Unsupervised Learning of Syntactic Knowledge: methods and measures

R. Basili (*), A. Marziali (*), M.T. Pazienza (*), P. Velardi(#)

(*) Dipartimento di Informatica, Sistemi e Produzione, Universita' di Roma Tor Vergata (ITALY) {basili,pazienza}@info.utovrm.it

(#) Istituto di Informatica, Universita' di Ancona (ITALY)
velardi@anvax1.cineca.it

Abstract

Supervised methods for ambiguity resolution learn in "*sterile*" environments, in absence of syntactic noise. However, in many language engineering applications manually tagged corpora are not available nor easily implemented. On the other side, the "*exportability*" of disambiguation cues acquired from a given, noise-free, domain (e.g. the Wall Street Journal) to other domains is not obvious.

Unsupervised methods of lexical learning have, just as well, many inherent limitations. First, the type of syntactic ambiguity phenomena occurring in real domains are much more complex than the standard V N PP patterns analyzed in literature. Second, especially in sublanguages, syntactic noise seems to be a *systematic* phenomenon, because many ambiguities occur within identical phrases. In such cases there is little hope to acquire a higher statistical evidence of the correct attachment. Class-based models may reduce this problem only to a certain degree, depending upon the richness of the sublanguage, and upon the size of the application corpus.

Because of these inherent difficulties, we believe that syntactic learning should be a gradual process, in which the most difficult decisions are made as late as possible, using increasingly refined levels of knowledge.

In this paper we present an incremental, class-based, unsupervised method to reduce syntactic ambiguity. We show that our method achieves a considerable compression of noise, preserving only those ambiguous patterns for which shallow techniques do not allow reliable decisions.

Unsupervised vs. supervised models of syntactic learning

Several corpus-based methods for syntactic ambiguity resolution have been recently presented in the literature. In (Hindle and Rooth, 1993) hereafter H&R, lexicalized rules are derived according to the probability of *noun-preposition* or *verb-preposition* bigrams for ambiguous structures like *verb-noun-preposition-noun* sequences. This method has been criticised because it does not consider the PP object in the attachment decision scheme. However collecting bigrams rather than trigrams reduces the well known problem of data sparseness.

In subsequent studies, trigrams rather than bigrams were collected from corpora to derive disambiguation cues. In (Collins and Brooks,1995) the problems of data sparseness is approached with a supervised back-off model, with interesting results. In (Resnik and Hearst, 1993) class-based trigrams are obtained by generalizing the PP head, using WordNet synonymy sets. In (Ratnaparkhi et al, 1994) word classes are derived automatically with a clustering procedure. (Franz, 1995) uses a loglinear model to estimate preferred attachments according to the linguistic features of co-occurring words (e.g. bigrams, the accompanying noun determiner, etc.). (Brill and Resnik, 1994) use transformation-based error-driven learning (Brill, 1992) to derive disambiguation rules based on simple context information (e.g. right and left adjacent words or POSs).

All these approaches need extensive collections of positive examples (i.e. hand corrected attachment instances) in order to trigger the acquisition process. Probabilistic, backed-off or loglinear models rely entirely on noise-free data, that is, correct parse trees or bracketed structures. In general the training set is the parsed Wall Street Journal (Marcus et al, 1993), with few exceptions, and the size of the training samples is around 10-20,000 test cases. Some methods do not require manually validated PP attachments, but word

collocations are collected from large sets of noise-free data. Unfortunately, in language engineering applications, manually tagged corpora are not widely available nor easily implemented[1]. On the other side, the "*exportability*" of disambiguation cues obtained in a given domain (e.g. WSJ) to other domains is not obvious.

Unsupervised methods have, on their side, serious limitations:

- First, the type of occurring syntactic ambiguity phenomena are in the average much more complex than the standard *verb-noun-preposition-noun* patterns analyzed in literature. H&R method has been proved very weak on complex phenomena like *verb-noun-preposition-noun-preposition-noun* sequences (see (Franz,1995)). Other methods (supervised or not) do not consider more complex ambiguous structures.
- Second, in real environments, and especially in sublanguages, syntactic noise seems to be a *systematic* phenomenon. Many ambiguities occur within several identical phrases, hence the "*wrong*" and the "*right*" associations may gain the same statistical evidence. Therefore, there are intrinsic limitations to the possibility of using purely statistical approaches to ambiguity resolution.

