Applying Alternating Structure Optimization to Word Sense Disambiguation

Rie Kubota Ando
IBM T.J. Watson Research Center
Yorktown Heights, NY 10598, U.S.A.
rie1@us.ibm.com

Abstract

This paper presents a new application of the recently proposed machine learning method *Alternating Structure Optimization (ASO)*, to word sense disambiguation (WSD). Given a set of WSD problems and their respective labeled examples, we seek to improve overall performance on that set by using all the labeled examples (irrespective of target words) for the entire set in learning a disambiguator for each individual problem. Thus, in effect, on each individual problem (e.g., disambiguation of "art") we benefit from training examples for other problems (e.g., disambiguation of "bar", "canal", and so forth). We empirically study the effective use of ASO for this purpose in the multi-task and semi-supervised learning configurations. Our performance results rival or exceed those of the previous best systems on several Senseval lexical sample task data sets.

1 Introduction

Word sense disambiguation (WSD) is the task of assigning pre-defined senses to words occurring in some context. An example is to disambiguate an occurrence of "bank" between the "money bank" sense and the "river bank" sense. Previous studies e.g., (Lee and Ng, 2002; Florian and Yarowsky, 2002), have applied supervised learning techniques to WSD with success.

A practical issue that arises in supervised WSD is the paucity of labeled examples (sense-annotated data) available for training. For example, the training set of the Senseval-2[1] English lexical sample task has only 10 labeled training examples per sense on average, which is in contrast to nearly 6K training examples per name class (on average) used for the CoNLL-2003 named entity chunking shared task[2]. One problem is that there are so many words and so many senses that it is hard to make available a sufficient number of labeled training examples for each of a large number of target words.

On the other hand, this indicates that the total number of available labeled examples (irrespective of target words) can be relatively large. A natural question to ask is whether we can effectively use *all* the labeled examples (irrespective of target words) for learning on each individual WSD problem.

Based on these observations, we study a new application of *Alternating Structure Optimization (ASO)* (Ando and Zhang, 2005a; Ando and Zhang, 2005b) to WSD. ASO is a recently proposed machine learning method for learning predictive structure (i.e., information useful for predictions) shared by multiple prediction problems via joint empirical risk minimization. It has been shown that on several tasks, performance can be significantly improved by a semi-supervised application of ASO, which obtains useful information from *unlabeled data* by learning automatically created prediction problems. In addition to such semi-supervised learning, this paper explores ASO *multi-task learning*, which learns a number of WSD problems simultaneously to exploit the inherent predictive structure shared by these WSD problems. Thus, in effect, each individual problem (e.g., disambiguation of "art") benefits from *labeled training examples for other problems* (e.g., disambiguation of "bar", disambiguation of "canal", and so forth).

The notion of benefiting from training data for other word senses is not new by itself. For instance,

[1] http://www.cs.unt.edu/~rada/senseval/. WSD systems have been evaluated in the series of Senseval workshops.

[2] http://www.cnts.ua.ac.be/conll2003/ner/

on the WSD task with respect to WordNet synsets, Kohomban and Lee (2005) trained classifiers for the top-level synsets of the WordNet semantic hierarchy, consolidating labeled examples associated with the WordNet sub-trees. To disambiguate test instances, these coarse-grained classifiers are first applied, and then fine-grained senses are determined using a heuristic mapping. By contrast, our approach does not require pre-defined relations among senses such as the WordNet hierarchy. Rather, we let the machine learning algorithm ASO automatically and implicitly find relations with respect to the disambiguation problems (i.e., finding shared predictive structure). Interestingly, in our experiments, seemingly unrelated or only loosely related word-sense pairs help to improve performance.

This paper makes two contributions. First, we present a new application of ASO to WSD. We empirically study the effective use of ASO and show that labeled examples of all the words can be effectively exploited in learning each individual disambiguator. Second, we report performance results that rival or exceed the state-of-the-art systems on Senseval lexical sample tasks.

2 Alternating structure optimization

This section gives a brief summary of ASO. We first introduce a standard linear prediction model for a single task and then extend it to a joint linear model used by ASO.

