# A system for summary-document similarity in notary domain

Carmine Cesarano, Antonino Mazzeo and Antonio Picariello
*Universitá di Napoli "Federico II", Italy*
*Dipartimento di Informatica e Sistemistica, via Claudio 21, 80125, Naples*
$\{$*carmine.cesarano, antonino.mazzeo, antonio.picariello*$\}$*@unina.it*

## Abstract

*In this paper we propose a methodology to perform a comparison between a legal document and its related hand-written summary. We thus describe the algorithms that verify when a human-produced summary is consistent with its source document. We first analyze both documents in order to extract only the relevant information then we compare such information in order to obtain a measure of correlation indicating the consistence between the two documents. Eventually, we briefly report on the performance of these algorithms.*

## 1 Introduction

Legal informatics research has been active for about 40 years: in particular, the current existence of huge legal text collections in several domain of interest is at the basis of the increasing interest of the scientific community in text information processing and retrieval particularly suited for legal documents. One of the promising research areas relies int the *acquisition of legal knowledge* and in *summarizing techniques* of documents and hypertext structures. The use of Pattern recognition techniques on the sentence level for the identification of concepts and document classification for automatic document description is described in several works, as SCISOR[6] and FASTUS [3]. In the system BREVIDOC, documents are automatically structured important sentences are extracted. These sentences are classified according to their relative importance [4]. From the NLP point of views, legal research concentrates on the automatic description of documents. In particular, the main focuses are: development of thesauri, machine learning for feature recognition, disambiguation of polysems, automatic clustering and neural networks. The most important systems are FLEXICON, KONTERM, ILAM, RUBRIC, SPIRE, the HYPO extension [1] and SALOMON. Automatic summarization and classification of documents have also been sufficiently analyzed [5] [2] [7]. The description of documents is done by *matching documents with a knowledge base*.

In this paper we describe a system that, using text information retrieval techniques, provide a solution for the evaluation of effectiveness of a summary to a given legal document. In order to describe our vision, we briefly introduce a motivating example. Let us consider the Italian juridic domain, and in particular the notary one: a notary is someone legally empowered to certify the legal validity of a document. Just to give an example in real estate market, in some countries such as Italy, when someone has the intention of buying a property – such as houses, pieces of lands and so on – a notary document, certifying the property transaction from an individual to another one, is signed. Successively, the notary has to accomplish some bureaucratic issues, and one of these is to send the signed *document* together with a *summary* to a National Conservatory. This organization uses the provided summary as an index for consultation issues and stores the signed notary document in an internal information system. When the notary document and the summary are sent to the National Conservatory, an officer is charged to check the correspondence between the summary and the document in order to verify if they contain the same juridic relevant information. We explicitly note that this checking operation becomes very time consuming when the number of documents/summaries becomes huge. In the previously described process, it is of a crucial importance that: i) the provided summary is well written and ii) above all it contains the same information of the original document and iii) it contains all the relevant information useful to facilitate the retrieving process. Note that the summary and the legal document are written in natural notary language and the same concepts are expressed with different words and different sentence structures in both documents.

In this paper, we describe a system that, using text information retrieval techniques, analyzes the notary document and the related summary and expresses a *similarity score* between the two documents. If the similarity value is above a given threshold the summary is considered valid, if the score is below to another threshold the summary is considered not valid and finally if the score is between the two thresholds, a human judgment is required. The paper is organized as follows: section 2 describes the theoretical background, section 3 presents a system overview, section 4 describes the main algorithms. The experimental results are reported in section 5; some conclusions and future works are reported in 6.

## 2 Theoretical Background

Let us give a formal description of the problem we are discussing in this work.