The nature of ambiguous phenomena in untagged corpora has not been studied in detail in the literature although one such analysis would be very useful on a language engineering standpoint. Accordingly, section 2 is devoted to an experimental analysis of complexity and recurrence of ambiguous phenomena in sublanguages. This analysis demonstrates that syntactic disambiguation in large cannot be afforded by the use of knowledge induced exclusively from the corpus. We think that corpus based techniques are useful to significantly *reduce*, not to *eliminate*, the ambiguous phenomena. In section 3, we describe an unsupervised, class-based, incremental, syntactic disambiguation method that is aimed at reducing noisy collocates, to the extent that this is allowed by the observation of corpus phenomena. The approach that we support is *to reduce syntactic ambiguity through an incremental process*. Decisions are deferred until enough evidence has been gained of a noisy phenomenon. First, a kernel of shallow grammatical competence is used to extract a collection of noise-prone syntactic collocates. Then, a global data analysis is performed to review local choices and derive new statistical distributions. This incremental process can be iterated to the point that the system reaches a kernel of "hard" cases for which there is no more evidence for a reliable decision. The output of the last iteration represents a less noisy environment on which additional learning process can be triggered (e.g. sense disambiguation, acquisition of subcategorization frames, ...). These later inductive phases may rely on some level of *a priori* knowledge, like for example the *naive* case relations used in the ARIOSTO_LEX system (Basili et al, 1993c , 1996).

[1] It is not just a matter of time, but also of required linguistic skills (see for example (Marcus et al, 1993)).

Complexity and recurrence of ambiguous patterns in corpora

In the previous section we pointed out that unsupervised lexical learning methods must cope with complex and repetitive ambiguities. We now describe an experiment to measure these phenomena in corpora. In this experiment, we wish to demonstrate that:

- The type of syntactic ambiguities are much more complex than V N PP or N N PP sentences. In a realistic environment, the correct attachment must be selected among several possibilities, not just two.
- The fundamental assumption of most common statistical analyses is that the events being analyzed (productive word pairs or triples in our case) are independent. Instead, ambiguous patterns are highly repetitive, especially in sublanguages. This means that in many cases, unless we work in absence of noise, the "*correct*" and "*wrong*" associations in an ambiguous phrase acquires the same or similar statistical evidence.

To conduct the experiment, we used a *shallow syntactic analyzer* (SSA) (Basili et al, 1994) to extract word associations from two very different corpora in Italian (a scientific corpus of environmental abstracts, called ENEA, and a legal corpus on taxation norms, called LD) [2].

Given a corpus, SSA produces an extensive database of *elementary syntactic links* (*esl*). Typical *esl* classes express the following dependency relations: noun-preposition-noun (N_P_N), verb-preposition-noun (V_P_N), adjective-conjunction-adjective (Adj_C_Adj) and others. An *esl* has the following structure:

$$esl(h, mod(p, w)),$$

where h is the head of the underlying phrasal structure and $mod(p, w)$ denotes the head modifier, and w as the modifier head.

Ambiguity is generated by multiple morphologic derivations and intrinsic language ambiguities (PP references, coordination, etc.). Given a sentence, SSA

[2] SSA is based on a DCG model with controlled skip rules

produces in general a noise-prone set of *esl's*, some of which represent colliding interpretations. The definition of Collision Set (CS) is the following:

DEF(Collision Set): A Collision Set (*CS*) is the set of syntactic groups, derived from a given sentence that share the same modifier, *mod*().

To smooth the weight of ambiguous *esl's* in lexical learning, each detected *esl* is weighted by a measure called *plausibility*. To simplify, the plausibility of a detected *esl* is roughly inversely proportional to the number of mutually excluding syntactic structures in the text segment that generated the *esl* (see (Basili et al, 1993a) for details).

In the following, we show examples of collision sets extracted from the LD (an English word by word translation is provided for the sentence fragments that generated a collision set). It is important to observe that the complexity does not arise simply from the number of colliding tuples but also from the structure of ambiguous patterns (e.g. non consecutive word strings, as in the second example). Bold characters identify the *mod*(p, w) shared by colliding tuples. Local plausibility values are reported on the right.

1. Examples of Simple Collision sets:

1.1 Minimal Attachment (consecutive word strings):

su richiesta del ministro per le finanze , il [(servizio di vigilanza sulle aziende) **di credito**] (* service of control of agencies **of credit**) controlla l'esattezza delle attestazioni contenute nel certificato .

g_N_p_N(2,azienda,di,credito) 0.333

g_N_p_N(4,vigilanza,di,credito) 0.333

g_N_p_N(6,servizio,di,credito) 0.333

1.2 Non-Minimal Attachment (non consecutive word strings)

i sostituti d imposta devono [(presentare la dichiarazione di-cui-a quarto comma dell'articolo 9, relativamente ai pagamenti fatti e agli utili distribuiti nell'anno 1974) **entro il 15- aprile- 1975**]. (* must present the declaration of which at comma 4th of item 9, relatively to the payment done and the profit distributed in the year 1974,**within april 15, 1974**)

g_N_p_N(17,articolo,entro,x_15_aprile_1975) 0.166

g_N_p_N(7,distribuire,entro,x_15_aprile_1975) 0.166

g_Adv_p_N(14,relativamente,entro,x_15_aprile_1975) 0.166

g_N_p_N(19,comma,entro,x_15_aprile_1975) 0.166

g_V_p_N(24,presentare,entro,x_15_aprile_1975) 0.166

To measure the complexity of the ambiguous structures, we collected from fragments of the two corpora all the ambigous collision sets, i.e. those with more than one *esl*. 10,433 collision sets were found in the ENEA corpus and 30,130 in the LD[3]. Figure 1 plots the percentage of colliding *esl's* vs. the cardinality of collision sets. The average size of ambiguous collision sets is about 4 in both corpora.

