsupervised English part-of-speech tagging, but there is still some work to be done to rival supervised methods such as CRFs, where 99.5% of output is tagged correctly (see Figure 2).

Unsupervised English POS Tagging

	24K tokens	48K tokens (2K sen.)
Haghighi and Klein (2006)		
Baseline (trigram), Treebank tagset		42.4%
Prototype-augmented, Treebank tagset		79.1%
Prototype-augmented, reduced tagset	82.2%	
Smith and Eisner (2005)		
CE, with 2125-entry tagging dictionary	79.5%	
CE, with 3362-entry tagging dictionary	88.1%	
CE, with 5406-entry tagging dictionary	90.4%	
Toutanova and Johnson (2008)		
Latent Dirichlet Allocation	93.4%	

Figure 3.   Tagging accuracy for three unsupervised POS systems. The reduced tagset used by Haghighi and Klein is that of Smith and Eisner. Contrastive Estimation results are given for three conditions corresponding to the amount of information in the tagging dictionary. Constructed from the full (labeled) corpus, the dictionary has 5406 word types; using only words that appeared at least twice, it has 2125 word types; and using only words that appeared in the first half of the data, it has 3362 word types.

3 Unsupervised Approaches to Morphology

So far we have considered the labeling of observed units (words) in sequences. Now we consider problems in which the boundaries between units are unobserved. *Segmentation* is problem of predicting these boundaries. The tasks we will consider concern morphological units.

Morphology refers to the internal structure of words. Consider the adjective *unsupervised*: it decomposes into a verb root, a suffix, and a prefix—*un-supervise-d*. Meaningful units within words (roots, prefixes, and suffixes) are known as *morphemes*. *Morpheme segmentation* is thus the task of identifying morpheme boundaries.

Morphological structure is not limited to morpheme boundaries: a richer morphological parse is $[_A$un-$[_A[_V$supervise$]$-d$]]$, which shows how the word arises from joining morphemes together, one at a time. The bare verb *supervise* is first joined with the *-d* suffix indicating completion (similar to the regular past tense suffix on verbs). But in this case, *supervised* is not a verb but an adjective derived from a verb, or *participle* (it modifies a noun, as in *the **supervised** algorithm*). Then, the *un-* prefix is added to negate the adjectival stem (the result remains an adjective). We know the ordering could not have been otherwise, because *unsupervise* is not a word. *Affix* is a general term for prefix or suffix, and the portion of the word an affix attaches to is called its *stem*. Modeling the hierarchical structure of morphology could be useful for tasks such as machine translation, in much the same way that syntactic parses can be exploited for these tasks.

While a morphological parser that recognizes this complex structure is not out of the realm of possibility, most unsupervised work to date has been focused on the segmentation task alone. The unsupervised methods discussed below seek to predict morpheme boundaries by looking for common prefixes and suffixes in unlabeled data. Some go one step further, explicitly modeling possible stem/affix pairings, or *paradigms*. As the morphology of English is rather paltry, we take an example from Spanish: the verb *hablar* 'to speak' is conjugated according to tense and grammatical characteristics of the subject—*hablo* '(I) speak', *hablamos* '(we) speak', *hablan* '(they) speak', etc. Since the root *habl-* can take one of many suffixes depending on the grammatical context, we refer to these suffixes as the *paradigm* of the verb. In some cases the paradigm generalizes to many different verbs (roots), and in other cases it is specific to a single root—it is irregular. Irregular patterns are a major challenge for broad-coverage morphological systems.

All of the models discussed herein assume *agglutinative* morphology. The morphology of an agglutinative language consists only of roots, prefixes, and suffixes. For other languages, such as Arabic and Turkish, morphological structure is more complex—and as a result, these approaches are less successful for these languages.

Our focus will be on three unsupervised approaches: the ParaMor algorithm for morphology induction (Monson et al., 2007, 2008a,b); a nonparametric Bayesian approach to word segmentation (Goldwater et al., 2006; Goldwater et al., in submission); and a nonparametric Bayesian approach to morpheme segmentation which makes use of multilingual data (Snyder and Barzilay, 2008). There are a number of other unsupervised approaches which have modeled morphology with some success (e.g. Yarowsky and Wicentowski, 2000; Goldsmith, 2001, 2006; Adler and Elhadad, 2006; Creutz and Lagus, 2007).

3.1 Learning Inflectional Paradigms: ParaMor

(Monson et al., 2007) describes ParaMor, an algorithm for unsupervised morphology induction. ParaMor studies the vocabulary in a corpus and infers paradigms based on counts of substrings of the word types. A ParaMor *paradigm* consists of a set of stems and a set of suffixes which can apply alternately to those stems. The approach is best at finding inflectional paradigms. The paradigms can be used to segment words into

their component morphemes.

3.1.1 The ParaMor Algorithm

We begin our discussion of ParaMor with a summary of the algorithm, which is described in (Monson et al., 2007). The algorithm simply requires a vocabulary of words observed in a corpus. It does not rely on a statistical model, but rather proposes paradigms based on type frequencies in the vocabulary. **Stage 1** induces these paradigms, by first extracting partial paradigms such that every stem in the paradigm occurred with every suffix (Algorithm 1); and then merging and filtering these partial paradigms to overcome sparsity and spurious segmentations. Merging is done with greedy hierarchical clustering, and spurious segmentations are removed by bounding the entropy of characters at the hypothesized morpheme boundaries. **Stage 2** then uses these paradigms straightforwardly to propose all possible segmentations of new words. The version described in (Monson et al., 2008b) can propose multiple segmentations at once, thereby accommodating agglutinative languages.

Algorithm 1 Step 1.1 of the ParaMor system for morphological induction. Given the vocabulary of a language, this algorithm produces a set of paradigms, where each paradigm $\langle T, F \rangle$ represents a pairing of a set of stems T with a set of suffixes F such that any of the stems in T can be suffixed with any of the suffixes in F. α is a free parameter which can be tuned.

$F^* = \{\text{all word-final character sequences observed in the vocabulary}\}$
$P^* = \{\}$
for all $f \in F^*$ **do**
    $T = \{\text{all candidate stems corresponding to } f\}$
    $P = \langle T, \{f\} \rangle$
    $P^* = P^* \cup \{P\}$
end for
for all $P = \langle T, F \rangle \in P^*$ **do**
    while $|T| > |F|$ **do**
        $f' = $ candidate suffix following the largest fraction q of stems in T
        if $q \leq \alpha$ **then**
            break
        end if
        $T = T \setminus \{\text{all candidate stems which cannot form a word with } f'\}$
    end while
end for
return P^*

To illustrate this, we use the inflectional paradigm for regular *-ar* verbs in Spanish. The present tense conjugation table looks like:

	'speak'	**'dance'**	**'buy'**	...	**generalization**
infinitive	hablar	bailar	comprar	...	t-ar
I	hablo	bailo	compro	...	t-o
we	hablamos	bailamos	compramos	...	t-amos
$\vdots$		$\vdots$			$\vdots$
they	hablan	bailan	compran	...	t-an
root	habl-	bail-	compr-	...	t

Note that the root never occurs on its own, but is determined by subtracting the frequent suffixes (across

verbs) from the vocabulary items.

If the vocabulary extracted from the corpus includes all conjugations of these two verbs (and several others fitting the pattern), Algorithm 1 will successfully extract the candidate paradigm $\langle\{\text{habl}, \text{bail}, \text{compr}, \ldots\}, \{\text{-ar}, \text{-o}, \text{-amos}, \ldots, \text{-an}\}\rangle$. It may also extract paradigms based on erroneous segmentations, such as $\langle\{\text{hab}, \text{bai}, \ldots\}, \{\text{-lar}, \text{-lo}, \text{-lamos}, \ldots, \text{-lan}\}\rangle$. The filtering step is designed to weed out these erroneous paradigms.

In practice, however, many candidate paradigms will only be partial: for instance, if the vocabulary contained *hablar*, *bailar*, *hablan*, *bailo*, *compro*, and *compramos* (along with overlapping inflections of other verbs from the same true paradigm), we would want the algorithm to generalize these to a single paradigm. (This amounts to smoothing, enabling the system to handle unseen words.) ParaMor's merging step is designed to accomplish this.

3.1.2   Results

The Morpho Challenge competition evaluated several morphology induction systems, measuring scores on two sets of tasks: the linguistic analysis task of identifying morpheme boundaries in words, and several IR tasks. A hybrid system combining the outputs of ParaMor and Morfessor (Creutz and Lagus, 2007)—another algorithm, designed to handle both inflectional and derivational morphology—did well in Morpho Challenge 2007 (Monson et al., 2007).

