Verb Sense and Subcategorization: Using Joint Inference to Improve Performance on Complementary Tasks

Galen Andrew, Trond Grenager, and Christopher Manning
Computer Science Department
Stanford University
Stanford, CA 94305-9040
{pupochik, grenager, manning}@cs.stanford.edu

Abstract

We propose a general model for joint inference in correlated natural language processing tasks when fully annotated training data is not available, and apply this model to the dual tasks of word sense disambiguation and verb subcategorization frame determination. The model uses the EM algorithm to simultaneously complete partially annotated training sets and learn a generative probabilistic model over multiple annotations. When applied to the word sense and verb subcategorization frame determination tasks, the model learns sharp joint probability distributions which correspond to linguistic intuitions about the correlations of the variables. Use of the joint model leads to error reductions over competitive independent models on these tasks.

1 Introduction

Natural language processing research has traditionally been divided into a number of separate tasks, each of which is believed to be an important subtask of the larger language comprehension or generation problem. These tasks are usually addressed separately, with systems designed to solve a single problem. However, many of these tasks are not truly independent; if solutions to one were known they would facilitate finding solutions to the others. For some sets of these problems, one would like to be able to do joint inference, where information of one kind can influence decisions about information of another kind and vice versa. For instance, information about named entities can usefully inform the decisions of a part-of-speech tagger, but equally, part-of-speech information can help a named entity recognizer. If one had a large corpus annotated with all the information types of interest, one could estimate a joint distribution over all of the variables simply by counting. However, it is more often the case that one lacks any jointly annotated corpus, or at least one that is sufficiently large, given that the joint distribution is necessarily sparser than the marginal distributions. It would therefore be useful to be able to build a model for this joint inference task using only partially supervised data. In this paper we examine these problems in the context of joint inference over verb senses and their subcategorization frames (SCFs).

System Name	Accuracy
kunlp	57.6
jhu-english-JHU-final	56.6
SMUls	56.3
LIA-Sinequa-Lexsample	53.5
manning-cs224n	52.3

Table 1: Performance of the top 5 Senseval-2 word sense disambiguation systems when considering accuracy only on the 29 verbs. Systems not guessing on all instances have been omitted.

1.1 Verb Sense and Subcategorization

Of the syntactic categories tested in the Senseval word sense disambiguation (WSD) competitions, verbs have proven empirically to be the most difficult. In Senseval-1, Kilgarriff and Rosenzweig (2000) found a 10-point difference between the best systems' performance on verbs compared with other parts-of-speech. In Senseval-2, Yarowsky and Florian (2002) also found that while accuracies of around 73% were possible for adjectives and nouns, even the most competitive systems have accuracies of around 57% when tested on verbs (see Table 1). A likely explanation for this discrepancy is that different senses of common verbs can occur in similar lexical contexts, thereby decreasing the effectiveness of "bag-of-words" models.

Verbs also pose serious challenges in a very different task: syntactic parsing. Verb phrases are syntactically complex and frought with pitfalls for automated parsers, such as prepositional phrase attachment ambiguities. These challenges may be partially mitigated by the fact that particular verbs often have strong preferences for particular SCFs. Unfortunately, it is not the case that each verb consistently takes the same SCF. More often, a verb has several preferred SCFs, with rarer forms also occurring, for example, in idioms. Jurafsky (1998) proposes us-

	Ø	NP	PP	NPPP	VPto	VPing
2:30:00	4	1	0	0	20	33
2:30:01	1	7	0	4	0	0
2:42:04	12	0	3	0	0	1

Table 2: The learned joint distribution over the senses and subcategorizations of the verb *begin* (in percent probability). Low probability senses and subcategorizations have been omitted.

ing a probabilistic framework to represent subcategorization preferences, where each lexical item has a corresponding distribution over the possible sets of arguments. Modeling these distributions may be useful: Collins (2003) has shown that verb subcategorization information can be used to improve syntactic parsing performance.