Of course SSA introduces additional noise due to its shallow nature (see referred papers for an evaluation of performances[4]), but as far as our experiment is concerned (measuring the complexity of collision sets) SSA still provides a good testbed. In fact, some *esl* can be missed in a collision set, or some spurious attachment can be detected, but in the average, these phenomena are sufficiently rare and in any case they tend to be equally probable.

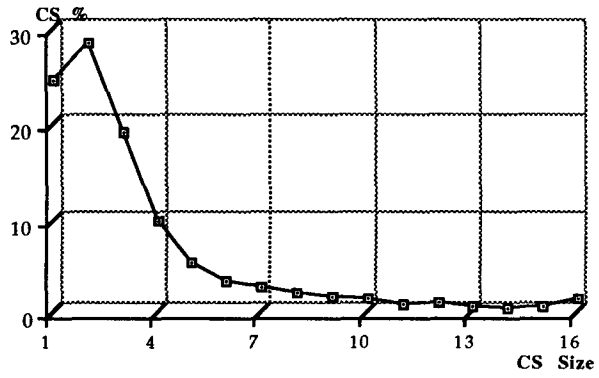


Figure 1: Percentage of collision sets Vs. number of colliding tuples for the LD.

In the second experiment we measure the recurrence of ambiguous patterns. This phenomenon is known to be typical in sublanguage, but was never analyzed in detail. A straightforward measure of recurrence is provided by the *average Mutual Information* of colliding *esl's*. This figure measures the probability of co-occurrence of two *esl's* in a collision set. If the Mutual Information is high, it means that the measured phenomena (productive word tuples) do *not* independently occur in collision sets, i.e. they systematically occur in reciprocal ambiguity in the corpus. The consequence is that statistically based lexical learning methods are faced not only with the problem of *data sparseness* (events that are never or rarely encountered), but also with the problem of *systematic ambiguity* (events

[3]The LD test corpus is larger, and in addition, the legal language is more verbous and less concise than the scientific style that characterizes the ENEA corpus.

[4]We measured an average of 80% precision and 75% recall over three corpora, one of which in English.

Table 1: Mutual Information of co-occurring esl's

	LD (30,130 CS)	ENEA (10,433 CS)
Average MI	13.65	12.9
σ	1.8	0.84
σ^2	3.2	.72
average frequency of *esl's*	1.9	1.43

Table 2: Mutual Information of *esl's* occurring with frequency higher than average

	LD	ENEA
Average MI	11.60	11.60
σ	2.05	1.12
σ^2	4.23	1.27

that occur always in the same sequence). This phenomenon is likely to be more relevant in sublanguages (medicine, law, engineering) than in narrative texts, but sublanguages are at the basis of many important applications.

The average Mutual Information was evaluated by first computing, in the standard way, the Mutual Information of all the pairs of *esl's* that co-occurred in at least one collision set:

$$MI(esl_i, esl_j) = \log_2 \frac{Prob(esl_i, esl_j)}{Prob(esl_i)Prob(esl_j)} \quad (1)$$

where the probability is evaluated over the space of collision sets with cardinality > 1.

Tables 1 and 2 summarize the results of the experiment.

Tables 1 and 2 show the average MI, standard deviation and variance for the two domains. The values in 1 shows that the *average MI* is *close to the perfect correlation*[5] and has a small variance, especially in the ENEA corpus that is in technical style. This result could be biased by the *esl's* occurring just once in the collision sets, hence we repeated the computation for the pair of *esl's* occurring at a frequency higher than the average (≥ 2, in both domains). The results are reported in Table 2. It is seen that the values remain rather high, still with a small variance.

Clustering the *esl's* would seem an obvious way to reduce this problem. Therefore, in a subsequent experiment we clustered the head of PPs in the collision sets using a set of high-level semantic tags (for a discussion on semantic tagging see (Basili et al, 1992, 1993b)[6]. For example, the *esl*

V_P_N(*to_present, within, april_15_1974*)

is generalized as:

V_P_N(*to_present, within, TEMPORAL_ENTITY*).

Because of sense ambiguity, the collision sets became 20,353 in the ENEA corpus, and 42,681 in the LD. The average frequency of *"right-generalized" esl's* is now 4.28 in the ENEA and 4.64 in the LD. The results are summarised in Table 3.

Table 3: Mutual Information of right-generalized *esl's* in two domains

	LD (all *esl's*)	LD (high freq.*esl's*)	ENEA (all *esl's*)	(hig
Average MI	11.5	7.99	11.00	
σ	3.10	2.15	2.65	
σ^2	9.62	4.66	7.05	

Notice that the phenomenon of systematic ambiguity is much less striking (lower MI and higher variance), though it is not eliminated. It is also important that the two corpora, though very different in style, behave in the same way as far as systematic ambiguity is concerned.

