A Hybrid Relational Approach for WSD – First Results

Lucia Specia
Núcleo Interinstitucional de Lingüística Computational – ICMC – University of São Paulo
Caixa Postal 668, 13560-970, São Carlos, SP, Brazil
lspecia@icmc.usp.br

Abstract

We present a novel hybrid approach for Word Sense Disambiguation (WSD) which makes use of a relational formalism to represent instances and background knowledge. It is built using Inductive Logic Programming techniques to combine evidence coming from both sources during the learning process, producing a rule-based WSD model. We experimented with this approach to disambiguate 7 highly ambiguous verbs in English-Portuguese translation. Results showed that the approach is promising, achieving an average accuracy of 75%, which outperforms the other machine learning techniques investigated (66%).

1 Introduction

Word Sense Disambiguation (WSD) is concerned with the identification of the correct sense of an ambiguous word given its context. Although it can be thought of as an independent task, its importance is more easily realized when it is applied to particular tasks, such as Information Retrieval or Machine Translation (MT). In MT, the application we are focusing on, a WSD (or *translation disambiguation*) module should identify the correct translation for a source word when options with different meanings are available.

As shown by Vickrey et al. (2005), we believe that a WSD module can significantly improve the performance of MT systems, provided that such module is developed following specific requirements of MT, e.g., employing multilingual sense repositories. Differences between monolingual and multilingual WSD are very significant for MT, since it is concerned only with the ambiguities that appear in the translation (Hutchins and Sommers, 1992).

In this paper we present a novel approach for WSD, designed focusing on MT. It follows a hybrid strategy, i.e., knowledge and corpus-based, and employs a highly expressive relational formalism to represent both the examples and background knowledge. This approach allows the exploitation of several knowledge sources, together with evidences provided by examples of disambiguation, both automatically extracted from lexical resources and sense tagged corpora. This is achieved using Inductive Logic Programming (Muggleton, 1991), which has not been exploited for WSD so far. In this paper we investigate the disambiguation of 7 highly ambiguous verbs in English-Portuguese MT, using knowledge from 7 syntactic, semantic and pragmatic sources.

In what follows, we first present some related approaches on WSD for MT, focusing oh their limitations (Section 2). We then give some basic concepts on Inductive Logic Programming and describe our approach (Section 3). Finally, we present our initial experiments and the results achieved (Section 4).

2 Related work

Many approaches have been proposed for WSD, but only a few are designed for specific applications, such as MT. Existing multilingual approaches can be classified as (a) knowledge-based approaches, which make use of linguistic knowledge manually codified or extracted from lexical resources (Pedersen, 1997; Dorr and Katsova, 1998); (b) corpus-based approaches, which make use of knowledge automatically acquired from text using machine learning algorithms (Lee, 2002; Vickrey et al., 2005); and (c) hybrid approaches, which employ techniques from the two other approaches (Zinovjeva, 2000).

Proceedings of the COLING/ACL 2006 Student Research Workshop, pages 55–60,
Sydney, July 2006. ©2006 Association for Computational Linguistics

Hybrid approaches potentially explore the advantages of both other strategies, yielding accurate and comprehensive systems. However, they are quite rare, even in monolingual contexts (Stevenson and Wilks, 2001, e.g.), and they are not able to integrate and use knowledge coming from corpus and other resources during the learning process.

In fact, current hybrid approaches usually employ knowledge sources in pre-processing steps, and then use machine learning algorithms to combine disambiguation evidence from those sources. This strategy is necessary due to the limitations of the formalism used to represent examples in the machine learning process: the propositional formalism, which structures data in attribute-value vectors.

Even though it is known that great part of the knowledge regarding to languages is relational (e.g., syntactic or semantic relations among words in a sentence) (Mooney, 1997), the propositional formalism traditionally employed makes unfeasible the representation of substantial relational knowledge and the use of this knowledge during the learning process.

According to the attribute-value representation, one attribute has to be created for every feature, and the same structure has to be used to characterize all the examples. In order to represent the syntactic relations between every pair of words in a sentence, e.g., it will be necessary to create at least one attribute for each possible relation (Figure 1). This would result in an enormous number of attributes, since the possibilities can be many in distinct sentences. Also, there could be more than one pair with the same relation.