It has also been recognized that a much more accurate prediction of verb subcategorization preference can be made if conditioned on the sense of the verb. Roland and Jurafsky (2002) conclude that for a given lexical token in English, verb sense is the best determiner of SCF, far outweighing either genre or dialect. Demonstrating the utility of this, Korhonen and Preiss (2003) achieve significant improvement at a verb subcategorization acquisition task by conditioning on the verb sense as predicted by a statistical word sense disambiguation system. Conversely, if different senses have distinct subcategorization preferences, it is reasonable to expect that information about the way a verb subcategorizes in a particular case may be of significant utility in determining the verb's sense. As an example, Yarowsky (2000) makes use of rich syntactic features to improve the performance of a supervised WSD system.

As an illustration of this correlation, Table 2 shows a learned joint distribution over sense and SCF for the common verb *begin*.[1] Its common senses, taken from WordNet, are as follows: sense 2:30:00, to initiate an action or activity, ("begin working"), sense 2:30:01, to set in motion or cause to start, ("to begin a war"), and sense 2:42:04, to have a beginning, ("the day began"). The SCFs shown here are a subset of the complete set of SCFs, described in Table 3. Note that the sense and SCF variables are highly correlated for this verb. Sense 2:30:00 occurs almost entirely with verb phrase arguments, sense 2:30:01 occurs almost entirely as a transitive verb, and sense 2:42:04 occurs as an intransitive verb (no arguments following the verb). It should be evident that the strong correlation between these two variables can be exploited to increase performance in the tasks of predicting their values in either direction, even when the evidence is weak or uncertain.

1.2 Learning a Joint Model

Performing joint inference requires learning a joint distribution over sense and SCF for each verb. In order to estimate the joint distribution directly from data we would need a large corpus that is annotated for both verb sense and SCF. Unfortunately, no such corpus of adequate size exists.[2] Instead, there are some corpora such as SemCor and Senseval-2 labeled for sense, and others that are parsed and from which it is possible to compute verb SCFs deterministically. In the current work we use two corpora to learn a joint model: Senseval-2, labeled for sense but not syntax, and the Penn Treebank, labeled for syntax but not sense. We do so by treating the two data sets as a single one with incompletely labeled instances. This partially labeled data set then yields a semi-supervised learning problem, suitable for the Expectation-Maximization (EM) algorithm (Dempster et al., 1977).

2 Tasks and Data Sets

We evaluate our system on both the WSD task and the verb SCF determination task. We describe each task in turn.

2.1 Word Sense Disambiguation

We used as our sense-annotated corpus the data sets from the English lexical sample portion of the Senseval-2 word sense disambiguation competition (Kilgarriff and Rosenzweig, 2000). This data set contains multiple instances of 73 different English word types, divided into training and testing examples. Each word type is marked for part of speech, so that the sense disambiguation task does not need to distinguish between senses that have different parts of speech. We selected from this data set all 29 words that were marked as verbs.

Each example consists of a marked occurrence of the target word in approximately 100 words of surrounding context. The correct sense of the word, marked by human annotators, is also given. Each instance is labeled with a sense corresponding to a *synset* from WordNet (Miller, 1995). The number of senses per word varies enormously: some words have more than 30 senses, while others have five

[1] We cannot show an empirical joint distribution because of the lack of a sufficiently large jointly annotated corpus, as discussed below.

[2] A portion of the Brown corpus has been used both in the construction of the SemCor word sense database and in the construction of the Penn Treebank, but coverage is very low, especially for sense markings, and the individual sentences have not to our knowledge been explicitly aligned.

or fewer. These "fine-grained" senses are also partitioned into a smaller number of "coarse-grained" senses, and systems are evaluated according to both metrics. The number of training and testing examples per word varies from tens to nearly a thousand. We used the same train/test division as in Senseval-2, so that our reported accuracy numbers are directly comparable with those of other Senseval-2 submissions, as given in Table 1.

2.2 Verb Subcategorization

We use as our SCF-annotated corpus sentences drawn from the Wall Street Journal section of the Penn Treebank. For each target verb we select sentences containing a form of the verb (tagged as a verb) with length less than 40 words. We select training examples from sections 2 through 21, and test examples from all other sections.[3]

There are many conceivable ways to partition the set of possible verb argument combinations into SCFs. One possible approach would be to use as the SCF representation the raw sequence of constituents occurring in the verb phrase. This is certainly an unbiased representation, but as there are many thousands of rewrites for VP in the Penn Treebank, data sparsity would present a significant problem. In addition, many of the variants do not contain useful information for our task: for example, we wouldn't expect to get much value from knowing about the presence or absence of an adverb in the phrase. Instead, we chose to use a small number of linguistically motivated SCFs which form a partition over the large space of possible verb arguments.