For example, consider the following sentence fragment:

... imposta sul reddito delle persone ... (... tax on the income of people ...)*

that occurs in the LD corpus almost 200 times. The global plausibility of the syntactic collocates (*i*) *imposta-di-persona* (*tax-of-people*) and (*ii*) *reddito-di-persona* (*income-of-people*) is (*i*) 91.66 and (*ii*) 93.69. Therefore a reliable decision is not allowed by the set of syntactic observations found in the corpus. Furthermore, similar sentences, like for example

*... imposta sul reddito delle societa'... (*tax on the income of companies...),*

always have a HUMAN_ENTITY as head modifier. Therefore, the fact that (*reddito di persona*) is correct cannot be captured even when comparing the generalized patterns (*reddito di* HUMAN_ENTITY) and (*imposta di* HUMAN_ENTITY).

[5] Two *esl's* occurring exactly as the average (1.9 in LD) are in perfect correlation when their MI is equal to 13.8.

[6] Class based approaches are widely employed. Clusters are created by means of distributional techniques in (Ratnaparkhi et al, 1994), while in (Resnik and Hearst, 1993) low level synonim sets in WordNet are used. Instead, we use high level tags (human, time, abstraction etc.), manually assigned in Italian domains and automatically assigned from WordNet in English domains. For sake of brevity, we do not re-discuss the matter here. See aforementioned papers.

The conclusion we may derive from these two experiments is that most syntactic disambiguation methods presented in literature are tested in an unrealistic environment. This does not mean that they don't work, but simply that their applicability to real domains is yet to be proven. Application corpora are noisy, may not be very large, and include repetitive and complex ambiguities that are an obstacle to reliable statistical learning.

The experiments also stress the importance of class based models of lexical learning. Clustering "*similar*" phenomena is an obvious way of reducing the problems just outlined. Unfortunately, Table 3 shows that generalization improves, but not eliminates, the problem of repetitive patterns.

An incremental architecture for unsupervised reduction of syntactic ambiguity

The previous section shows that we need to be more realistic in approaching the problem of syntactic ambiguity resolution in large. Certain results can be obtained with purely statistical methods, but there are many complex cases for which there seems to be a clear need for less shallow techniques.

The approach that we have undertaken is to attack the problem of syntactic ambiguity through increasingly refined learning phases. The first stage is *noise compression*, in which we adopt an incremental syntactic learning method, to create a more suitable framework for subsequent steps of learning. Noise compression is performed essentially by the use of shallow NLP and statistical techniques. This method is described hereafter, while the subsequent steps, that use deeper (rule-based) levels of knowledge, are implemented into the ARIOSTO_LEX lexical learning system, described in (Basili et al., 1993b, 1933c and 1996).

A feedback algorithm for noise reduction

The process of incremental noise reduction works as follows:

1. First, use a surface grammatical competence (i.e. SSA) to derive the (noise prone) set of observations.
2. Cluster the collocational data according to semantic categories.
3. Apply class based disambiguation operators to reduce the initial source of noise, by first disambiguating the non-persistent ambiguity phenomena.
4. Derive new statistical distributions.
5. Repeat step 2.-4. on the remaining (i.e. persistent) ambiguous phenomena.

The incremental disambiguation activity stops when no more evidence can be derived to solve new ambiguous cases.

In order to accomplish the outlined noise reduction process we need: (*i*) a disambiguation operator and (*ii*) a disambiguation strategy to eliminate at each step "*some*" noisy collocations.

The class based disambiguation operator is the *Mutual Conditioned Plausibility (MCPl)* (Basili et al.,1993a). Given an *esl*, the value of its corresponding $MCPl$ is defined by the following:

DEF(*Mutual Conditioned Plausibility*): The Mutual Conditioned Plausibility (MCPl) of a prepositional attachment $esl(w, mod(p, n))$, is:

$$MCPl(esl(w, mod(p, n))) = \frac{\sum_{y \in \Gamma} pl(esl(w, mod(p, y)))}{\sum_{\forall h, y \in \Gamma} pl(esl(h, mod(p, y))) \sum_{\forall y} pl(esl(w, mod(p, y)))} \quad (2)$$

where Γ is the high level semantic tag assigned to the modifier n and $pl()$ is the plausibility function. Examples of the generalized *esl's* were presented in the previous section. For example to the computation of the $MCPl$ of $esl(reddito, (di, persona))$ contribute *esl's* like $esl(reddito, (di, professionista))$, $esl(reddito, (di, azienda))$ where $professionista, persona$ and $azienda$ are instances of $HUMAN\_ENTITIY$.

After a first scan of the corpus by the SSA and after the computation of global MCPl values, a *primary knowledge base* is available. This knowledge is fully corpus driven, and it is obtained without a preliminary training set of hand tagged patterns. Each *esl* in a collision set has its own MCPl value, that has been globally derived from the corpus. The MCPl is thus employed to remove the less plausible attachments proposed by the grammar, with a consequent reduction in size of the related collision sets. When more than one *esl* remain in a collision set the system *is not forced to decide*, and a further disambiguation step is attempted later.

