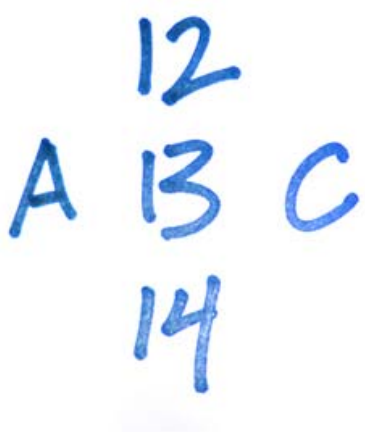
Relaxation Labeling Processes

The importance of contextual information



c → cat
→ circus

i → sin
→ fine

e → red
→ read

festival

graphics

The labeling problem involves a set of objects $\mathbf{B} = \{b_1, \ldots, b_n\}$ and a set of possible labels $\mathbf{\Lambda} = \{1, \ldots, m\}$.[1] The purpose is to label each object of $\mathbf{B}$ with one label of $\mathbf{\Lambda}$. To accomplish this, two sources of information are exploited. The first one relies on *local* measurements which capture the salient features of each object viewed in isolation; classical pattern recognition techniques can be practically employed to carry out this task. The second source of information, instead, accounts for possible interactions among nearby labels and, in fact, incorporates all the contextual knowledge about the problem at hand. This is quantitatively expressed by means of a real-valued four-dimensional matrix of compatibility coefficients $R = \{r_{ij}(\lambda, \mu)\}$. The coefficient $r_{ij}(\lambda, \mu)$ measures the strength of compatibility between the hypotheses "b_i has label λ" and "b_j has label μ:" high values correspond to compatibility and low values correspond to incompatibility.

The initial local measurements are assumed to provide, for each object $b_i \in \mathbf{B}$, an m-dimensional vector $\bar{p}_i^{(0)} = (p_i^{(0)}(1), \ldots, p_i^{(0)}(m))^T$ (where "T" denotes the usual transpose operation), such that $p_i^{(0)}(\lambda) \geq 0$, $i = 1 \ldots n$, $\lambda \in \mathbf{\Lambda}$, and $\sum_\lambda p_i^{(0)}(\lambda) = 1$, $i = 1 \ldots n$. Each $p_i^{(0)}(\lambda)$ can be regarded as the initial, non-contextual degree of confidence of the hypothesis "b_i is labeled with label λ." By simply concatenating $\bar{p}_1^{(0)}, \bar{p}_2^{(0)}, \ldots, \bar{p}_n^{(0)}$ we obtain a weighted labeling assignment for the objects of $\mathbf{B}$ that will be denoted by $\bar{p}^{(0)} \in \mathbb{R}^{nm}$. A relaxation labeling process takes as input the initial labeling assignment $\bar{p}^{(0)}$ and iteratively updates it taking into account the compatibility model R.

At this point, we introduce the space of weighted labeling assignments:

$$\mathbb{IK} = \left\{ \bar{p} \in \mathbb{R}^{nm} \mid p_i(\lambda) \geq 0,\ i = 1 \ldots n,\ \lambda \in \mathbf{\Lambda} \right.$$
$$\left. \text{and} \sum_{\lambda=1}^{m} p_i(\lambda) = 1,\ i = 1 \ldots n \right\}$$

which is a linear convex set of $\mathbb{R}^{nm}$. Every vertex of $\mathbb{IK}$ represents an *unambiguous* labeling assignment, that is one which assigns exactly one label to each object. The set of these labelings will be denoted by $\mathbb{IK}^*$:

$$\mathbb{IK}^* = \{\bar{p} \in \mathbb{IK} \mid p_i(\lambda) = 0 \text{ or } 1,\ i = 1 \ldots n,\ \lambda \in \mathbf{\Lambda}\}.$$

Moreover, a labeling $\bar{p}$ in the interior of $\mathbb{IK}$ (i.e., $0 < p_i\ (\lambda) < 1$, for all i and λ) will be called *strictly ambiguous*.