With some improvements to ParaMor, detailed in (Monson et al., 2008b), the hybrid system won Morpho Challege 2008, beating the other systems on every language in terms of F_1 score on the linguistic analysis task. Winning scores were in the 50% range for English, German, Finnish, and Turkish, and about 40% for Arabic (Monson et al., 2008a). The system did worst for Arabic, which is unsurprising due to Arabic's complex morphology. The authors speculate that updating ParaMor to handle prefixes as well as suffixes would help for Arabic, English, and German. As for the IR evaluation, the ParaMor+Morfessor system had the best average precision (over several IR tasks) for English and German, whereas another system did better for Finnish.

3.1.3   Discussion

ParaMor succeeds at learning inflectional paradigms from raw text, and does so with a relatively simple algorithm. This simplicity is attractive, as many of the statistical models used for morphology (including the ones described below) are quite difficult to understand and implement. That ParaMor learns paradigms, and not just segmentations, is another advantage, as induced paradigms are more easily interpreted than highly abstract probability distributions.

3.1.4   Language Variation

When moving to other languages, it is difficult to see how ParaMor can be extended to achieve better accuracy or accommodate more complex morphological phenomena. The authors speculate that it could be extended to handle prefixes as well as suffixes, but this still assumes agglutinative morphology. While the model relies heavily on type frequency (of suffix strings with respect to the stem types they can attach to), ParaMor fails to exploit other statistical properties such as token frequency. Both type and token frequency are known to be important in language; for instance, it is well known that the most frequent verbs (such as *be*, *have*, *make*, *do*, and *go* in English) are most likely to be morphologically irregular (Bybee, 1985).

Finally, the authors note that ParaMor is less successful for derivational morphology, which is likely part of the reason that combining ParaMor and Morfessor was necessary to achieve unsupervised state-of-the-art results in the Morpho Challenge. Ideally, one system would be able to learn both inflectional and derivational phenomena, though it is an open question whether statistical models would benefit from encoding this distinction explicitly.

3.2   A Bayesian Approach to Segmentation

Next we turn to the task of *word segmentation*, which is a form of morphological segmentation (only for a particular class of morphemes, i.e. words). This is necessary for processing speech, as well as text for languages such as Chinese where word boundaries are not marked orthographically. We discuss the Bayesian approach introduced in (Goldwater et al., 2006) and elaborated at length in (Goldwater et al., in submission).

3.2.1   Two Nonparametric Models

Goldwater et al. (2006) introduce two Bayesian models for word segmentation: a unigram model which uses the Dirichlet Process; and a bigram model using a Hierarchical Dirichlet Process (Teh et al., 2006).

The generative story for the unigram model is as follows: the document is generated as a sequence of words, which are unobserved. For each word, there is some probability that it is one of the words seen previously, and there is some probability that it is novel. The probability that it is novel is proportional to some hyperparameter α_0. If it is novel, we then generate the characters (or phonemes) in the word according to some unigram distribution P_0. If it is not novel, then we choose a previously-generated word with probability proportional to the number of times that word was generated previously. This procedure characterizes a *Dirichlet Process* (DP), and we write

$$
\begin{aligned}
U &\sim DP(\alpha_0, P_0) \\
w_i | U &\sim U
\end{aligned}
$$

where the *base distribution* P_0 is the unigram distribution over characters for a new word, and the *concentration parameter* α_0 affects the preference for sparsity of words. The probability that ℓ will be the character sequence drawn for the ith word is given by

$$
P(w_i = \ell | \mathbf{w}_1^{i-1}) = \frac{n_\ell + \alpha_0 \cdot P_0(w_i = \ell)}{i - 1 + \alpha_0}
$$

where n_ℓ is the number of times the word ℓ has been drawn previously. This model is *nonparametric* because the number of parameters is free to grow with the size of the data: larger corpora will have larger vocabularies, and each word in the vocabulary corresponds to a parameter. The Dirichlet Process accomplishes this by always reserving some probability mass for additional parameters—in principle accommodating an infinite number of parameters. This is feasible, however, because the probability of adding a new parameter gets increasingly unlikely as the model grows (how fast it does so is controlled by the concentration parameter).

Because of a useful property called *exchangeability*, it is possible to sample words at random from a Dirichlet Process, conditioning them on all other words—i.e. $P(w_i = \ell | \mathbf{w}_1^{i-1}, \mathbf{w}_{i+1}^{|\mathbf{w}|})$ with the same right-hand side as the equation above, except replacing i in the denominator with $|\mathbf{w}|$. The authors exploit this property, deriving and implementing a Gibbs sampler to perform inference in the model.

In the *Hierarchical Dirichlet Process*, or HDP (Teh et al., 2006), a distribution (call it U) sampled from a Dirchlet Process is used as the base distribution for another Dirichlet Process. The unigram DP model

described above serves as the basis for the bigram model, but with an added layer: first, $w_i|w_{i-1}$ is chosen to be novel or previously seen—i.e. it is decided whether the bigram has been seen, given the previous word. This is proportional to the number of previous bigrams starting with w_{i-1}. The preference for novel vs. seen *bigrams* is affected by a second concentration parameter, α_1. If the bigram is previously seen, then w_i is chosen with probability equal to the proportion of time it has followed w_{i-1} in the past. Otherwise, w_i is chosen from the unigram DP distribution as before (parameterized with α_0 and the unigram character distribution)—except now the choice of word w_i is dictated by the proportion of previously seen bigram types having that choice as their second word. The hierarchical procedure is thus

$$
\begin{aligned}
U &\sim DP(\alpha_0, P_0) \\
B_{\ell'} &\sim DP(\alpha_1, U) \; \forall \ell' \\
w_i|w_{i-1} = \ell', B_{\ell'} &\sim B_{\ell'} \; \forall \ell'
\end{aligned}
$$

Intuitively, U is a generalized distribution over words given some previous word, and $B_{\ell'}$ is a more specific distribution taking into account that the previous word was observed to be ℓ'. This ability to generalize over distributions thought to be similar/related (but not identical) is the essence of the HDP.

As with the unigram DP model, a Gibbs sampler is used for inference. Refer to (Goldwater et al., 2006, in submission) for further details on the DP and HDP models—including how they treat utterance boundaries—and the corresponding sampling procedures.

3.2.2 Results

The following table, reproduced from (Goldwater et al., in submission), compares the performance of several word segmentation systems on a corpus of child-directed speech which has phonemic transcriptions. Goldwater et al.'s models are the unigram Dirichlet Process model (DP) and the bigram Hierarchical Dirichlet Process model (HDP).[3] The others are the unigram MBDP-1 model from (Venkataraman, 2001), and the unigram and bigram varieties of the n-gram segmentation system (NGS) from (Brent, 1999). Each model's performance is measured in terms of precision, recall, and F-score according to three different criteria: the words whose boundaries were correctly identified (P, R, F); the individual boundaries themselves (BP, BR, BF); the induced lexicon (vocabulary) of the corpus (LP, LR, LF).

Model	P	R	F	BP	BR	BF	LP	LR	LF
					Performance measure				
NGS-u	67.7	**70.2**	68.9	80.6	**84.8**	82.6	52.9	51.3	52.0
MBDP-1	67.0	69.4	68.2	80.3	84.3	82.3	53.6	51.3	52.4
DP	61.9	47.6	53.8	**92.4**	62.2	74.3	57.0	**57.5**	57.2
NGS-b	68.1	68.6	68.3	81.7	82.5	82.1	54.5	57.0	55.7
HDP	**75.2**	69.6	**72.3**	90.3	80.8	**85.2**	**63.5**	55.2	**59.1**

The unigram DP does well in terms of lexicon induction and in terms of boundary precision, but trails the other models significantly with respect to a majority of metrics. It is thus under-predicting boundaries. In contrast, the bigram HDP model fared best for a majority of performance metrics, including F-score, and is not too far behind the highest score on any of the remaining four metrics.

---

[3]In addition to the concentration parameter(s), both models have hyperparameters $p_\#$, the stop probability for a sequence of phonemes; and $p_\$$, the probability of an utterance boundary. Hyperparameter settings for these results were as follows: for the unigram DP model, $p_\# = .5$, $p_\$ = .5$, and $\alpha_0 = 20$; and for the bigram HDP model, $p_\# = .2$, $p_\$ = .5$, $\alpha_0 = 3000$, and $\alpha_1 = 100$.

3.2.3 Discussion

Goldwater et al.'s approach to word segmentation is attractive for many reasons. Their use of the Hierarchical Dirichlet Process enables backoff from a bigram model to a unigram model. The models can be trained using standard sampling techniques, with only the unsegmented text as input. Unfortunately, these techniques are difficult to implement—but this is a general problem with Bayesian inference, not unique to the task at hand. Because the only input is raw text, and because the model only represents phonemes/characters and words, this approach contains no inherent bias towards any particular type of language—though performance for some languages might benefit from an even richer model. Unfortunately, the model was only tested on an English corpus of child-directed speech; it would be nice to see how it performs (a) for longer utterances, (b) for orthographic word segmentation (e.g. English text with spaces removed), and (c) for other languages.

