ISTITUTO DI STUDI E ANALISI ECONOMICA

String Matching Algorithms. An Application to ISAE and ISTAT Firms' Registers

by

Emma De Angelis
ISAE, piazza dell'Indipendenza, 4, 00185 Roma, Italia
e-mail: e.deangelis@isae.it

Carmine Pappalardo
ISAE, piazza dell'Indipendenza, 4, 00185 Roma, Italia
e-mail: c.pappalardo@isae.it

Working paper n. 115
July 2009

The Series "*Documenti di Lavoro" of the Istituto di Studi e Analisi Economica* - Institute for Studies and Economic Analyses (ISAE) hosts the preliminary results of the research projects carried out within ISAE. The diffusion of the papers is subject to the favourable opinion of an anonymous referee, whom we would like to thank. The opinions expressed are merely the Authors' own and in no way involve the ISAE responsability.

The series is meant for experts and policy-makers with the aim of submitting proposals and raising suggestions and criticism.

La serie "Documenti di Lavoro" dell'Istituto di Studi e Analisi Economica ospita i risultati preliminari di ricerche predisposte all'interno dell'ISAE. La diffusione delle ricerche è autorizzata previo il parere favorevole di un anonimo esperto della materia che qui si ringrazia. Le opinioni espresse nei "Documenti di Lavoro" riflettono esclusivamente il pensiero degli autori e non impegnano la responsabilità dell'Ente.

La serie è destinata agli esperti e agli operatori di politica economica, al fine di formulare proposte e suscitare suggerimenti o critiche.

Stampato presso la sede dell'Istituto

ISAE - Piazza dell'Indipendenza, 4 - 00185 Roma.
Tel. +39-06444821; www.isae.it

ABSTRACT

The aim of the paper is to develop algorithms for the matching of strings from two different registers of Italian enterprises. This makes effective the possibility of integrating the firm-level information collected by ISAE (Institute for Studies and Economic Analyses) with the one gathered by the Italian NSI. The name of the company is the reference information used for string matching. The first procedure is based on a backward recursive application of the critical factorization method. In the second algorithm, a shifting rule is defined based on the positions of blanks within the reference text T. About 80% of the units in ISAE register of business enterprises have been exactly matched. This suggests that the role of firms' demography should be adequately taken into account to get satisfactory results and confirms the reliability of the proposed algorithms. Some structural characteristics of ISAE sampled firms can be now accounted for.

NON-TECHNICAL SUMMARY

The aim of the paper is to identify the number as large as possible of business activities common to two archives of business activities: the first archive is the one managed by ISAE (Institute for Studies and Economic Analyses), the second one consists of the official statistical repository concerning the universe of productive units administered by the Italian National Statistical Institute. The main purpose of the experiment is to make effective the possibility of integrating the overall firm-level information collected by ISAE with the one gathered by other Institutions.

Two different procedures based on matching algorithms are developed. The basic idea concerning the first algorithm consists in factorizing the pattern P into three parts. The searching phase entails the comparison of the sequence of the three sub-strings from the left to the right in the reference text T. Specifically, an explicit shifting rule, common to many string matching procedures, is replaced by the sequential factorization method we suggest. The goal of the second algorithm is to deal with all cases in which the source of the mismatch is the different order of the substrings included in the company names. This represents the main source of mismatch once the first algorithm is used. In the second one, comparisons are performed for each sequence of characters delimited by blanks in the ISAE company name. To additionally select within the list of firms picked out from the first step, the same procedure applies for comparisons to the string variable reporting the street address and ZIP code in both archives of enterprises. This proves to increase the odds to identify the same productive unit in both firms' registers.

The use of both procedures provided satisfactory results, suggesting that the role of firms' demography is a key item to be adequately taken into account. Some structural characteristics of ISAE sampled firms, not available in the absence of our research, can be now accounted for.

ALGORITMI DI *STRING MATCHING*. UNA APPLICAZIONE AI REGISTRI DELLE IMPRESE ISAE E ISTAT

SINTESI

In questo lavoro si descrivono gli algoritmi di *string matching* utilizzati per l'abbinamento di stringhe provenienti da due diversi registri delle imprese manifatturiere italiane. Da un lato si considera l'archivio ISAE (Istituto di Studi e Analisi Economica) delle imprese manifatturiere; dall'altro, il database ufficiale dell'universo delle imprese attive (ASIA) reso disponibile dall'Istituto nazionale di statistica. L'obiettivo della ricerca consiste nell'associare a ogni impresa dell'archivio ISAE il corrispondente codice impresa ISTAT. Il risultato finale, se soddisfacente, rende effettiva la possibilità di integrare le informazioni *micro* presenti nei campioni ISAE con i dati a livello di impresa disponibili presso altre istituzioni.

Entrambi gli algoritmi si applicano alla ragione sociale dell'impresa. Il primo si basa su una procedura di fattorizzazione ricorsiva del *pattern* tratto dal registro delle imprese ISAE. La seconda procedura consente di tener conto dei casi in cui il *mismatch* sia dovuto al differente ordine delle sotto-stringhe che costituiscono la ragione sociale nei due archivi di riferimento.

Nel complesso, circa 8.000 unità produttive del registro delle imprese ISAE (pari al 78% del totale delle imprese) sono state correttamente identificate nel corrispondente archivio ASIA. Questo risultato conferma l'affidabilità degli algoritmi proposti e indica come il ruolo della demografia di impresa debba essere attentamente considerato per ottenere risultati soddisfacenti.

CONTENTS

1 INTRODUCTION

In this paper the procedures developed for the matching of strings from two different registers of Italian enterprises are described. The first archive is the one managed by ISAE (Institute for Studies and Economic Analyses), which is the national Institution in charge of carrying out business tendency surveys (BTS) in Italy. The second one consists of the official statistical repository concerning the universe of productive units active in the country in a given year (ASIA). This is administered by the Italian National Statistical Institute (ISTAT) and is updated on a yearly basis.

A systematically updated statistical business register is a basic tool in constructing an integrated economic information system serving multiple purposes as sample design, studies on enterprise demographics, identification of the country's installed productive base in terms of location, size, and type of activity. Specifically, it is also aimed at identifying the information links pertaining to each surveyed enterprise across several firm-level inquiries carried out by various Institutions.

Some aspects are to be considered. Firstly, ISAE is part of the Italian National Statistical System (SISTAN)[1]. Secondly, in both the enterprise archives cited above, the firm is considered as the basic unit of observation[2]. Thirdly, ASIA is the source relating to enterprise identification for the whole of ISAE business surveys on the manufacturing sector.

In spite of this, all firms selected to enter into ISAE panels are not identified by the same ASIA identification code. Enterprises are labeled using a codification process specific to ISAE sample surveys which is not consistent with the one used in ISTAT firms' register. Further, ISAE archive of enterprises does not include neither the EIN (Employer Identification Number) nor the VAT code (both available in ASIA), which should allow to uniquely identify any business activity.