Now, let $\bar{p} \in \mathbb{K}$ be any labeling assignment. To develop a relaxation algorithm that updates $\bar{p}$ in accordance with the compatibility model, we need to define, for each object $b_i \in \mathbf{B}$ and each label $\lambda \in \mathbf{\Lambda}$, what is called a *support* function. This should quantify the degree of agreement between the hypothesis that b_i is labeled with λ, whose confidence is expressed by $p_i(\lambda)$, and the context. This measure is commonly defined as follows (see, e.g., [4, 20, 21] for alternative definitions):

$$q_i(\lambda; \bar{p}) = \sum_{j=1}^{n} \sum_{\mu=1}^{m} r_{ij}(\lambda, \mu) p_j(\mu). \qquad (1)$$

In practical applications, some simplifying assumptions are made. First, it is usually assumed that objects interact only within a small neighborhood; for each $i = 1 \cdots n$ we will denote by Δ_i the neighborhood of object b_i, that is the set of relative positions that are supposed to influence the object on site i.

support formula (2) is replaced with

$$q_{i\lambda}^{(t)} = \sum_{\delta \in \Delta_i} \sum_{\mu=1}^{m} r_{\delta\lambda\mu} p_{i+\delta,\mu}^{(t)}. \tag{3}$$

Now, $r_{\delta\lambda\mu}$ denotes the compatibility between labels λ and μ, when μ is at offset δ from λ.

We will find it convenient to abandon the matrix notation for compatibilities, regarding them as real vectors: $\boldsymbol{r} \in R^D$, with $D = |\Delta| m^2$.

Putting together the $q_i(\lambda; \bar{p})$'s, as for the $p_i(\lambda)$'s, we obtain an nm-dimensional support vector that will be denoted by $\bar{q}(\bar{p})$.[2] Support factors have an obvious interpretation: $q_i(\lambda)$ is high when high-confidence neighboring labels are "compatible" with λ on b_i; conversely, it is low when high-confidence neighboring labels are "incompatible" with λ. Furthermore, notice that low-confidence nearby labels have little or no influence on the support measure, and this is what one should expect.

The above discussion suggests a way to properly adjust the labeling $\bar{p}$: increase $p_i(\lambda)$ when $q_i(\lambda)$ is high and decrease it when $q_i(\lambda)$ is low. This naturally leads to the following updating rule

$$p_i(\lambda) := p_i(\lambda) q_i(\lambda) \Big/ \sum_{\mu=1}^{m} p_i(\mu) q_i(\mu) \qquad (2)$$

where the denominator serves simply to ensure that the updated vectors are still in IK. Formulas (1) and (2) define the original nonlinear relaxation operator of Rosenfeld et al. [1] which was in fact originally motivated by making recourse to the simple-minded, heuristic arguments just developed.

The relaxation algorithm will be best viewed as a continuous mapping $\mathcal{T}$ of the assignment space onto itself. It starts out with $\bar{p}^{(0)}$ and iteratively produces a sequence of points $\bar{p}^{(0)}, \bar{p}^{(1)}, \bar{p}^{(2)}, \ldots \in \mathbb{K}$, where $\bar{p}^{(t+1)} = \mathcal{T}(\bar{p}^{(t)})$, $t \geq 0$. The process continues until (at least in theory) a fixed, or equilibrium, point is reached, which means that $\mathcal{T}(\bar{p}^{(t)}) = \bar{p}^{(t)}$, for some t. It can be easily shown that a labeling $\bar{p}$ is an equilibrium point for $\mathcal{T}$ if and only if the following relation holds [22]:

$$q_i(\lambda) = c_i \text{ whenever } p_i(\lambda) > 0, \quad i = 1 \ldots n, \ \lambda \in \mathbf{\Lambda} \tag{3}$$

for some nonnegative constants $c_1, \ldots, c_n$ (note that unambiguous labelings are therefore equilibrium points for $\mathcal{T}$; the converse, of course, need not be true).

Consistency and its properties

In this section, we briefly review Hummel and Zucker's theory of constraint satisfaction [6] which commences by providing a general definition of consistency. By analogy with the unambiguous case, which is more easily understood, a weighted labeling assignment $\bar{p} \in \mathbb{IK}$ is said to be *consistent* if

$$\sum_{\lambda=1}^{m} p_i(\lambda) q_i(\lambda; \bar{p}) \geq \sum_{\lambda=1}^{m} v_i(\lambda) q_i(\lambda; \bar{p}), \quad i = 1 \ldots n$$

for all $\bar{v} \in \mathbb{IK}$. Furthermore, if strict inequalities hold in (4), for all $\bar{v} \neq \bar{p}$, then $\bar{p}$ is said to be *strictly consistent*. It can be seen that a necessary condition for $\bar{p}$ to be strictly consistent is that it is an unambiguous one, that is $\bar{p} \in \mathbb{IK}^*$. Consistency is also usefully characterized by the following condition: $(\bar{v} - \bar{p}) \cdot \bar{q}(\bar{p}) \leq 0$ for all $\bar{v} \in \mathbb{IK}$, where "$\cdot$" denotes the standard inner product operator.

