Word Sense Disambiguation using Inductive Logic Programming

Lucia Specia[1], Ashwin Srinivasan[2],[3], Ganesh Ramakrishnan[2], and Maria das Graças Volpe Nunes[1]

[1] ICMC - University of São Paulo, Trabalhador São-Carlense, 400, São Carlos, 13560-970, Brazil
{lspecia, gracan}@icmc.usp.br
[2] IBM India Research Laboratory, Block 1, Indian Institute of Technology, New Delhi 110016, India
[3] Dept. of Computer Science and Engineering & Centre for Health Informatics, University of New South Wales, Sydney, Australia
{ashwin.srinivasan, ganramkr}@in.ibm.com

Abstract. The identification of the correct sense of a word is necessary for many tasks in automatic natural language processing like machine translation, information retrieval, speech and text processing. Automatic Word Sense Disambiguation (WSD) is difficult and accuracies with state-of-the art methods are substantially lower than in other areas of text understanding like part-of-speech tagging. One shortcoming of these methods is that they do not utilize substantial sources of background knowledge, such as semantic taxonomies and dictionaries, which are now available in electronic form (the methods largely use shallow syntactic features). Empirical results from the use of Inductive Logic Programming (ILP) have repeatedly shown the ability of ILP systems to use diverse sources of background knowledge. In this paper we investigate the use of ILP for WSD in two different ways: (a) as a stand-alone constructor of models for WSD; and (b) to build interesting features, which can then be used by standard model-builders such as SVM. Our investigation is in the form of experiments that examine a monolingual WSD task using the 32 English verbs contained in the SENSEVAL-3 benchmark data; and a bilingual WSD task using 7 highly ambiguous verbs in machine translation from English to Portuguese. Background knowledge available is from eight sources that provide a wide range of syntactic and semantic information. For both WSD tasks, experimental results show that ILP-constructed models and models built using ILP-generated features have higher accuracies than those obtained using a state-of-the art feature-based technique equipped with shallow syntactic features. This suggests that the use of ILP with diverse sources of background knowledge can provide one way for making substantial progress in the field of automatic WSD.

1 Introduction

Word Sense Disambiguation (WSD) aims to identify the correct sense of an ambiguous word in a sentence. Usually described as an "intermediate task" [36]—that is, not an end in itself—it is necessary in most natural language tasks like machine translation, information retrieval, speech and text processing, and so on. That it is extremely

difficult, possibly impractical, to completely solve WSD is a long-standing view [2] and accuracies with state-of-the art methods are substantially lower than in other areas of text understanding. Part-of-speech tagging accuracies, for example, are now over 95%; in contrast, the best WSD results are still below 80%.

The principal approach adopted for the automatic construction of WSD models is a "shallow" one. In this, sample data consisting of sentences with the ambiguous words and their correct sense are represented using features capturing some limited context around the ambiguous words in each sentence. For example, features may denote two to three words on either side of an ambiguous word and the part-of-speech tags of those words. Sample data represented in this manner are then used by a statistical model constructor to build a general predictive model for disambiguating words. Results from the literature on benchmark data like those provided under the various SENSEVAL competitions[4] suggest that support vector machines (SVMs) yield models with one of the highest accuracies. Despite some improvements made in the accuracy of predictions, it is generally thought that significant progress in automatic WSD would require a "deep" approach in which access to substantial body of linguistic and world knowledge could assist in resolving ambiguities. However, the incorporation of large amounts of domain knowledge has been hampered by the following: (a) access to such information in electronic form suitable for constructing models; and (b) modeling techniques capable of utilizing diverse sources of domain knowledge. The first of these difficulties is now greatly alleviated by the availability in electronic form of very large semantic lexicons like WordNet [16], dictionaries, parsers, grammars and so on. In addition, there are now very large amounts of "shallow" data in the form of electronic text corpora from which statistical information can be readily extracted. Using these diverse sources of information is, however, beyond the capabilities of existing general-purpose statistical methods that have been used for WSD. Arguably, Inductive Logic Programming (ILP) systems provide the most general-purpose framework for dealing with such data: there are explicit provisions made for the inclusion of background knowledge of any form; the representation language is powerful enough to capture the contextual relationships that arise; and modeling is not restricted to being of a particular form (for example, classification only).

In this paper, we investigate the use of ILP for WSD in two different ways : (a) the construction of models that can be used directly to disambiguate words; and (b) the construction of interesting features that can be used by standard feature-based algorithms such as SVMs to build models to disambiguate verbs. We call the two different kinds of models "ILP models" and "ILP-assisted models". In each case, background knowledge is from eight different sources that provide syntactic and semantic information that could be useful for disambiguation. The purpose of our investigation is to examine whether using an ILP system equipped with these diverse sources of background information can substantially improve the predictive accuracy of WSD models. Our investigation is in the form of an empirical evaluation of ILP models and ILP-assisted models on WSD data arising from two different tasks: (1) monolingual disambiguation of 32 English verbs contained in SENSEVAL-3; and (2) bilingual disambiguation of the Portuguese sense of 7 highly ambiguous English verbs in an English-to-Portuguese machine translation task.

The rest of the paper is organized as follows. In Section 2 we present some related work on WSD. The specification of ILP implementations that construct ILP models

[4] see: http://www.senseval.org

and features for use in ILP-assisted models is in Section 3. The experimental evaluation comprising our investigation is described in Section 4. This includes materials (Section 4.1) and methods (Section 4.2). Results are presented in Section 5. Section 6 concludes the paper.