2.1 Standard linear prediction models

In the standard formulation of supervised learning, we seek a *predictor* that maps an input vector (or *feature vector*) $\mathbf{x} \in \mathcal{X}$ to the corresponding output $y \in \mathcal{Y}$. For NLP tasks, binary features are often used – for example, if the word to the left is "money", set the corresponding entry of $\mathbf{x}$ to 1; otherwise, set it to 0. A k-way classification problem can be cast as k binary classification problems, regarding output $y = +1$ and $y = -1$ as "in-class" and "out-of-class", respectively.

Predictors based on *linear prediction models* take the form: $f(\mathbf{x}) = \mathbf{w}^T\mathbf{x}$, where $\mathbf{w}$ is called a *weight vector*. A common method to obtain a predictor $\hat{f}$ is regularized *empirical risk minimization*, which minimizes an empirical loss of the predictor (with regularization) on the n labeled training examples $\{(\mathbf{X}_i, Y_i)\}$:

$$\hat{f} = \arg\min_f \left(\sum_{i=1}^{n} L(f(\mathbf{X}_i), Y_i) + r(f) \right) . \quad (1)$$

A *loss function* $L(\cdot)$ quantifies the difference between the prediction $f(\mathbf{X}_i)$ and the true output Y_i, and $r(\cdot)$ is a regularization term to control the model complexity.

2.2 Joint linear models for ASO

Consider m prediction problems indexed by $\ell \in \{1, \ldots, m\}$, each with n_ℓ samples $(\mathbf{X}_i^\ell, Y_i^\ell)$ for $i \in \{1, \ldots, n_\ell\}$, and assume that there exists a low-dimensional predictive structure shared by these m problems. Ando and Zhang (2005a) extend the above traditional linear model to a joint linear model so that a predictor for problem ℓ is in the form:

$$f_\ell(\Theta, \mathbf{x}) = \mathbf{w}_\ell^T\mathbf{x} + \mathbf{v}_\ell^T\Theta\mathbf{x} \,, \;\; \Theta\Theta^T = \mathbf{I} \,, \quad (2)$$

where $\mathbf{I}$ is the identity matrix. $\mathbf{w}_\ell$ and $\mathbf{v}_\ell$ are weight vectors specific to each problem ℓ. Predictive structure is parameterized by the *structure matrix* Θ shared by all the m predictors. The goal of this model can also be regarded as learning a common good feature map $\Theta\mathbf{x}$ used for all the m problems.

2.3 ASO algorithm

Analogous to (1), we compute Θ and predictors so that they minimize the empirical risk summed over all the problems:

$$[\hat{\Theta}, \{\hat{f}_\ell\}] = \arg\min_{\Theta, \{f_\ell\}} \sum_{\ell=1}^{m} \left(\sum_{i=1}^{n_\ell} \frac{L(f_\ell(\Theta, \mathbf{X}_i^\ell), Y_i^\ell)}{n_\ell} + r(f_\ell) \right) . \quad (3)$$

It has been shown in (Ando and Zhang, 2005a) that the optimization problem (3) has a simple solution using *singular value decomposition (SVD)* when we choose square regularization: $r(f_\ell) = \lambda\|\mathbf{w}_\ell\|_2^2$ where λ is a regularization parameter. Let $\mathbf{u}_\ell = \mathbf{w}_\ell + \Theta^T\mathbf{v}_\ell$. Then (3) becomes the minimization of the joint empirical risk written as:

$$\sum_{\ell=1}^{m} \left(\sum_{i=1}^{n_\ell} \frac{L(\mathbf{u}_\ell^T\mathbf{X}_i^\ell, Y_i^\ell)}{n_\ell} + \lambda\|\mathbf{u}_\ell - \Theta^T\mathbf{v}_\ell\|_2^2 \right) . \quad (4)$$

This minimization can be approximately solved by repeating the following alternating optimization procedure until a convergence criterion is met:

Nouns	art, authority, bar, bum, chair, channel, child, church, circuit, day, detention, dyke, facility, fatigue, feeling, grip, hearth, holiday, lady, material, mouth, nation, nature, post, restraint, sense, spade, stress, yew
Verbs	begin, call, carry, collaborate, develop, draw, dress, drift, drive, face, ferret, find, keep, leave, live, match, play, pull, replace, see, serve strike, train, treat, turn, use, wander wash, work
Adjectives	blind, colourless, cool, faithful, fine, fit, free, graceful, green, local, natural, oblique, simple, solemn, vital

Figure 1: Words to be disambiguated; Senseval-2 English lexical sample task.