**Definition 2.1** (Notary document). *A notary document i.e. act is a set of attributes and their corresponding values:*

$$\mathcal{A} = \{< Attributes_i, Values_i >\}, i \in [0, N] \quad (1)$$

*each $Attributes_i$ being a concept contained in the legal document.*

A summary is a subset of significant attributes of a notary document:

**Definition 2.2** (Summary). *A summary is a subset of a notary document,*

$$\mathcal{S} = \{< Attributes_k, Values_k >\} \subset \mathcal{A} \quad (2)$$

$k \in K$, $K \subset [0, N]$.

Note that some of these attributes are necessary to understand what the notary document describes and some other attributes describe details that could be missed in the summary. Moreover the same attributes could be expressed using different words used to express the same concepts. In this way, the relation $\mathcal{S} \subseteq \mathcal{A}$ is always satisfied. By the way, in some applications, as described in the previous section, two kinds of documents are provided and we have to verify if $\mathcal{S} \subseteq \mathcal{A}$. The problem is not simple to solve for a variety of reasons: i) it is possible to have different *names of attributes* in $\mathcal{S}$ and in $\mathcal{A}$, with the same semantic content; ii) it is possible to have different but *similar values of attributes* in $\mathcal{S}$ and in $\mathcal{A}$.
For these reasons, we can provide the following problem:

**Definition 2.3** ($\mathcal{S}$ and $\mathcal{A}$ matching problem). *Let us consider two documents $\mathcal{A}$ and $\mathcal{S}$. A $\mathcal{S}$ and $\mathcal{A}$ matching problem consists into finding the grade of information content of $\mathcal{S}$ that is contained in $\mathcal{A}$.*

We first define a metric $\mu$ that measures the distance between the two collections $\mathcal{A}$ and $\mathcal{S}$, as follows:

**Definition 2.4** (($\mathcal{S}$, $\mathcal{A}$)Distance). *Let us consider a notary document $\mathcal{A}$ and a summary $\mathcal{S}$; the distance between$\mathcal{A}$ and $\mathcal{S}$ is the function:*

$$\mu : (\mathcal{A}, \mathcal{S}) \rightarrow [0, 1] \quad (3)$$

Note that in this model, the value "0" is reached when the document-summary couple are totally different and the value "1" is obtained when the document-summary couple shares exactly the same concepts.

The $\mu$ function may be obtained in a variety of ways. In particular, we can calculate the total score

$$\mu = \frac{1}{|M|} \sum_i \alpha_i \mu_i \quad (4)$$

where $\alpha_i$ is a weight, ranging in the interval [0,1], associated to each attribute for a given type of notary document, and $M$ is a normalizing factor.
Each $\mu_i$ is calculated as described in the following: let us consider two couples $\langle Attiribute_i, Value_i \rangle$ and $\langle Attribute_j, Value_j \rangle$. Whenever is satisfied that $Attribute_i = Attribute_j \wedge Value_i = Value_j$ , $\mu_i = 1$; $\mu_i \in ]0, 1[$ if the attributes are equal and some differences are encountered in the related values. Alternatively, two possible solutions may be provided: a) to consider the attribute values as different ($\mu_i = 0$); b) to grade the distance, using an appropriate metric such as Levenshtein distance, determining the similarity of the two strings. Solution a) is a pessimistic approach and avoid to confuse a location, such as "Columbia", with another location, "Colombia"; solution b) tries to recover common typos mistakes. By the way, our system may be properly configured and both the solutions may be adopted depending on the criticality of the application.

In case the two attributes are equal but the values are different, the metric could be refined considering a dictionary such as Wordnet or Wordnet derived national projects (such as Italwordnet [9] or Jurwordnet [10]), and using the number of "vertical hops" among the concepts in the semantic network. In fact, if we consider the semantic hierarchy built around a generic $word_i$, we obtain the following structure:

$$\begin{aligned} H(w_i) \quad = \quad & \{s_i, h^1_{0,1}, \ldots, h^1_{0,m}, h^2_{1,1}, \ldots, h^2_{1,n}, \ldots, \\ & h^2_{m,1}, \ldots, h^2_{m,p}, m_i, m^1_{0,1}, \ldots, m^2_{m,p} \\ & hy^1_{0,1}, \ldots, hy^1_{0,m}, hy^2_{1,1}, \ldots, hy^2_{1,n}, my^1_{0,1}, ..\} \end{aligned}$$

where: $s_i$ is the set of the synsets associated to the word $w_i$; $h^j_{k,l}$ is the hypernym of j-th level (w.r.t. the root of the hierarchy) of the l-th synset associated to the k-th noun; $m_i$ is the set of meronyms associated to $s_i$; $m^j_{k,l}$ is the set of meronyms associated to $h^j_{k,l}$; $hy^j_{k,l}$ is the hyponym of j-th level of the l-th synset associated to the k-th noun; and $my^j_{k,l}$ is the set of meronyms associated to $hy^j_{k,l}$.

The two values $Value_i$, $Value_j$ are related if and only if the correspondent hierarchies share at least one element. In this case the following relation is satisfied $\mu_i = \frac{1}{VerticalHops+1}$. The Vertical Hops variable measures the minimum number of levels dividing the common element. For example if two words share an element,

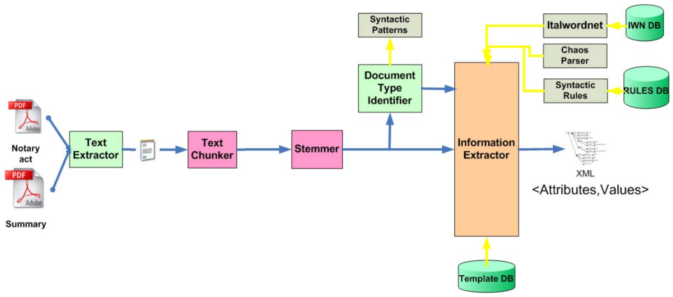


**Figure 1. The proposed system**

that is situated on the first level of the first hierarchy, (the share element is an hypernym/meronim/sinonim of the first level) and the same element is situated on a second level of the second hierarchy (the share element is an hypernym/meronim/synonim of the second level), the value of $VerticalHops$ is equal to 1.

## 3 System Overview

Figure 1 shows at a glance of the proposed system that we will briefly describe.

*Text Extractor* is a module able to extract the plain text from the source file preserving the document format. The input of the module is a structured file, such as a pdf file, and the output is a formatted text.

*Text Chunker* is a module able to cut the document into a set of elements (i.e. paragraphs) on which further processing will be performed. The subdivision of the document into simple chunks of text permit to render more accurate the syntactic and semantic analysis. This operation is accomplished using, on one hand, a text linearization process – that transforms the formatted document into a sequence of strings containing also spaces and punctuation marks – and, on the other hand, the paragraphation process that, using well defined syntactic and formatting rules, identifies the subset of strings that are candidates to form a paragraph. A simple criterion used to identify a candidate paragraph may be, for example, to find a sequence of strings that are located before a chain of “dot”-”space”- “enter”. The input of this module is a formatted text document, the output is a set of paragraphs.

*Stemmer* is a module used for removing the commoner morphological and inflexional forms from words in Italian language [11]. The input of this module is a set of paragraphs and the output is the same set of paragraphs with some added information i.e. the stemmed words.

*Document Type Identifier* is a module able to quick classify the notary document in one of the well know categories. This process is performed using a syntactic pattern analyzer that processes the first part (usually the first two paragraphs) of the document where is mandatory to specify the document typology e.g. buying selling documents, constitution of a company documents, house bank loan documents and so on. The input of this module is the first $k$ paragraphs of the document and the output is an identifier indicating the type of notary document that was submitted to the system.

*Information Extractor* is the core of the system. This module is able to analyze both syntactically and semantically the set of paragraphs, with the aim of finding *all the relevant information* that are contained in the document. This module extracts information and is driven by the preliminary classification that has been performed by the *Document Type Identifier* module. The output of this module is a couple of xml files i.e. one for the notary document and the other one for the summary. The details about the algorithms will be described in section 4.