2 Models for Word Sense Disambiguation

The earliest computer-executable models for WSD are manually constructed, capturing specific aspects of human disambiguation expertise in symbolic structures like semantic networks [25] and semantic frames [5, 6, 14]. Early reports also exist of sub-symbolic neural networks [4]. Most of these techniques appear to have suffered from the important difficulty in manual acquisition of expert knowledge identified by Feigenbaum (and somewhat anticipated, in the WSD context [2]), resulting in their application being limited to very small subsets of the languages.

The development of machine readable resources like lexical databases, dictionaries and thesauri has provided a turning point in automatic processing of natural language, enabling the development of techniques that used linguistic and extra-linguistic information extracted automatically from these resources [12, 27, 1, 35]. While the resources provided ready access to large bodies of knowledge, the actual disambiguation models continued to be manually codified. This changed with the use of statistical and machine-learning techniques for constructing models. The characteristic of these methods is the use of a corpus of examples of disambiguation to construct automatically models for disambiguation. The most common of these "corpus-based" techniques employ statistical methods that construct models based on features representing frequencies estimated from a corpus. For example, these may be the frequencies of some words on either side of the ambiguous word [37, 18, 28, 23]. While techniques using such "shallow" features that refer to the local context of the ambiguous word have yielded the best models, the accuracies obtained are low, and significant improvements do not appear to be forthcoming.

More sophisticated corpus-based approaches such as [34] try to incorporate deeper knowledge using machine readable resources. These are special-purpose methods aimed at specific tasks and it is not clear how they could be scaled-up for use across a wide range of WSD tasks. ILP provides a general-purpose approach that can be tailored to a variety of NLP tasks by the incorporation of appropriate background knowledge. To date, [30] appears to be the only work dealing with the use of ILP for WSD. The work here extends this substantially in terms of experimental results; and in exploring alternate ways of using ILP for WSD.

3 Inductive Logic Programming

Functionally, Inductive Logic Programming (ILP) can bee largely characterised by two classes of programs. The first, predictive ILP, has been concerned with constructing models (sets of rules; or first-order variants of classification or regression trees) for discriminating accurately amongst two sets of examples ("positive" and "negative"). The partial specifications provided by [19] have formed the basis for deriving programs in this class. We refer the reader to [21] for definitions of the logical terms used below:

- B is background knowledge consisting of a finite set of clauses $= \{C_1, C_2, \ldots\}$

- E is a finite set of examples $= E^+ \cup E^-$ where:
  - *Positive Examples.* $E^+ = \{e_1, e_2, \ldots\}$ is a non-empty set of definite clauses;
  - *Negative Examples.* $E^- = \{\overline{f_1}, \overline{f_2} \ldots\}$ is a set of Horn clauses (this may be empty)
- H, the output of the algorithm given B and E is acceptable if the following conditions are met:
  - *Prior Satisfiability.* $B \cup E^- \not\models \square$
  - *Posterior Satisfiability.* $B \cup H \cup E^- \not\models \square$;
  - *Prior Necessity.* $B \not\models E^+$
  - *Posterior Sufficiency.* $B \cup H \models e_1 \wedge e_2 \wedge \ldots$

The second category of ILP programs, descriptive ILP, has been concerned with identifying relationships that hold amongst the background knowledge and examples, without a view of discrimination. The partial specifications for programs in this class are based on the description in [20]:

- B is background knowledge consisting of a finite set of clauses $= \{C_1, C_2, \ldots\}$
- E is a finite set of examples (this may be empty)
- H, the output of the algorithm given B and E is acceptable if the following condition is met:
  - *Posterior Sufficiency.* $B \cup H \cup E \not\models \square$

The idea of using a feature-based model constructor that uses first-order features can be traced back at least to the LINUS program [11]. More recently, the task of identifying good features using a first-order logic representation has been the province of programs developed under the umbrella of "propositionalization" (see [9] for a review). Programs in this class are not easily characterised as either predictive or descriptive ILP and we have not found explicit specifications for them within the ILP literature. Conceptually, solutions involve two steps: (1) a feature-construction step that identifies (within computational reason) all the features that are consistent with the constraints provided by the background knowledge. This is characteristic of a descriptive ILP program; and (2) a feature-selection step that retains some of the features based on their utility in classifying the examples. This is characteristic of a predictive ILP program. To this extent, we present partial specifications for feature construction that reflect a combination of the two dominant categories of ILP programs:

- B is background knowledge consisting of a finite set of clauses $= \{C_1, C_2, \ldots\}$
- E is a finite set of examples $= E^+ \cup E^-$ where:
  - *Positive Examples.* $E^+ = \{e_1, e_2, \ldots\}$ is a non-empty set of definite clauses;
  - *Negative Examples.* $E^- = \{\overline{f_1}, \overline{f_2} \ldots\}$ is a set of Horn clauses (this may be empty)
- $\mathcal{H}$ is the set of definite clauses, constructible with predicates, functions and constants in $B \cup E$; $\mathcal{F}$ the set of features constructible using a set of individuals and B; and $\tau : \mathcal{H} \mapsto \mathcal{F}$ a function that maps a definite clause $h \in \mathcal{H}$ to a feature $f \in \mathcal{F}$.
- $F = \{f_1, f_2, \ldots\} \subseteq \mathcal{F}$, the output of the algorithm given B and E is acceptable for any set $H = \{h_1, h_2, \ldots\} \subseteq \mathcal{H}$ if the following conditions are met:
  - *Posterior Sufficiency.* $B \cup \{h_i\} \models e_1 \vee e_2 \vee \ldots$, where $\{e_1, e_2, \ldots\} \subseteq E^+$

- $f_i = \tau(h_i)$

The reader would have noted the principal differences in the three Posterior Sufficiency constraints. For feature construction—at least for the purposes of this paper—clauses identified are required to entail at least one positive example given B. Obviously more would be better, but the specification here is a minimal one. This is not the case for descriptive ILP, and clearly insufficient for the predictive case.