This prevents any integration between firm-level information collected by the two Institutions. The lack of the one-to-one relationship between the two datasets involves two main drawbacks. From one side, it could likely lead to a lower consistency of overall statistical information. On the other one, it inhibits the comparison of the two kinds of micro-level statistical information (both

---

1 ISTAT acts as co-ordinator of the National Statistical System (SISTAN), a network which involves the whole Public Administration at central, regional and local levels. It is aimed at assuring homogeneous methods and processes for official statistics.

2 ISAE business surveys have been recently restructured and the firm is selected as the observation unit, cf. Malgarini *et al.* (2005).

quantitative and qualitative) on the same issue (i.e., current assessment of industrial activity, short-run production prospects, etc.), which, if feasible, would allow for additional reliability checks and support a broader interpretation of macro-data.

Our aim is to identify the number as large as possible of business activities common to both archives. This makes effective the possibility of integrating the overall firm-level information collected by ISAE on the manufacturing sector with the one gathered by the Italian national statistical institute. Two different procedures based on matching algorithms are developed. In both methods, the name of the company is the reference information used for string matching.

The experiment consists in combining each occurrence of ISAE register of firms (pattern *P*, hereafter) with the one of the ASIA database (text *T*, subsequently). As it is common to the majority of matching exercises, each algorithm consists of two phases: the pre-processing stage is devoted to re-arrange the data so to make the search process more efficient. The searching phase consists in finding in the text *T* all the occurrences of the pattern *P*, or the specific sequence in which the string has been partitioned.

The basic idea underlying the first algorithm relates to factorizing the pattern *P* into three parts. This partition is obtained through the backward recursive application of the critical factorization method. The subsequent searching phase entails the comparison of the sequence of the three sub-strings in the reference text *T*, from the left to the right. Each sub-string is compared irrespective of the outcome concerning the remaining ones.

The rationale of the second algorithm is to compare each pattern *P* with all the available texts T_i (i=1,…, n) where n is the size of the selected universe. So, the searching phase is of the kind *1:n*. For each pair (P, T_i), all the comparisons between each word in *P* (denoted as a sequence of characters delimited by blanks) and all the n strings contained in the text T_i are considered. The algorithm is based on a shifting rule. The comparison starts from the left to the right on the basis of k jumps, where k is the number of blanks in T_i. Further, to select within the several firms from ASIA register those which are linked to the single pattern *P*, similar contrasts are also carried out with respect to the variable reporting the street address and ZIP code in both archives of enterprises. The sequential matching of both company name and street address is shown to more likely identify the same productive unit in both firms' registers.

In the period in which this analysis is carried out, ISAE register covers about 10,200 company names. It consists of a set of units increasing over time, as it includes both the new companies and the ones excluded from the surveys over past time periods. The matching exercise has been firstly carried out with respect to the universe of manufacturing firms operating in the year 2004 (which

sum to about 600,000 units). After the sequence of two algorithms is run, the firms exactly matched in both archives amount to about 6,800 units. Once firms operating in two additional time periods are accounted for (respectively, in the years 1999 and 2006), more than 1,200 firms are additionally identified (corresponding to about 35% of the unmatched enterprises). We consider these preliminary findings as satisfactory. Overall, about 8,000 units of ISAE register of business enterprises have been exactly matched (78% of the total). This suggests that the role of firms' demography should be adequately taken into account to get satisfactory results and, furthermore, it confirms the reliability of the proposed algorithms.

The paper is organized as follows. Section 2 presents some basic string definitions and describes the mechanics of some widely used string-matching algorithms. The structure of the two matching procedures proposed in the paper is described in Section 3. Section 4 summarizes the final results. Section 5 concludes.

2 STRING DEFINITIONS AND MAIN ALGORITHMS

2.1 Definitions

A string x can be considered as a sequence of characters from a given alphabet Σ. Its length, denoted by $|x|$, is defined as the number of characters in the string itself. For instance, x=(FDETEFET) is a string from an alphabet Σ ={D, E, F, T} which length is 8. An empty string is a string which length is zero. The i-th character of the string x is indicated by $x[i]$. The substring $x[i]x[i+1]…x[j]$ obtained from string x is denoted as $x[i…j]$, where $i < j$. As an example, if the string x is FDETEFET, then substring x[2…5] is DETE.

A substring u is a prefix of the string x if there exists a string v (also an empty string) such that $x=uv$. As an example, u=(F, FD, FDE,…, FDETEFET) include all prefixes for x. Similarly, a substring v is a suffix of string x if there exists a string u (also an empty string) such that $x=uv$. For instance, for the same string x, v=(T, ET, FET,…, DETEFET, FDETEFET) are all prefixes of x.

A *period* of a string x is an integer p, $0<p\leq|x|$, such that $x[i]=x[i+p]$ for i=1,2,...,$|x|-p$. For example, with respect to the string x = *aabaabaabaab*, ones get:

aabaabaabaab → period = 3
aabaabaabaab → period = 6
aabaabaabaab → period = 9
aabaabaabaab → period = 12

The smallest period of x is denoted as *per*(x). In the above example, *per*(x) = 3. Given the string x and an integer k, the k-times repetitions of string x is denoted by x^k. For instance, for string x=(FDETEFET) and integer k=2, x^2 = (FDETEFET FDETEFET). Based on the period definition, a string x of length l is *periodic* if *per(x)* $\leq l/2$; otherwise, the string is said to be *non-periodic*.

A string x is called *basic* if it cannot be written as a power of another string. In our example, there exists no string w and no integer k such that $x=w^k$ is satisfied. A string z is a *border* of another string x if there exist two substrings, u and v, such that $x= uz = zv$. In this case, z is both prefix and suffix of x and, additionally, $|u| = |v|$ is the period of x. A reverse of a string x of length l, denoted by x^R, is the mirror image of x such that $x^R = x[l]\ x[l\text{-}1]\ldots\ x[1]$.

A *maximal suffix* is a suffix which is lexicographically maximal with respect to all suffixes of the string x. If x = FDETEFET, the set of its suffixes is {T, ET, FET,…, DETEFET, FDETEFET}, but the set of its sorted suffixes (based on the lexicographic order) is (TEFET, T, FET, FDETEFET, ETEFET, ET, EFET, DETEFET). According to the definition, the *maximal suffix* of the string x is denoted as *MaxSuf*(x) = (TEFET). If *MaxSuf*(x) = x, the string is said to be *self-maximal*. For instance, in the case of x = TTEETE, its *maximal suffix* is (TTEETE) and hence x is *self-maximal* [3].