After the first scan of the corpus by means of the SSA grammar, the corpus is re-written as a set of possibly ambiguous Collision Sets, i.e. if C is the corpus and CS_i a Collision Set, we have:

$$C = CS_0 \cup CS_1 \cup ... \cup CS_i \cup ...CS_N$$
$$CS_i \cap CS_j = \{\emptyset\}, \qquad \text{for } i \neq j, i, j = 0, 1, 2, ...N$$

where N is the total number of collision sets found in the corpus.

The cardinality of a generic collision set is directly proportional to the degree of ambiguity of its members. The feedback algorithm tries to reduce the cardinality

Table 4: A general feedback algorithm for noise reduction

(1) Use SSA to derive all the syntactic observations O from the corpus;
Set the initial performance index PFC' to 0;
(2) REPEAT

(2.1) Substitute PCF with PCF'
(2.2) Evaluate the MCPl for each $esl \in O$
(2.3) Use MCPl on a subset of the corpus (testset) and evaluate the current performance index PCF'
(2.3) IF PCF' > PCF THEN:
(2.3.1) Rewrite the collision sets of O removing hell *esl's* into a new set of observation O'
(2.3.2) Replace O with O'
UNTIL PCF' > PCF
(3) STOP

Table 5: Disambiguation Algorithm: Learning Phase

Let $CS = \{ e_1, e_2, \ldots e_N \}$ be any collision set in the corpus, where $e_i's$ are *esl's*
Let $\frac{1}{N}$ be the *prior probability* (*pprior*).
Let $MCPl(e_i)$ be the Mutual Conditional Plausibility (2) of e_i
The *posterior probability* of e_i, $ppost_i$, is defined as

$$ppost_i = \frac{MCPl(e_i)}{\sum_{j=1}^{N} MCPl(e_j)}$$

Let $\sigma \in [0,1]$ be a given learning threshold.

For each e_i in CS DO:
IF $\frac{ppost_i}{pprior} < 1 - \sigma$ THEN
REMOVE e_i from CS, i.e. PUT it in the *hell* set
OTHERWISE e_i is a *limbo esl*.
IF $\forall i \neq j$ e_i is in *hell*
MOVE e_j in the *paradise set*

of all CS_i step by step: *esl* with "*lower*" MCPl values (as globally derived from all the corpus) are filtered out; the MCPl values are then redistributed among the remaining *esl's*. In a picturesque way, we can say that discarded *esl's* are damned (the *hell* is the right place), while survived *esl's* are waiting for next judgment (the *limbo* is the right place for this wait state); at the end of the algorithm, if there is a single winner *esl*, it will gain the *paradise*. Persistently ambiguous *esl* of the corpus may remain still ambiguous within the corresponding collision sets: limbo will be their place forever. The algorithm will try to obtain as many paradise *esl's* (i.e. singleton CS) as possible but is robust against persistently ambiguous phenomena.

The general feedback algorithm is illustrated in Table 4. It should be noted that the above feedback strategy has three main phases: (step 2.2) statistical induction of syntactic preference scores; (step 2.3) testing phase (which is necessary in order to quantify the performance of disambiguation criteria derived from the current statistical distributions); (step 2.3.1) learning phase, to filter out the syntactically odd *esl's* (i.e. *esl* with locally low MCPl values).

Learning and Testing disambiguation cues

According to the *disambiguate as late as possible* strategy, the learning and testing phases have different objectives:

- During the learning phase, the objective is to take only highly reliable decisions, by eliminating those *esl's* with a very low plausibility, while delaying unreliable choices.
- During the test phase, the objective is to evaluate the ability of the system at separating, within each collision set, correct from wrong attachment candidates.

This results in two different disambiguation algorithms: the learning phase is used only to remove hell *esl's* from the collision sets, without forcing any paradise choice (e.g. a maximum likelihood candidate). In the test phase *esl's* are classified as (locally) correct and wrong according to their relative values of MCPl.

The learning phase, called i_{th} -learning step, is guided by the following algorithm:

1. Identify all Collision Sets of the corpus, CS_i, $i = 1, 2, \ldots N$;
2. Apply the preference criterion to each CS_i in order to classify *hell*, *limbo* or *paradise esl's*;
3. Redistribute plausibility values among the limbo *esl's* of each CS_i;

Step 2 is further specified in Table 5.

In step 3 of the Learning algorithm, the new plausibility values are redistributed among the survived *esl's* according to the following rule:

$$pl_{i+1}(esl(h, mod(p, w))) = pl_i \frac{pl_i(CS_i)}{pl_{i+1}(CS_{i+1})} \quad (3)$$

where i is the learning step and $CS_{i+1} (\subseteq CS_i)$ does not contain *esl's* that have been placed in *hell* during step i.

After each learning step the upgraded plausibility values provide newer MCPl scores that are more reliable because the hell *esl's* have been discarded.

Table 6: Disambiguation Algorithm: Learning Phase

Let $CS = \{ e_1, e_2, \ldots e_N \}$ be any collision set the test set and $Ncases$ be the number of test cases.
Let $\frac{1}{N}$ be the *prior probability* (*pprior*).
Let $MCPl(e_i)$ be the Mutual Conditional Plausibility (2) of e_i
The *posterior probability* of e_i, $ppost_i$, is defined as $ppost_i = \frac{MCPl(e_i)}{\sum_{j=1}^{N} MCPl(e_j)}$
Let $\tau \in [0,1]$ be a given test threshold.