1. Fix $(\Theta, \{\mathbf{v}_\ell\})$, and find m predictors $\{\mathbf{u}_\ell\}$ that minimizes the joint empirical risk (4).

2. Fix m predictors $\{\mathbf{u}_\ell\}$, and find $(\Theta, \{\mathbf{v}_\ell\})$ that minimizes the joint empirical risk (4).

The first step is equivalent to training m predictors independently. The second step, which couples all the predictors, can be done by setting the rows of Θ to the most significant *left singular vectors* of the *predictor (weight) matrix* $\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_m]$, and setting $\mathbf{v}_\ell = \Theta\mathbf{u}_\ell$. That is, the structure matrix Θ is computed so that the projection of the predictor matrix $\mathbf{U}$ onto the subspace spanned by Θ's rows gives the best approximation (in the least squares sense) of $\mathbf{U}$ for the given row-dimension of Θ. Thus, intuitively, Θ captures the commonality of the m predictors.

ASO has been shown to be useful in its *semi-supervised learning* configuration, where the above algorithm is applied to a number of *auxiliary problems* that are *automatically created* from the unlabeled data. By contrast, the focus of this paper is the *multi-task learning* configuration, where the ASO algorithm is applied to a number of *real problems* with the goal of improving overall performance on these problems.

3 Effective use of ASO on word sense disambiguation

The essence of ASO is to learn information useful for prediction (predictive structure) shared by multiple tasks, assuming the existence of such shared structure. From this viewpoint, consider the target words of the Senseval-2 lexical sample task, shown in Figure 1. Here we have multiple disambiguation tasks; however, at a first glance, it is not entirely clear whether these tasks share predictive structure (or are related to each other). There is no direct semantic relationship (such as synonym or hyponym relations) among these words.

Local context	word uni-grams in 5-word window, word bi- and tri-grams of (w_{-2}, w_{-1}), $(w_{+1}, w_{+2}), (w_{-1}, w_{+1})$, $(w_{-3}, w_{-2}, w_{-1}), (w_{+1}, w_{+2}, w_{+3})$, $(w_{-2}, w_{-1}, w_{+1}), (w_{-1}, w_{+1}, w_{+2})$.
Syntactic	full parser output; see Section 3 for detail.
Global	all the words excluding stopwords.
POS	uni-, bi-, and tri-grams in 5-word window.

Figure 2: Features. w_i stands for the word at position i relative to the word to be disambiguated. The 5-word window is $[-2, +2]$. Local context and POS features are position-sensitive. Global context features are position insensitive (a bag of words).

The goal of this section is to empirically study the effective use of ASO for improving overall performance on these seemingly unrelated disambiguation problems. Below we first describe the task setting, features, and algorithms used in our implementation, and then experiment with the Senseval-2 English lexical sample data set (with the official training / test split) for the development of our methods. We will then evaluate the methods developed on the Senseval-2 data set by carrying out the Senseval-3 tasks, i.e., training on the Senseval-3 training data and then evaluating the results on the (unseen) Senseval-3 test sets in Section 4.

Task setting In this work, we focus on the Senseval *lexical sample task*. We are given a set of target words, each of which is associated with several possible senses, and their labeled instances for training. Each instance contains an occurrence of one of the target words and its surrounding words, typically a few sentences. The task is to assign a sense to each test instance.

Features We adopt the feature design used by Lee and Ng (2002), which consists of the following four types: (1) *Local context*: n-grams of nearby words (position sensitive); (2) *Global context*: all the words (excluding stopwords) in the given context (position-insensitive; a bag of words); (3) *POS*: parts-of-speech n-grams of nearby words; (4) *Syn-*

tactic relations: syntactic information obtained from parser output. To generate syntactic relation features, we use the Slot Grammar-based full parser ESG (McCord, 1990). We use as features syntactic relation types (e.g., subject-of, object-of, and noun modifier), participants of syntactic relations, and bigrams of syntactic relations / participants. Details of the other three types are shown in Figure 2.