It seems like it would be straightforward to adapt these models to the standard morpheme segmentation task, simply by changing the character n-gram distribution P_0 to prefer shorter (i.e. morpheme-length rather than word-length) strings, and possibly changing the concentration parameters α_0 and α_1. Additionally, we envision that the HDP might be useful for encoding generalizations over morphophonological variants, e.g. discovering that the English phoneme sequences /s/, /z/, and /əz/—or their orthographic counterparts *s* and *es*—are in fact variants of the same (plural) suffix. In a similar vein, Johnson (2008) showed that a nonparametric model for Sesotho (a morphologically complex language) is better for word segmentation when it takes morphology into account. Thus, we feel that nonparametric Bayesian methods are a promising direction for unsupervised learning of morphology as well as word segmentation.

3.3 Improving Segmentation with Multilingual Input

Snyder and Barzilay (2008) offer an approach that uses parallel multilingual data to improve training of a morpheme segmentation model. The model as described only considers data from two languages at once (experiments are done with several different language pairs). However, they claim it is extensible to multiple languages. Their generative model represents corresponding morphemes in both languages with what they call *abstract morphemes*, as well as so-called *stray morphemes* that only occur in one of the two languages. Improvement over a monolingual baseline is demonstrated for aligned phrases from the Hebrew Bible with parallel data for Hebrew, Aramaic, Arabic, and English.

3.3.1 A Multilingual Model

Figure 4 shows an example parallel phrase with morphological alignments in the four languages:
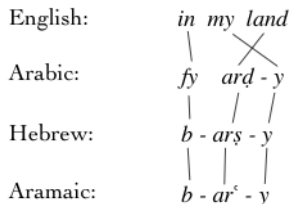

Figure 4. Morphological alignments across four languages. Reproduced from (Snyder and Barzilay, 2008).

In this example, each of the parallel phrases has three morphemes, all of which are aligned to morphemes in the other phrases—thus, for this example there are three abstract morphemes in the model:

$\langle in, fy, b, b\rangle$, $\langle my, y, y, y\rangle$, etc. Unaligned morphemes would surface in the model as stray morphemes. The model does not allow for one-to-many alignments, nor does it distinguish between prefixes, suffixes, and stems/standalone words.[4]

A nonparametric Bayesian model is used for the abstract and stray morphemes and how they surface in phrases. We now address the generative process and sampling procedure for this model.

Generative process: We adapt their notation, using the variable ℓ which can take the value from the set $\mathfrak{L}$ of languages being modeled. Subscripts always index the language.

1. Draw a distribution over possible morphemes in each language. The Dirichlet Process prior is parameterized with a language-specific base distribution P_ℓ, which "can encode prior knowledge about the properties of morphemes in each [language], such as length and character n-grams": $L_\ell | \alpha, P_\ell \sim DP(\alpha, P_\ell) \ \ \forall \ell \in \mathfrak{L}$

2. Draw a distribution over **abstract morphemes** relating the languages:

$$A | \alpha', P' \quad \sim \quad DP(\alpha', P')$$
$$\langle m_1, m_2, \ldots, m_{|\mathfrak{L}|}\rangle \quad \sim \quad A$$

Variables of the form m_ℓ are simply strings in language ℓ; thus, an abstract morpheme is a tuple of strings across multiple languages. In their phonetic model, P' is defined so as to take into account known sound correspondences between language pairs (using a linguistically-informed edit distance metric).

3. For a parallel phrase:

(a) Draw the number of language-unique **stray morphemes** that will be used from each language (n_ℓ corresponding to language ℓ), as well as the number of **abstract morphemes** having a cross-lingual correspondence that will be used (n_*): $n_*, n_1, n_2, \ldots, n_{|\mathfrak{L}|} \sim Poisson(\lambda)$

(b) Draw stray and abstract morphemes for each of the languages in the phrase:

$$m_\ell^1, \ldots, m_\ell^{n_\ell} \quad \sim \quad L_\ell \ \ \forall \ell \in \mathfrak{L} \ [\textbf{stray morphemes}]$$
$$\langle m_1^{(1)}, m_2^{(1)}, \ldots\rangle, \ldots, \langle m_1^{(n_*)}, m_2^{(n_*)}, \ldots\rangle \quad \sim \quad A \ [\textbf{abstract morphemes}]$$

The phrase now contains $n_\ell + n_*$ morphemes in language ℓ: $m_\ell^1, \ldots, m_\ell^{n_\ell}$ and $m_\ell^{(1)} \ldots m_\ell^{(n_*)}$. The abstract morphemes constitute implicit alignments; the stray morphemes are equivalent to unaligned morphemes. So far, these occur in no particular order, and word boundaries are not defined.

(c) Draw the ordering of morphemes for each language in the phrase. We will use μ_ℓ to denote the ordered morphemes for language ℓ: $\mu_\ell^1 \ldots \mu_\ell^{n_\ell + n_*} \sim ORDER | m_\ell^1, \ldots, m_\ell^{n_\ell}, m_\ell^{(1)} \ldots m_\ell^{(n_*)}$, where $ORDER$ is defined uniformly over all permutations—i.e. $ORDER(\bullet | \overrightarrow{\mathbf{m}}) = \frac{1}{|\overrightarrow{\mathbf{m}}|!}$. Note that word boundaries remain undefined.

(d) Determine which pairs of adjacent ordered morphemes should be fused as part of the same word: $w_\ell^1 \ldots w_\ell^{s_\ell} \sim FUSE | \mu_\ell^1 \ldots \mu_\ell^{n_\ell + n_*}$, where s_ℓ is the size of the phrase for language ℓ. $FUSE$ is defined uniformly, i.e. $FUSE(\bullet | \overrightarrow{\mu}) = \frac{1}{2^{|\overrightarrow{\mu}|-1}}$.

Sampling algorithm: Unfortunately, the paper provides few details regarding the sampling technique, except that it is a "blocked and collapsed Gibbs sampler" which marginalizes over implicit morpheme alignments.

---

[4]Only concatenative morphology is addressed—that is, every morpheme is assumed to occur on its own or concatenated linearly with other morphemes. Arabic, Hebrew, and Aramaic have nonconcatenative morphemes, but the experiments only consider concatenative segmentations.

3.3.2   Results

The model was evaluated by extracting short parallel phrases (using automatic alignments) for training. For the testing phase, the system proposed segmentations for phrases in a single language, which were then evaluated against a gold standard. Several different models were evaluated on Hebrew and Arabic: a monolingual model, serving as a baseline; models trained with a second language, using a length-based edit distance metric; and models trained with a second Semitic language, using a phonological edit distance metric.

The best result was for Arabic, when trained with Arabic+Hebrew and the phonological edit distance: F_1 was 72.20 (67.75 precision, 77.29 recall). This was an improvement of about 9 points over the monolingual baseline, and 7 points over another state-of-the-art system, Morfessor (Creutz and Lagus, 2007). Without the phonological similarity metric, Arabic+Hebrew, Arabic+Aramaic, and Arabic+English had similar scores (F_1 between 68 and 69). The scores when tested on Hebrew were about 10 points less than for Arabic, and the F_1 score for the Arabic+Hebrew+phonology when tested on Hebrew only slightly outpaced that of Morfessor (63.87 vs. 61.29), though it did surpass that of the monolingual baseline (59.78).

Thus, this model not only demonstrates the value of multilingual input for morpheme segmentation, but is also competitive with the state-of-the-art for both Hebrew and Arabic. The authors further conclude that a related language is somewhat more useful than an unrelated language when similar structure can be exploited (as enabled by the phonological similarity metric).

3.3.3   Discussion

We feel that Snyder and Barzilay's model is an exciting start to leveraging multilingual information to improve statistical morphological analysis. Overall, the separation of abstract morphemes (cross-linguistically relevant) from stray morphemes (language-specific) feels intuitive. However, we would like to identify several aspects of the model which we believe could be improved.

First, we feel that their choice of distributions for a couple of variables is suspect—particularly $ORDER$ and $FUSE$, which are assumed to be uniform. Experimentation with more sophisticated distributions is probably warranted. For instance, $FUSE$ might be structured so as to encapsulate the distinction between prefixes, roots, and suffixes (morphemes which pattern as suffixes are unlikely to follow a word boundary, and so forth).

Second, this model is probably not robust to language-specific morphophonological variation: if there is a morphological correspondence between two languages, but morphophonological variants pattern differently in the two languages, then the paper's definition of abstract morpheme would not seem particularly suitable. Third, unlike ParaMor (§3.1 above), Snyder and Barzilay's approach doesn't model paradigms—in this model, morphemes are related across languages, but not within a language. A generalization of this model that might remove these two deficiencies would be to allow each abstract morpheme to consist of multiple realizations in each language. Thus, morphophonological variants, as well as paradigmatic variants (e.g. inflectional suffixes on verbs in a language) could all be encapsulated in a single abstract morpheme relating them to corresponding variants in other languages.