We still need to clarify the meanings of $\mathcal{F}$, $\mathcal{H}$ and τ. For the purposes of this paper, we will assume that the boolean values $FALSE$ and $TRUE$ are represented by 0 and 1; the features f_i are functions of the form $f_i : \mathcal{X} \mapsto \{0, 1\}$; and examples E are some subset of the binary relation $\mathcal{X} \times \mathcal{Y}$, where $\mathcal{X}$ denotes the set of individuals and $\mathcal{Y}$ some finite set of classes. Positive and negative examples are represented by the predicate $class : \mathcal{X} \times \mathcal{Y} \mapsto \{0, 1\}$ and we will take each $h_i \in \mathcal{H}$ to be a definite clause $class(X, y_k) \leftarrow cp_i(X)$, where X is a variable and y_k is some class in $\mathcal{Y}$. Here, adopting terminology from [26], $cp_i : \mathcal{X} \mapsto \{0, 1\}$ is a "context predicate" and corresponds to a conjunction of literals that evaluates to true or false for any particular individual x. With these preliminaries in place, given $h_i : class(X, y_k) \leftarrow cp_i(X)$, $f_i(x) = \tau(h_i) = 1$ $\mathrm{i}ff$ $cp_i(x) = 1$ (and 0 otherwise)[5].

Given a set of examples represented by individuals and their classes, a program for feature construction that minimally satisfies this specification would proceed as follows. First, a set of clauses H is identified for the individuals. Each clause in this set entails at least one positive example, given the background knowledge B. Next, each clause h_i in H is converted into a boolean feature f_i that takes the value 1 (or 0) for any individual for which the body of the clause is true (if the body is false). Thus, the set of clauses H gives rise to a boolean vector for each individual in the set of examples. Examples in the WSD context are shown in Figure 1.

Clause:

$$h_1 : \quad class(X, voltar) : -has\_expression(X, \text{'come back '}, voltar), \; has\_pos(X, pcwr\_4, nn)$$

Feature:

$$f_1(X) = \begin{cases} 1 & has\_expression(X, \text{'come back '}, voltar) \wedge has\_pos(X, pcwr\_4, n) \; = \; 1 \\ 0 & \text{otherwise} \end{cases}$$

Fig. 1. Example of a boolean feature constructed from a clause for WSD. The clause shown here in Prolog syntax identifies the Portuguese sense of the English verb 'to come'. The meanings of the predicate symbols *has_expression* and *has_pos* are explained in Section 4.

[5] In [26] the f_i are denoted by $f_{i,Y}$. The notation here is more in line with the machine learning literature.

4 Empirical Evaluation

Our objectives are to evaluate empirically the use of ILP in constructing models for WSD. Specifically, we intend to investigate the performance of two kinds of models:

1. *ILP models*. These are models constructed by an ILP system for predicting the correct sense of a word. The models are to be constructed by an implementation conforming to the specification for predictive ILP systems in Section 3.
2. *ILP-assisted models*. These are models for predicting the correct sense of a word that, in addition to existing shallow features, use features constructed by an ILP system. The models are to be constructed by an implementation conforming to the specification for feature construction in Section 3.

4.1 Materials

Data

Monolingual task. Data consist of the 32 verbs from the SENSEVAL-3 competition. SENSEVAL[6] is a joint evaluation effort for WSD and related tasks. We use all the verbs of the English lexical sample task from the third and last edition of the competition: *activate*, *add*, *appear*, *ask*, *begin*, *climb*, *decide*, *eat*, *encounter*, *expect*, *express*, *hear*, *lose*, *mean*, *miss*, *note*, *operate*, *play*, *produce*, *provide*, *receive*, *remain*, *rule*, *smell*, *suspend*, *talk*, *treat*, *use*, *wash*, *watch*, *win*, and *write*. The number of examples for each verb varies from 40 to 398 (average of 186). The number of senses varies from 3 to 12 with an average of 7 senses. The average accuracy of the majority class is about 55%. We refer the reader to [15] for more information about the SENSEVAL-3 data.

Bilingual task. Data consist of 7 highly frequent and ambiguous verbs: *come*, *get*, *give*, *go*, *look*, *make*, and *take*. The sample corpus comprises around 200 English sentences for each verb extracted from a corpus of fiction books, with the verb translation automatically annotated by a system previously developed [31]. In that corpus, the number of translations varies from 5 to 17, with an average of 11 translations. The average accuracy of the majority class is about 54%.