Implementation Our implementation follows Ando and Zhang (2005a). We use a modification of the Huber's robust loss for regression: $L(p, y) = (\max(0, 1-py))^2$ if $py \geq -1$; and $-4py$ otherwise; with square regularization ($\lambda = 10^{-4}$), and perform empirical risk minimization by *stochastic gradient descent (SGD)* (see e.g., Zhang (2004)). We perform one ASO iteration.

3.1 Exploring the multi-task learning configuration

The goal is to effectively apply ASO to the set of word disambiguation problems so that overall performance is improved. We consider two factors: *feature split* and *partitioning of prediction problems*.

3.1.1 Feature split and problem partitioning

Our features described above inherently consist of four *feature groups*: local context (LC), global context (GC), syntactic relation (SR), and POS features. To exploit such a natural feature split, we explore the following extension of the joint linear model:

$$f_\ell(\{\Theta_j\}, \mathbf{x}) = \mathbf{w}_\ell^T \mathbf{x} + \sum_{j \in F} \mathbf{v}_\ell^{(j)^T} \Theta_j \mathbf{x}^{(j)} , \qquad (5)$$

where $\Theta_j \Theta_j^T = \mathbf{I}$ for $j \in F$, F is a set of disjoint feature groups, and $\mathbf{x}^{(j)}$ (or $\mathbf{v}_\ell^{(j)}$) is a portion of the feature vector $\mathbf{x}$ (or the weight vector $\mathbf{v}_\ell$) corresponding to the feature group j, respectively. This is a slight modification of the extension presented in (Ando and Zhang, 2005a). Using this model, ASO computes the structure matrix Θ_j for each feature group separately. That is, SVD is applied to the sub-matrix of the predictor (weight) matrix corresponding to each feature group j, which results in more focused dimension reduction of the predictor matrix. For example, suppose that $F = \{\mathrm{SR}\}$. Then, we compute the structure matrix Θ_{SR} from the corresponding sub-matrix of the predictor matrix $\mathbf{U}$, which is the gray region of Figure 3 (a). The structure matrices Θ_j for $j \notin F$ (associated with the white regions in the figure) should be regarded as being fixed to the zero matrices. Similarly, it is possible to compute a structure matrix from a subset of the predictors (such as noun disambiguators only), as in Figure 3 (b). In this example, we apply the extension of ASO with $F = \{\mathrm{SR}\}$ to three sets of problems (disambiguation of nouns, verbs, and adjectives, respectively) separately.

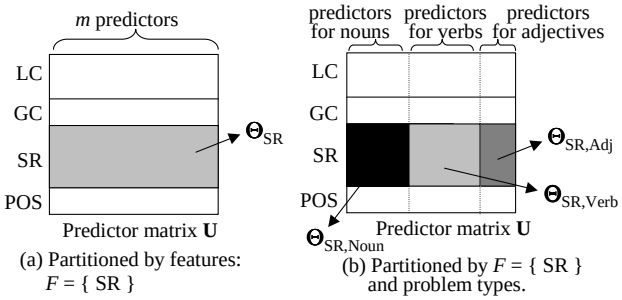

Figure 3: Examples of feature split and problem partitioning.

To see why such partitioning may be useful for our WSD problems, consider the disambiguation of "bank" and the disambiguation of "save". Since a "bank" as in "money bank" and a "save" as in "saving money" may occur in similar global contexts, certain global context features effective for recognizing the "money bank" sense may be also effective for disambiguating "save", and vice versa. However, with respect to the position-sensitive local context features, these two disambiguation problems may not have much in common since, for instance, we sometimes say "the bank announced", but we rarely say "the save announced". That is, whether problems share predictive structure may depend on feature types, and in that case, seeking predictive structure for each feature group separately may be more effective. Hence, we experiment with the configurations with and without various feature splits using the extension of ASO.

Our target words are nouns, verbs, and adjectives. As in the above example of "bank" (noun) and "save" (verb), the predictive structure of global context features may be shared by the problems irrespective of the parts of speech of the target words. However, the other types of features may be more dependent on the target word part of speech. There-

fore, we explore two types of configuration. One applies ASO to all the disambiguation problems at once. The other applies ASO separately to each of the three sets of disambiguation problems (noun disambiguation problems, verb disambiguation problems, and adjective disambiguation problems) and uses the structure matrix Θ_j obtained from the noun disambiguation problems only for disambiguating nouns, and so forth.