Fourth, like the other unsupervised approaches to morphology we have seen, this model is designed for agglutinative languages. Performance with nonconcatenative morphology in Semitic languages is not evaluated, and presumably would not be very good. Any attempt to model morphological processes besides prefixation and suffixation would likely necessitate significant changes to this model.

Fifth, the model is evaluated on the level of phrases—parallel phrases are automatically aligned. We wonder how complete these alignments are; if there are unaligned phrases, then these are ignored in training, though

presumably they could provide useful information under another model (or learning procedure). This would be especially problematic if certain morphemes tend to occur in unaligned phrases. Additionally, it's unclear whether at test time this would scale up to full sentences, given that this adds significant uncertainty with respect to the $FUSE$ and $ORDER$ distributions (which are assumed to be uniform, giving equal probability to an exponential number of possible configurations).

3.4   Summary of Morphology

We have seen three approaches to unsupervised learning and annotation of morphological structure:

- **ParaMor** (Monson et al., 2007, 2008a,b), a deterministic procedure to extract inflectional paradigms from a corpus vocabulary; these paradigms can then be used for morpheme segmentation.

- Two **nonparametric Bayesian models** for word segmentation (Goldwater et al., 2006; Goldwater et al., in submission): a unigram model using the Dirichlet Process, and a bigram model using the Hierarchical Dirichlet Process.

- A **multilingual Bayesian model** which leverages morpheme correspondences from parallel phrases to improve morpheme segmentation (Snyder and Barzilay, 2008).

First, we compare the inputs and outputs of these three approaches:

MORPHOLOGY	Monson et al.	Goldwater et al.	Snyder & Barzilay
Phrase/Document-Level			
Unsegmented text		●	
Parallel sentences			◖
Phrasal aligner			◖
Word-Level			
Vocabulary (list of word types)	●		
Sub-Word–Level			
Paradigms	◖		
Segmentations	◗	○	◗
Phonetic correspondences			(●)

Legend		
	training	test
input	◖	◗
output	◖	◗

All three methods produce segmentations at test time. However, they require different forms of input. The Goldwater et al. approach needs only a raw corpus; the Monson et al. approach needs only the vocabulary (word types) from a corpus; and the Snyder and Barzilay approach requires parallel phrases, optionally with phonetic correspondences between related languages for best performance. Of these, parallel corpora are the most difficult to acquire in general, but seem to offer useful information when available for a language. Unlike methods for the other tasks described in this paper, these do not rely on the availability of parses, stemmers, or part-of-speech tag sets for data (though these might in principle provide information relevant to morphology).

The evaluations for the three models used different data and methodologies, so it is not possible to compare them quantitatively with each other. Qualitatively, they take advantage of data in different ways. The Monson et al. approach uses a simple algorithm to extract interpretable paradigms for inflectional morphology.

The other two approaches seem more adept at learning the statistical patterns necessary for segmentation tasks, but due to the nature of the Bayesian models, estimating and interpreting the parameters themselves is more difficult. The generative stories for these two models are fairly intuitive, though we suspect they might be improved by adding more structure.

4   Unsupervised Acquisition of Lexical Resources

So far we have seen unsupervised solutions to several word-level and sub–word-level prediction tasks, including part-of-speech tagging and morpheme segmentation. Next we consider the problem of building linguistic resources that can be useful for various tasks pertaining to words. In particular, unsupervised methods are described for constructing bilingual lexicons from monolingual data (Haghighi et al., 2008); inferring semantic roles for verbs and their corresponding syntactic realizations (Grenager and Manning, 2006); and identifying related events that occur sequentially in narratives (Chambers and Jurafsky, 2008).

4.1   Constructing Bilingual Lexicons

4.1.1   Something from Nothing: A CCA-based Model

Haghighi et al. (2008) present a method for learning bilingual translation lexicons from monolingual corpora. They do not use any data from each language other than monolingual sources of text, such as Wikipedia articles, news articles, and a single side of Europarl data. Word types in each language are characterized by "purely monolingual features, such as orthographic substrings and word context counts." To learn translations the authors define and train a generative model based on Canonical Correlation Analysis.

Surprisingly, the authors present a method that can learn a translation lexicon between two divergent languages using only monolingual corpora. When the two languages are historically related, they can improve performance by adding orthographic features.

The lexicon produced is at most 1-1, meaning that each word on either side of the lexicon is either matched with exactly one word from the other side of lexicon, or not matched at all. The authors' choice of matching allows unmatched words, and disallows one-to-many mappings. The motivation behind such simplifying assumptions is to allow comparison to previous work.

Their model represents each word type as a feature vector derived only from the word-type and monotext. They use two types of features: orthographic features which consist of character n-grams in the word, and context features which represent counts of words that occur nearby in text.
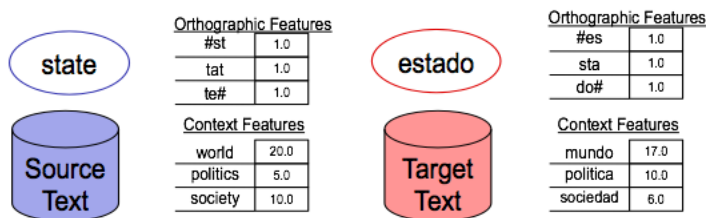

Figure 5. Reproduced from (Haghighi et al., 2008) slides. Features on both sides are completely monolingual.

After computing feature vectors for words which lie in different vector spaces, they define a generative model

19

over "(1) a source lexicon, (2) a target lexicon, and (3) a matching between them." Their model is based on Canonical Correlation Analysis (CCA), which projects corresponding data points (vectors for word types) in two spaces (the two languages) to a common latent space. Inference is done using a variant of the Viterbi-EM algorithm. They compare to a standard baseline: the EDITDIST baseline, which is the maximum bipartite matching with weights on the partition edges as normalized edit distances.
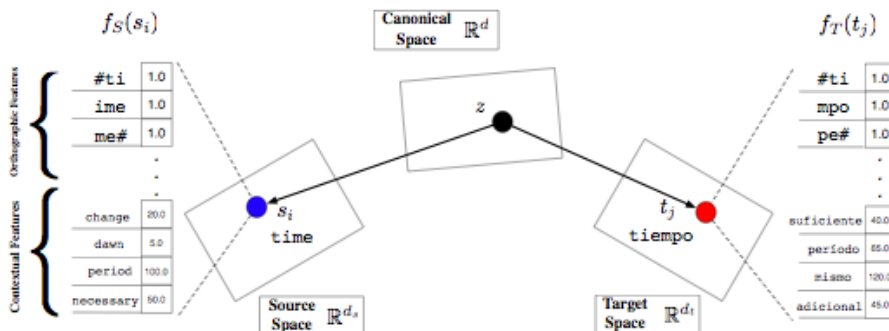

Figure 6. Reproduced from (Haghighi et al., 2008). Illustration of Matching Canonical Correlation Analysis model. The latent concept $z_{i,j}$ originates in the canonical space. The observed word vectors in the source and target spaces are generated independently given this concept.

The authors construct their models such that if two words are truly translations, "it will be better to relate their feature vectors through the latent space than to explain them independently." However, if a source word is not a translation of any of the target words, it is possible for it to be left out of the matching and thus be generated independently.

For experiments they build English-Spanish, English-French, English-Chinese, and English-Arabic lexicons using monolingual data from Wikipedia articles, Europarl data[5], and several other corpora. The authors evaluated their results using the standard F_1 measure. Precision is the proportion of correct proposed translations, and recall is the proportion of proposed translations that were possible. Their system beats any other previous unsupervised results.

An important note about the training data is that while the method presented makes no use of document-level or sentence-level alignments, the models learned perform significantly better when the monolingual text on either side of the training corpus are largely translations of each other. Specifically, this method of constructing bilingual lexicons may not work when the monolingual text on either side of the are from different domains. Their models perform significantly worse on corpora where the two monolingual sides contain no parallel sentences: 49% and 62% F_1 for two corpora without parallel sentences, versus 72% and 77% F_1 for corpora with parallel sentences.

4.1.2 Language Variation

The authors mention that "while orthographic features are clearly effective for historically related language pairs, they are more limited for other language pairs, where we need to appeal to other clues."

Haghighi et al. explored how system performance varies for language pairs other than English-Spanish. On the English-French task, their system achieves 88.3% precision at 0.33 recall (see Figure 7 for more performance measures). This result shows that their system can handle language pairs for which it was not initially designed.

---

[5]**Europarl** is a parallel corpus extracted from the proceedings of the European Parliament.

Languages	$p_{0.1}$	$p_{0.25}$	$p_{0.33}$	$p_{0.50}$	Best-F_1
EN-ES	91.4	94.3	92.3	89.7	63.7
EN-FR	94.5	89.1	88.3	78.6	61.9
EN-CH	60.1	39.3	26.8	——	30.8
EN-AR	70.0	50.0	31.1	——	33.1

Figure 7. Reproduced from (Haghighi et al., 2008). Orthographic features are less applicable for languages that are not historically related. The columns represent recall at different values of precision.