Background Knowledge To achieve accurate disambiguation in both tasks is believed to require a variety of syntactic and semantic information. In what follows, we describe the background knowledge available for the tasks and illustrate it using the following sentence (assuming that we are attempting to determine the sense of 'coming'):

> "If there is such a thing as reincarnation, I would not mind coming back as a squirrel".

B0. *Shallow features*. Features corresponding to the predicates in B1-B5, below, conveying the same information, but represented by means of attribute-value vectors.

B1. *Bag-of-words*. The 5 words to the right and left of the verb, extracted from the corpus and represented using definitions of the form $has\_bag(sentence, word)$. For example:

[6] http://www.senseval.org

$$has\_bag(snt1, mind).$$
$$has\_bag(snt1, not). \ldots$$

B2. *Narrow context.* Lemmas of 5 content words to the right and left of the verb, extracted from the corpus, previously lemmatized by MINIPAR [13]. These are represented using definitions of the form $has\_narrow(sentence, wordposition, word)$. For example:

$$has\_narrow(snt1, first\_content\_word\_left, mind).$$
$$has\_narrow(snt1, first\_content\_word\_right, back). \ldots$$

B3. *Part-of-speech tags.* Part-of-speech (POS) tags of 5 content words to the right and left of the verb, are obtained using MXPOST [26] and represented using definitions of the form: $has_pos(sentence, wordposition, pos)$. For example:

$$has\_pos(snt1, first\_content\_word\_left, nn).$$
$$has\_pos(snt1, first\_content\_word\_right, rb). \ldots$$

B4. *Subject-Object relations.* Subject and object syntactic relations with respect to the verb. These were obtained from parsing sentences using MINIPAR and represented using definitions of the form $has\_rel(sentence, type, word)$. For example:

$$has\_rel(snt1, subject, i).$$
$$has\_rel(snt1, object, nil). \ldots$$

B5. *Word collocations.* 11 collocations with respect to the verb, extracted from the corpus: 1st preposition to the right, 1st and 2nd words to the left and right, 1st noun, 1st adjective, and 1st verb to the left and right. These are represented using definitions of the form $has\_collocation(sentence, collocation\_type, collocation)$. For example:

$$has\_collocation(snt1, first\_word\_right, back).$$
$$has\_collocation(snt1, first\_word\_left, mind). \ldots$$

B6. *Verb restrictions.* Selectional restrictions of the verbs, defined in terms of the semantic features of their arguments in the sentence, extracted using LDOCE [24]. WordNet relations are used when the restrictions imposed by the verb are not part of the description of its arguments, but can be satisfied by synonyms or hyperonyms of those arguments. A hierarchy of feature types is used to account for restrictions established by the verb that are more generic than the features describing its arguments in the sentence. These are represented by definitions of the form $satisfy\_restrictions(sentence, rest\_subject, rest\_object)$. For example:

$$satisfy\_restrictions(snt1, [human], nil).$$
$$satisfy\_restrictions(snt1, [animal, human], nil).$$

B7. *Dictionary definitions.* A relative count of the overlapping words in dictionary definitions of each of the possible translations of the verb (from [22]) and the words surrounding it in the sentence. These are represented by facts of the form $has\_highest\_overlap(sentence, translation)$. For example:

$$has\_highest\_overlap(snt1, voltar).$$

B8. *Phrasal verbs.* Phrasal verbs possibly occurring in a sentence, according to the list of phrasal verbs given by dictionaries and the context of the verb (5 surrounding words). These are represented by definitions of the form *has_expression*(*sentence*, *verbal_expression*). For example:

$$has\_expression(snt1, \text{'come back'}).$$

Of these definitions, B0 is intended for use by a feature-based model constructor. B1–B8 are intended for use by an ILP system. The ILP implementation we use is capable of exploring intensional definitions of each of B1–B8. However, it is more efficient to represent the definitions in an extensional form (that is, as a set of ground facts). For the tasks here, the background knowledge B1–B8 amount to about 204,000 ground facts for the monolingual task and 24,000 for the bilingual task. For comparison, the monolingual task has about 10 times more background facts than the carcinogenesis benchmark described in [33]; and about 20 times more facts than the mutagenesis benchmark [7]. The bilingual task is comparable to these two benchmarks.

Algorithms We use implementations within the ILP system Aleph [32] to construct disambiguation models and to construct features. Feature-based model construction is performed by a linear SVM (the specific implementation used is the one provided in the WEKA toolbox called SMO[7]). For convenience, we will call the Aleph implementation the "ILP learner" and the SVM implementation the "feature-based learner."

4.2 Method

We adopt the following method:

For each verb in each task (that is, 32 verbs in the monolingual task and 7 verbs in the bilingual task):

1. Obtain the best possible model using the feature-based learner and the features in B0. Call this the "baseline model"[8].
2. Obtain the best possible model using the ILP learner, equipped with background knowledge definitions B1–B8. Call this the "ILP model".
3. Construct at most k features using the ILP learner, equipped with background knowledge definitions B1–B8. Call these features "B9".
4. Obtain the best model possible using the feature-based learner with features in B0 and B9. Call this the "ILP-assisted model".
5. Compare the performance of the baseline model against that of the ILP model and the ILP-assisted model.