Thus, we explore combinations of two parameters. One is the set of feature groups F in the model (5). The other is the partitioning of disambiguation problems.

3.1.2 Empirical results

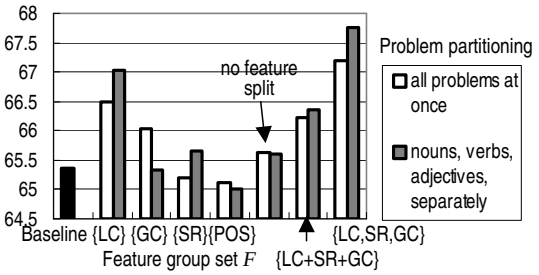

Figure 4: F-measure on Senseval-2 English test set. Multi-task configurations varying feature group set F and problem partitioning. Performance at the best dimensionality of Θ_j (in $\{10, 25, 50, 100, \cdots\}$) is shown.

In Figure 4, we compare performance on the Senseval-2 test set produced by training on the Senseval-2 training set using the various configurations discussed above. As the evaluation metric, we use the F-measure (micro-averaged)[3] returned by the official Senseval scorer. Our baseline is the standard *single-task* configuration using the same loss function (modified Huber) and the same training algorithm (SGD).

The results are in line with our expectation. To learn the shared predictive structure of local context (LC) and syntactic relations (SR), it is more advantageous to apply ASO to each of the three sets of problems (disambiguation of nouns, verbs, and adjectives, respectively), separately. By contrast, global context features (GC) can be more effectively exploited when ASO is applied to all the disambiguation problems at once. It turned out that the configuration $F = \{\mathrm{POS}\}$ does not improve the performance over the baseline. Therefore, we exclude POS from the feature group set F in the rest of our experiments. Comparison of $F = \{\mathrm{LC+SR+GC}\}$ (treating the features of these three types as one group) and $F = \{\mathrm{LC}, \mathrm{SR}, \mathrm{GC}\}$ indicates that use of this feature split indeed improves performance. Among the configurations shown in Figure 4, the best performance (67.8%) is obtained by applying ASO to the three sets of problems (corresponding to nouns, verbs, and adjectives) separately, with the feature split $F = \{\mathrm{LC}, \mathrm{SR}, \mathrm{GC}\}$.

ASO has one parameter, the dimensionality of the structure matrix Θ_j (i.e., the number of left singular vectors to compute). The performance shown in Figure 4 is the ceiling performance obtained at the best dimensionality (in $\{10, 25, 50, 100, 150, \cdots\}$). In Figure 5, we show the performance dependency on Θ_j's dimensionality when ASO is applied to all the problems at once (Figure 5 left), and when ASO is applied to the set of the noun disambiguation problems (Figure 5 right). In the left figure, the configuration $F = \{\mathrm{GC}\}$ (global context) produces better performance at a relatively low dimensionality. In the other configurations shown in these two figures, performance is relatively stable as long as the dimensionality is not too low.

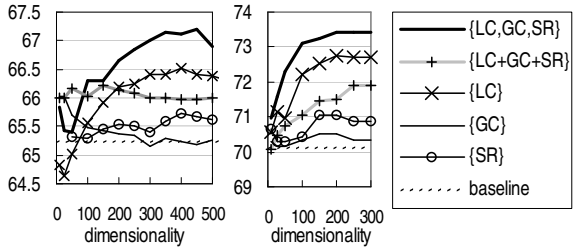

Figure 5: Left: Applying ASO to all the WSD problems at once. Right: Applying ASO to noun disambiguation problems only and testing on the noun disambiguation problems only. x-axis: dimensionality of Θ_j.

3.2 Multi-task learning procedure for WSD

Based on the above results on the Senseval-2 test set, we develop the following procedure using the feature split and problem partitioning shown in Figure 6. Let $\mathcal{N}$, $\mathcal{V}$, and $\mathcal{A}$ be sets of disambiguation problems whose target words are nouns, verbs, and adjectives, respectively. We write $\Theta_{(j,s)}$ for the struc-

[3] Our precision and recall are always the same since our systems assign exactly one sense to each instance. That is, our F-measure is the same as 'micro-averaged recall' or 'accuracy' used in some of previous studies we will compare with.