We chose as a starting point the SCF partition specified in Roland (2001). These SCFs are defined declaratively using a set of tgrep expressions that match appropriate verb phrases.[4] We made significant modifications to the set of SCFs, and also simplified the tgrep expressions used to match them.

One difference from Roland's SCF set is that we analyze verb particles as arguments, so that several SCFs differ only in the existence of a particle. This is motivated by the fact that the particle is a syntactic feature that provides strong evidence about the verb sense. One might argue that the presence of a particle should be considered a lexical feature modeled independently from the SCF, but the distinction is blurry, and we have instead combined the variables in favor of model simplicity. A second difference is

Subcat	Description
∅	No arguments
NP	Transitive
PP	Prepositional phrase
NP PP	Trans. with prep. phrase
VPing	Gerundive verb phrase
NP VPing	Perceptual complement
VPto	Intrans. w/ infinitival VP
NP VPto	Trans. w/ infinitival VP
S for to	Intrans. w/ *for* PP and infin. VP
NP SBAR	Trans. w/ finite clause
NP NP	Ditransitive
PRT	Particle and no args.
NP PRT	Transitive w/ particle
PP PRT	Intrans. w/ PP and particle
VP PRT	Intrans. w/ VP and particle
SBAR PRT	Intrans. w/ fin. clause and part.
Other	None of the above

Table 3: The 17 subcategorization frames we use.

that unlike Roland, we do not put passive verb constructions in a separate "passive" SCF, but instead we undo the passivization and put them in the underlying category. Although one might expect that passivization itself is a weak indicator of sense, we believe that the underlying SCF is more useful. Our final set of SCFs is shown in Table 3.

Given a sentence annotated with a syntactic parse, the SCF of the target verb can be computed by attempting to match each of the SCF-specific tgrep expressions with the verb phrase containing the target verb. Unlike those given by Roland, our tgrep expressions are not designed to be mutually exclusive; instead we determine verb SCF by attempting matches in a prescribed sequence, using "if-then-else" logic.

3 Model Structure and Inference

Our generative probabilistic model can be thought of as having three primary components: the sense model, relating the verb sense to the surrounding context, the subcategorization model, relating the verb subcategorization to the sentence, and the joint model, relating the sense and SCF of the verb to each other. More formally, the model is a factored representation of a joint distribution over these variables and the data: the verb sense (V), the verb SCF (C), the unordered context "bag-of-words" (W), and the sentence as an ordered sequence of words (S). The joint distribution $\mathrm{P}(V, C, W, S)$ is then factored as

$$\mathrm{P}(V)\mathrm{P}(C|V)\mathrm{P}(S|C)\prod_i \mathrm{P}(W_i|V)$$

[3] Sections 2 through 21 of the WSJ are typically used for training PCFG parsers, and section 23 is typically used for testing. Because of sparse data we drew our test examples from all non-training sections.

[4] tgrep is a tree node matching program written by Richard Pito, distributed with the Penn Treebank.

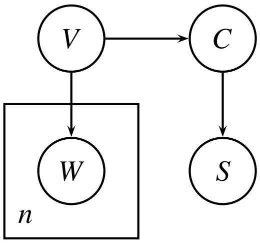

Figure 1: A graphical representation of the combined sense and subcategorization probabilistic model. Note that the box defines a *plate*, indicating that the model contains n copies of this variable.

where W_i is the word type occurring in each position of the context (including the target sentence itself). The first two terms together define a joint distribution over verb sense (V) and SCF (C), the third term defines the subcategorization model, and the last term defines the sense model. A graphical model representation is shown in Figure 1.