One concern is how their system performs on language pairs where orthographic features are less applicable. Results on disjoint English-Chinese and English-Arabic are given as EN-CH and EN-AR in Figure 7, both using only context features. In these cases, their system performed less favorably, with 26.8% precision and 31.0% precision at 0.33 recall, respectively. Lacking orthographic features, performance on EN-CH and EN-AR suffered compared to EN-ES.

We now move onto two unsupervised tasks relating to verbs: Semantic role labeling for verbs, and Narrative event chain extraction.

4.2   Identifying Subcategorization Frames for Verbs

Grenager and Manning (2006) present an unsupervised method to determine the *semantic roles* of a verb's dependents. For example, the verb *give* is associated with three semantic roles: that of the giver, the recipient, and the thing given. These can all appear as arguments in sentence. Consider the sentence *Sarah Palin's speech today gave me a stomachache.* The verb has three arguments which correspond to the three semantic roles of *give*: *Sarah Palin's speech*, *me*, and *a stomachache.* These arguments are all noun phrases. A fourth noun phrase, *today*, is called an *adjunct* because it is not a core semantic role of the verb, but can modify events in general. Adjuncts can generally be omitted without altering the grammaticality of the sentence. Moreover, depending on the verb and the construction, not all arguments need be expressed: a corresponding passive sentence using only two is *Today I was given a stomachache.*

Grenager and Manning train a Bayesian model which relates a verb, its semantic roles, and their possible syntactic realizations. In doing so, they address three problems: learning all the semantic roles associated with a particular verb, given a variety of sentences using that verb; generalizing over the syntactic realizations of a verb in relation to its arguments and adjuncts; and labeling the noun phrases in a sentence with verb-specific semantic roles. Their unsupervised approach stands in contrast to current supervised methods of learning dependents' semantic roles, which are driven by hand-tagged corpora such as PropBank. PropBank is a verb ontology covering a subset of the Penn Treebank. Verb instances are annotated with a sense ID, and arguments are tagged so as to be consistent across all instances of that verb sense. There are loose correspondences between verbal arguments—ARG0 is generally used for intransitive subjects, ARG1 for active transitive subjects, and higher arguments for objects—however, there is no standard correspondence between arguments and semantic roles that holds across verbs. Unfortunately labeled datasets such as PropBank are too sparse, providing motivation for unsupervised techniques where data is abundant.

Their system relies on dependency parses of the sentences in the corpus. (These parses can be obtained by running an English dependency parser, or by running a phrase structure parser and using deterministic rules to transform phrase structure parses into dependency parses.) Dependencies are labeled with syntactic relations such as SUBJ (subject), NP#1 (first noun phrase after the verb), NP#2 (second noun phrase after the verb), and PREP_x (prepositional modifier with preposition x).

In training, the model associates each verb with a small number of syntactic roles, with identifiers similar to PropBank's ARG0, ..., ARG5. ARGM is used for all adjuncts. Note that the argument numbers do not rep-

resent explicit correspondences across verbs; thus *give*.ARG2 may bear no relation to *tell*.ARG2 (though their procedure for assigning identifiers to roles may result in some correlations across verbs—see the paper for details of their procedure). Then, the model scores *linkings*, or mappings between a verb's arguments/adjuncts and their syntactic realizations (see Figure 9 below). These linkings are useful for the task of semantic role labeling.

4.2.1   Generative Process

The learning method is based on a structured probabilistic model of the domain. A graphical representation of the model is shown in Figure 8. The model encodes a "joint probability distribution over the elements of a single verb instance, including the verb type, the particular linking, and for each dependent of the verb, its syntactic relation to the verb, semantic role, and head word."
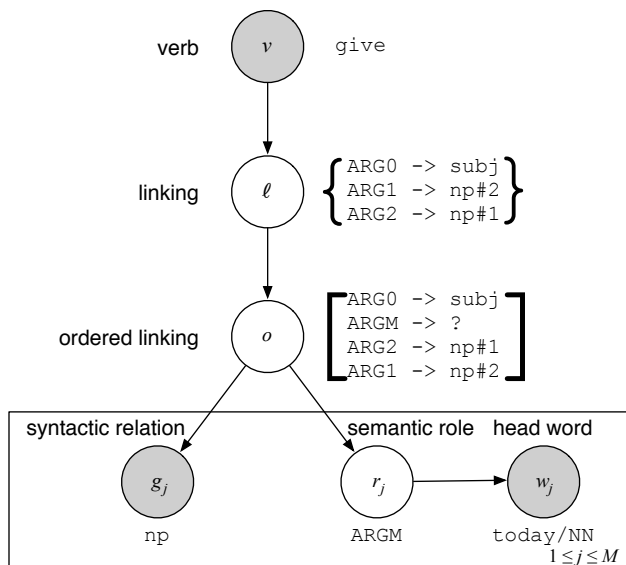

Figure 8. Adapted from (Grenager and Manning, 2006). A graphical representation of the verb linking model, with example values for each variable. The rectangle is a *plate*, indicating that the model contains multiple copies of the variables shown within it: in this case, one for each dependent (indexed by j). Variables observed during learning are shaded. Included are example values for a ditransitive sentence with the verb *give* and temporal adjunct *today*.

What follows is a description of the generative process for the model, which we have adapted from (Grenager and Manning, 2006):

1. Begin by generating the verb v.

2. Conditioned on the choice of v, generate a linking ℓ, which defines both the set of roles to be expressed, as well as the syntactic relations that express them.

3. Conditioned on choice of linking, now generate an *ordered* linking o, giving a final position in the dependent list for each role and relation in the linking ℓ, while also optionally inserting one or more adjunct roles.

4. Now iterate through each of the M dependents $1 \leq j \leq M$, generating each in turn. For each core

argument, the semantic role r_j and syntactic relation g_j are completely determined by the ordered linking o, so it remains only to sample the syntactic relation for the adjunct roles.

5. Finally, generate the head word of each dependent, conditioned on the semantic role of that dependent.

The authors mention a potential improvement to the generative process by adding word classes. Examples of potentially useful word classes include PERSON, OBJECT, PLACE, and EVENT. In such a scenario the word class would be generated conditioned on the semantic role, and the choice of head word would then be conditioned on the word class. This modification might improve results by moderating the effects of sparsity during training.

4.2.2   Evaluation

The authors use the EM algorithm to train the above model, with training details found in the paper. Three English newswire corpora are used for evaluation (an automatic parser was used for corpora without gold standard parse trees). The authors evaluate the performance of their learned models by using them to predict the semantic roles of constituents which were already identified as dependents of a verb instance. They report two results: "*coarse roles*, in which all of the adjunct roles are collapsed to a single ARGM role," as opposed to *core roles*, in which performance is evaluated for core semantic roles only. The authors do not report results on the all roles task, since their model does not distinguish between different types of adjunct roles. For each task they report precision, recall, and F_1. In the classification task, they compare their system to an informed baseline, which is computed by labeling each dependent with a role that is a deterministic function of its syntactic relation.

Their best system was trained on 1000 verb instances per verb type when available, and achieved an F_1 score of 0.897 on the coarse roles task, as opposed to an F_1 of 0.856 for the deterministic baseline. Figure 9 shows the verbs that were most improved by the model as compared to the baseline.

Verb ($\triangle$ **F1**)		Learned Linkings
give (+.436)	.57	{0=subj,1=np#2,2=np#1}
	.24	{0=subj,1=np#1}
	.13	{0=subj,1=np#1,2=to}
work (+.206)	.45	{0=subj}
	.09	{0=subj,2=with}
	.09	{0=subj,2=for}
	.09	{0=subj,2=on}
pay (+.178)	.47	{0=subj,1=np#1}
	.21	{0=subj,1=np#1,2=for}
	.10	{0=subj}
	.07	{0=subj,1=np#2,2=np#1}
look (+.170)	.28	{0=subj}
	.18	{0=subj,2=at}
	.16	{0=subj,2=for}
rise (+.160)	.25	{0=subj,1=np#1,2=to}
	.17	{0=subj,1=np#1}
	.14	{0=subj,2=to}
	.12	{0=subj,1=np#1,2=to,3=from}

Figure 9. Learned linkings for the most improved verbs over the baseline. Arguments are abbreviated as integers (ARG0 as 0, etc.). Reproduced from (Grenager and Manning, 2006).

Overall, the authors show that it is possible to automatically extract the verb structure from unlabeled text by using a bayesian model trained with EM. More work needs to be done to improve model performance, but the presented model is a good starting point.

23

4.2.3 Language Variation

The mode of (Grenager and Manning, 2006) relies on the availability of a syntactic parse for all the sentences in the corpus. Parsers may not be available in all languages. Dan Klein's thesis explores unsupervised parsing (Klein, 2005) but assumes POS tags. It's unclear whether parsing can really be done well from scratch with minority languages. It's also unclear how to evaluate parsers for languages that do not have some tagged data available, e.g. the TreeBank. This problem is further exacerbated by a potential lack of linguistic knowledge about the language. This is a general problem with minority languages, where lack of linguistic data represents problems evaluating unsupervised methods.