The following details are relevant:

(a) The SENSEVAL-3 benchmark specifies 34% of the data that are to be used to estimate the performance of disambiguation models. For uniformity, we randomly

[7] http://www.cs.waikato.ac.nz/~ml/weka/

[8] The term "baseline" is not used in a pejorative sense: models constructed with shallow features of the form in B0 in fact represent the state-of-the-art, and any other techniques would have to perform at least as well as these.

use 34% of the bilingual data for performance evaluation (the test set). The remaining 66% in each task is available for model construction (the training set). Performance will be measured by the accuracy of prediction on the test set (that is, the percentage of test examples whose sense is predicted correctly).

(b) The ILP learner constructs a set of clauses in line with the specifications for predictive ILP as described in Section 3. Positive examples for the ILP learner are provided by the correct sense (or translation in the bilingual case) of the verb in a sentence. Negative examples are generated automatically using all other senses (or translations). The specifications do not, however, describe how the clauses constructed are to be used to predict the sense or translation of verbs in the test data. We use the following method. Clauses are evaluated in order of their identification by the ILP learner and the class of an example is determined by the first clause for which literals in the body are satisfied by the example. If no such clause exists, then the example is assigned the majority class, as computed on the training data. In effect, this treats the clauses as a decision list, with the addition to the end of a default rule assigning the majority class.

(c) For each verb and task, constructing the "best possible model" requires determining optimal values for some parameters of the feature-based or ILP learner. We estimate these values using an instance of the method proposed in [8] that proceeds as follows. First, we decide on the relevant parameters. Second, we obtain, using the training set only, unbiased estimates of the predictive accuracy of the models for each verb arising from systematic variation across some small number of values for these parameters. Values that yielded the best average predictive accuracy across all verbs are taken to be optimal ones. This procedure is not perfect: correctly, optimal values may change from one verb to another; and even if they did not, the results obtained may be a local maximum (that is, better models may result from further informed variation of values).

(d) The principal parameter for the feature-based learners concerns the extent of feature-selection to be performed. Values experimented with were: selecting $50, 100, 150, 200, 250, 500$ or all features (we use the subset feature selection method within WEKA). For the monolingual task the best average accuracy for baseline models was obtained with 150 features; and with 250 features for the ILP-assisted case. For the bilingual task, the best average accuracy for baseline models used all features. The ILP-assisted models case required 500 features. For the the ILP-learner, the principal parameters selected were: the choice between a greedy and non-greedy rule construction strategy (as implemented by the `induce` and `induce_max` procedures within Aleph); the maximal length of clauses; and the minimum accuracy of clauses. Maximal clause lengths examined were 4 and 8. Minimum clause accuracies examined were 1.0 and 0.8. For the monolingual task, the best average accuracies were obtained with the non-greedy strategy, in conjunction with a maximal clause length of 8 literals and minimal clause accuracy of 1.0. The bilingual task also required the non-greedy strategy and a maximal clause length of 8 literals, but a minimal clause accuracy of 0.8.

(e) In all cases, the value of k (the maximum number of features constructed) is 5000.

(f) Comparison of performance is done using the Wilcoxon signed-rank test [29]. This is a non-parametric test of the null hypothesis that there is no significant difference between the median performance of a pair of algorithms. The test works by ranking the absolute value of the differences observed in performance of the

pair of algorithms. Ties are discarded and the ranks are then given signs depending on whether the performance of the first algorithm is higher or lower than that of the second. If the null hypothesis holds, the sum of the signed ranks should be approximately 0. The probabilities of observing the actual signed rank sum can be obtained by an exact calculation (if the number of entries is less than 10), or by using a normal approximation.

5 Results and Discussion

Figures 2 and 3 tabulate the performance of baseline, ILP, and ILP-assisted models—these two collectively termed ILP-based models—on the two disambiguation tasks. It is also standard practice to include the performance of a classifier that simply predicts the most frequent sense of the verb. The principal details in these tabulations are these: (1) The "majority class" classifier clearly performs poorest; (2) For both tasks, the accuracies of the baseline models are usually lower than the ILP-based models. Discarding ties, the baseline model has the highest accuracy only for 5 of the 32 verbs in the monolingual task and for 0 of the 7 verbs in the bilingual task; (3) ILP models and ILP-assisted models appear to be comparable in their performance in the monolingual task, while ILP models are uniformly better than ILP-assisted models for the bilingual task.

We turn now to the question of whether the differences observed between the models are in fact significant. The probabilities calculated by using the Wilcoxon test are shown in Fig. 4. The tabulations suggest that one or the other of the ILP-based models perform substantially better than the baseline or majority class models. However, they also suggest that a simple choice between ILP and ILP-assisted models is not evident: ILP-assisted models appear to be the best choice for the monolingual task and it is evident that ILP models are uniformly best for the bilingual task.[9]

It is curious that the two ILP-based approaches are comparable on the monolingual task and are completely incommensurate on the bilingual task. Closer study of the performance of the ILP model reveals the substantial role of the default rule predicting the majority class (as described in Section 4.2). Removal of this rule lowers the ILP column's median accuracy by about 11% for the monolingual task and 8% for the bilingual task (the two ILP-based methods are then comparable on the bilingual task). Since it is not evident that the use of the default rule will always yield such beneficial results to the ILP model, and ILP-assisted models do not require such a rule, the ILP-assisted approach probably represent a more reliable route for constructing WSD models.