The model assumes the following generative process for a data instance of a particular verb. First we generate the sense of the target verb. Conditioned on the sense, we generate the SCF of the verb. (Note that the decision to generate sense and then SCF is arbitrary and forced by the desire to factor the model; we discuss reversing the order below.) Then, conditioned on the sense of the verb, we generate an unordered collection of context words. (For the Senseval-2 corpus, this collection includes not only the words in the sentence in which the verb occurs, but also the words in surrounding sentences.) Finally, conditioned on the SCF of the verb, we generate the immediate sentence containing the verb as an ordered sequence of words.

An apparent weakness of this model is that it double-generates the context words from the enclosing sentence: they are generated once by the sense model, as an unordered collection of words, and once by the subcategorization model, as an ordered sequence of words. The model is thus deficient in that it assigns a large portion of its probability mass to impossible cases: those instances which have words in the context which do not match those in the sentence. However because the sentences are always observed, we only consider instances in the set of consistent cases, so the deficiency should be irrelevant for the purpose of reasoning about sense and SCF.

We discuss each of the model components in turn.

3.1 Verb Sense Model

The verb sense component of our model is an ordinary multinomial Naive Bayes "bag-of-words" model: $\mathrm{P}(V)\prod_i \mathrm{P}(W_i|V)$. We learn the marginal over verb sense with maximum likelihood estimation (MLE) from the sense annotated data. We learn the sense-conditional word model using smoothed MLE from the sense annotated data, and to smooth we use Bayesian smoothing with a Dirichlet prior. The free smoothing parameter is determined empirically, once for all words in the data set. In the independent sense model, to infer the most likely sense given a context of words $\mathrm{P}(S|\mathbf{W})$, we just find the V that maximizes $\mathrm{P}(V)\prod_i \mathrm{P}(W_i|V)$. Inference in the joint model over sense and SCF is more complex, and is described below.

In order to make our system competitive with leading WSD systems we made an important modification to this basic model: we added relative position feature weighting. It is known that words closer to the target word are more predictive of sense, so it is reasonable to weight them more highly. We define a set of "buckets", or partition over the position of the context word relative to the target verb, and we weight each context word feature with a weight given by its bucket, both when estimating model parameters at train time and when performing inference at test time. We use the following 8 relative position buckets: $(-\infty, -6]$, $[-5, -3]$, -2, -1, 1, 2, [3, 5], and $[6, \infty)$. The bucket weights are found empirically using a simple optimization procedure on k-fold training set accuracy. In ablation tests on this system we found that the use of relative position feature weighting, when combined with corresponding evidence attenuation (see Section 3.3) increased the accuracy of the standalone verb sense disambiguation model from 46.2% to 54.0%.

3.2 Verb Subcategorization Model

The verb SCF component of our model $\mathrm{P}(S|C)$ represents the probability of particular sentences given each possible SCF. Because there are infinitely many possible sentences, a multinomial representation is infeasible, and we instead chose to encode the distribution using a set of probabilistic context free grammars (PCFGs). A PCFG is created for each possible SCF: each PCFG yields only parse trees in which the distinguished verb subcategorizes in the specified manner (but other verbs can parse freely). Given a SCF-specific PCFG, we can determine the probability of the sentence using the *inside algorithm*, which sums the probabilities of all possible trees in the grammar producing the sentence. To do this, we modified the exact PCFG parser of Klein and Manning (2003). In the independent SCF model, to infer the most likely SCF given a sentence $\mathrm{P}(C|S)$, we just find the C that maximizes

$P(S|C)P(C)$. (For the independent model, the SCF prior is estimated using MLE from the training examples.) Inference in the joint model over sense and SCF is more complex, and is described below.

Learning this model, SCF-specific PCFGs, from our SCF-annotated training data, requires some care. Commonly PCFGs are learned using MLE of rewrite rule probabilities from large sets of tree-annotated sentences. Thus to learn SCF-specific PCFGs, it seems that we should select a set of annotated sentences containing the target verb, determine the SCF of the target verb in each sentence, create a separate corpus for each SCF of the target verb, and then learn SCF-specific grammars from the SCF-specific corpora. If we are careful to distinguish rules which dominate the target verb from those which do not, then the grammar will be constrained to generate trees in which the target verb subcategorizes in the specified manner, and other verbs can occur in general tree structures. The problem with this approach is that in order to create a broad-coverage grammar (which we will need in order for it to generalize accurately to unseen test instances) we will need a very large number of sentences in which the target verb occurs, and we do not have enough data for this approach.