A maximal suffix of a string has many properties. First of all, if the condition *MaxSuf* (x) ≤ $|x|/2$ is satisfied, then the maximal suffix of x is called the *short* maximal suffix. If x = ABCDDA, the maximal suffix DDA satisfies the relation *MaxSuf* (x) ≤ $|x|/2$, and DDA is short-maximal for x. Secondly, a relevant property (used in the "Two-Way" algorithm) is the following. Let consider a string x = (u, v) where v = *MaxSuf*(x) and u is the remaining part of x. If u is not-empty, then the maximal suffix v does not overlap with respect to u. More extensively, none suffix of u equals a prefix of v [4].

As it will be clear below, exact string matching algorithms generally consist of two main parts: the pre-processing and the searching phase. The former broadly consist of a specific treatment of the text to be matched and is

[3] The *maximal suffix* and *the period* have important relations. *Maximal suffix* strings allow us to compute the period quickly.

[4] For a proof of this property, cf. Chen-Cheng (2008).

preliminary to the *true* procedure. It plays a very important role in both reducing the time of running and maximizing the number of matched cases.

The string decomposition $x = (u, v)$ is an example of *string factorization*. It is widely used in the pre-processing phases of many algorithms. The pre-processing phase of the first algorithm developed in this paper basically consists in selecting an efficient *factorization* for each string x. Our criterion for string decomposition is based on an extension of the Critical Factorization method proposed by Chrochemore and Perrin (1991).

Let (u, v) be a factorization of x and suppose that an overlap exists between u and v. It means that a substring w occurs at both sides of the cut between u and v with a possible overflow on either side. This substring is known as *repetition* in (u, v). The smallest length of a repetition in (u, v) is called the *local period* and is denoted by $r(u, v)$. Each factorization (u, v) of x has at least one repetition[5].

A factorization (u, v) of x such that $r(u, v)=per(x)$ is called a *critical factorization* of x (see Charras and Leqroc, 2004; Crochemore and Ritter, 1994). The critical factorization of string x is computed as follows: let "$\leq$" and "$\subseteq$" be the alphabetic orders induced, respectively, by the conventional and inverse order of the alphabet Σ. Let v and v' be the maximal suffix of a nonempty pattern x, respectively, for "$\leq$" and "$\subseteq$" orders. The identity $x = uv = u'v'$ is always satisfied. If $|v| < |v'|$, then (u, v) is a critical factorization, otherwise the pair $(u'v')$ is selected[6]. For example, in the case of the string used above, x=FDETEFET, the factorization according to "$\leq$" is x = (**FDE**, ***TEFET***) while x = (**F**, ***DETEFET***) is the one obtained on the basis of the "$\subseteq$" order. Thus, the first factorization is critical.

2.2 Some interesting algorithms

In the following, widely used algorithms to solve the exact string matching problem are presented according to the characteristics of the pre-processing phase. Based on a general terminology, we define as pattern P the string that we are interested in comparing with a given sequence of characters we denote as text T.

---

5 It can be easily seen that $1 \leq r(u,v) \leq |x|$.

6 In other terms, to compute the critical factorization (x_l,x_r) of string x, we first compute the maximal suffix z for the order "$\leq$" and the maximal suffix $\check{z}$ for the reverse order "$\subseteq$". Then the ordered sequence (x_l,x_r) is chosen such that $|x_l|=max\{|z|,|\check{z}|\}$.

The Brute Force algorithm is one of simplest algorithms for exact string matching. In the general case of a given a text *T* of length *n* and a pattern *P* of length *m*, the algorithm compares all *n*-*m*+1 locations in text *T* which possibly match those of *P*, skipping the pattern to the right one step at time. It does not allow for a pre-processing phase and basically compares the pattern *P* with every substring of *T*.

It should be considered that more efficient exact string matching algorithms avoid comparing pattern *P* with all substrings of *T*. In almost every exact string matching algorithm, a window in *T* with the same size of the pattern *P* is opened. If the window exactly matches the pattern, a combination is found. Otherwise, it will be necessary to move the pattern to the right and, there are many methods to determine the number of steps to slide the pattern.

One of this procedure is the Suffix-to-prefix rule, in which the shifting occurs so that the longest suffix of the window in *T* matches a prefix of *P*. If the suffix coincides with the window, an exact matching is found and no sliding happens.

In the Two Way algorithm (Crochemore and Perrin, 1991), the pattern *P* is factorized in two parts x_l and x_r such that $x=x_l x_r$. This is the pre-processing phase of the algorithm, which consists then in choosing a good *factorization* ($x_l x_r$). To compute the critical factorization (x_l,x_r) of *P* the maximal suffix *z* of *P* is first computed for the order $\leq$, and then the maximal suffix ž for the reverse order $\subseteq$. Then (x_l,x_r) is chosen such that $|x_l|=max\{|z|,|ž|\}$. The searching phase of the Two Way algorithm consists in first comparing the character of x_r from left to right, then the character of x_l from right to left. When a mismatch occurs in scanning the *k*-th character of x_r, then a shift of length *k* is performed. When a mismatch occurs when scanning x_l, or when an occurrence of the pattern is found, then a shift of length *per*(x) is performed.

The goal of the encoding algorithm is to shorten both lengths of pattern *P* and text *T* (Apostolico *et al.*, 1997). In the general case, both lengths of the encoded pattern *P* and text *T* are much shorter than the original ones. This significantly improves the searching time. The algorithm sequence is as follows:

Step 1: Choose a character *x* whose frequency in *P* is as small as possible and is not included in the following cases: *i*) *x* only appears once in *P*, *ii*) *x* appears twice and adjacent in *P*.

Step 2 – encoding phase: Let *P*' be the longest substring of *P* that begins and ends with *x*. If the length of the substring between two nearest *x* is greater than 0, the substring is replaced with its length. As an example, let *P* = *aacdstaaceedrcab* and the character *c* is selected. Then *P*'=*cdstaaceedrc* and, in encoded form, P'_{en} = c5c4c. The same method applies to encode *T*.

Step 3 – searching phase: Use the Quick Search Algorithm to find all occurrences of P'_{en} in T'_{en} from left to right. Record the position i's where P'_{en} occurs in T'_{en}.

Step 4 – examination phase: For every position i, align P with the substring in T to examine whether there exists an exact matching or not.

A way to encode the pattern P is to use the value of the character which is defined in ASCII code (American Standard Code for Information Interchange). For example, *Val*(a)=97, *Val*(b)=98, *Val*(c)=99.

An exact string matching algorithm based on this encoding method is divided into two stages as follows. In the pre-processing phase, to each character is attributed a value (based on the ASCII table), and the total value of the string S is computed, $TVal(S) = \Sigma(s_i)$. As an example, if S = abcd, $TVal(S)$ = 489.

In the searching phase, $TVal(T[i+j\ldots i+m-k])$ is computed for $1 \leq i \leq n-m$, k=1 when $i < n-m$ (otherwise k=0) and j=1 when $i=n-m$ (otherwise j=0), where n is the length of T and m the size of pattern P. If $TVal(T[i+j\ldots i+m-k]) = TVal(P)$, then i is a possible position to have an exact matching. Otherwise, the window is ignored and the sliding window is shifted by only one position to the right. When $TVal(T[i+j\ldots i+m-k]) = TVal(P)$, then a specific algorithm is used to check if this position is an exact matching position or not.