Sentence: *John gave to Mary a big cake.*			
$verb_1$-$subj_1$	$verb_1$-obj_1	mod_1-obj_1	…
give-john	give-cake	big-cake	…

Figure 1. Attribute-value vector for syntactic relations

Given that some types of information are not available for certain instances, many attributes will have null values. Consequently, the representation of the sample data set tends to become highly sparse. It is well-known that sparseness on data ensue serious problems to the machine learning process in general (Brown and Kros, 2003). Certainly, data will become sparser as more knowledge about the examples is considered, and the problem will be even more critical if relational knowledge is used.

Therefore, at least three relevant problems arise from the use of a propositional representation in corpus-based and hybrid approaches: (a) the limitation on its expressiveness power, making it difficult to represent relational and other more complex knowledge; (b) the sparseness in data; and (c) the lack of integration of the evidences provided by examples and linguistic knowledge.

3 A hybrid relational approach for WSD

We propose a novel hybrid approach for WSD based on a relational representation of both examples and linguistic knowledge. This representation is considerably more expressive, avoids sparseness in data, and allows the use of these two types of evidence during the learning process.

3.1 Sample data

We address the disambiguation of 7 verbs selected according to the results of a corpus study (Specia, 2005). To build our sample corpus, we collected 200 English sentences containing each of the verbs from a corpus comprising fiction books. In a previous step, each sentence was automatically tagged with the translation of the verb, part-of-speech and lemmas of all words, and subject-object syntactic relations with respect to the verb (Specia et al., 2005). The set of verbs, their possible translations, and the accuracy of the most frequent translation are shown in Table 1.

Verb	# Translations	Most frequent translation - %
come	11	50.3
get	17	21
give	5	88.8
go	11	68.5
look	7	50.3
make	11	70
take	13	28.5

Table 1. Verbs and their possible senses in our corpus

3.2 Inductive Logic Programming

We utilize Inductive Logic Programming (ILP) (Muggleton, 1991) to explore relational machine learning. ILP employs techniques of both Machine Learning and Logic Programming to build first-order logic theories from examples and background knowledge, which are also represented by means of first-order logic clauses. It allows the efficient representation of substantial knowledge about the problem, and allows this knowledge to be used during the learning process. The general idea underlying ILP is:

Given:

- a set of positive and negative examples $\boldsymbol{E} = \boldsymbol{E}^+ \cup \boldsymbol{E}^-$

- a predicate $\boldsymbol{p}$ specifying the target relation to be learned

- knowledge $\boldsymbol{K}$ of a certain domain, described according to a language $\boldsymbol{L_k}$, which specifies which other predicates $\boldsymbol{q_i}$ can be part of the definition of $\boldsymbol{p}$.

The goal is: to induce a hypothesis (or theory) $\boldsymbol{h}$ for $\boldsymbol{p}$, with relation to $\boldsymbol{E}$ and $\boldsymbol{K}$, which covers most of the $\boldsymbol{E}^+$, without covering the $\boldsymbol{E}^-$, that is, $\boldsymbol{K} \wedge \boldsymbol{h} \models \boldsymbol{E}^+$ and $\boldsymbol{K} \wedge \boldsymbol{h} \not\models \boldsymbol{E}^-$.

To implement our approach we chose Aleph (Srinivasan, 2000), an ILP system which provides a complete relational learning inference engine and various customization options. We used the following options, which correspond to the Progol mode (Muggleton, 1995): bottom-up search, non-incremental and non-interactive learning, and learning based only on positive examples. Fundamentally, the default inference engine induces a theory iteratively by means of the following steps:

1. One instance is randomly selected to be generalized.
2. A more specific clause (bottom clause) explaining the selected example is built. It consists of the representation of all knowledge about that example.
3. A clause that is more generic than the bottom clause is searched, by means of search and generalization strategies (best first search, e.g.).
4. The best clause found is added to the theory and the examples covered by such clause are removed from the sample set. If there are more instances in the sample set, return to step 1.