Because we want to maximize the use of the available data, we must instead make use of *every verb occurrence* when learning SCF-specific rewrite rules. We can accomplish this by making a copy of each sentence for each verb occurrence (not just the target verb), determining the SCF of the distinguished verb in each sentence, partitioning the sentence copies by distinguished verb SCF, and learning SCF-specific grammars using MLE. Finally, we change the lexicon by forcing the distinguished verb tag to rewrite to only our target verb. The method we actually use is functionally equivalent to this latter approach, but altered for efficiency. Instead of making copies of sentences with multiple verbs, we use a dense representation. We determine the SCF of each verb in the sentence, and then annotate the verb and all nonterminal categories occurring above the verb in the tree, up to the root, with the SCF of the verb. Note that some nonterminals will then have multiple annotations. Then to learn a SCF-specific PCFG, we count rules that have the specified SCF annotation as rules which can dominate the distinguished verb, and then count all rules (including the SCF-specific ones) as general rules which cannot dominate the distinguished verb.

3.3 The Joint Model

Given a fully annotated dataset, it is trivial to learn the parameters of the joint distribution over verb sense and SCF $P(V, C)$ using MLE. However, because we do not have access to such a dataset, we instead use the EM algorithm to "complete" the missing annotations with expectations, or soft assignments, over the values of the missing variable (we present the EM algorithm in detail in the next section). Given this "completed" data, it is again trivial to learn the parameters of the joint probability model using smoothed MLE. We use simple Laplace add-one smoothing to smooth the distribution.

However, a small complication arises from the fact that the marginal distributions over senses and SCFs for a particular verb may differ between the two halves of our data set. They are, after all, wholly different corpora, assembled by different people for different purposes. For this reason, when testing the system on the sense corpus we'd like to use a sense marginal distribution trained from the sense corpus, and when testing the system on the SCF corpus we'd like to use a SCF marginal distribution trained from the SCF corpus. To address this, recall from above that the factoring we choose for the joint distribution is arbitrary. When performing sense inference we use the model $P_v(V)P_j(C|V)$ where $P_j(C|V)$ was learned from the complete data, and $P_v(V)$ was learned from the sense-marked examples only. When performing SCF inference we use the equivalent factoring $P_c(C)P_j(V|C)$, where $P_j(V|C)$ was learned from the complete data, and $P_c(C)$ was learned from the SCF-annotated examples only.

We made one additional modification to this joint model to improve performance. When performing inference in the model, we found it useful to differentially weight different probability terms. The most obvious need for this comes from the fact that the sense-conditional word model employs relative position feature weighting, which can change the relatively magnitude of the probabilities in this term. In particular, by using feature weights greater than 1.0 during inference we overestimate the actual amount of evidence. Even without the feature weighting, however, the word model can still overestimate the actual evidence given that it encodes an incorrect independence assumption between word features (of course word occurrence in text is actually very highly correlated). The PCFG model also suffers from a less severe instance of the same problem: human languages are of course not context free, and there is in fact correlation between supposedly independent tree structures in different parts of the tree. To remedy this evidence overconfidence, it is helpful to attenuate or downweight the evidence terms accordingly. More generally, we place weights on each of the probability terms used in inference calculations, yielding models of the following form:

$$P(V)^{\alpha(v)}P(C|V)^{\alpha(c)}P(S|C)^{\alpha(s)}[\prod_i P(W_i|V)]^{\alpha(w)}$$

These $\alpha(\cdot)$ weights are free parameters, and we find them by simple optimization on k-fold accuracy. In ablation tests on this system, we found that term weighting (particularly evidence attenuation) increased the accuracy of the standalone sense model from 51.9% to 54.0% at the fine-grained verb sense disambiguation task.

We now describe the precise EM algorithm used. Prior to running EM we first learn the independent sense and SCF model parameters from their respective datasets. We also initialize the joint sense and SCF distribution to the uniform distribution. Then we iterate over the following steps:

- E-step: Using the current model parameters, for each datum in the sense-annotated corpus, compute expectations over the possible SCFs, and for each datum in the SCF-annotated corpus, compute expectations over the possible senses.
- M-step: use the completed data to reestimate the joint distribution over sense and SCF.