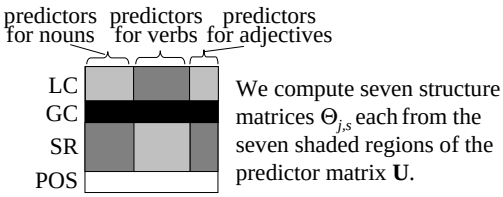

Figure 6: Effective feature split and problem partitioning.

ture matrix associated with the feature group j and computed from a problem set s. That is, we replace Θ_j in (5) with $\Theta_{(j,s)}$.

- Apply ASO to the three sets of disambiguation problems (corresponding to nouns, verbs, and adjectives), separately, using the extended model (5) with $F = \{\mathrm{LC}, \mathrm{SR}\}$. As a result, we obtain $\Theta_{(j,s)}$ for every $(j,s) \in \{\mathrm{LC}, \mathrm{SR}\} \times \{\mathcal{N}, \mathcal{V}, \mathcal{A}\}$.
- Apply ASO to all the disambiguation problems at once using the extended model (5) with $F = \{\mathrm{GC}\}$ to obtain $\Theta_{(\mathrm{GC}, \mathcal{N}\cup\mathcal{V}\cup\mathcal{A})}$.
- For a problem $\ell \in P \in \{\mathcal{N}, \mathcal{V}, \mathcal{A}\}$, our final predictor is based on the model:

$$f_\ell(\mathbf{x}) = \mathbf{w}_\ell^T \mathbf{x} + \sum_{(j,s)\in T} {\mathbf{v}_\ell^{(j,s)}}^T \Theta_{(j,s)} \mathbf{x}^{(j)} \ ,$$

  where $T = \{(\mathrm{LC}, P), (\mathrm{SR}, P), (\mathrm{GC}, \mathcal{N}\cup\mathcal{V}\cup\mathcal{A})\}$. We obtain predictor $\hat{f}_\ell$ by minimizing the regularized empirical risk with respect to $\mathbf{w}_\ell$ and $\mathbf{v}_\ell$.

We fix the dimension of the structure matrix corresponding to global context features to 50. The dimensions of the other structure matrices are set to 0.9 times the maximum possible rank to ensure relatively high dimensionality. This procedure produces 68.1% on the Senseval-2 English lexical sample test set.

3.3 Previous systems on Senseval-2 data set

Figure 7 compares our performance with those of previous best systems on the Senseval-2 English lexical sample test set. Since we used this test set for the development of our method above, our performance should be understood as the *potential performance*. (In Section 4, we will present evaluation results on the *unseen* Senseval-3 test sets.) Nevertheless, it is worth noting that our potential performance (68.1%) exceeds those of the previous best systems.

ASO multi-task learning (optimum config.)	*68.1*
classifier combination [FY02]	66.5
polynomial KPCA [WSC04]	65.8
SVM [LN02]	65.4
Our single-task baseline	*65.3*
Senseval-2 (2001) best participant	64.2

Figure 7: Performance comparison with previous best systems on Senseval-2 English lexical sample test set. FY02 (Florian and Yarowsky, 2002), WSC04 (Wu et al., 2004), LN02 (Lee and Ng, 2002)

Our single-task baseline performance is almost the same as LN02 (Lee and Ng, 2002), which uses SVM. This is consistent with the fact that we adopted LN02's feature design. FY02 (Florian and Yarowsky, 2002) combines classifiers by linear average stacking. The best system of the Senseval-2 competition was an early version of FY02. WSC04 used a polynomial kernel via the kernel Principal Component Analysis (KPCA) method (Schölkopf et al., 1998) with nearest neighbor classifiers.

4 Evaluation on Senseval-3 tasks

In this section, we evaluate the methods developed on the Senseval-2 data set above on the standard Senseval-3 lexical sample tasks.

4.1 Our methods in multi-task and semi-supervised configurations

In addition to the multi-task configuration described in Section 3.2, we test the following semi-supervised application of ASO. We first create auxiliary problems following Ando and Zhang (2005a)'s partially-supervised strategy (Figure 8) with distinct feature maps Ψ_1 and Ψ_2 each of which uses one of $\{\mathrm{LC}, \mathrm{GC}, \mathrm{SR}\}$. Then, we apply ASO to these auxiliary problems using the feature split and the problem partitioning described in Section 3.2.