For each CS and for each $e_i \in CS$ DO:
  IF $\frac{ppost_i}{pprior} > 1 + \tau$ THEN
    IF e_i is correct, i.e. manually validated, THEN
      $++TruePositives$;
    OTHERWISE
      $++FalsePositives$;
  OTHERWISE IF $\frac{ppost_i}{pprior} < 1 - \tau$ THEN
    IF e_i is correct THEN
      $++FalseNegatives$;
    OTHERWISE
      $++TrueNegatives$;
  $++Ncases$

$$precision = \frac{TruePositives+TrueNegatives}{TruePositives+TrueNegatives+FalsePositives+FalseNegatives}$$

$$recall = \frac{TruePositives+TrueNegatives}{Ncases}$$

$$coverage = \frac{TruePositives+TrueNegatives+FalsePositives+FalseNegatives}{Ncases}$$

The evaluation of each learning step is carried on by testing the syntactic disambiguation on a selected set of corpus sentences where ambiguities have been manually solved.

The general test algorithm is defined in Table 6.

In Table 6, notice that *precision* and *recall* evaluate the ability of the system *both* at eliminating truly wrong *esl's* and accepting truly correct *esl's*, since, as remarked in section 2, our objective is noise compression, rather than full syntactic disambiguation. Notice also that, because of their different classification objectives, learning and testing use different decision thresholds.

Experimental Results.

To evaluate numerically the benefits of the feedback algorithm, several experiments and performance indexes have been evaluated. The corpus selected for experimenting the incremental technique is the LD: the size of the corpus is about 500,000 words. The SSA grammar in LD has about 25 DCG rules and it generates

Table 7: Performance values of the MCPl without learning

τ	Coverage	Recall	Precision
0.0	99.8%	0.75	0.749
0.05	95.0%	0.72	0.75
0.1	87.4%	0.69	0.79
0.2	77.8%	0.62	0.80
0.5	49.9%	0.42	0.84

240,493 *esl's* from the whole corpus. Of these only 10% of *esl's* are initially unambiguous, while all the remaining are limbo *esl's*. A testset of 1,154 hand corrected collision sets was built. 5,285 different *esl's* are in the testset. An average of 25.9% correct groups have been found in the testset, again demonstrating a great level of ambiguity in the source data.

At first, we need to study the system *classification parameters*, σ and τ (see Tables (5) and (6)) . During the learning phase, we wish to eliminate as many hell *esl's* as possible, because the more noise has been eliminated from the source syntactic data, the more reliable is the application of the later inductive operators (i.e. ARIOSTO lexical learning system). However we know from the experiments in section 2 that the competence that we are using (shallow NLP and statistical operators) is insufficient to cope with highly repetitive ambiguities. The threshold σ is therefore a crucial parameter, because it must establish the best trade-off between precision of choices (i.e. it must classify as *hell* truly noisy *esl's*) and impact on noise compression (i.e. it must remove as much noise as possible).

Table 7 shows the results.

To select the best value for σ, we measured the values of *recall* and *precision* (defined in Table 6) according to different values for τ. These measures have been derived from the early (thus noisy) state of knowledge where just the SSA grammar, and no learning, was applied to the corpus.

According to the results of Table 7, $\tau = 0.2$ was selected for the better trade-off between *recall*, *precision* and *coverage*. The learning steps have then be performed with a threshold value $\sigma = 0.2$ over the LD corpus. In each phase the corresponding *recall* and *precision* have been measured.

The results of the experiment are summarised in Figure 2. Figure 2.A plots *recall* versus *precision* that have been obtained in the early (prior to learning) stage (Step 0), after 1 (Step 1) and 2 (Step 2) learning iterations. Each measure is evaluated for a different value of the testing threshold τ, that varies from 0.5 to 0.0 from left to right in Fig. 2.A.

Figure 2.B plots the *Information Gain* (Kononenko and Bratko, 1991) an information theory index that, roughly speaking, measures the quality of the statistical distributions of the correct vs. wrong *esl's*. Fig-

Table 8: Performance values of the LA without learning

τ	Coverage	Recall	Precision
0.0	100%	0.610	0.610
0.05	96.5%	0.594	0.615
0.1	93.8%	0.578	0.616
0.2	86.4%	0.544	0.631
0.5	71.9%	0.465	0.647

ure 2.C measures the *Data Compression*, that is the mere reduction of *els's* in the corpus. The compression is measured as the ratio between hell's *els's* and the number of the observed *esl's*. Figure 2.D plots the *Coverage*, i.e. the number of decided cases over the total number of possible decisions. Finally, Table 8 reports the performance (at the Step 0 phase) of the H&R Lexical Association (LA) [7]. We experiment this disambiguation operator just because the H&R method has, among the others, the merit of being easily reproducible.