Once the two xml files have been generated, they are submitted to *Feature Matcher* module that, using the metric (4), provides a similarity score about the submitted documents.

## 4 Algorithms

In this section, the set of algorithms, used to extract information from the documents, are described. In particular, we’ll focus our attention only on two main algorithms: the *Information Extractor* and *Feature Matcher*.

The *Information Extraction* module, as described previously, has the main goal of extracting information from a notary document or a summary on the base of the typology of document that has been submitted to the system. The structure of the algorithm is following described:

Algorithm IE($D,Paragraph,StemmedTokens,type$)
$D$ is the input document
$Paragraph$ is the set of paragraphs of the documents
$StemmedTokens$ contains the stemmed tokens of a single paragraph
$type$ is the typology of notary act
**begin**
$\mathcal{A} := getListOfAttributes(type)$ //get the list of attributes for a given category
$\mathcal{V} := \emptyset$ //is the set of values
**for each** attribute $a \in \mathcal{A}$ do
$R := getRules(a)$
**for each** rule $r \in R$ do
**for each** $s \in$ MatchRules(Paragraph,StemmedTokens,RuleSchema(r)) do
$(a, v) := ContentExtraction(s, r)$
$StoreResults(a, v)$
**end for**
**end for**
**end for**
**end**

where the function $getListOfAttributes$ returns, for a given type of notary document, the list of attributes on which a set of rules are defined. Such rules are able to identify the attributes and their values within the text. As shown in the algorithm, $getRules$ returns the appropriate rules and $MatchingRules$ retrieves those sentences matching the schema of a rule. Eventually, the selected sentences are analyzed by the $ContentExtraction$ module that retrieves the attribute-value couples. A generic rule is a combination of token patterns and/or syntactic patterns. An example of the first type is:

*(((via—Corso—C.so—Piazza—P.zza—Viale—V.le—v.le—*

*Galleria—Vicolo) ((((s*[A-Z]pPunct)+) ((s+(di—del))?s*p Upperw*) +)—((s+(di—del)) ?s*pUpperw*)+)))*[1]
This rule is able to pick up an address; while an example of the second type is a syntactic tree [8] able to retrieve for instance the attributes "acquirente"-"venditore" (buyer and seller). [2].

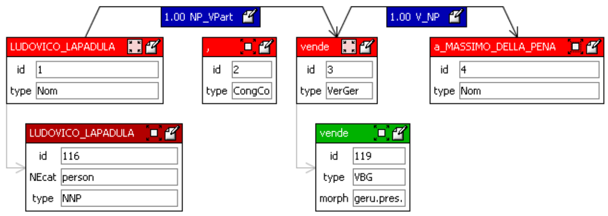


**Figure 2. Syntactic rules**

The *Feature Matcher* module has the task to compare the document and the relative summary starting from the set of attribute-value couples. The following algorithm describes this task:

```
Algorithm FeatureMatcher(A, S)
    A is the set of < Attribute_i , Value_i > couples of the notary act
    S is the set of < Attribute_j , Value_j > couples of the summary
begin
    MatchingF := ∅ //is the set of matching features
    Weights := 0//is the set of weights associated to attributes
    Score := 0//is the comparison score
    NoMatchingF := ∅ //is the set of no matching features
    H_{Values_S} := ∅//is the hierarchy built on a value of S
    H_{Values_A} := ∅//is the hierarchy built on a value of A
    for each attributes Attribute_i ∈ A do
        α:=getFeatureWeight(Attribute_i)
        Weights:=Weights+α
        for each attributes Attribute_j ∈ S do
            if (Attribute_i==Attribute_j)
                if (Value_i==Value_j)
                    MatchingF=MatchingF⋃ < a_i, v_i, α, 1 >
                    Score:=Score+1
                else
                    if (LD(Value_i, Value_j)< thr &
                            SubS(Value_i, Value_j) >0)
                        MatchingF=MatchingF⋃
                            < a_i, v_i, α, 0.8 >
                        Score:=Score+0,8
                    else
                        H_{valueS}:=CreateSemanticHierarchy(value_i)
                        H_{valueA}:=CreateSemanticHierarchy(value_j)
                        if(MatchHierarchy(H_{valueS},H_{ValueA}))>0
                            P:=1/(HierachyDepth+1)
                            MatchingF=MatchingF⋃
                                    < a_i, v_i, α, P >
                            Score:=Score+P
                        else
                        NoMatchingF=NoMatchingF⋃
                                    < a_i, v_i, α, 0 >
                        Score:=Score+0
                        end if
                    end if
                end if
            end if
        end for
    end for
    Score=Score/M*100
end
```