3 STRING MATCHING ALGORITHMS APPLIED TO FIRMS' REGISTERS

In this section, we describe the structure of the procedures developed for the matching of strings coming from two different registers of enterprises. The first archive is the one managed by ISAE, which is the national institution in charge of carrying out business tendency surveys (BTS) in Italy. The second one consists of the official statistical repository concerning the universe of productive units operating in the country in a given year (ASIA). This is administered by the Italian NSI (ISTAT) and is updated on a yearly basis.

A number of issues are to be pointed out. Firstly, ISAE is part of the Italian National Statistical System (SISTAN), which is coordinated by ISTAT. Secondly, ASIA is the source of data relating to the identification of enterprises participating to the whole of ISAE business surveys on the manufacturing

sector. As a result, in both the enterprise archives, the firm is considered as the basic unit of observation. In spite of this, all firms selected to enter into ISAE panels are not identified by the same ASIA identification code. Enterprises are labeled using a codification system specific to ISAE surveys and not consistent with the one used in ASIA.

ISAE archive of enterprises does not include neither the EIN (Employer Identification Number) nor the VAT code (both available in ASIA), which should allow to uniquely identify any business activity. Therefore, no exact matching at firm-level data is feasible. The result may be of a lower consistency of overall statistical information. Furthermore, it inhibits the comparison of two kinds of information (*hard* and *soft* micro-data) on the same issue (i.e., current assessment of industrial activity, short-run production prospects, orders, etc.); if feasible, it would allow for additional reliability checks and support a broader interpretation of macroeconomic figures.

To tackle the above drawbacks, our aim is to assign to each firm entering into ISAE business surveys the corresponding identification code used in ISTAT enterprises' register. This makes effective the possibility of integrating the overall firm-level information on the manufacturing sector collected by ISAE with the one gathered by the Italian national statistical institute. To identify the larger number of business activities common to both archives, two different procedures based on string matching algorithms are developed. In both cases, the name of company is the reference information used for string matching.

A plausible starting point is to consider that each firm must be uniquely identified by its company name. ISAE panels of enterprises are extracted from the ISTAT firms' register. This should allow for the perfect comparability of both datasets, and no need for string matching algorithms should be necessary. As a matter of fact, ISAE register of enterprises is currently subject to various updating procedures (i.e., following activities of merger and acquisitions between firms), which could lead to changes in the name of the manufacturer, legal form, street address and localization of the productive unit. These revisions are undertaken independently of the bringing up-to-date of the official register ASIA of business enterprises operating in a given year. Thus, each company is likely to be included in both archives even though the corresponding name would not be reported in the same way.

Several ways are used to shorten specific part of the company name, such as the kind of activity (*Ind.* in spite of *Industry*), the legal form (*SPA* rather than *S.P.A.*, *Spa*, *spa*), the name(s) of the entrepreneur(s) (*Mario Rossi & Company* rather than *Mario Rossi & C.*), and the street address (*Strada* instead of *Str.*), so that company information in both registers tend to be fairly similar rather than exactly the same.

Due to the presence of so many heterogeneities, to achieve pattern matching between each occurrence of ISAE register (*P*) and the ASIA database (*T*) is the main problem to be dealt with. It has been solved using specific tools based on string-matching algorithms. As it is common to the majority of matching exercises, each algorithm consists of two phases: the so-called pre-processing stage, which is devoted to re-arrange the data, usually collecting useful information on the pattern *P* and to make the search process more efficient; the searching phase, which is to find in the text *T* all the occurrences of the pattern *P*, or the specific sequence in which the string has been partitioned.

3.1 First algorithm

The basic idea underlying the first algorithm consists in factorizing the pattern *P* into three parts, *z*, *y*, *v* (where *u*=*zy* and *P*=*uv*). The subsequent searching phase entails the comparison of the sequence of the three sub-strings *z*, *y*, *v* from the left to the right in the reference text *T*. Each sub-string is compared regardless of the outcome concerning the remaining ones.

The pre-processing phase is peculiar to this procedure as it includes a number of processes to be performed before the searching stage. The first concerns string normalization, which consists in uppercase all alphanumeric characters and, further, in removing all blanks and special fonts, including punctuation marks. In the second step, the two archives we are dealing with are disaggregated on a regional basis and, subsequently, aligned one below the other. Thirdly, all company names are sorted by alphabetical order (see Tab.1)[7].

In a successive step, a subsequence of characters *P*' from the pattern *P* is selected. We adopt a criterion similar to the data encoding approach used to solve the exact string matching problem described in the previous section. Instead of selecting a substring from all characters between the first and the last selected item *x*, in our case the first |*P*'| characters are selected within the whole pattern *P*, moving from the left to the right.

The rationale relies on the assumption that, in the case the same business enterprise is part of both archives, the two corresponding names (if different) are more likely to vary in their final part (right-end side). As in the encoding method, to consider substrings for the matching of the initial part of the sequence of characters would improve the speed of searching process. In our application, the selected length of *P*' is 15 characters. The criteria adopted to identify the size of *P*' are described at the end of this section.

[7] The overall procedure is written using Visual Basic for Application (VBA).

Tab. 1 Firms' registers appended and alphabetically sorted

Source register	Row number	Company name
…	…	…
ASIA	n-2	ACEM DI BINELLI MARIA
ASIA	n-1	ACHILLI DORIANO & ZERBINI GUIDO SNC
ISAE	n	ACHILLI DORIANO & ZERBINI GUIDO SNC
ASIA	n+1	ADEA EDIZIONI DI MAGGIO MAURO
…	…	…
ASIA	m-3	CELASCHI SPA
ASIA	m-2	CENCI S.R.L.
ASIA	m-1	CORRADINI AUTOGRU S.R.L.
ISAE	m	CORRADINI SPA
ASIA	m+1	CORTI GIAN PIETRO
…	…	…
ASIA	k-1	RDB SPA
ISAE	k	RDB SPA
ASIA	k+1	REALFOOD 3 S.R.L.
…	…	…

The final stage of pre-processing is to split the pattern *P*' into three parts instead of the usual two, as in the 'Two-Way' algorithm. This partition is obtained through the recursive application of the critical factorization method where its second run applies to pattern *P*' net of the critical suffix computed in the first step, *P*'/*v*. On the assumption that the critical factorization in the first stage is (*u*,*v*), in the successive phase the same procedure applies to the prefix *u*. It is a *non-empty* substring and cannot overlap with *v*, according to the properties of the factorization theorem.