Note that the difference between the multi-task and semi-supervised configurations is the source of information. The multi-task configuration utilizes the *label information* of the training examples that are labeled for the rest of the multiple tasks, and the semi-supervised learning configuration exploits a large amount of *unlabeled data*.

1. Train a classifier C_1 only using feature map Ψ_1 on the labeled data for the target task.
2. Auxiliary problems are to predict the labels assigned by C_1 to the unlabeled data, using the other feature map Ψ_2.
3. Apply ASO to the auxiliary problems to obtain Θ.
4. Using the joint linear model (2), train the final predictor by minimizing the empirical risk for fixed Θ on the labeled data for the target task.

Figure 8: Ando and Zhang (2005a)'s ASO semi-supervised learning method using partially-supervised procedure for creating relevant auxiliary problems.

	#words	#train	avg #sense per word	avg #train per sense
English	73	8611	10.7	10.0
Senseval-3 data sets				
English	57	7860	6.5	21.3
Catalan	27	4469	3.1	53.2
Italian	45	5145	6.2	18.4
Spanish	46	8430	3.3	55.5

Figure 9: Data statistics of Senseval-2 English lexical sample data set (first row) and Senseval-3 data sets. On each data set, # of test instances is about one half of that of training instances.

4.2 Data and evaluation metric

We conduct evaluations on four Senseval-3 lexical sample tasks (English, Catalan, Italian, and Spanish) using the official training / test splits. Data statistics are shown in Figure 9. On the Spanish, Catalan, and Italian data sets, we use part-of-speech information (as features) and unlabeled examples (for semi-supervised learning) provided by the organizer. Since the English data set was not provided with these additional resources, we use an in-house POS tagger trained with the PennTree Bank corpus, and extract 100K unlabeled examples from the Reuters-RCV1 corpus. On each language, the number of unlabeled examples is 5–15 times larger than that of the labeled training examples. We use syntactic relation features only for English data set. As in Section 3, we report micro-averaged F measure.

4.3 Baseline methods

In addition to the standard single-task supervised configuration as in Section 3, we test the following method as an additional baseline.

Output-based method The goal of our multi-task learning configuration is to benefit from having the labeled training examples of a number of words. An alternative to ASO for this purpose is to use directly as features the output values of classifiers trained for disambiguating the other words, which we call 'output-based method' (cf. Florian et al. (2003)). We explore several variations similarly to Section 3.1 and report the ceiling performance.

4.4 Evaluation results

Figure 10 shows F-measure results on the four Senseval-3 data sets using the official training / test splits. Both ASO multi-task learning and semi-supervised learning improve performance over the single-task baseline on all the data sets. The best performance is achieved when we combine multi-task learning and semi-supervised learning by using all the corresponding structure matrices $\Theta_{(j,s)}$ produced by both multi-task and semi-supervised learning, in the final predictors. This combined configuration outperforms the single-task supervised baseline by up to 5.7%.

Performance improvements over the supervised baseline are relatively small on English and Spanish. We conjecture that this is because the supervised performance is already close to the highest performance that automatic methods could achieve. On these two languages, our (and previous) systems outperform inter-human agreement, which is unusual but can be regarded as an indication that these tasks are difficult.

The performance of the output-based method (baseline) is relatively low. This indicates that output values or proposed labels are not expressive enough to integrate information from other predictors effectively on this task. We conjecture that for this method to be effective, the problems are required to be more closely related to each other as in Florian et al. (2003)'s named entity experiments.

A practical advantage of ASO multi-task learning over ASO semi-supervised learning is that shorter computation time is required to produce similar performance. On this English data set, training for multi-task learning and semi-supervised learning takes 15 minutes and 92 minutes, respectively, using a Pentium-4 3.20GHz computer. The computation time mostly depends on the amount of the data on which auxiliary predictors are learned. Since our experiments use unlabeled data 5–15 times larger than labeled training data, semi-supervised learning takes longer, accordingly.