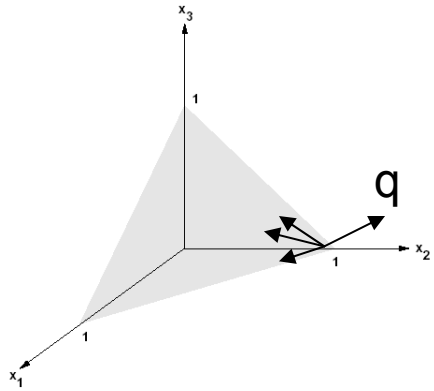
Geometrical interpretation

Given a labeling $\bar{p} \in \mathbb{IK}$, the tangent set at $\bar{p}$, denoted by $T_{\bar{p}}$, is defined as the set of possible directions along which one can move an infinitesimal amount away from $\bar{p}$, while remaining in $\mathbb{IK}$. It turns out that the tangent set at $\bar{p}$ is given by:

$$T_{\bar{p}} = \left\{ \bar{d} \in \mathbb{R}^{nm} \;\middle|\; \sum_{\lambda=1}^{m} d_i(\lambda) = 0, \quad i = 1 \ldots n, \right.$$

$$\left. p_i(\lambda) = 0 \Rightarrow d_i(\lambda) \geq 0, \quad i = 1 \ldots n, \;\; \lambda \in \Lambda \right\}.$$

Owing to the convexity of $\mathbb{IK}$, all the tangent vectors at $\bar{p}$ are of the form $\gamma(\bar{v} - \bar{p})$, for some $\gamma \geq 0$ and $\bar{v} \in \mathbb{IK}$. Accordingly, consistency is equivalent to the condition $\bar{d} \cdot \bar{q}(\bar{p}) \leq 0$, for all $\bar{d} \in T_{\bar{p}}$.



Theorem 3.1. *A labeling* $\bar{p} \in \mathbb{IK}$ *is consistent if and only if for all* $i = 1 \ldots n$ *the following conditions hold*:

1) $q_i(\lambda) = c_i$, *whenever* $p_i(\lambda) > 0$
2) $q_i(\lambda) \leq c_i$, *whenever* $p_i(\lambda) = 0$

for some nonnegative constants $c_1, \ldots, c_n$.

Corollary 3.2. *Let* $\bar{p} \in \mathbb{IK}$ *be consistent. Then* $\bar{p}$ *is a fixed point for the nonlinear relaxation operator* $\mathcal{T}$. *Moreover, if* $\bar{p}$ *is strictly ambiguous the converse also holds.*

The “average local consistency”

In [6], Hummel and Zucker introduced the *average local consistency*, defined as

$$A(\bar{p}) = \sum_{i=1}^{n} \sum_{\lambda=1}^{m} p_i(\lambda) q_i(\lambda) \qquad (5)$$

and proved the following fundamental result.

Theorem 3.3 (Hummel-Zucker, [6]). *Suppose that the compatibility matrix R is symmetric (i.e., $r_{ij}(\lambda, \mu) = r_{ji}(\mu, \lambda)$ for all i, j, λ, μ). Then any local maximum $\bar{p} \in \mathbb{IK}$ of A is consistent.*

Note that, in general, the converse of Theorem 3.3 need not be true since, to prove this, second-order derivative information would be required. However, the next proposition asserts that, by demanding that $\bar{p}$ be strictly consistent, this *does* happen.

Proposition 3.4. *Let $\bar{e} \in \mathbb{K}^*$ be strictly consistent, and suppose that R is symmetric. Then $\bar{e}$ is a strict local maximum of the average local consistency A.*

The relaxation algorithm: the symmetric case

Theorem 4.1 (Baum-Eagon [24]). *Let $P(\bar{x})$ be a homogeneous polynomial in the variables $\{x_i(\lambda)\}$ with nonnegative coefficients, and let $\bar{x}$ be a point of the domain $\mathbb{K}$. Define the mapping $\mathcal{M}$ as follows:*

$$(\mathcal{M}(\bar{x}))_i(\lambda) = x_i(\lambda)\frac{\partial P(\bar{x})}{\partial x_i(\lambda)} \Bigg/ \sum_{\mu=1}^{m} x_i(\mu)\frac{\partial P(\bar{x})}{\partial x_i(\mu)} \ . \tag{7}$$

Then $P(\mathcal{M}(\bar{x})) > P(\bar{x})$, unless $\mathcal{M}(\bar{x}) = \bar{x}$.