Specifically, the second recursive cutting provides a maximal suffix *y* for *u*, so that *u*=(*z*,*y*) is critical and *z* is a prefix for *P*'. The sequence of three substrings such that *P*'=*zyv* is held fixed except when a *singular* decomposition arises. As the maximum suffix must always include a sequence of characters, the case of singularity only concerns *u*=ø (in which case, *P*'=*v*) or *z*=ø (*P*'=*yv*). On such occasions, the first non empty substring is shifted to the left replacing the empty one[8].

The results of the overall pre-processing steps are organized in a table in which both factorized patterns *P*' (ISAE register) and texts *T* (ISTAT register) are sorted in alphabetical order. It represents the starting point of the searching

[8] Let string *x*=ab. If Σ={a,b} then *u*= ø and *v*=ab. If Σ={b,a} then *u*=a and *v*=b. So, according to the twice application of the critical factorization, we obtain z=ø, *y*=a and *v*=b. It is to avoid choosing the fault decomposition that is *u*=ø and *v*=ab. This is also a rationale for the backward sequential cut used in the pre-processing phase, since critical factorization picks the shorter maximal suffix, thus increasing the possibilities that the other two substrings would be not empty.

phase. Specifically, given the alphabetical sorting, the searching process concerning each pattern *P*' is always limited to the texts *T* immediately previous and subsequent the pattern itself. If ISAE pattern is at position *n* of the table, the comparison occurs, respectively, with the text *T* at position *n*-1 and the one at position *n*+1. In the case of two patterns at row *i*=*n, m*, for *m*>*n*, no comparison occurs between P_i' and T_j if $j \in [n+2, m-2]$, where *j* is the row of the text *T*, for $j \neq i$ (see Table 1). Furthermore, when two consecutive patterns at positions *n*, *n*+1 are considered, then evaluation is only performed for the pairs (P_n',T_{n-1}) and (P_{n+1}',T_{n+2}). This set of rules noticeably improves the speed of the searching and follows as results of the pre-processing stage.

In the second stage, the searching process starts in correspondence of each pattern *P*' factorized in the pre-processing table. A window of size |*P*'| (starting from the left) is opened in the text *T* positioned immediately previous and subsequent the pattern *P*'. Each window is successively separated into three parts with exactly the same dimension of substrings *z*, *y*, *v* of pattern *P*', respectively (see Table 2). As an example, ISAE pattern *P*' is at position *n* of the table: a substring of the same length is selected in both texts *T* positioned, respectively, at *n*-1 and *n*+1 and, sequential cuts also applies on the basis of |*z*|, |*y*| and |*v*|.

The comparison is such that the sequence of each of the three substrings *z*, *y*, *v* is held fixed and, each of them is compared with the one in *T* which has the same position. Thus, the evaluation is performed with respect to each substring and it is independent on the outcome of the other contrasts[9].

When each pair of substrings (*P*' and the text window *T*, respectively) is identical, exact matching occurs. The correspondence of the individual character for pair of substrings is not accounted for. If a mismatch occurs in the position of at least one item, the whole substring is not matched. Thus, the result is of dichotomous kind (match/does not match).

In other terms, we first determine whether *P*' occurs in *T*. If the answer is "does not match", we are certain that *P* does not occur in that position of *T*. On the contrary, if *P*' occurs in *T*, the probabilities that *P* is matched with *T* increases. It rises as we move from the left-side combined substring up to the (third) right-side matched substring. As discussed before, string heterogeneities grow moving towards to the right side of each text. We assume that the case "does not match" occurs when the prefix *z* of *P*' is not matched. If a perfect

[9] This method can be considered as a simplified version of the "suffix to prefix" rule but without any shifting of the pattern *P*' to the right. In other terms, we are assuming that, in the case of matching, the window in the text *T* exactly matches the pattern *P*' and there is no need to move the pattern to the right.

correspondence for *z* is found in the text *T*, then the more the number of other substrings matched, the greater the probability of overall exact string matching.

Tab. 2 **Backward recursive factorization**

Row number	Company name	Normalized company name	Text window *T*/ Pattern *P'*	Factorization		
				z	*y*	*v*
…						
n-1	ACHILLI DORIANO & ZERBINI GUIDO SNC	ACHILLIDORIANOEZERBINIGUIDOSNC	ACHILLIDORIANOE	ACHILLID	O	RIANOE
				⇑	⇑	⇑
n	ACHILLI DORIANO & ZERBINI GUIDO SNC	ACHILLIDORIANOEZERBINIGUIDOSNC	ACHILLIDORIANOE	ACHILLID	O	RIANOE
				⇓	⇓	⇓
n+1	ADEA EDIZIONI DI MAGGIO MAURO	ADEAEDIZIONIDIMAGGIOMAURO	ADEAEDIZIONIDIM	ADEAEDIZ	I	ONIDIM
…	…	…	…	…	…	…
m-1	CORRADINI AUTOGRU S.R.L.	CORRADINIAUTOGRUSRL	CORRADINIAUT	CORR	ADINI	AUT
				⇑	⇑	⇑
m	CORRADINI SPA	CORRADINISPA	CORRADINISPA	CORR	ADINI	SPA
				⇓	⇓	⇓
m+1	CORTI GIAN PIETRO	CORTIGIANPIETRO	CORTIGIANPIE	CORT	IGIAN	PIE
…						

A scalar is assigned to each comparison. Their sequence is called 'score-array' while the sum of the array elements is the 'score' (Table 3). In the case of matching, the assigned value increases moving from the prefix to the suffix of *P'*: 5 is the score if substring *z* is matched, 7 if *y* is matched and 9 if *v* is

Tab. 3 **Score-arrays and scores**

Score-array	Score	Description
(1, 1, 1)	**3**	None of the strings is matched
(5, 7, 9)	**21**	*z, y, v are matched*
(5, 7, 0)	**12**	*z and y are matched. v is an empty string*
(5, 7, 1)	**13**	*z, y are matched, v is not matched*
(5, 0, 0)	**5**	*z is matched. y and v are empty strings*
(5, 1, 0)	**6**	*z is matched, y is not matched, v is an empty string*
(5, 1, 1)	**7**	*z is matched, y and v are not matched*
(5, 1, 9)	**15**	*z, v are matched, y is not matched*
(1, ., .)	**-**	*"does not match" case*

matched as well. 0 is the value if the comparison includes at least one empty string so that, the score for the comparison of prefix *z* is always different from zero. Additionally, 1 is the score assigned to the comparison in the case of mismatch between any pair of substrings.

The cases in which the prefix *z* of *P*' is not matched are identified by score-arrays including 1 in the first position and, they are labeled as "does not match". As a result, 21, 12 and 5 is the value of the score in the case of exact matching of, respectively, overall substrings, the first two substrings and the prefix of *P*'. Any other result signals the occurrence of a mismatch or an empty string. Some score-arrays and the corresponding value of the score are listed in the Table 3. An example of the results obtained using this algorithm is presented in Table 4.