For the monolingual task, we are also able to compare the performance of ILP-based models to those of models produced by the best supervised techniques for the same data. SENSEVAL's evaluation software provides estimates on the performance

[9] We note here that repeated cross-comparisons of this form will yield occasions on which one or the other model will seem better. For repeated comparisons of a given pair of algorithms on different random samples of data, it is possible to apply a correction (known as the Bonferroni adjustment) for this problem. The situation of repeated comparisons of different pairs of algorithms on a given set of data (as is here) does not, on the surface, appear to be amenable to the same correction. However, the spirit of the correction and the small number of verbs in the bilingual case suggests caution in interpreting the probabilities tabulated.

Verb	**Senses**	**Accuracy**			
		Majority class	Baseline	ILP	ILP-assisted
activate	5	82.46±3.56	**85.09**±3.34	52.63±4.68	83.33±3.49
add	6	45.80±4.35	**82.44**±3.32	73.28±3.87	**82.44**±3.32
appear	3	44.70±4.33	68.18±4.05	**87.88**±2.84	71.21±3.94
ask	6	27.78±3.99	**53.17**±4.45	40.48±4.37	50.00±4.45
begin	4	59.74±5.59	57.14±5.64	55.84±5.66	**74.03**±5.00
climb	5	55.22±6.08	71.64±5.51	59.70±5.99	**83.58**±4.53
decide	4	67.74±5.94	**77.42**±5.31	**77.42**±5.31	**77.42**±5.31
eat	7	**88.37**±3.46	**88.37**±3.46	83.72±3.98	87.21±3.60
encounter	4	50.77±6.20	**73.85**±5.45	67.69±5.80	72.31±5.55
expect	3	74.36±4.94	75.64±4.86	79.49±4.57	**92.31**±3.02
express	4	69.09±6.23	67.27±6.33	70.91±6.12	**72.73**±6.01
hear	7	46.88±8.82	53.13±8.82	65.62±8.40	**65.63**±8.40
lose	9	52.78±8.32	**58.33**±8.22	55.56±8.28	**58.33**±8.22
mean	7	52.50±7.90	**77.50**±6.60	55.00±7.87	70.00±7.25
miss	8	33.33±8.61	36.67±8.80	**56.67**±9.05	33.33±8.61
note	3	38.81±5.95	58.21±6.03	82.09±4.68	**88.06**±3.96
operate	5	16.67±8.78	72.22±10.56	**83.33**±8.78	77.78±9.80
play	12	46.15±6.91	**53.85**±6.91	46.15±6.91	**53.85**±6.91
produce	6	52.13±5.15	63.83±4.96	**75.53**±4.43	67.02±4.85
provide	6	85.51±4.24	**89.86**±3.63	88.41±3.85	**89.86**±3.63
receive	9	88.89±6.05	88.89±6.05	**92.59**±5.04	88.89±6.05
remain	3	78.57±4.90	84.29±4.35	80.00±4.78	**87.14**±4.00
rule	5	50.00±9.13	66.67±8.61	**86.67**±6.21	83.33±6.80
smell	7	40.74±6.69	**79.63**±5.48	68.52±6.32	77.78±5.66
suspend	7	35.94±6.00	**60.94**±6.10	**60.94**±6.10	57.81±6.17
talk	9	72.60±5.22	**73.97**±5.14	**73.97**±5.14	**73.97**±5.14
treat	9	28.07±5.95	40.35±6.50	**57.89**±6.54	47.37±6.61
use	5	71.43±12.07	85.71±9.35	**92.86**±6.88	**92.86**±6.88
wash	12	67.65±8.02	70.59±7.81	61.76±8.33	**73.53**±7.57
watch	7	74.51±6.10	74.51±6.10	**76.47**±5.94	74.51±6.10
win	7	44.74±8.07	52.63±8.10	47.37±8.10	**60.53**±7.93
write	8	26.09±9.16	52.17±10.42	**56.52**±10.34	34.78±9.93
Mean	7	55.31	68.56	69.15	71.97
Median	6	52.31	71.11	69.71	74.03

Fig. 2. Estimates of accuracies of disambiguation models on the monolingual task. "Senses" refers to the numbers of possible senses of each verb. The column labeled "Majority class" gives the accuracy of models that simply predict the most common sense of each verb. The entries in boldface represent the highest accuracy obtained for a verb.

Verb	**Translations**	**Accuracy**			
		Majority class	Baseline	ILP	ILP-assisted
come	11	50.30±7.62	67.44±7.15	**86.67**±5.07	76.74±6.44
get	17	21.00±6.70	32.43±7.70	**51.28**±8.00	40.54±8.07
give	5	88.80±4.81	97.67±2.30	**97.78**±2.20	95.35±3.21
go	11	68.50±6.78	72.34±6.52	**85.71**±5.00	78.72±5.97
look	7	50.30±7.45	77.78±6.20	**82.98**±5.48	82.22±5.70
make	11	70.00±7.25	75.00±6.85	**76.19**±6.57	75.00±6.85
take	13	28.50±8.24	46.67±9.11	**62.50**±8.56	60.00±8.94
Mean	11	53.91	67.05	77.59	72.65
Median	11	50.30	72.34	82.98	76.74

Fig. 3. Estimates of accuracies of disambiguation models on the bilingual task. "Translations" refers to the numbers of possible translations of each verb into Portuguese.