We run EM to convergence, which for our dataset occurs within 6 iterations. Additional iterations do not change the accuracy of our model. Early stopping of EM after 3 iterations was found to hurt k-fold sense accuracy by 0.1% and SCF accuracy by 0.2%. Early stopping of EM after only 1 iteration was found to hurt k-fold sense accuracy by a total of 0.2% and SCF accuracy by 0.4%. These may seem like small differences, but significant relative to the advantages given by the joint model (see below).

In the E-step of EM, it is necessary to do inference over the joint model, computing posterior expectations of unknown variables conditioned on evidence variables. During the testing phase, it is also necessary to do inference, computing *maximum a posteriori* (MAP) values of unknown variables conditioned on evidence variables. In all cases we do exact Bayesian network inference, which involves conditioning on evidence variables, summing over extraneous variables, and then either maximizing over the resulting factors of query variables, or normalizing them to obtain distributions of query variables. At test time, when querying about the MAP sense (or SCF) of an instance, we chose to maximize over the marginal distribution, rather than maximize over the joint sense and SCF distribution. We found empirically that this gave us higher accuracy at the individual tasks. If instead we were doing joint prediction, we would expect high accuracy to result from maximizing over the joint.

4 Results and Discussion

In Figures 2, 3 and 4 we compare the performance of the independent and joint models on the verb sense disambiguation and verb SCF determination problems, evaluated using both 10-fold cross-validation accuracy and test set accuracy. In Figure 2, we report the performance of a system resulting from doing optimization of free parameters (such as feature and term weights) on a per-verb basis. We also provide a baseline computed by guessing the most likely class.

Although the parameter optimization of Figure 2 was performed with respect to 10-fold cross-validation on the training sets, its lower performance on the test sets suggests that it suffers from overfitting. To test this hypothesis we also trained and tested on the test sets a version of the system with corpus-wide free parameter optimization, and the results of this test are shown in Figure 3. The lower gap between the training set cross-validation and test set performance on the WSD task confirms our overfitting hypothesis. However, note that the gap between training set cross-validation and test set performance on the SCF determination task persists (although it is diminished slightly). We believe that this results from the fact that there is significant data drift between the training sections of the WSJ in the Penn Treebank (sections 2 through 21) and all other sections.

Using corpus-wide optimization, the joint model improves sense disambiguation accuracy by 1.9% over the independent model, bringing our system to 55.9% accuracy on the test set, performance that is comparable with that of the state of the art systems on verbs given in Table 1. The joint model reduces sense disambiguation error by 4.1%. On the verb SCF determination task, the joint model yields a 2.1% improvement in accuracy over the independent model, reducing total error by 5.1%.

We also report results of the independent and joint systems on each verb individually in Table 4 Not surprisingly, making use of the joint distribution was much more helpful for some verbs than others.

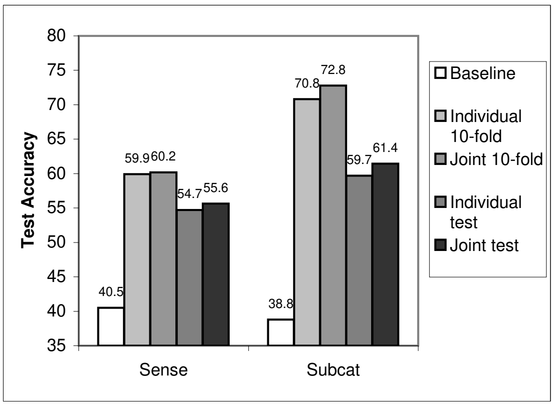

Figure 2: Chart comparing results of independent and joint systems on the verb sense and SCF tasks, evaluated with 10-fold cross-validation on the training sets and on the test sets. The baseline shown is guessing most likely class. These systems used per-verb optimization of free parameters.