Tab. 4 **First Algorithm: examples of pattern matching**

Score-array	Score		*z*		*y*		*v*
(5, 7, 9)	**21**	**5**	ACHILLID	**7**	O	**9**	RIANOE
			⇑		⇑		⇑
			ACHILLID		O		RIANOE
			⇓		⇓		⇓
(1, 1, 1)	**3**	**1**	ADEAEDIZ	**1**	I	**1**	ONIDIM
(5, 7, 9)	**21**	**5**	CELA	**7**	SCHI	**9**	SPA
			⇑		⇑		⇑
			CELA		SCHI		SPA
			⇓		⇓		⇓
(1, 1, 0)	**2**	**1**	CENC	**1**	ISRL	**0**	
(5, 7, 1)	**13**	**5**	CORR	**7**	ADINI	**1**	AUT
			⇑		⇑		⇑
			CORR		ADINI		SPA
			⇓		⇓		⇓
(1, 1, 1)	**3**	**1**	CORT	**1**	IGIAN	**1**	PIE
(5, 7, 9)	**21**	**5**	RDB	**7**	SP	**9**	A
			⇑		⇑		⇑
			RDB		SP		A
			⇓		⇓		⇓
(1, 1, 1)	**3**	**1**	REA	**1**	LF	**1**	O

In the use of the above algorithm for the matching of firms' registers, the length of *P*' is based on some empirical evidences that have been obtained using the ISAE archive of normalized company names. We basically focus on the descriptive statistics obtained from the distribution of the patterns' length. Their average length is of 15 characters and we take this as the reference size for *P*'. It is interesting to consider that the median is close but slightly lower than the sample mean while the mode is somewhat below 10 characters.

On the basis of some empirical proofs carried out respect to a reference region, the size of *P*' at 15 characters has proved to get the maximum number of successful matching, as shown in Table 5. Further, it emerges that the number of occurrences in which the suffix of the pattern is matched increases as the length of *P*' reduces, even though with a higher number of discrepancies (reaching 24% in the case of score=21, as the length of the pattern decreases from 15 to 10). On the contrary, as the length of *P*' rises, the first part of the pattern is more likely to be found in the text. Overall, for the pilot sample we are considering, the total number of exact matching is (locally) maximized if pattern length is set out at 15 characters.

Tab. 5 Successful matching as a function of pattern's length *P*'

|*P*'|	O	M	O – M	S	O	M	O – M	S	O	M	O – M	S	T	R
	Score: 21				Score: 12				Score: 5					
10	469	356	113	24,09	59	50	9	15,25	5	4	1	20,00	533	410
15	**266**	**259**	**7**	**2,63**	**123**	**123**	**0**	**0,00**	**39**	**39**	**0**	**0,00**	**428**	**421**
21	170	156	14	8,24	131	120	11	8,40	91	79	12	13,19	392	355
27	110	100	10	9,09	137	125	12	8,76	131	111	20	15,27	378	336

Note: |*P*'|: Pattern *P*' length; O: Output; M: Matched cases; S: % of mismatch; T: Overall output; R: Total number of matched patterns.

As described above, this procedure is entirely based on the variable "company name" and on the assumption that the sequence of characters, for each name of the enterprise, is approximately identical in both firms' registers. It is likely that some pairs of (*P*, *T*) in our dataset might not meet this requirement. One of these cases occurs when the full company name is not recorded in exactly the same order in both business registers (i.e., *Mario Rossi & C.* vis-à-vis *Rossi Mario & C.*).

3.2 Second algorithm

The goal of the second algorithm is to deal with all cases in which the source of the mismatch is the different order of the substrings for the company names to be compared. The rationale of the second algorithm is to match each pattern *P* with all the available texts T_i (i=1,…, n). So, the searching phase is of the kind *1:n*, and this noticeably increases the time of searching. For each pair (*P*, T_i), all the comparisons between each word of *P* and all the words contained in the text T_i are considered. We label as *word* any sequence of characters delimited by *blanks* and which length is more than 3 characters.

A *word* is analogous to the substring as defined in the first algorithm but, in this second procedure, it is obtained on the basis of the occurrence of blanks inside each string (both in the pattern *P* and the text *T*), rather than using any factorisation method. Each substring of *P* (delimited by blanks) is then compared with each substring of *T* (delimited by blanks). Thus, the pre-processing phase does not account for any preliminary normalization of the strings, except to restrict to one the number of blanks between two consecutive *words*.

Tab. 6 Selected code for the second string-matching algorithm

```
For rigae = 2 To FineRE
For rigad = 2 To FineRD

If Len(Trim(myVal1)) = 0 Then
sp1 = 0
Else
sp1 = Len(Trim(myVal1)) - Len(Replace(myVal1, " ", "")) + 1
End If
If Len(Trim(myVal2)) = 0 Then
sp2 = 0
Else
sp2 = Len(Trim(myVal2)) - Len(Replace(myVal2, " ", "")) + 1
End If

For a = 1 To sp1
        S1 = Split(myVal1, Chr(32))
        Str1 = S1(a - 1)
If Len(Str1) <= 3 Then GoTo Piccola

            For b = 1 To sp2
                S2 = Split(myVal2, Chr(32))
                Str2 = S2(b - 1)

If Len(Str2) <= 3 Then GoTo piccolaa
If Str1 = Str2 Then
If ActiveCell = "" Then
…
End If
End If
piccolaa:
                 Next b
Piccola:
Next a

Next rigad
Next rigae
```

Furthermore, the second algorithm accounts for a kind of shifting of substrings of *P* with respect to T_i (from the left to the right): the swinging is based on the positions of blanks within the text *T*[10]. The shifting rule works as follows: each substring of the pattern to be found in the text is positioned for comparison at position *t*+1 if the blank in text *T* occurs at position *t*. The comparison starts from the left to the right on the basis of *k* jumps, where *k* is the number of blanks in *T*. A sequence of the Visual Basic code is presented in Table 6.

As a general outcome, this method would lead to select a large number of firms from the ASIA business register which potentially match with the individual pattern *P*. To additionally select within the list of firms picked out from the first step, the same procedure applies for comparisons to the string variable reporting the street address and ZIP code in both archives of enterprises. The sequential matching of both company name and street address is shown to more likely identify the same productive unit in both firms' registers. An example of the final results using the second string-matching algorithm is presented in Table 7, where substrings of the pattern *P* (bold font) are those to be found in the text *T* (underlined font).