Theorem 5.1. *The nonlinear relaxation operator $\mathcal{T}$ is a growth transformation for the average local consistency A, provided that compatibility coefficients are nonnegative and symmetric.*

More explicitly, the preceding theorem asserts that the nonlinear relaxation scheme strictly increases the average local consistency on each iteration, i.e.,

$$A(\bar{p}^{(t+1)}) > A(\bar{p}^{(t)}), \quad t = 0, 1, \ldots \tag{9}$$

until a fixed point is reached. Even more interestingly, from (8) we can assert that $A(\bar{p}^{(t)})$ is also smaller than the value of A at each labeling assignment lying on the segment joining $\bar{p}^{(t)}$ to $\bar{p}^{(t+1)}$, for each time step $t \geq 0$.

Theorem 5.2. *Let $\bar{e} \in \mathbb{K}^*$ be strictly consistent and suppose that the compatibility matrix R is nonnegative and symmetric. Then $\bar{e}$ is an asymptotically stable equilibrium point for the nonlinear relaxation scheme $\mathcal{T}$ and, consequently, is a local attractor.*

The asymmetric case

Theorem 6.3. *Let $\bar{p}^{(0)}$ be a strictly ambiguous labeling, and suppose that the sequence $\{\bar{p}^{(t)}\}$ produced by the nonlinear relaxation process $\mathcal{T}$ converges to the fixed point $\bar{p}^* \in \mathbb{IK}$. Then $\bar{p}^*$ is consistent.*

Theorem 6.4. *Let $\bar{e} \in \mathbb{IK}^*$ be a strictly consistent labeling. Then $\bar{e}$ is an asymptotically stable equilibrium point for the nonlinear relaxation scheme $\mathcal{T}$.*

Learning the compatibility coefficients

The learning problem

The learning algorithm developed in this paper is based on the assumption that a set of instances of the problem we intend to solve is available. To be more specific, it is supposed that a number of learning samples exist:

$$L = \{L_1, \cdots, L_N\},$$

where each sample $L_\gamma (\gamma = 1 \cdots N)$ is a set of labeled objects of the form

$$L_\gamma = \{(b_i^\gamma, \lambda_i^\gamma) : 1 \leq i \leq n_\gamma, b_i^\gamma \in \boldsymbol{B}, \lambda_i^\gamma \in \Lambda\}.$$

Clearly, the b_i's can well be feature vectors describing real objects.

For each $\gamma = 1 \cdots N$ let $\boldsymbol{p}^{(L_\gamma)} \in R^{n_\gamma m}$ denote the unambiguous labeling assignment for the objects of L_γ, that is

$$p_{i\alpha}^{(L_\gamma)} = \begin{cases} 0, & \text{if } \alpha \neq \lambda_i^\gamma, \\ 1, & \text{if } \alpha = \lambda_i^\gamma. \end{cases}$$

Furthermore, suppose that we have some mechanism for constructing an initial labeling $\boldsymbol{p}^{(I_\gamma)}$ on the basis of the objects in L_γ, and let $\boldsymbol{p}^{(F_\gamma)}$ denote the labeling produced by the relaxation algorithm when $\boldsymbol{p}^{(I_\gamma)}$ is given as input. The same mechanism for deriving the initial labelings should be used both in the "learning" and in the "testing" phases.

A relaxation process is a function that, given as input a vector of compatibility coefficients $\boldsymbol{r}$ and an initial labeling $\boldsymbol{p}^{(I)}$, produces iteratively the final labeling $\boldsymbol{p}^{(F)}$, i.e., $\boldsymbol{p}^{(F)} \leftarrow \mathrm{Relax}(\boldsymbol{r}, \boldsymbol{p}^{(I)})$. In our approach we consider the relaxation operator as a function of the compatibility coefficients only, the initial labeling being considered as a constant. To emphasize this dependence we will write $p_{i\lambda}^{(F)}(\boldsymbol{r})$ to denote the $i\lambda$ component of the final labeling.