	methods	English	Catalan	Italian	Spanish
ASO	multi-task learning	73.8 (+0.8)	89.5 (+1.5)	63.2 (+4.9)	89.0 (+1.0)
	semi-supervised learning	73.5 (+0.5)	88.6 (+0.6)	62.4 (+4.1)	88.9 (+0.9)
	multi-task+semi-supervised	**74.1 (+1.1)**	**89.9 (+1.9)**	**64.0 (+5.7)**	**89.5 (+1.5)**
baselines	output-based	73.0 (0.0)	88.3 (+0.3)	58.0 (-0.3)	88.2 (+0.2)
	single-task supervised learning	*73.0*	*88.0*	*58.3*	*88.0*
previous systems	SVM with LSA kernel [GGS05]	73.3	89.0	61.3	88.2
	Senseval-3 (2004) best systems	72.9 [G04]	85.2 [SGG04]	53.1 [SGG04]	84.2 [SGG04]
	inter-annotator agreement	67.3	93.1	89.0	85.3

Figure 10: Performance results on the Senseval-3 lexical sample test sets. Numbers in the parentheses are performance gains compared with the single-task supervised baseline (italicized). [G04] Grozea (2004); [SGG04] Strapparava et al. (2004).

GGS05 combined various kernels, which includes the LSA kernel that exploits unlabeled data with global context features. Our implementation of the LSA kernel with our classifier (and our other features) also produced performance similar to that of GGS05. While the LSA kernel is closely related to a special case of the semi-supervised application of ASO (see the discussion of PCA in Ando and Zhang (2005a)), our approach here is more general in that we exploit not only unlabeled data and global context features but also the labeled examples of other target words and other types of features. G04 achieved high performance on English using regularized least squares with compensation for skewed class distributions. SGG04 is an early version of GGS05. Our methods rival or exceed these state-of-the-art systems on all the data sets.

5 Conclusion

With the goal of achieving higher WSD performance by exploiting all the currently available resources, our focus was the new application of the ASO algorithm in the multi-task learning configuration, which improves performance by learning a number of WSD problems simultaneously instead of training for each individual problem independently. A key finding is that using ASO with appropriate feature / problem partitioning, labeled examples of seemingly unrelated words can be effectively exploited. Combining ASO multi-task learning with ASO semi-supervised learning results in further improvements. The fact that performance improvements were obtained consistently across several languages / sense inventories demonstrates that our approach has broad applicability and hence practical significance.

References

Rie Kubota Ando and Tong Zhang. 2005a. A framework for learning predictive structures from multiple tasks and unlabeled data. *Journal of Machine Learning Research*, 6(Nov):1817–1853. An early version was published as IBM Research Report (2004).

Rie Kubota Ando and Tong Zhang. 2005b. High performance semi-supervised learning for text chunking. In *Proceedings of ACL-2005*.

Radu Florian and David Yarowsky. 2002. Modeling consensus: Classifier combination for word sense disambiguation. In *Proceedings of EMNLP-2002*.

Radu Florian, Abe Ittycheriah, Hongyan Jing, and Tong Zhang. 2003. Named entity recognition through classifier combination. In *Proceedings of CoNLL-2003*.

Cristian Grozea. 2004. Finding optimal parameter settings for high performance word sense diambiguation. In *Proceedings of Senseval-3 Workshop*.

Upali S. Kohomban and Wee Sun Lee. 2005. Learning semantic classes for word sense disambiguation. In *Proceedings of ACL-2005*.

Yoong Keok Lee and Hwee Tou Ng. 2002. An empirical evaluation of knowledge sources and learning algorithms for word sense disambiguation. In *Proceedings of EMNLP-2002*.

Michael C. McCord. 1990. Slot Grammar: A system for simpler construction of practical natural language grammars. *Natural Language and Logic: International Scientific Symposium, Lecture Notes in Computer Science*, pages 118–145.

Bernhard Schölkopf, Alexander Smola, and Klaus-Rober Müller. 1998. Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, 10(5).

Carlo Strapparava, Alfio Gliozzo, and Claudio Giuliano. 2004. Pattern abstraction and term similarity for word sense disambiguation: IRST at Senseval-3. In *Proceedings of Senseval-3 Workshop*.

Dekai Wu, Weifeng Su, and Marine Carpuat. 2004. A kernel PCA method for superior word sense disambiguation. In *Proceedings of ACL-2004*.

Tong Zhang. 2004. Solving large scale linear prediction problems using stochastic gradient descent algorithms. In *ICML 04*, pages 919–926.