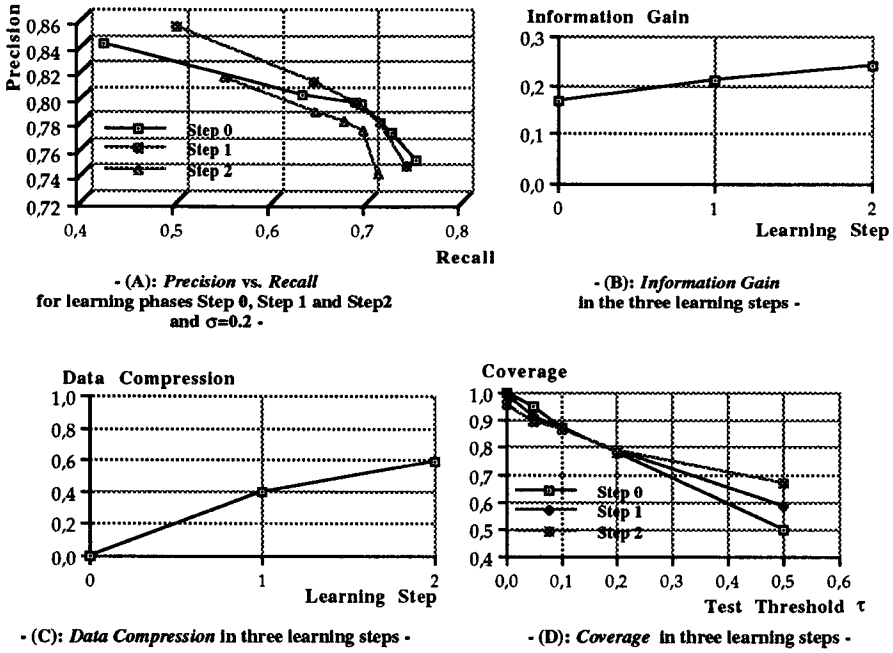
The first four figures give a global overview of the method. In Fig. 2.A (Step 1), a significant improvement in *precision* can be observed. For $\tau = 0.5$ the improvement in *recall* (.5) and *precision* (.85) is more sensible. Furthermore a better coverage (60 %) is shown in Fig. 2.D (Step 1). A further index to evaluate the status of the system knowledge about the PP-attachment problem is the *Information Gain* ((Kononenko and Bratko, 1991) and (Basili et al, 1996)). The posterior probability (see algorithms in Table 5 and 6) improves over the *"blind"* prior probability as much as it increases the confidence of correct *esl's* and decreases the confidence of wrong *esl's*. The improvement is quantified by means of the number of saved bits necessary to describe the correct decisions when moving from prior to posterior probability. The *Information Gain* does not depend on the selected thresholds, since it acts on all the probability values, and it is related to the complexity of the learning task. It *gives a measure of the global trend of the statistical decision model.* A significant improvement measured over the testset (12% to 24% relative increment) is shown by Fig. 2.B as a result of the learning steps. As discussed in (Basili et al.,1994), the Information Gain produces performance results that may contrast with precision and recall.

In fact, in the learning step 2, we observed decreased performance of precision and recall. The overlearning effect is common of feedback algorithms. Furthermore, the small size of the corpus is likely to anticipate this phenomenon. The problem is clearly due to the highly repetitive ambiguities. The system quickly removes from the corpus syntactically wrong *esl's* with low MCPl. But now let's consider a collision set with two *esl's* that almost constantly occur together. Their MCPl tends to acquire exactly the same value. Thus, they will stay in the limbo forever. But if one of the two, accidentally the wrong, has an even minimal additional evidence with respect to its competitor, this initially small advantage may be emphasized by the plausibility redistribution rule 3[8]. Hence once the learning algorithm reaches the *"hard cases"* and is still forced to discriminate, it gets at stuck, and may take accidental decisions. This phenomenon occurs very early in our domains, and this could be easily foreseen according to the high correlation between *esl's* that we measured.

For the current experimental setup, our data show a significant reduction of noise with a significant 40% compression of the data after step 1, and a correspondent slight improvement in precision-recall, given the complexity of the task (see the Lexical Association performance in Table 8, for a comparison). However, the phenomena that we analyzed in Section 2 have a negative impact on the possibility of a longer incremental learning process. We do not believe that experimenting over different domains would give different results. In fact, the Legal and Environmental sublanguages are very different in style, and not so narrow in scope. Rather, we believe that the size of the corpora may be in fact too small. We could hope in a higher variability of language patterns by training over 1-2 million words corpora.

[7] Unlike H&R, we did not use the *t-score* as a decision criteria, but forced the system to decide according to different values of the thresholds τ for sake of readability of the comparison. Technical details of our treatment of the LA operator within our grammatical framework can be found in (Basili et al,1994),

[8] whereas, for more independent phenomena, 3 should emphasize the right attachments.