3.3 Knowledge sources

The choice, acquisition, and representation of syntactic, semantic, and pragmatic knowledge sources (KSs) were our main concerns at this stage. The general architecture of the system, showing our 7 groups of KSs, is illustrated in Figure 2.

Several of our KSs have been traditionally employed in monolingual WSD (e.g., Agirre and Stevenson, 2006), while other are specific for MT. Some of them were extracted from our sample corpus (Section 3.1), while others were automatically extracted from lexical resources[1]. In what follows, we briefly describe, give the generic definition and examples of each KS, taking sentence (1), for the "to come", as example.

(1) "If there is such a thing as reincarnation, I would not mind *coming* back as a squirrel".

KS_1: Bag-of-words – a list of ±5 words (lemmas) surrounding the verb for every sentence (*sent_id*).

bag(sent_id, list_of_words).
bag(sent1,[mind, not, will, i, reincarnation, back, as, a, squirrel])

KS_2: Part-of-speech (POS) tags of content words in a ±5 word window surrounding the verb.

has_pos(sent_id, word_position, pos).
has_pos(sent1, first_content_word_left, nn).
has_pos(sent1, second_content_word_left, vbp).

KS_3: Subject and object syntactic relations with respect to the verb under consideration.

has_rel(sent_id, subject_word, object_word).
has_rel(sent1, i, nil).

KS_4: Context words represented by 11 collocations with respect to the verb: 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.

has_collocation(sent_id, collocation_type, collocation)
has_collocation(sent1, word_right_1, back).
has_collocation(sent1, word_left_1, mind). …

KS_5: Selectional restrictions of verbs and semantic features of their arguments, given by LDOCE. Verb restrictions are expressed by lists of semantic features required for their subject and object, while these arguments are represented with their features.

rest(verb, subj_restrition, obj_ restriction ,translation)
rest(come, [], nil, voltar).
rest(come, [animal,human], nil, vir). ...

feature(noun, sense_id, features).
feature(reincarnation, 0_1, [abstract]).
feature(squirrel, 0_0, [animal]).

The hierarchy for LDOCE feature types defined by Bruce and Guthrie (1992) is used to account for restrictions established by the verb for features that are more generic than the features describing the words in the subject / object roles in the sentence.

Ontological relations extracted from WordNet (Miller, 1990) are also used: if the restrictions imposed by the verb are not part of the description of its arguments, synonyms or hypernyms of those arguments that meet the restrictions are considered.

relation(word1, sense_id1, word2 ,sense_id2).
hyper(reincarnation, 1, avatar, 1).
synon(rebirth, 2, reincarnation, -1).

KS_6: Idioms and phrasal verbs, indicating that the verb occurring in a given context could have a specific translation.

[1] Michaelis® and Password® English-Portuguese Dictionaries, LDOCE (Procter, 1978), and WordNet (Miller, 1990).