$LD$ being a function that computes the Levenshtein distance on a set of string values; $SubS$ a function verifying that a value is a substring of another one and $MatchingF$ a set containing the attribute-value couple the $\alpha$ weight and the matching probability $\mu_i$. The global score is obtained using the score function (4) multiplied by 100.

## 5 Experimental results

We have conducted two kinds of experiments aiming to evaluate the effectiveness of the proposed methodology. The first set of experiments tries to evaluate the precision and the recall in terms of extracted features (i.e. attribute-value couples) from a set of notary documents. The second set of experiments tries to evaluate the same parameters in the comparison between a given summary and the set of documents present into the document collection. Such collection is composed of 100 notary documents belonging to three categories: buying-selling document, enterprise foundation document, bank loan document. Each document has been labeled by a notary practitioner in order to highlight the main information characterizing the document. Note that the various parameters in the algorithms such as $thr$, $M$ and $\alpha_i$ of equation (4) are domain dependent and have been empirically set.

### 5.1 Feature extraction evaluation

For the first experiment, we have compared the results obtained by the proposed algorithms on the labeled corpus and we have evaluated the precision and recall values. The precision and the recall are defined as follow:

$$recall = \frac{R}{R + RNR} \times 100 \quad precision = \frac{R}{R + NR} \times 100 \tag{5}$$

where $R$ is the set of retrieved features, $RNR$ is the number of Relevant features Not Retrieved by the system and $NR$ is the number of Not-Relevant features retrieved by the system. The results for 20 documents (about 7 documents for each category) are shown in figure 5.1a

The first seven documents are buying-selling notary documents, the second seven documents are enterprise setting-up notary documents and the last six documents are bank loan notary documents. As shown in table 5.1 we obtain, on the average, good results in terms of precision and recall; we obtain excellent performances in extracting information from buying-selling documents. The same experiment has been conducted on the correspondent set of summaries in order to evaluate the effectiveness of the extracted information. The results are shown in figure 5.1b

On the average, the precision and recall values of the extracted features are very high for both original documents and summary. These results confirm that the selected fea-

[1]Note that "via", "Corso", "C.so" etc, are the way of denoting street, avenue and so on in Italian while the second part of the rule denotes the opportunity to have a sequence of letters and numbers preceded by some punctuated abbreviation.

[2]Figure 2 reports a simple rule that is triggered when a sentence contains the verb "vende" (sell) or one of its conjugation. In this case "Ludovico La Padula" is the proper noun of the noun phrase and is the seller, while "Massimo della Pena", who is the direct objective of the phrase, is the buyer