A quadratic error function

Within this framework, a natural way to derive compatibility coefficients is to choose them so that $\boldsymbol{p}^{(F_\gamma)}$ be as close as possible to the desired labeling $\boldsymbol{p}^{(L_\gamma)}$, for each $\gamma = 1 \cdots N$. To do so, we can define an error function measuring the loss incurred when $\boldsymbol{p}^{(F_\gamma)}$ is obtained instead of $\boldsymbol{p}^{(L_\gamma)}$, and attempt to minimize it. As an example, a quadratic error function may be adopted:

$$E_\gamma^{(Q)}(\boldsymbol{r}) = \frac{1}{2} \sum_{i=1}^{n_\gamma} \sum_{\lambda=1}^{m} (p_{i\lambda}^{(F_\gamma)}(\boldsymbol{r}) - p_{i\lambda}^{(L_\gamma)})^2, \tag{4}$$

which measures the (squared) Euclidean distance between $\boldsymbol{p}^{(L_\gamma)}$ and $\boldsymbol{p}^{(F_\gamma)}$, when $\boldsymbol{r}$ is used. Also, the total error achieved can be defined as

$$E^{(Q)}(\boldsymbol{r}) = \sum_{\gamma=1}^{N} E_\gamma^{(Q)}(\boldsymbol{r}). \tag{5}$$

A logarithmic error function

An alternative error function comes from information theory. Notice, in fact, that both $\boldsymbol{p}^{(F_\gamma)}$ and $\boldsymbol{p}^{(L_\gamma)}$ are composed of n_γ discrete probability distributions: $\boldsymbol{p}_i^{(F_\gamma)}$ and $\boldsymbol{p}_i^{(L_\gamma)}$, respectively $(i = 1 \cdots n_\gamma)$. Of course, we wish that each $\boldsymbol{p}_i^{(F_\gamma)}$ be as close as possible to $\boldsymbol{p}_i^{(L_\gamma)}$. A well-known information–theoretic divergence measure between two probability distributions is Kullback's I directed divergence [33], which has been successfully employed also in certain connectionist learning procedures [34], [35]. Kullback's divergence is defined as[2]

$$I(\boldsymbol{p}_i^{(L_\gamma)} | \boldsymbol{p}_i^{(F_\gamma)}(\boldsymbol{r})) = \sum_{\lambda=1}^{m} p_{i\lambda}^{(L_\gamma)} \ln \frac{p_{i\lambda}^{(L_\gamma)}}{p_{i\lambda}^{(F_\gamma)}(\boldsymbol{r})}. \qquad (6)$$

Since $\boldsymbol{p}_i^{(L_\gamma)}$ is a simple class indicator vector, containing all zeros except at the position corresponding to λ_i^γ, (6) reduces to

$$I(\boldsymbol{p}_i^{(L_\gamma)}|\boldsymbol{p}_i^{(F_\gamma)}(\boldsymbol{r})) = -\ln p_{i\lambda_i^\gamma}^{(F_\gamma)}(\boldsymbol{r}). \tag{7}$$

The "logarithmic" error achieved for sample γ is

$$E_\gamma^{(I)}(\boldsymbol{r}) = -\sum_{i=1}^{n_\gamma} \ln p_{i\lambda_i^\gamma}^{(F_\gamma)}(\boldsymbol{r}) \tag{8}$$

and the total error is

$$E^{(I)}(\boldsymbol{r}) = \sum_{\gamma=1}^{N} E_\gamma^{(I)}(\boldsymbol{r}). \tag{9}$$

In the following, E will be used to denote either $E^{(Q)}$ or $E^{(I)}$.

The learning algorithm

One popular algorithm for solving linearly constrained minimization problems is Rosen's *gradient projection method* [24]. It is basically an extension of the steepest descent procedure for unconstrained problems, to accommodate the presence of constraints. Here, we make use of a simplified form of the algorithm developed in [25].

The algorithm begins with an initial feasible point $\boldsymbol{r}^{(0)}$ and iteratively produces a sequence of points $\{\boldsymbol{r}^{(k)}\}$ so that the objective function E decreases:

$$E(\boldsymbol{r}^{(k+1)}) \leq E(\boldsymbol{r}^{(k)}). \tag{11}$$

At the kth stage a new point is derived according to the following formula

$$\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \rho_k \boldsymbol{u}^{(k)}, \tag{12}$$

where $\boldsymbol{u}^{(k)}$ is the projection of the gradient of the objective function E onto the intersection of hyperplanes defined by the active constraints (i.e. the constraints that are satisfied as equalities by the current point $\boldsymbol{r}^{(k)}$), and ρ_k is a suitable positive step length, determined so that the new point remains feasible.