- (A): *Precision* vs. *Recall* for learning phases Step 0, Step 1 and Step2 and σ=0.2 -

- (B): *Information Gain* in the three learning steps -

- (C): *Data Compression* in three learning steps -

- (D): *Coverage* in three learning steps -

Figure 2: Incremental Learning: Experimental Results

Further improvements could also be obtained using a more refined discriminator than MCPl, but there is no free lunch. If the corpus is our unique source of knowledge, it is not possible to learn things for which there is no evidence. Only if we can rely on some a-priori model of the world, even a *naive* model[9] to guide difficult choices, then we can hope in a better coverage of repetitive phenomena.

Conclusions

As a conclusion we may claim that corpus-driven lexical learning should result from the interaction of cooperating inductive processes triggered by several knowledge sources. The described method is a combination of numerical techniques (e.g. the probability driven MCPl disambiguation operator) and some logical devices:

- a shallow syntactic analyzer that embodies a *surface and portable grammatical competence* helpful in triggering the overall induction;
- a *naive* semantic type system to obviate the problem of data sparseness and to give the learning system some *explanatory power*

The interaction of such components has been exploited in an incremental process. In the experiments, the performance over a typical NLP task [10] (i.e. PP-disambiguation) has been significantly improved by this a cooperative approach. Moreover, on the language engineering standpoint the main consequences are a significant *data compression* and a corresponding improvement of the overall *system efficiency.*

One of the purposes of this paper was to show that, despite the good results recently obtained in the field of corpus-driven lexical learning, we must still demonstrate that NLP techniques, after the advent of lexical statistics, are industrially competitive. And one good way for doing so, is by measuring ourselves with the full complexities of language. More effort should thus be devoted in evaluating the performance of lexical learning methods in real world, noisy domains.

REFERENCES

(Basili et al.,1992) Basili, R., Pazienza, M.T., Velardi, P., *Computational Lexicons: the Neat Examples and the Odd Exemplars*, Proc. of Third Int. Conf. on Applied Natural Language Processing, Trento, Italy, 1-3 April, 1992.

(Basili et al,1993a) Basili, R., A. Marziali, M.T. Pazienza, *Modelling syntactic uncertainty in lexical acquisition from texts*, Journal of Quantitative Linguistics, vol.1, n.1, 1994.

(Basili et al,1993b) Basili, R., M.T. Pazienza, P. Velardi, *What can be learned from raw texts ?*, Journal of Machine Translation, 8:147-173,1993.

(Basili et al,1993c) Basili, R., M.T. Pazienza, P. Velardi, *Acquisition of selectional patterns*, Journal of Machine Translation, 8:175-201,1993.

(Basili et al.,1994a) Basili, R., M.T. Pazienza, P.Velardi, *A (not-so) shallow parser for collocational analysis*, Proc. of Coling '94, Kyoto, Japan, 1994.

(Basili et al.,1994b) Basili, R., M.H.Candito, M.T. Pazienza, P. Velardi, *Evaluating the information gain of probability-based PP-disambiguation methods*, Proc. of International Conference on New Methods in Language Processing, Manchester, September 1994.

(Basili et al.,1996), Basili, R., M.T. Pazienza, P.Velardi, *An Empirical Symbolic Approach to Natural Language Processing*, Artificial Intelligence, to appear on vol. 85, August 1996

(Brill 1992) Brill, E., *A simple rule-based part of speech tagger*, in Proc. of the 3rd Conf. on Applied Natural Language Processing, ACL, Trento Italy

(Brill and Resnik,1994) Brill E., Resnik P., *A rule-based approach to prepositional phrase attachment disambiguation*, in Proc. of COLING 94, 1198-1204

(Collins and Brooks,1995) Collins M. and Brooks J., *Prepositional Phrase Attachment trough a Backed-off Model*, 3rd. Workshop on Very Large Corpora, MT, 1995

(Franz,1995), Franz A., *A statistical approach to learning prepositional phrase attachment disambiguation*, in Proc. of IJCAI Workshop on New Approaches to Le arning for Natural Language Processing, Montreal 1995.

(Hindle and Rooth,1993) Hindle D. and Rooth M., *Structural Ambiguity and Lexical Relations*, Computational Linguistics, 19(1): 103-120.

(Kononenko and Bratko, 1991) Kononenko I., I. Bratko, *Information-Based Evaluation Criterion for Classifier's Performance*, Machine Learning, 6,67-80, 1991.

(Marcus et al, 1993) Marcus M., Santorini B. and Marcinkiewicz M., *Building a large annotated corpus in English: The Penn Tree Bank*, Computational Linguistics, 19(2): 313-330.

(Ratnaparkhi et al, 1994), Ratnaparkhi, Rynar and Roukos, *A maximum entropy model for prepositional phrase attachment.* In ARPA Workshop on Human language Technology, plainsboro, NJ, 1994.

(Resnik and Hearst, 1993) Resnik P. and Hearst M., *Structural Ambiguity and Conceptual Relations*, in Proc. of 1st Workshop on Very Large Corpora, 1993.

[9] like for example the coarse selectional restrictions used by the ARIOSTO_LEX system (see refereed papers)

[10] although inherently hard for an unsupervised noise-prone framework