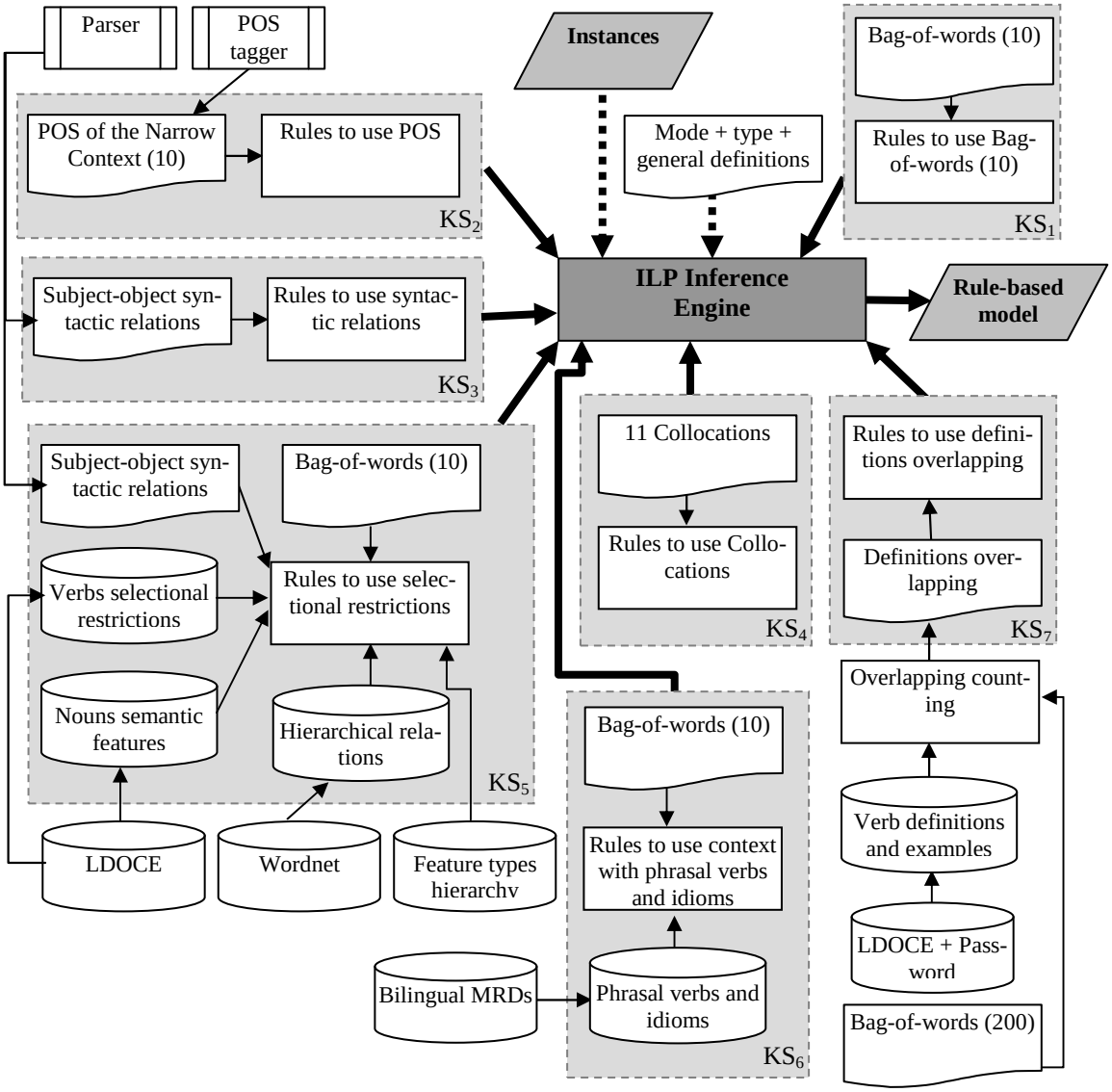

Figure 2. System architecture

```
        exp(verbal_expression, translation)
exp('come about', acontecer).
exp('come about', chegar).                    …
```

KS7: A count of the overlapping words in dictionary definitions for the possible translations of the verb and the words surrounding it in the sentence, relative to the total number of words.

```
    highest_overlap(sent_id, translation, overlapping).
highest_overlap(sent1, voltar, 0.222222).
highest_overlap(sent2, chegar, 0.0857143).
```

The representation of all KSs for each example is independent of the other examples. Therefore, the number of features can be different for different sentences, without resulting in sparseness in data.

In order to use the KSs, we created a set of rules for each KS. These rules are not dependent on particular words or instances. They can be very simple, as in the example shown below for bag-of-words, or more complex, e.g., for selectional restrictions. Therefore, KSs are represented by means of rules and facts (rules without conditions), which can be intensional, i.e., it can contain variables, making the representation more expressive.

```
has_bag(Sent,Word) :-
        bag(Sent,List), member(Word,List).
```

Besides the KSs, the other main input to the system is the set of examples. Since all knowledge about them is expressed by the KSs, the representation of examples is very simple, containing only the example identifier (of the sentence, in our case, such as, “sent1”), and the class of that example (in

our case, the translation of the verb in that sentence).

```
sense(sent_id,translation).
sense(sent1,voltar).
sense(sent2,ir).
```

In Aleph's default induction mode, the order of the training examples plays an important role. One example is taken at a time, according to its order in the training set, and a rule can be produced based on that example. Since examples covered by a certain rule are removed from the training set, certain examples will not be used to produce rules. Induction methods employing different strategies in which the order is irrelevant will be exploited in future work.