Algorithm 1:

Input: An initial feasible compatibility vector $\boldsymbol{r}^{(0)}$;

Output: An "optimal" compatibility vector.

1) $k := 0$;
2) determine the indices of active constraints, that is $J^{(k)} = \{(d, \alpha, \beta) : r^{(k)}_{d\alpha\beta} = 0\}$;
3) evaluate the vector $\boldsymbol{u}^{(k)}$, as follows:

$$u^{(k)}_{d\alpha\beta} = \begin{cases} \dfrac{\partial E(\boldsymbol{r}^{(k)})}{\partial r_{d\alpha\beta}}, & \text{if } (d, \alpha, \beta) \notin J^{(k)}, \\ 0, & \text{if } (d, \alpha, \beta) \in J^{(k)}; \end{cases}$$

4) if $u^{(k)} \neq \mathbf{0}$
   - 4.1) determine a suitable step length ρ_k;
   - 4.2) move to the next point using the relation $\boldsymbol{r}^{(k+1)} = \boldsymbol{r}^{(k)} - \rho_k \boldsymbol{u}^{(k)}$;
   - 4.3) $k := k + 1$;
   - 4.4) goto 2);

5) else
   - 5.1) if $\partial E(\boldsymbol{r}^{(k)})/\partial r_{d\alpha\beta} \geq 0 \ \forall (d, \alpha, \beta) \in J^{(k)}$ EXIT;
   - 5.2) else
     - 5.2.1) delete from $J^{(k)}$ the index corresponding to the most negative value;
     - 5.2.2) goto 3);

An application: Part-of-speech Disambiguation

In this application, the objects to be labeled are words of a sentence $\boldsymbol{W} = w_1 \cdots w_n$ and the labels are the parts-of-speech. The word labeling task can be accomplished by a two-step procedure. First, by means of some local analysis, one derives an initial labeling assignment $\boldsymbol{p}^{(0)}$. The simplest way of doing this consists of using a dictionary look-up which provides for each word the list of its potential parts-of-speech, but more sophisticated methods that exploit the orthographic structure of words have been developed [54]. Due to the presence of homographs in natural language, that is words belonging to more than one syntactic class, local information does not suffice to achieve good labeling results; therefore, in the second step of the word-labeling procedure, contextual constraints are taken into account. This task can be accomplished by a relaxation labeling process [55], where the compatibility coefficients express the strength of agreement between neighboring syntactic classes.

In the experiments presented here, the initial labelings $\boldsymbol{p}^{(I)}$'s were constructed by uniformly distributing the probability mass among the labels found into a dictionary look-up. More precisely, let $\Lambda_i \subseteq \Lambda$ be the set of possible labels for word w_i, as found by consulting the dictionary; then

$$p_{i\lambda}^{(0)} = \begin{cases} 1/|\Lambda_i|, & \text{if } \lambda \in \Lambda_i, \\ 0, & \text{otherwise.} \end{cases}$$

The final labelings $\boldsymbol{p}^{(F)}$'s, instead, were obtained by stopping the relaxation process after the first iteration. The neighborhood chosen for disambiguation contained only the right offset position (i.e., $\Delta_i = \{+1\}$), while the label set Λ consisted of the main parts-of-speech: verb, noun, adjective, adverb, determiner, conjunction, preposition, pronoun, plus a special miscellaneous label.

In the first phase of our experiments, we took a 3,500-word sample text containing sentences extracted from some issues of the EEC Italian Official Journal. This was part of a larger corpus that was subject to a semi-automatic labeling within the ESPRIT Project 860 "Linguistic analysis of the European languages" [56]. We divided the sample text into three separate parts. The first one (containing about 1,500 words) was used to derive two different statistical compatibility vectors to be used as the initial points for the learning algorithm. More specifically, we determined correlation-based coefficients

$$r^{(0)}_{\delta\lambda\mu} = 1 + \frac{P_\delta(\lambda, \mu) - P(\lambda)P(\mu)}{\sqrt{(P(\lambda) - P(\lambda)^2)(P(\mu) - P(\mu)^2)}} \tag{29}$$

and Peleg's compatibilities [18]

$$r^{(0)}_{\delta\lambda\mu} = \frac{P_\delta(\lambda, \mu)}{P(\lambda)P(\mu)}. \tag{30}$$