There are obstacles to using the above model for languages other than English: rules to go from a syntactic parse to dependency parse may not be readily available for the language of interest. Ultimately, the authors need a dependency parse, which may be hard to obtain for some languages.

Another consideration is that languages that have rich morphology may already decorate the roles of each dependent by way of morphological markers, thus making the generation of the ordering less important. Such problems can be overcome by simple modifications of the model. Finally, the verb that is generated in the generative process is a verb stem, which requires stemming (lemmatization). Stemmers are not available for most languages.

The next task is a new one, introduced by (Chambers and Jurafsky, 2008). It is a good example of how unsupervised learning can lead to new and exciting tasks that have not received enough attention to motivate labeled data sets.

4.3 Relating Events within Narratives

A somewhat different attempt to extract a statistical characterization of verbs is described in (Chambers and Jurafsky, 2008). Here, the goal is to construct a graph of events that tend to pattern together in *narratives*, or complex events. For example, *arrest*, *charge*, *plead*, *testify*, *convict*, and *acquit* refer to parts of a conventional narrative of CRIMINAL PROSECUTION. Likewise, the verbs *code*, *compile*, *execute*, and *debug* can refer to different aspects of a PROGRAMMING narrative.[6] Both of these examples track related, often temporally-ordered events involving a common participant/protagonist. For the verbs listed for the PROGRAMMING narrative, the protagonist is always the agent: the person who writes code is (most likely) the one compiling, executing, and debugging it. In contrast, the common protagonist in the CRIMINAL PROSECUTION events— the accused—takes the role of agent for some verbs (*plead*, *testify*) but the object/theme role for others (*arrest*, *convict*, *acquit*).

Chambers and Jurafsky show that some of the structure of these narratives can be extracted automatically from parsed text. They define a *narrative chain* as a partially-ordered sequence of events that tend to occur with a common protagonist. Their goal, then, is to identify the related events in narratives and their ordering. This ordering is not necessarily absolute: *debug* and *execute* might occur in either order (or multiple times, even). Observe, also, that the component events and their relative order are not the only things we know about narratives: we know, for instance, that *convict* and *acquit* are mutually exclusive, and a complete natural language understanding system would need to know this as well for full inferential power.

4.3.1 Basic Model

Using a typed dependency parser, the authors extract from each document *verb.dependencyType* relations for all verbs and all arguments occurring in the document. Each relation is a *perspectivized event*: it characterizes

---

[6]Such narratives are known in various disciplines as *frames*, *schemas*, or *scripts*.

an event from the perspective of one of its participants (the participant filling the role corresponding to the dependency type). The authors use a coreference resolution package to identify which verbs share participants. For example, an individual accused of a crime might be the protagonist associated with event roles *arrest*.OBJECT and *plead*.SUBJECT.

The unsupervised learning step is identifying correlated events across documents, where correlation is defined by sharing a protagonist. *Pointwise mutual information* (PMI) is used as a formal measure of the correlation between perspectivized events. In general, PMI measures the independence of two random events by dividing their joint probability by the product of their marginal probabilities:

$$PMI(X,Y) = \log \frac{P(X,Y)}{P(X)P(Y)}$$

A PMI score of $PMI(x,y) = \log 1 = 0$ indicates that two outcomes can be predicted independently without loss of information. If this is true for all outcomes, then the two events X and Y are independent. A positive PMI score indicates that the joint outcome is *more* likely than the product of the marginals would suggest; a negative score indicates that the joint outcome is *less* likely than the product of the marginals would suggest.

PMI is similar to the TF.IDF metric used in information retrieval. The intuition behind TF.IDF is that if two terms tend to occur together across documents more than they occur independently, they are probably related. For this scenario (assuming probabilities are estimated directly from counts), the log denominator for highly correlated terms would be smaller than the value of the log numerator, and therefore highly correlated terms would have a high (positive) PMI log score. Inversely correlated terms would have a negative PMI score.

We write $v_i.t_i$ for a perspectivized event, where v_i is a verbal word type (e.g. *arrest*) and t_i is a dependency type observed for a participant of the verb (e.g. OBJECT). Chambers and Jurafsky score the correlation between two perspectivized events $v_1.t_1$ and $v_2.t_2$ with $PMI(v_1.t_1, v_2.t_2)$. They define the joint probability of two perspectivized events as the fraction out of all coreferential perspectivized events in a document:

$$P(v_1.t_1, v_2.t_2) = \frac{\text{number of times } v_1.t_1 \text{ and } v_2.t_2 \text{ corefer}}{\text{number of times any pair of event arguments } v_1'.t_1' \text{ and } v_2'.t_2' \text{ corefer } (\forall v_1', v_2', t_1', t_2')}$$

Thus, more closely correlated events (as linked by their participants) will have higher PMI scores.

Given PMI scores between all pairs of events, one can look at a set of events in a document and predict which other events are most likely to accompany them (summing over the pairwise PMI scores between observed events and every unobserved event and taking the argmax yields the most likely unobserved event). A ranked list of likely next events can be computed in this way, where the best guess (the argmax) has ranking 1. These rankings are used to evaluate performance by way of a *narrative cloze task*, which consists of stripping an event from a document, and measuring the predicted rank of that event given the other events in the document. When tested on the Gigaword corpus, the results demonstrate that—with sufficient training data—the protagonist-based model does better than a baseline which only uses verb coocurrence counts to measure event relatedness.

4.3.2   Modeling Temporal Ordering

In their second model, Chambers and Jurafsky (2008) try to predict the pairwise ordering of correlated events. They use a binary temporal classifier based on (Chambers et al., 2007) to predict a BEFORE or OTHER relation between two events—in other words, does one of them tend to precede the other? The

classifier is trained with a (supervised) SVM using time-annotated data from the Timebank corpus [7], on which it achieves 72.1% accuracy. The classifier was then asked to predict the ordering of narrative-related events from the basic model.

The temporal narrative chains are evaluated by randomly ordering the events in a document, and checking whether the model predicts a lower coherence score for that ordering than the actual ordering. The coherence score is based on summing log differences in counts of BEFORE?(a, b) versus BEFORE?(b, a) for all events a, b (see the paper for details). The overall accuracy is 75% (versus a random baseline of 50%); accuracy tends to be better for larger chains.

For qualitative evaluation of the model, directed graphs of temporally-ordered narrative sequences can be constructed by performing agglomerative clustering on PMI scores for event pairs. Two examples are given, with mostly reasonable predictions made by the model. The more successful of the two is reproduced in Figure 10.
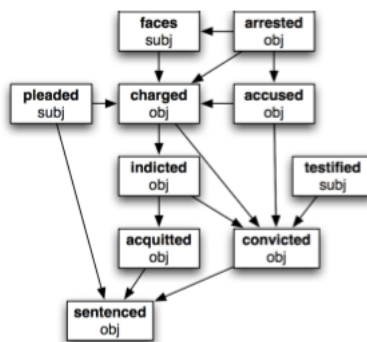

Figure 10. Learned structure of the CRIMINAL PROSECUTION narrative. Arrows represent temporal relations. Reproduced from (Chambers and Jurafsky, 2008).

4.3.3 Discussion

This paper is notable for introducing a new task—learning of narrative event sequences—which essentially concerns semantics, but can be acquired in an unsupervised fashion using clever statistical relatedness criteria. The observation that sequentially related events tend to share participants would seem to be fairly general in language.[8] However, because their approach relies on so much (manually or automatically) annotated structure—parses, coreference links, and (for their second model) a temporal classifier—extracting event sequences for resource-poor languages will require a considerable amount of unsupervised NLP just for these prerequisite tasks. And of course, syntactic parses alone are not likely to suffice in a morphologically-rich language.

Since event sequence learning is a new task, its practical import is not immediately clear. One imagines that it could be used alongside other methods to build rich semantic representations for deep language processing. The authors also suggest that the narrative event chains might be used to learn semantic roles, which we think is a promising idea worthy of further investigation.

---

[7]The Timebank corpus is a subset of the Treebank, with events annotated according to temporal ordering

[8]It would be interesting to evaluate this task on Chinese, as it is quite common to omit some of a verb's arguments when they can be inferred from context.