	Majority class	Baseline	ILP
Baseline	< 0.001, 0.020	–	–
ILP	< 0.001, 0.020	0.849, 0.020	–
ILP-assisted	< 0.001, 0.020	0.037, 0.075	0.134, 0.020

Fig. 4. Probablities of observing the differences in accuracies for the monolingual and bilingual tasks, under the assumption that median accuracies of the pair of algorithms being compared are equal. Each entry consists of a pair of probability estimates, corresponding to the mono and bilingual tasks.

of the systems according to two different levels of sense distinction: fine and coarse-grained. The former comprises average accuracies in the normally understood sense. Comparative results are shown in Fig. 5. Syntalex-1 to Syntalex-4 approaches are presented in [17]. Syntalex-1 uses bagged decision trees with narrow context part-of-speech features. Syntalex-2 uses bagged decision trees, but with broad context part-of-speech features. Syntalex-3 uses an ensemble of bagged decision trees with narrow context part-of-speech features and bigrams. Syntalex-4 uses the same features as Syntalex-3, but with unified decision trees. CLaC1 and CLaC2 are presented in [10]. CLaC1 uses a Naive Bayes algorithm with a dynamically adjusted context window around the target word. CLaC2 uses a Maximum Entropy learner instead, and also syntactic features and the hyperonyms of the neighbor words. Finally, MC-WSD [3] is a multi-class averaged perceptron classifier with one component trained on the data provided by SENSEVAL and other trained on WordNet glosses. Syntactic and narrow context features are explored. As we can see, among all the approaches, our ILP models are outperformed only by MC-WSD for fine-grained distinctions and therefore it is evident that the ILP-based models are comparable to the state-of-the-art in the field. In practice, we would expect that all these methods would be able to use features constructed by an ILP system. Improvements in their performance similar to those seen from the baseline classifier may then follow.

Models	Accuracy
MC-WSD	72.50
ILP-assisted	71.97
ILP	69.15
Syntalex-3	67.60
Syntalex-1	67.00
CLaC1	67.00
Syntalex-2	66.50
CLaC2	66.00
Syntalex-4	65.30

Fig. 5. Comparative average fine-grained accuracies of the best models reported for the SENSEVAL-3 competition.

6 Concluding Remarks

Word sense disambiguation, a necessary component for a variety of natural language processing tasks, remains amongst the hardest to model adequately. It is of course possible that the vagaries of natural language may place a limit on the accuracy with which a model could identify correctly the sense of an ambiguous word, but it is not clear that this limit has been reached with the modelling techniques that constitute the current state-of-the-art. The performance of these techniques depends largely on the adequacy of the features used to represent the problem. As it stands, these features are usually hand-crafted and largely of a syntactic nature. For substantial, scalable progress it is believed that knowledge that accounts for more elaborate semantic information needs to be incorporated: however, no adequate general-purpose techniques have been forthcoming. In this paper, we have investigated the use of Inductive Logic Programming as a mechanism for incorporating multiple sources of syntactic and semantic information into the construction of models for WSD. The investigation has been in the form of empirical studies of using ILP to construct models for monolingual and bilingual WSD tasks and the results suggest that the use of ILP can improve predictive accuracies. These studies represent the first extensive application of ILP to the task of constructing WSD models.

We believe much of the gains observed with ILP stems from the use of substantial amounts of background knowledge. For the work here, this knowledge has been obtained by translations of information in standard corpora or electronic lexical resources. This is promising, as it suggests that these translators, in conjunction with ILP, may provide a set of tools for the automatic incorporation of deep knowledge into the construction of general WSD models. Turning specifically to the tasks addressed here, further improvements could be achieved with the inclusion of other kinds of background knowledge. For example, for the bilingual task, the "translation context" for a verb may help greatly. This refers to the translations into the target language of the words forming the context of the verb.

The use of other ILP implementations may also provide improvements in predictive accuracies, thus strengthening the case for the use of ILP further. On the basis of results here there is little to chose between ILP-models and ILP-assisted models, although we believe that the latter may provide a more reliable approach for constructing WSD models. There does not appear to be any inherent limitation in using

a feature-based representation for verb disambiguation: a finding that may extend to other WSD tasks. The key is to get a good set of features, and results here suggest that ILP could provide a reliable method of identifying these.