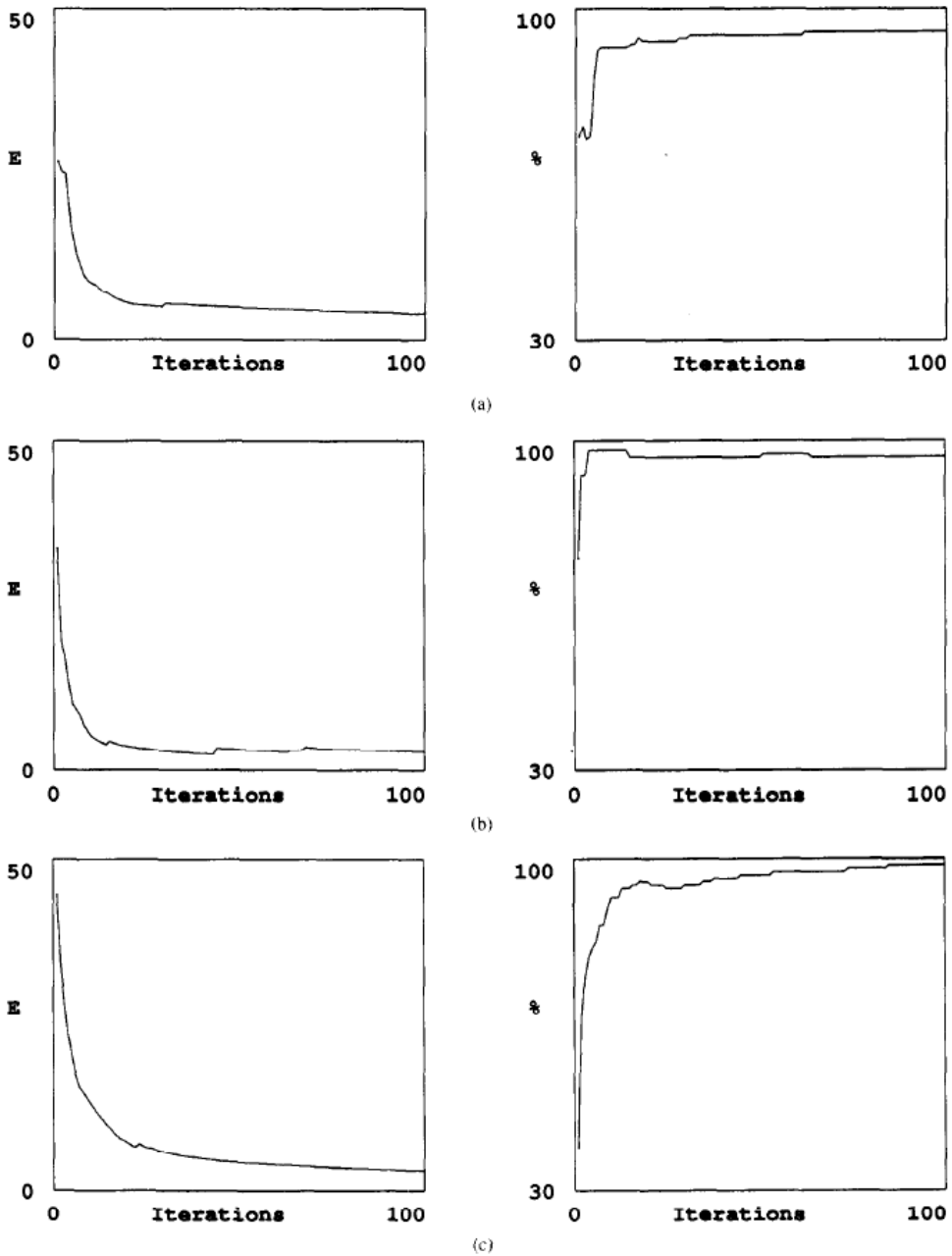


Fig. 1. Behavior of the quadratic error function (on the left side) and the corresponding disambiguation accuracy (on the right side) during the learning process, using different starting points: (a) Peleg's measure; (b) correlation; (c) random point.