4.4 Summary of Lexical Resources

The inputs and outputs for the lexical resource acquisition techniques detailed above are as follows:

Lexical Resources	Haghighi et al.	Grenager & Manning	Chambers & Jurafsky [9]	
Phrase/Document-Level				
Typed dependency parses		●	●	●
Coreference resolver			●	●
Temporal classifier				●
Word-Level				
Event relatedness scores			○	○
Predicted event ordering				○
Per-word translation lexicon	○			
Word context counts	●			
Per-verb semantic roles		○		
Verb argument frames		○		
Stemmer		●		
Sub–Word-Level				
Character n-grams	●			

<center>●= input, ○= output</center>

Given the abundance of unannotated text data, unsupervised approaches are very appealing for natural language processing. In this section we presented three unsupervised systems for the tasks relating to lexical resource acquisition. Some of these systems achieve close to state-of-the-art results in domains previously dominated by fully supervised approaches. The contributions we reviewed are:

- a model which learns translation lexicons without relying on parallel corpora (Haghighi et al., 2008);

- a Bayesian model which relates a verb, its semantic roles, and their possible syntactic realizations, which is useful for semantic role labeling (Grenager and Manning, 2006); and

- a method for automatically constructing a graph of events that tend to pattern together in *narratives*, or complex sequences of events (Chambers and Jurafsky, 2008).

Each of the three approaches presented in this section had their models initially inspired by English, but have reasonable extensions available for other languages. For lexical translation, (Haghighi et al., 2008) present orthographic features which improve performance when the two languages are historically related. However, word context features work well for many divergent language pairs.

Both methods presented in (Grenager and Manning, 2006) and (Chambers and Jurafsky, 2008) require parsers, which may not be easily available in all languages. (Chambers and Jurafsky, 2008) further assumes the existence of a coreference resolution system, as well as a temporal classifier. This makes the task of porting narrative event chain extraction to non-English even more difficult.

5 Overall discussion

Next we take a broader view, drawing connections among the three types of unsupervised tasks reviewed above: sequence tagging, morphological segmentation, and lexical resource acquisition. We will focus on

---

[9](Chambers and Jurafsky, 2008) presents two models, which are summarized in separate columns.

three considerations: the types of input data required to produce output for various tasks; the relative gain in semantic complexity of the output over the input; and the adaptability of the methodology to different languages.

5.1 Inputs and Outputs

In Figure 11 (p. 28) we reproduce the input/output information from the three topics, concatenated together in a single table.

	Sequences/POS		Morphology			Lexical Resources			
	S&E	H&K	M+	G+	S&B	H+	G&M	C&J₁	C&J₂
Phrase/Document-Level									
Unsegmented corpus				●					
Typed dependency parses							●	●	●
Coreference resolver								●	●
Temporal classifier									●
Parallel sentences					◖				
Phrasal aligner					◖				
Word-Level									
Word tokens in sequence	●	●							
Discrete set of labels/tags	●	●							
Per-type possible tags	●								
Per-label prototypes		●							
Per-token labels	○	○							
Event relatedness scores								○	○
Predicted event ordering									○
Per-word translation lexicon						○			
Word context counts		●				●			
Per-verb semantic roles							○		
Verb argument frames							○		
Stemmer							●		
Vocabulary (list of word types)			●						
Sub-Word–Level									
Paradigms			◖						
Segmentations			◗	○	◗				
Phonetic correspondences					(●)				
Character n-grams	(●)	●				●			

	Legend	
	training	test
input	◖	◗
output	◖	◗

Figure 11. Inputs and outputs for all papers discussed. Rows represent information while columns correspond to papers. (Paper citations are abbreviated with the first letter of the authors' names, with '+' short for 'et al.'.) Chambers and Jurafsky (2008) describe two models, which correspond to the eighth and ninth columns, respectively.

This table exposes some patterns that are worth noting. First, we see that most of the annotation types (rows) are relevant to only one category of problems (column groups). For instance, among the papers we looked at, parses were only used as input for the lexical resource acquisition tasks, and POS tags were only produced as output of the sequence tagging models. One imagines that both parses and POS tags might

be useful as input for morphological tasks, but neither was exploited in the approaches we considered. The two exception were character-level n-gram features and word context counts, which are easily extracted from corpora.

Second, we note that the lexical resource acquisition approaches involved the greatest diversity of types of data. This is not surprising, given the diversity and semantic complexity of tasks under this category.

Finally, many of the unsupervised procedures nevertheless make use of supervised tools (such as parsers) for preprocessing. We would like to see more work that chains together unsupervised systems, better approximating what would be necessary for processing of resource-poor languages.

5.2   Semantic value added

Next, we examine the extent to which the aforementioned procedures for sequence labeling, morphological analysis, and lexical resource building constitute the acquisition of *semantic* information. Intuitively, information about semantic roles or translations is more "semantic" than information about syntax or parts of speech, which are more "grammatical". In turn, grammatical information is closer to semantics than raw text represented in n-gram or word context features. We refer to this qualitative notion of semantic complexity/richness as *semanticity*. While there is no obvious way to quantify the semanticity of a type of linguistic annotation, we construct a numerical semanticity scale based on our own intuitions about these relative differences. This scale is shown in Figure 12. Though it is obviously subjective, we feel the scale nevertheless has value in elucidating some of the contrasts between tasks and techniques.

Using this scale, we rank each type of input and output information relevant to the methods we have discussed (see the table in the previous section). Then, for each approach, we take the difference between the maximum input semanticity score and the maximum output semanticity scores to yield a *semantic gain* value. These numbers are summarized in the following table:
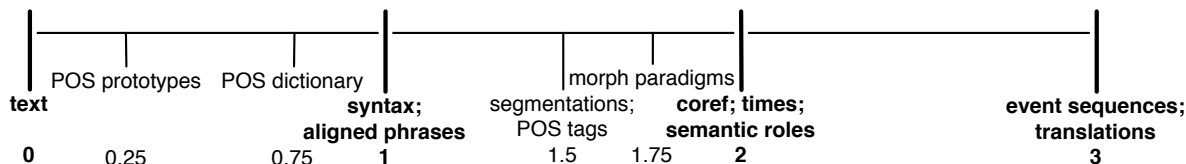

Figure 12. Semantic complexity rankings for different types of data annotations

	Sequences/POS		*Morphology*			*Lexical Resources*		
	S&E	H&K	M+	G+	S&B	H+	G&M	C&J
Input semanticity	.75	.25	0	0	1	0	1	2
Output semanticity	1.5	1.5	1.75	1.5	1.5	3	2	3
Semantic gain	.75	1.25	1.75	1.5	.5	3	1	1

The biggest "win" semantically is multilingual lexicon induction from raw text (Haghighi et al., 2008), which corresponds to our intuitions. The next biggest gains are with Monson et al.'s morphological paradigm induction procedure, and Goldwater et al.'s Bayesian word segmentation model. In contrast, the multilingual morphology approach (Snyder and Barzilay, 2008) do not achieve much gain in semanticity, as it requires aligned parallel phrases as input—already a fairly sophisticated type of information relative to knowledge of morpheme boundaries.

Of course, this semantic gain rating is just one dimension for comparison among approaches. We hasten to add that it does not take into account differences in performance among the approaches (as this is not comparable across tasks), nor does it necessarily represent the relative costs of obtaining gold-standard annotations.

5.3   Training procedures

Of three Bayesian models we examined, the two morphology ones (of Goldwater et al. and Snyder and Barzilay) used Gibbs sampling to infer parameters for latent variables. The third Bayesian approach, that of Grenager and Manning, used the EM algorithm for learning. EM-like methods for log-linear models were seen in other papers—local gradient ascent in (Haghighi and Klein, 2006), and Contrastive Estimation (Smith and Eisner, 2005). Other techniques include Canonical Correlation Analysis in (Haghighi and Klein, 2006), pointwise mutual information scores in (Chambers and Jurafsky, 2008), and Singular Value Decomposition in (Haghighi and Klein, 2006) for obtaining prototype similarity scores. Finally, Monson et al. introduced a novel algorithm for morphological paradigm induction that uses substring counts over word types.

5.4   Language variation

Many of the presented papers require multiple inputs in addition to large amounts of text. When moving to new languages, some resources may not be readily available and unsupervised tools for their production may not exist. For example, in the case of (Smith and Eisner, 2005) a tagging dictionary that defines a set of possible tags for each word is required. This may be difficult or expensive to produce for some languages, which is why the authors investigate the performance of their algorithms under degradation of the dictionary. Similarly, (Haghighi and Klein, 2006) require a few exemplar words for each POS tag in order to categorize all words. This small amount of labeled data is reasonable for languages such as English, but may pose problems in other less common languages.

The methods of (Chambers and Jurafsky, 2008) and (Grenager and Manning, 2006) require syntactic parses of the training data. Obtaining parse trees is not an easy problem for languages where treebanks do not exist. Klein (2005) and Smith (2006) explore unsupervised extraction of parse trees, but it remains unclear whether unsupervised parsing can really be done well from scratch with minority languages. This problem may be further exacerbated if there is a lack of linguistic knowledge about the language. While we hope unsupervised methods will eventually be able to learn linguistic structure with very little human knowledge of the language, we believe that continued construction of (and evaluation with respect to) annotated corpora in additional languages is necessary to determine whether unsupervised methods can generalize to many different types of languages.