References

1. Agirre, E. and Rigau, G. Word Sense Disambiguation Using Conceptual Density. 16th International Conference on Computational Linguistics, Copenhagen (1996).
2. Bar-Hillel, Y. Automatic Translation of Languages. In F. Alt, D. Booth, and R. E. Meagher (eds), Advances in Computers. Academic Press, New York (1960).
3. Ciaramita, M. and Johnson, M. Multi-component Word Sense Disambiguation. SENSEVAL-3: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona (2004) 97–100.
4. Cottrell, G. W. A Connectionist Approach to Word Sense Disambiguation. Research Notes in Artificial Intelligence. Morgan Kaufmann, San Mateo (1989).
5. Hayes, P.J. A Process to Implement Some Word Sense Disambiguation. Working paper 23, Institut pour les Etudes Semantiques et Cognitives, Universite de Geneve, Geneve (1976).
6. Hirst, G. Semantic Intepretation and the Resolution of Ambiguity. Studies in Natural Language Processing. Cambridge Universisty Press, Cambridge (1987).
7. King, R., Muggleton, S., Srinivasan, A., and Sternberg, M. Structure-activity relationships derived by machine learning: The use of atoms and their bond connectivities to predict mutagenicity by inductive logic programming. National Academy of Sciences, 93 (1996) 438–442.
8. Kohavi, R., and John, G.H. Automatic Parameter Selection by Minimizing Estimated Error. 12th International Conference on Machine Learning, Morgan Kaufmann, San Francisco, CA (1995).
9. Kramer, S., Lavrac, N., and Flach, P. Propositionalization Approaches to Relational Data Mining. Relational Data Mining, S. Dzeroski and N. Lavrac (eds), Springer (2001) 262–291.
10. Lamjiri, A., Demerdash, O., Kosseim, F. Simple features for statistical Word Sense Disambiguation. SENSEVAL-3: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona (2004) 133–136.
11. Lavrac, N., Dzeroski, S., and Grobelnik, M. Learning nonrecursive definitions of relations with LINUS. Technical report, Jozef Stefan Institute (1990).
12. Lesk, M. Automated Sense Disambiguation Using Machine-readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. SIGDOC Conference, Toronto (1986) 24–26.
13. Lin, D. Principle based parsing without overgeneration. 31st Annual Meeting of the Association for Computational Linguistics, Columbus (1993) 112–120.
14. McRoy, S. Using Multiple Knowledge Sources for Word Sense Discrimination. Computational Linguistics, 18(1) (1992) 1–30.
15. Mihalcea, R., Chklovski, T., Kilgariff, A. The SENSEVAL-3 English Lexical Sample Task. SENSEVAL-3: 3rd International Workshop on the Evaluation of Systems for Semantic Analysis of Text, Barcelona (2004) 25–28.
16. Miller, G.A., Beckwith, R.T., Fellbaum, C.D., Gross, D., Miller, K. Wordnet: An On-line Lexical Database. International Journal of Lexicography, 3(4) (1990) 235–244.
17. Mohammad, S. and Pedersen, T. Complementarity of Lexical and Simple Syntactic Features: The SyntaLex Approach to SENSEVAL-3. SENSEVAL-3: 3rd International Workshop on the Evaluation of Systems for the Semantic Analysis of Text, Barcelona (2004) 159–162.
18. Mooney, R.J. Inductive Logic Programming for Natural Language Processing. 6th International Workshop on Inductive Logic Programming, Stockolm, LNAI 1314, Springer-Verlag (1997) 3–24.

19. Muggleton, S. Inductive Logic Programming: derivations, successes and shortcomings. SIGART Bulletin 5(1) (1994) 5–11.
20. Muggleton, S. and Raedt, L. D. Inductive logic programming: Theory and methods. Journal of Logic Programming 19,20 (1994) 629–679.
21. Nienhuys-Cheng, S. and de Wolf, R. Foundations of Inductive Logic Programming. Springer, Berlin (1997).
22. Parker, J.; Stahel, M. Password: English Dictionary for Speakers of Portuguese. Martins Fontes, São Paulo (1998).
23. Pedersen, T. A Baseline Methodology for Word Sense Disambiguation. 3rd International Conference on Intelligent Text Processing and Computational Linguistics, Mexico City (2002).
24. Procter, P. (editor). Longman Dictionary of Contemporary English. Longman Group, Essex (1978).
25. Quillian, M.R. A Design for an Understanding Machine. Colloquium of semantic problems in natural language. Cambridge University, Cambridge (1961).
26. Ratnaparkhi, A. A Maximum Entropy Part-Of-Speech Tagger. Empirical Methods in NLP Conference, University of Pennsylvania (1996).
27. Resnik, Philip. Disambiguating Noun Groupings with Respect to WordNet Senses. 3rd Workshop on Very Large Corpora, Cambridge (1995) 54–68.
28. Schutze, H. Automatic Word Sense Discrimination. Computational Linguistics, 24(1) (1998) 97–124.
29. Siegel, S. Nonparametric Statistics for the Behavioural Sciences. McGraw-Hill, New York (1956).
30. Specia, L.: A Hybrid Relational Approach for WSD - First Results. Student Research Workshop at Coling-ACL, Sydney (2006) 55–60.
31. Specia, L, Nunes, M.G.V., and Stevenson, M. Exploiting Parallel Texts to Produce a Multilingual Sense-tagged Corpus for Word Sense Disambiguation. RANLP-05, Borovets (2005) 525–531.
32. Srinivasan, A. The Aleph Manual. Available at http://www.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/ (1999).
33. Srinivasan, A. and King, R. Carcinogenesis predictions using ILP. 7th International Workshop on Inductive Logic Programming, Prague, LNAI 1297, Springer-Verlag, (1997) 273–287.
34. Stevenson, M. and Wilks, Y. The Interaction of Knowledge Sources for Word Sense Disambiguation. Computational Linguistics, 27(3) (2001) 321–349.
35. Wilks, Y. and Stevenson, M. Combining Independent Knowledge Sources for Word Sense Disambiguation. 3rd Conference on Recent Advances in Natural Language Processing, Tzigov Chark, (1997) 1-7.
36. Wilks, Y. and Stevenson, M. The Grammar of Sense: Using Part-of-speech Tags as a First Step in Semantic Disambiguation. Natural Language Engineering, 4(1) (1998) 1–9.
37. Yarowsky, D. Unsupervised Word Sense Disambiguation Rivaling Supervised Methods. 33rd Annual Meeting of the Association for Computational Linguistics, Cambridge (1995) 189–196.