In order to produce a theory, Aleph also requires "mode definitions", i.e., the specification of the predicates ***p*** and ***q*** (Section 3.2). For example, the first mode definition below states that the predicate ***p*** to be learned will consist of a clause *sense(sent_id, translation)*, which can be instantiated only once (*1*). The other two definitions state the predicates ***q***, *has_colloc(sent_id, colloc_id, colloc)*, with at most *11* instantiations, and *has_bag(sent_id, word)*, with at most *10* instantiations. That is, the predicates in the conditional piece of the rules in the theory can consist of up to 11 collocations and a bag of up to 10 words. One mode definition must be created for each KS.

```
:- modeh(1,sense(sent,translation)).
:- modeb(11,has_colloc(sent,colloc_id,colloc)).
:- modeb(10,has_bag(sent,word)).    …
```

Based on the examples and background knowledge, the inference engine will produce a set of symbolic rules. Some of the rules induced for the verb "to come", e.g., are illustrated in the box below.

```
1. sense(A, sair) :-
     has_collocation(A, preposition_right, out).
2. sense(A, chegar) :-
     satisfy_restrictions(A, [animal,human],[concrete]);
     has_expression(A, 'come at').
3. sense(A, vir) :-
     satisfy_restriction(A, [human],[abstract]),
     has_collocation(A, word_right_1, from).
```

The first rule checks if the first preposition to the right of the verb is "out", assigning the translation "sair" if so. The second rule verifies if the subject-object arguments satisfy the verb restrictions, i.e, if the subject has the features "animal" or "human", and the object has the feature "concrete". Alternatively, it verifies if the sentence contains the phrasal verb "come at". Rule 3 also tests the verb selectional restrictions and the first word to the right of the verb.

4 Experiments and results

In order to assess the accuracy of our approach, we ran a set of initial experiments with our sample corpus. For each verb, we ran Aleph in the default mode, except for the following parameters:

set(evalfn, posonly): learns from positive examples.
set(search, heuristic): turns the search strategy heuristic.
set(minpos, 2): establishes as 2 the minimum number of positive examples covered by each rule in the theory.
set(gsamplesize, 1000): defines the number of randomly generated negative examples to prune the search space.

The accuracy was calculated by applying the rules to classify the new examples in the test set according to the order these rules appeared in the theory, eliminating the examples (correctly or incorrectly) covered by a certain rule from the test set. In order to cover 100% of the examples, we relied on the existence of a rule without conditions, which generally is induced by Aleph and points out to the most frequent translation in the training data. When this rule was not generated by Aleph, we add it to the end of theory. For all the verbs, however, this rule only classified a few examples (form 1 to 6).

In Table 2 we show the accuracy of the theory learned for each verb, as well as accuracy achieved by two propositional machine learning algorithms on the same data: Decision Trees (C4.5) and Support Vector Machine (SVM), all according to a 10-fold cross-validation strategy. Since it is rather impractical to represent certain KSs using attribute-value vectors, in the experiments with SVM and C4.5 only low level features were considered, corresponding to $\mathbf{KS_1}$, $\mathbf{KS_2}$, $\mathbf{KS_3}$, and $\mathbf{KS_4}$. On average, Our approach outperforms the two other algorithms. Moreover, its accuracy is by far better than the accuracy of the most frequent sense baseline (Table 1).

For all verbs, theories with a small number of rules were produced (from 19 to 33 rules). By looking at these rules, it becomes clear that all KSs are being explored by the ILP system and thus are potentially useful for the disambiguation of verbs.