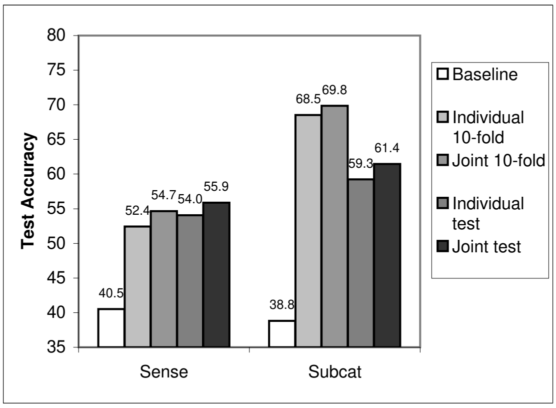

Figure 3: Chart comparing results of independent and joint systems on the verb sense and SCF tasks. These systems used corpus-wide optimization of free parameters.

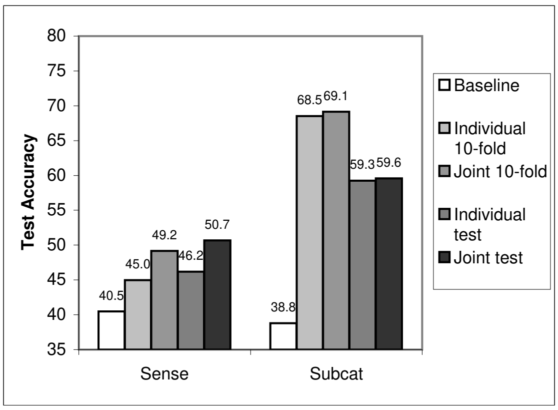

Figure 4: Chart comparing results of independent and joint systems on the verb sense and SCF tasks. This system has no relative position word feature weighting and no term weighting.

Verb	**Indep Sense**	**Joint Sense**	**Indep Subcat**	**Joint Subcat**
begin	76.8	84.3	57.0	63.3
call	39.4	42.4	44.9	49.0
carry	45.5	40.9	63.3	70.0
collaborate	90.0	90.0	100.0	100.0
develop	42.0	39.1	69.7	69.7
draw	29.3	26.8	72.7	63.6
dress	59.3	59.3	NA	NA
drift	43.8	40.6	50.0	50.0
drive	45.2	52.4	54.5	54.5
face	81.7	80.6	82.4	82.4
ferret	100.0	100.0	NA	NA
find	23.5	29.4	61.1	64.8
keep	46.3	58.2	52.1	53.5
leave	47.0	54.5	36.4	40.0
live	62.7	65.7	85.7	85.7
match	57.1	54.8	58.3	66.7
play	42.4	45.5	66.7	61.9
pull	28.3	26.7	44.4	55.6
replace	57.8	62.2	56.0	60.0
see	40.6	39.1	53.6	55.1
serve	60.8	52.9	72.0	72.0
strike	37.0	27.8	50.0	50.0
train	55.6	55.6	40.0	40.0
treat	52.3	54.5	69.2	76.9
turn	29.9	29.9	46.3	50.0
use	65.8	68.4	69.7	68.8
wander	78.0	80.0	NA	NA
wash	50.0	41.7	0.0	0.0
work	41.7	43.3	67.9	66.1

Table 4: Comparison of the performance of the independent and joint inference models on the verb sense and SCF tasks,evaluated on the Senseval-2 test set, for each of the 29 verbs in the study. These results were obtained with no per-verb parameter optimization. Note the great variation in problem difficulty and joint model performance across verbs.

For example, on the verbs *begin*, *drive*, *find*, *keep*, *leave*, and *work*, the joint model gives a greater than 5% accuracy boost on the WSD task. In contrast, for some other verbs, the joint model showed a slight decrease in accuracy on the test set relative to the independent model.

We present a few representative examples where the joint model makes better decisions than the individual model. In the sentence

> … prices *began* weakening last month after Campeau hit a cash crunch.

the sense model (based on bag-of-words evidence) believes that the sense 2:42:04 is most likely (see Table 2 for senses and joint distribution). However, the SCF model gives high weight to the frames VPto and VPing, which when combined with the joint distribution, give much more probability to the sense 2:30:00. The joint model thus correctly chooses sense 2:30:00. In the sentence

> ... before *beginning* a depressing eight-year slide that continued through last year.

the sense model again believes that the sense 2:42:04 is most likely. However, the SCF model correctly gives high weight to the NP frame, which when combined with the joint distribution, gives much more probability to the sense 2:30:01. The joint model thus correctly chooses sense 2:30:01.