Tab. 7 Examples of string matching results using the second algorithm

Outcome	Register	Company name	Street address
	ISAE	**VALOBRA VIRGILIO**	VIA **CARDUCCI**
Not matched	ASIA	PIZZERIA VALOBRA S.N.C. DI PATTARO SABINA E ROSANNA	V. VALOBRA 175
Matched	ASIA	DITTA VIRGILIO VALOBRA DI GUGLIELMO VALOBRA	V. CARDUCCI 5 7 SS
	ISAE	**OLCESE RICCI** SRL	VIA **CAORSI**
Matched	ASIA	OLCESE & RICCI SRL	V. CAORSI 49 A
Not matched	ASIA	OLCESE SPA	V. PIRANESI 44 A
Not matched	ASIA	OLCESE & C. SNC	V. ROMA 1
Not matched	ASIA	OLCESE DANILO	V. CESAREA 89 R
Not matched	ASIA	OLCESE ILARIO MAURILIO	V. DUCA CANEVARO 6
Not matched	ASIA	G.I.O.M. DI OLCESE MARIO	V. XXV APRILE 37 1
Not matched	ASIA	OLCESE IMMOBILIARE S.R.L.	VIA SANTO SPIRITO 14
Not matched	ASIA	PASTICCERIA ROBERTO DI OLCESE MARINA & C. S.A.S.	VIA CESAREA 2/32

[10] We include the blank as an additional character in our alphabet Σ.

4 RESULTS

In this section, final results concerning the overall matching procedure applied to ISAE and ISTAT firms' registers on the manufacturing sector are presented. It should be noted that the ISAE register is systematically updated over time. This process is mainly due to the usual demography of enterprises. Additionally, on some occasions, firms are no longer available to take part in the surveys. In both cases, firms leaving the reference sample are replaced by new productive units. The latter are randomly drawn from a sub-sample of the ISTAT register of firms which characteristics are similar to the discarded businesses (in terms of the sector of economic activity, size and localization).

As a result, the ISAE firms' archive consists of a set of enterprises increasing over time, as it includes both the new companies and the ones currently excluded which entered into ISAE surveys carried out in the previous years. In the period in which this analysis is carried out, ISAE register includes about 10,200 company names. At first, the matching exercise is managed with respect to the universe of manufacturing firms operating in Italy in the year 2004, which sums to about 600,000 units.

After the sequence of two algorithms is run, all the matched patterns are checked, so to validate the outcome that each pair of company names identifies the same enterprise. As a result, the ISTAT identification code is uniquely associated to the ID used in ISAE firms' register. Overall, the manufacturing firms which are exactly matched in both archives amount to about 6,800 units.

At this stage of the analysis, we do not find any correspondence for about 3,400 items in the ISAE archive of enterprises. Several reasons are possible. First of all, we find that mis-reporting of the company names occasionally affects ISAE business database, so that also the second algorithm might not be able to identify any correspondence. Secondly, firms' demography significantly impacts the results of the matching procedure, as a number of issues other than plants closure may be considered under this topic, as the change of the company name, transformation of legal form, variation of street address and localization of the company.

To mostly account for the relevance of birth and mortality rates along the first half of 2000's, we attempt to combine the remaining unmatched firms with the ones included in the ASIA releases corresponding to two different time periods, respectively the years 1999 and 2006. We consider that the preliminary findings are satisfactory, since more than 1,200 firms are exactly identified (about 35% of the unmatched enterprises).

Overall, about 8,000 units of ISAE firms' register are exactly matched (78% of the total items). This result suggests that the role of firms' demography should be adequately accounted for to obtain a satisfactory outcome and, furthermore, it confirms the reliability of the proposed algorithms. This allows for a broader integration of ISAE microdata with the corresponding firm-level information collected by ISTAT (e.g., PRODCOM databases, external trade, etc.). Furthermore, a re-organization by sectors of ISAE firms' register, consistent with several ATECO classifications, can now be achieved.

Additionally, some structural characteristics of ISAE sampled firms (and of the ISAE register), not feasible in the absence of our research, can be now accounted for. We focus on a restricted set of structural aspects of firms taken from the several ASIA releases: they concern the number of local units per each enterprise, the overall number of employees in local units, the legal form and the time of start up of the enterprise. In the following, we present some descriptive statistics which relates to the subset of firms participating, for at least for one time, in ISAE business surveys along the year 2007.

Concerning the number of local units, 89% of the firms taking part to the survey do not present any related local unit. Within this group, 28.4% are localized in the Centre of Italy, 30% in the North-East, 17.4% in the North-West and the remaining part in the South. When enterprises with at least one plant are concerned, the average number of local units is 2.6 per each firm, which increases to about 3 in the North of Italy and reduces to 1.7 in the Southern regions.

At the establishment level, the number of employees amounts, on average, to about 137. For each enterprise, 56% of the total employment is engaged at the establishment level. This ratio increases to 60% in the North-West of Italy (where the average number of employees in local units rises to 248 units), while it results below the mean in the remaining regions (about 54%). Further, it generally emerges that the number of employees engaged at establishment level increases with the share of local units. The incidence in terms of the overall number of employees shows the same behaviour.

As regard the legal forms, the public limited companies cover about 70% of the whole sample of firms. Its incidence is larger in the North-East regions (30%), lower in the North-West ones (20%). The share of limited, general and commercial partnership amounts to 25% of the overall sample, and it presents its greater incidence in the Centre and North-East regions (more than 25%). If the total number of employees is accounted for, the stock companies show the greater size (101 employees, on average) followed by the cooperative societies (58 employees). Firm size for all the other legal forms is substantially below the 15 employees.

As the date of starting-up is available, we can approximately infer the age of the enterprise. On average, with regard to the 2007 sampled units, the age is of about 29 years, where the older companies are localized in the North-West of Italy (34 years) and the younger ones in the Centre and South regions. Generally, it emerges that the years of activity and the number of local units is positively correlated. Based on the 2007 sample data, the subset of enterprises aged 29 present 1 local unit; the age of businesses rises up to 57 in the case of firms including 10 local establishments. Similar relationships could also be found between the age and the size of the firm (Jovanovich, 1982; Harho *et al.* 1998).

5 CONCLUSIONS

In this paper the procedures developed for the matching of strings from two different registers of Italian enterprises are described. The first archive is the one managed by ISAE (Institute for Studies and Economic Analyses). The second one consists of the official statistical repository concerning the universe of productive units administered by the Italian National Statistical Institute.

Our aim is to identify the number as large as possible of business activities common to both archives. This allows for the possibility of integrating the overall firm-level information collected by ISAE with the one gathered by other Institutions. The experiment consists in combining each occurrence of ISAE register of firms with the one of the ASIA database.

Two different procedures based on matching algorithms are developed. The basic idea concerning the first algorithm consists in factorizing the pattern *P* into three parts. The searching phase entails the comparison of the sequence of the three sub-strings from the left to the right in the reference text *T*. Each sub-string is compared regardless of the outcome concerning the remaining ones. The goal of the second algorithm is to deal with all cases in which the source of the mismatch is the different order of the substrings in the company names. So, the searching phase is of the kind *1:n* and this noticeably increases the time of searching. Comparisons are performed for each sequence of characters delimited by *blanks* in the ISAE company name. The evaluation starts from the left to the right on the basis of a number of jumps equal to the number of *blanks* in the reference text *T*.