5 Conclusion and future work

We presented a hybrid relational approach for WSD designed for MT. One important characteristic of our approach is that all the KSs were

Verb	Aleph Accuracy	C4.5 Accuracy	SVM Accuracy
come	0.82	0.55	0.6
Get	0.51	0.36	0.45
Give	0.96	0.88	0.88
Go	0.73	0.73	0.72
look	0.83	0.66	0.84
make	0.74	0.76	0.76
Take	0.66	0.35	0.41
Average	0.75	0.61	0.67

Table 2. Results of the experiments with Aleph

automatically extracted, either from the corpus or machine-readable lexical resources. Therefore, the work could be easily extended to other words and languages.

In future work we intend to carry out experiments with different settings: (a) combinations of certain KSs; (b) other sample corpora, of different sizes, genres / domains; and (c) different parameters in Aleph regarding search strategies, evaluation functions, etc. We also intend to compare our approach with other machine learning algorithms using all the KSs employed in Aleph, by preprocessing the KSs in order to extract binary features that can be represented by means of attribute-value vectors. After that, we intend to adapt our approach to evaluate it with standard WSD data sets, such as the ones used in Senseval[2].

References

E. Agirre and M. Stevenson. 2006 (to appear). Knowledge Sources for Word Sense Disambiguation. In *Word Sense Disambiguation: Algorithms, Applications and Trends*, Agirre, E. and Edmonds, P. (Eds.), Kluwer.

M.L. Brown, J.F. Kros. 2003. Data Mining and the Impact of Missing Data. *Industrial Management and Data Systems*, 103(8):611-621.

R. Bruce and L. Guthrie. 1992. Genus disambiguation: A study in weighted performance. In *Proceedings of the 14th COLING*, Nantes, pp. 1187-1191.

B.J. Dorr and M. Katsova. 1998. Lexical Selection for Cross-Language Applications: Combining LCS with WordNet. In *Proceedings of AMTA'1998*, Langhorne, pp. 438-447.

W.J. Hutchins and H.L. Somers. 1992. *An Introduction to Machine Translation*. Academic Press, Great Britain.

H. Lee. 2002. Classification Approach to Word Selection in Machine Translation. In *Proceedings of AMTA'2002*, Berlin, pp. 114-123.

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

G.A. Miller, R.T. Beckwith, C.D. Fellbaum, D. Gross, K. Miller. 1990. WordNet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235-244.

R.J. Mooney. 1997. Inductive Logic Programming for Natural Language Processing. In *Proceedings of the 6th International ILP Workshop*, Berlin, pp. 3-24.

S. Muggleton. 1991. Inductive Logic Programming. *New Generation Computing*, 8 (4):295-318.

S. Muggleton. 1995. Inverse Entailment and Progol. *New Generation Computing Journal*, 13: 245-286.

B.S. Pedersen. 1997. *Lexical Ambiguity in Machine Translation: Expressing Regularities in the Polysemy of Danish Motion Verbs*. PhD Thesis, Center for Sprogteknologi, Copenhagen.

P. Procter (editor). 1978. *Longman Dictionary of Contemporary English.* Longman Group, Essex, England.

L. Specia. 2005. A Hybrid Model for Word Sense Disambiguation in English-Portuguese MT. In *Proceedings of the 8th CLUK*, Manchester, pp. 71-78.

L. Specia, M.G.V Nunes, M. Stevenson. 2005. Exploiting Parallel Texts to Produce a Multilingual Sense-tagged Corpus for Word Sense Disambiguation. In *Proceedings of RANLP-05*, Borovets, pp. 525-531.

A. Srinivasan. 2000. *The Aleph Manual. Technical Report*. Computing Laboratory, Oxford University. URL: http://web.comlab.ox.ac.uk/oucl/research/areas/machlearn/Aleph/aleph_toc.html.

M. Stevenson and Y. Wilks. 2001 The Interaction of Knowledge Sources for Word Sense Disambiguation. *Computational Linguistics*, 27(3):321-349.

D. Vickrey, L. Biewald, M. Teyssier, and D. Koller. 2005. Word-Sense Disambiguation for Machine Translation. In *Proceedings of HLT/EMNLP-05*, Vancouver.

N. Zinovjeva. 2000. *Learning Sense Disambiguation Rules for Machine Translation*. Master's Thesis, Department of Linguistics, Uppsala University.