Given the amount of information contained in the joint distribution it is surprising that the joint model doesn't yield a greater advantage over the independent models. It seems to be the case that the word sense model is able to capture much of the SCF information by itself, without using an explicit syntactic model. This results from the relative position weighting, since many of our SCFs correlate highly with the presence of small sets of words in particular positions (for instance, the infinitival "to", prepositions, and pronouns). We tested this hypothesis by examining how the addition of SCF information affected performance of a weaker sense model, obtained by removing feature and term weighting. The results are shown in Figure 4. Indeed, when using this weaker word sense model, the joint model yields a much larger 4.5% improvement in WSD accuracy.

5 Future Work

We can imagine several modifications to the basic system that might improve performance. Most importantly, more specific use could be made of SCF information besides modeling its joint distribution with sense, for example conditioning on headwords of (perceived) arguments, especially particles and prepositions. Second, although we made some attempt at extracting the "underlying" SCF of verbs by analyzing passive constructions separately, similar analysis of other types of movement such as relative clauses may also be useful. Third, we could hope to get some improvement from changing our model structure to address the issue of double-generation of words discussed in section 3. One way this could be done would be to use a parser only to estimate the probability of the sequence of word *tags* (i.e., parts of speech) in the sentence, then to use a sense-specific lexicon to estimate the probability of finding the words under the tags.

Although we chose WSD and SCF determination as a test case, the approach of this paper is applicable to other pairs of tasks. It may also be possible to improve parsing accuracy on verb phrases or other phrases, by simultaneously resolving word sense ambiguities, as attempted unsuccessfully by Bikel (2000). This work is intended to introduce a general methodology for combining disjoint NLP tasks that is of use outside of these specific tasks.

6 Acknowledgements

This paper is based on work supported in part by the Advanced Research and Development Activity (ARDA)'s Advanced Question Answering for Intelligence (AQUAINT) Program, and by the National Science Foundation under Grant No. IIS-0085896, as part of the Knowledge Discovery and Dissemination program. We additionally thank the reviewers for their insightful comments.

References

Daniel M. Bikel. 2000. A statistical model for parsing and word-sense disambiguation. *Joint SIGDAT Conference on Empirical Methods in Natural Language Processing and Very Large Corpora*.

Michael Collins. 2003. Head-driven statistical models for natural language parsing. *To appear in Computational Linguistics*.

A.P. Dempster, N.M. Laird, and D.B. Rubin. 1977. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B (Methodological)*, 39(1):1–38.

Daniel Jurafsky. 1998. A probabilistic model of lexical and syntactic access and disambiguation. *Cognitive Science*, 20(2):139–194.

Adam Kilgarriff and Joseph Rosenzweig. 2000. Framework and results for english senseval. *Computers and the Humanities*, 34(1-2):15–48.

Dan Klein and Christopher Manning. 2003. Accurate unlexicalized parsing. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics*.

Anna Korhonen and Judita Preiss. 2003. Improving subcategorization acquisition using word sense disambiguation. In *Proceedings of the 41st Annual Meeting of the Association for Computational Linguistics, Sapporo, Japan*, pages 48–55.

G.A. Miller. 1995. Wordnet: A lexical database for english. *Communications of the ACM*, 38(11):39–41.

Douglas Roland and Daniel Jurafsky. 2002. Verb sense and verb subcategorization probabilities. In Paola Merlo and Suzanne Stevenson, editors, *The Lexical Basis of Sentence Processing*, chapter 16. John Benjamins, Amsterdam.

Douglas Roland. 2001. *Verb Sense and Verb Subcategorization Probabilities*. Ph.D. thesis, University of Colorado.

David Yarowsky and Radu Florian. 2002. Evaluating sense disambiguation across diverse parameter spaces. *Natural Language Engineering*, 8(4):293–310.

David Yarowsky. 2000. Hierarchical decision lists for word sense disambiguation. *Computers and the Humanities*, 34(1-2).