After the sequence of two algorithms is run, the firms exactly matched in both archives amount to about 6,800 units. Considering also those identified in two additional time periods (respectively, 1999 and 2006), about 8,000 units of ISAE register of business enterprises have been exactly matched (78% of the total). This suggests that the role of firms' demography should be adequately taken into account to get satisfactory results and, furthermore, it confirms the reliability of the proposed algorithms. Some structural characteristics of ISAE sampled firms, not available in the absence of our research, can be now accounted for.

REFERENCES

Apostolico, A., Landau, G.M. and Skiena, S. (1997), "Matching for run-length encoded strings", Journal of Complexity, Vol. 15, 1997, pp. 4-16.

Charras, C. and Lecroq, T. (2004), "Handbook of Exact String Matching Algorithms" King's College London Publications.

Chen-Cheng, Y. (2008), "A Combination of Berry Ravindran Algorithm and Two Way Algorithm for Exact String Matching", Degree Thesis, Computer Science and Information Engineering Department National Chi-Nan University.

Chrochemore, M. and Perrin, D. (1991), "Two-Way String-Matching", Journal of the ACM, 38(3): 651-675.

Crochemore, M. and Ritter, W. (1994), "Text Algorithms", Oxford University Press.

Harho, D., Stahl, K. and Woywode, M. (1998), "Legal form, growth and exit of West German firms. Empirical results for manifacturing, construction, trade and service industries", *Journal of Industrial Economics*, 46(4):453 488.

Jovanovich, B. (1982), "Selection and the evolution of industry", Econometrica, Vol. 50, No. 3, May, 649-70.

Malgarini, M., Margani, P. and Martelli, B. (2005), "Re-engineering the ISAE Manufacturing Survey", ISAE Working Paper No. 47.

Working Papers available:		
n. 75/07	R. BASILE	Intra-distribution dynamics of regional per-capita income in Europe: evidence from alternative conditional density estimators
n. 76/07	M. BOVI	National Accounts, Fiscal Rules and Fiscal Policy Mind the Hidden Gaps
n. 77/07	L. CROSILLA S. LEPROUX	Leading indicators on construction and retail trade sectors based on ISAE survey data
n. 78/07	R. CERQUETI M. COSTANTINI	Non parametric Fractional Cointegration Analysis
n. 79/07	R. DE SANTIS C. VICARELLI	The “deeper” and the “wider” EU strategies of trade integration
n. 80/07	S. de NARDIS R. DE SANTIS C. VICARELLI	The Euro’s Effects on Trade in a Dynamic Setting
n. 81/07	M. BOVI R. DELL’ANNO	The Changing Nature of the OECD Shadow Economy
n. 82/07	C. DE LUCIA	Did the FED Inflate a Housing Price Bubble? A Cointegration Analysis between the 1980s and the 1990s
n. 83/07	T. CESARONI	Inspecting the cyclical properties of the Italian Manufacturing Business survey data
n. 84/07	M. MALGARINI	Inventories and business cycle volatility: an analysis based on ISAE survey data
n. 85/07	D. MARCHESI	The Rule Incentives that Rule Civil Justice
n. 86/07	M. COSTANTINI S. de NARDIS	Estimates of Structural Changes in the Wage Equation: Some Evidence for Italy
n. 87/07	R. BASILE M. MANTUANO	La concentrazione geografica dell’industria in Italia: 1971-2001
n. 88/07	S. de NARDIS R. DE SANTIS C. VICARELLI	The single currency’s effects on Eurozone sectoral trade: winners and losers?
n. 89/07	B.M. MARTELLI G. ROCCHETTII	Cyclical features of the ISAE business services series

Working Papers available:		
n. 90/08	M. MALGARINI	Quantitative inflation perceptions and expectations of Italian Consumers
n. 91/08	P. L. SCANDIZZO M. VENTURA	Contingent valuation of natural resources: a case study for Sicily
n. 92/08	F. FULLONE B.M. MARTELLI	Re-thinking the ISAE Consumer Survey Processing Procedure
n. 93/08	M. BOVI P. CLAEYS	Treasury v dodgers. A tale of fiscal consolidation and tax evasion
n. 94/08	R. DI BIASE	Aliquote di imposta sul lavoro dipendente: analisi per figure tipo e con dati campionari
n. 95/08	M. BOVI	The “Psycho-analysis” of Common People’s Forecast Errors. Evidence from European Consumer Surveys
n. 96/08	F. BUSATO A. GIRARDI A. ARGENTIERO	Technology and non-technology shocks in a two-sector economy
n. 97/08	A. GIRARDI	The Informational Content of Trades on the EuroMTS Platform
n. 98/08	G. BRUNO	Forecasting Using Functional Coefficients Autoregressive Models
n. 99/08	A. MAJOCCHI A. ZATTI	Land Use, Congestion and Urban Management
n. 100/08	A. MAJOCCHI	Theories of Fiscal Federalism and the European Experience
n. 101/08	S. de NARDIS C. PAPPALARDO C. VICARELLI	The Euro adoption’s impact on extensive and intensive margins of trade: the Italian case
n. 102/08	A. GIRARDI P.PAESANI	Structural Reforms and Fiscal Discipline in Europe
n. 103/08	S. TENAGLIA M. VENTURA	Valuing environmental patents legal protection when data is not availableby
n. 104/08	P. L. SCANDIZZO M. VENTURA	A model of public and private partnership through concession contracts

Working Papers available:		
n. 105/08	M. BOSCHI A. GIRARDI	The contribution of domestic, regional and international factors to Latin America's business cycle
n. 106/08	T. CESARONI	Economic integration and industrial sector fluctuations: evidence from Italy
n. 107/08	G. BOTTONE	Human Capital: an Institutional Economics point of view
n. 108/09	T. CESARONI M. MALGARINI L. MACCINI	Business cycle stylized facts and inventory behaviour: new evidence for the Euro area
n. 109/09	G. BOTTONE	Education in Italy: is there any return?
n. 110/09	S. de NARDIS C. PAPPALARDO	Export, Productivity and Product Switching: the case of Italian Manufacturing Firms
n. 111/09	M. BOVI R. CERQUETI	Why is the Tax Evasion so Persistent?
n. 112/09	B. ANASTASIA M. MANCINI U. TRIVELLATO	Il sostegno al reddito dei disoccupati: note sullo stato dell'arte. Tra riformismo strisciante,inerzie dell'impianto categoriale e incerti orizzonti di *flexicurity*
n. 113/09	A. ARGENTIERO	Some New Evidence on the Role of Collateral: Lazy Banks or Diligent Banks?
n. 114/09	M. FIORAMANTI	Estimation and Decomposition of Total Factor Productivity Growth in the EU Manufacturing Sector: a Stochastic Frontier Approach