6   Conclusion

We reviewed several unsupervised approaches to ubiquitous problems in natural language. These fall into three broad categories:

- tagging words in **sequences** (Smith and Eisner, 2005; Haghighi and Klein, 2006);

- predicting **segmentations** of morphemes (Monson et al., 2007, 2008a,b; Snyder and Barzilay, 2008) and words (Goldwater et al., 2006; Goldwater et al., in submission); and

- building **lexical resources** that encode patterns associated with verbs (Grenager and Manning, 2006; Chambers and Jurafsky, 2008) and bilingual translations for words in general (Haghighi et al., 2008).

In our analysis, we examined several aspects of these approaches: the types of models and training procedures used; their required inputs and produced outputs; the extent to which the output is richer than the input in terms of semantics; and the degree to which the models can be applied to other languages.

References

Meni Adler and Michael Elhadad. An unsupervised morpheme-based HMM for Hebrew morphological disambiguation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the ACL*, pages 665–672, Sydney, 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P06-1084.

Brent Berlin and Paul Kay. *Basic color terms: their universality and evolution.* University of California Press, Berkeley, 1969.

Michael R. Brent. An efficient, probabilistically sound algorithm for segmentation andword discovery. *Mach. Learn.*, 34(1-3):71–105, 1999. ISSN 0885-6125. doi: 10.1023/A:1007541817488. URL http://www.springerlink.com/content/g840p555100429vj.

Joan L. Bybee. *Morphology: a study of the relation between meaning and form.* Typological studies in language. John Benjamins, Amsterdam, 1985. ISBN 0915027372.

Nathanael Chambers and Dan Jurafsky. Unsupervised learning of narrative event chains. In *Proceedings of ACL-08: HLT*, pages 789–797, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P08/P08-1090.

Nathanael Chambers, Shan Wang, and Dan Jurafsky. Classifying temporal relations between events. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 173–176, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P/P07/P07-2044.

Mathias Creutz and Krista Lagus. Unsupervised models for morpheme segmentation and morphology learning. *ACM Transactions on Speech and Language Processing*, 4(1):3, 2007. ISSN 1550-4875. doi: 10.1145/1187415.1187418. URL http://portal.acm.org/ft_gateway.cfm?id=1217101&type=pdf&coll=GUIDE&dl=GUIDE&CFID=10012239&CFTOKEN=87673495.

John Goldsmith. Unsupervised learning of the morphology of a natural language. *Computational Linguistics*, 27(2):153–198, 2001. ISSN 0891-2017. doi: 10.1162/089120101750300490. URL http://portal.acm.org/ft_gateway.cfm?id=972668&type=pdf&coll=GUIDE&dl=&CFID=10012198&CFTOKEN=64466389.

John Goldsmith. An algorithm for the unsupervised learning of morphology. *Natural Language Engineering*, 12(4):353–371, 2006. ISSN 1351-3249. doi: 10.1017/S1351324905004055. URL http://hum.uchicago.edu/~jagoldsm/Papers/algorithm.pdf.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. A Bayesian framework for word segmentation: exploring the effects of context. URL http://homepages.inf.ed.ac.uk/sgwater/papers/journal-wordseg-hdp.pdf. In submission.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*, pages 673–680, Sydney, Australia, July 2006. Association for Computational Linguistics. doi: 10.3115/1220175.1220260. URL http://www.aclweb.org/anthology/P06-1085.

Trond Grenager and Christopher D. Manning. Unsupervised discovery of a statistical verb lexicon. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 1–8, Sydney, Australia, July 2006. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/W/W06/W06-1601.

Trond Grenager, Dan Klein, and Christopher Manning. Unsupervised learning of field segmentation models for information extraction. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 371–378, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219886. URL `http://www.aclweb.org/anthology/P05-1046`.

Aria Haghighi and Dan Klein. Unsupervised coreference resolution in a nonparametric Bayesian model. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 848–855, Prague, Czech Republic, June 2007. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P07/P07-1107`.

Aria Haghighi and Dan Klein. Prototype-driven learning for sequence models. In *Proceedings of the Human Language Technology Conference of the NAACL, Main Conference*, pages 320–327, New York City, USA, June 2006. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/N/N06/N06-1041`.

Aria Haghighi, Percy Liang, Taylor Berg-Kirkpatrick, and Dan Klein. Learning bilingual lexicons from monolingual corpora. In *Proceedings of ACL-08: HLT*, pages 771–779, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P08/P08-1088`.

Mark Johnson. Unsupervised word segmentation for Sesotho using adaptor grammars. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 20–27, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W08/W08-0704`.

Dan Klein. *The unsupervised learning of natural language structure*. Ph.D. dissertation, Stanford University, 2005. URL `http://www.cs.berkeley.edu/~klein/papers/klein_thesis.pdf`.

George Lakoff. *Women, fire, and dangerous things: what categories reveal about the mind*. University of Chicago Press, Chicago, 1987. ISBN 0226468038.

Percy Liang, Slav Petrov, Michael I. Jordan, and Dan Klein. The infinite PCFG using hierarchical Dirichlet processes. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, 2007. URL `http://aclweb.org/anthology/D07-1072`.

Bernard Merialdo. Tagging English text with a probabilistic model. *Computational Linguistics*, 20(2): 155–171, 1994. ISSN 0891-2017. URL `http://aclweb.org/anthology/J94-2001`.

Christian Monson, Jaime Carbonell, Alon Lavie, and Lori Levin. ParaMor: Finding paradigms across morphology. In Alessandro Nardi and Carol Peters, editors, *Working Notes for the CLEF 2007 Workshop: CLEF at Morpho Challenge 2007*, Budapest, September 2007. URL `http://www.clef-campaign.org/2007/working_notes/monsonCLEF2007.pdf`.

Christian Monson, Jaime Carbonell, Alon Lavie, and Lori Levin. ParaMor and Morpho Challenge 2008. In *Working Notes for the CLEF 2008 Workshop: Morpho Challenge at CLEF 2008*, Aarhus, Denmark, September 2008a. URL `http://www.clef-campaign.org/2008/working_notes/Monson-paperCLEF2008.pdf`.

Christian Monson, Alon Lavie, Jaime Carbonell, and Lori Levin. Evaluating an agglutinative segmentation model for ParaMor. In *Proceedings of the Tenth Meeting of ACL Special Interest Group on Computational Morphology and Phonology*, pages 49–58, Columbus, Ohio, June 2008b. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/W/W08/W08-0708`.

Hoifung Poon and Pedro Domingos. Joint unsupervised coreference resolution with Markov Logic. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 650–659, Honolulu, Hawaii, October 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/D08-1068`.

Eleanor Rosch. Principles of categorization. In Eleanor Rosch and B. B. Lloyd, editors, *Cognition and categorization*, pages 27–48. Lawrence Erlbaum Associates, Hillsdale, New Jersey, 1978.

Noah A. Smith. *Novel estimation methods for unsupervised discovery of latent structures in natural language text*. Ph.D. dissertation, Johns Hopkins University, October 2006. URL `http://www.cs.cmu.edu/~nasmith/papers/smith.thesis06.pdf`.

Noah A. Smith and Jason Eisner. Contrastive estimation: training log-linear models on unlabeled data. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 354–362, Ann Arbor, Michigan, June 2005. Association for Computational Linguistics. doi: 10.3115/1219840.1219884. URL `http://www.aclweb.org/anthology/P05-1044`.

Benjamin Snyder and Regina Barzilay. Unsupervised multilingual learning for morphological segmentation. In *Proceedings of ACL-08: HLT*, pages 737–745, Columbus, Ohio, June 2008. Association for Computational Linguistics. URL `http://www.aclweb.org/anthology/P/P08/P08-1084`.

Yee Whye Teh, Michael I. Jordan, Matthew J. Beal, and David M. Blei. Hierarchical Dirichlet Processes. *Journal of the American Statistical Association*, 101:1566–1581, 2006. doi: 10.1198/016214506000000302. URL `http://www.ingentaconnect.com/content/asa/jasa/2006/00000101/00000476/art00023`.

Kristina Toutanova and Mark Johnson. A Bayesian LDA-based model for semi-supervised part-of-speech tagging. In J.C. Platt, D. Koller, Y. Singer, and S. Roweis, editors, *Advances in Neural Information Processing Systems 20*, pages 1521–1528, Cambridge, MA, 2008. MIT Press. URL `http://books.nips.cc/papers/files/nips20/NIPS2007_0964.pdf`.

Anand Venkataraman. A statistical model for word discovery in transcribed speech. *Computational Linguistics*, 27(3):352–372, 2001. ISSN 0891-2017. URL `http://aclweb.org/anthology/J01-3002`.

David Yarowsky and Richard Wicentowski. Minimally supervised morphological analysis by multimodal alignment. In *Proceedings of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 207–216, Hong Kong, October 2000. Association for Computational Linguistics. doi: 10.3115/1075218.1075245. URL `http://www.aclweb.org/anthology/P00-1027`.