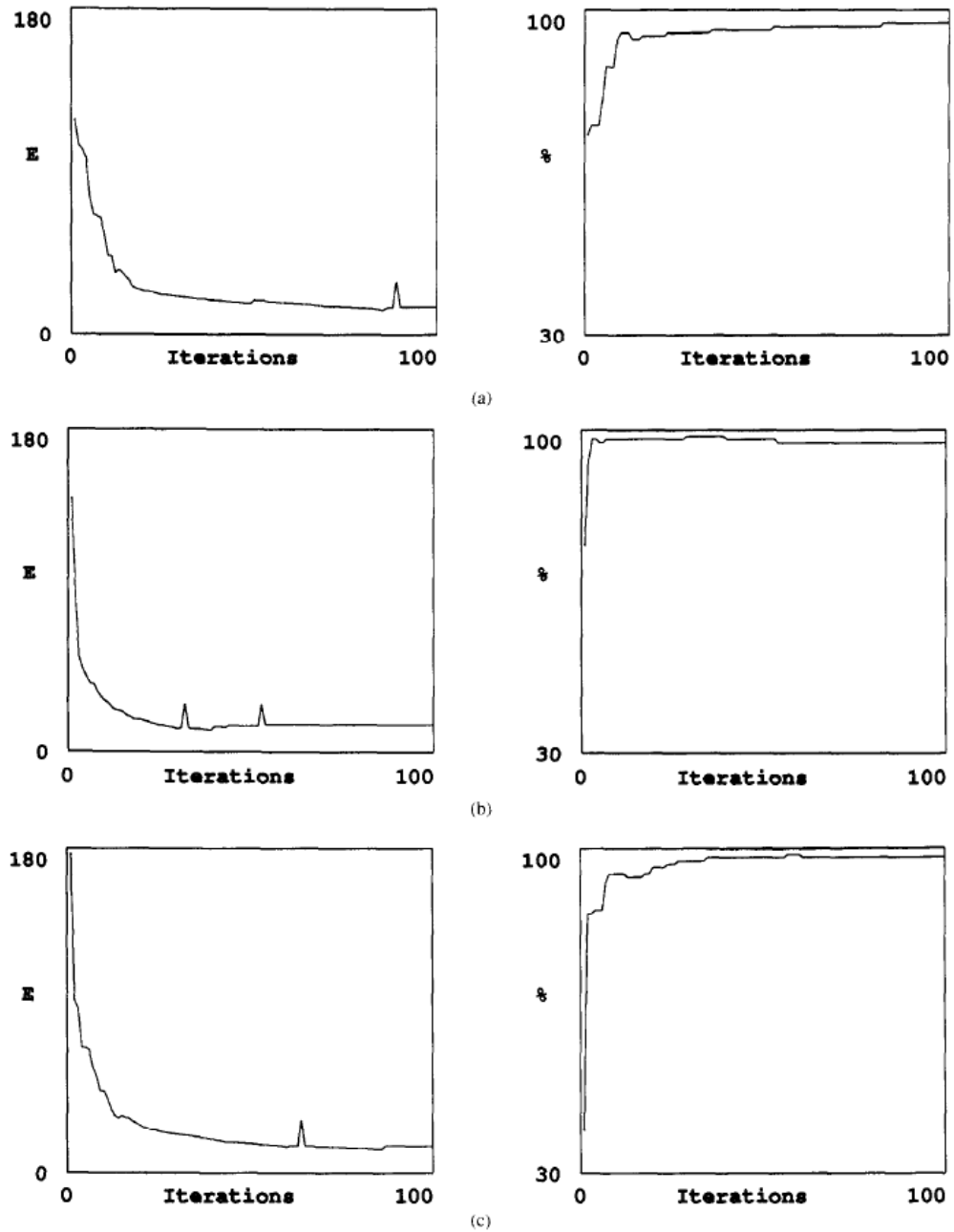


Fig. 2. Behavior of the logarithmic error function (on the left side) and the corresponding disambiguation accuracy (on the right side) during the learning process, using different starting points: (a) Peleg's measure; (b) correlation; (c) random point.

Generalization results

TABLE I

DISAMBIGUATION ACCURACY OF RELAXATION LABELING OVER A 1,000-WORD TEST SAMPLE, USING BOTH THE INITIAL POINTS AND THE BEST POINTS FOUND BY THE LEARNING ALGORITHM

	Initial Points	Optimal Points	
		Quadratic Error	Logarithmic Error
Peleg	72.0%	88.2%	92.6%
Correlation	73.5%	93.4%	94.1%
Random	42.6%	89.7%	91.9%

References

M. Pelillo. The dynamics of nonlinear relaxation labeling processes. *Journal of Mathematical Imaging and Vision*, 1997.

M. Pelillo and M. Refice. Learning compatibility coefficients for relaxation labeling processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1994.