| Document | TP | RNR | NR | Recall | Precision |
|---|---|---|---|---|---|
| Doc 1 | 28 | 0 | 1 | 100 | 96.55 |
| Doc 2 | 28 | 0 | 1 | 100 | 96.55 |
| Doc 3 | 22 | 0 | 1 | 100 | 95.65 |
| Doc 4 | 19 | 1 | 2 | 95 | 90.48 |
| Doc 5 | 29 | 4 | 5 | 87.88 | 85.29 |
| Doc 6 | 26 | 1 | 0 | 96.30 | 100 |
| Doc 7 | 20 | 1 | 1 | 95.24 | 95.24 |
| Doc 8 | 31 | 3 | 8 | 91.18 | 79.49 |
| Doc 9 | 17 | 2 | 1 | 89.47 | 94.4 |
| Doc 10 | 24 | 0 | 0 | 100 | 100 |
| Doc 11 | 20 | 0 | 2 | 100 | 90.91 |
| Doc 12 | 27 | 2 | 1 | 93.1 | 96.43 |
| Doc 13 | 18 | 0 | 3 | 100 | 85.71 |
| Doc 14 | 20 | 2 | 1 | 90.91 | 95.24 |
| Doc 15 | 22 | 1 | 0 | 95.65 | 100 |
| Doc 16 | 25 | 1 | 0 | 96.15 | 100 |
| Doc 17 | 14 | 6 | 2 | 70 | 87.5 |
| Doc 18 | 22 | 0 | 1 | 100 | 95.65 |
| Doc 19 | 19 | 6 | 1 | 76 | 95 |
| Doc 20 | 16 | 4 | 5 | 80 | 76.19 |

| Document | TP | RNR | NR | Recall | Precision |
|---|---|---|---|---|---|
| Sum 1 | 15 | 0 | 1 | 100 | 93,75 |
| Sum 2 | 12 | 2 | 2 | 85,71 | 85,71 |
| Sum 3 | 14 | 2 | 3 | 87,5 | 82,3 |
| Sum 4 | 14 | 0 | 2 | 100 | 87,5 |
| Sum 5 | 15 | 2 | 3 | 88,23 | 83,33 |
| Sum 6 | 15 | 3 | 2 | 83,33 | 88,23 |
| Sum 7 | 13 | 2 | 3 | 86,66 | 81,25 |
| Sum 8 | 16 | 1 | 2 | 94,11 | 88,88 |
| Sum 9 | 13 | 2 | 3 | 86,66 | 81,25 |
| Sum 10 | 12 | 3 | 2 | 80 | 85,71 |
| Sum 11 | 11 | 2 | 3 | 84,61 | 78,57 |
| Sum 12 | 15 | 2 | 3 | 88,23 | 83,33 |
| Sum 13 | 14 | 3 | 0 | 82,35 | 100 |
| Sum 14 | 13 | 1 | 0 | 92,85 | 100 |
| Sum 15 | 16 | 0 | 2 | 100 | 88,88 |
| Sum 16 | 14 | 1 | 0 | 93,33 | 100 |
| Sum 17 | 12 | 3 | 1 | 80 | 92,3 |
| Sum 18 | 15 | 1 | 2 | 93,75 | 88,23 |
| Sum 19 | 14 | 2 | 1 | 87,5 | 93,33 |
| Sum 20 | 14 | 1 | 1 | 93,33 | 93,33 |

**Figure 3. Notary document(a), Summary(b): Precision and recall values**

| | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 | 16 | 17 | 18 | 19 | 20 |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 76 | 13 | 23 | 13 | 33 | 23 | 13 | 13 | 13 | 12 | 14 | 31 | 32 | 31 | 13 | 23 | 22 | 23 | 8 | 25 |
| 2 | 4 | 87 | 8 | 17 | 34 | 31 | 24 | 9 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 |
| 3 | 5 | 15 | 80 | 5 | 5 | 14 | 57 | 14 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 6 | 5 | 5 | 5 | 5 |
| 4 | 4 | 5 | 4 | 98 | 5 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 4 | 4 | 4 | 4 |
| 5 | 4 | 4 | 4 | 4 | 76 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 6 | 4 | 12 | 4 | 4 | 4 | 89 | 5 | 12 | 5 | 5 | 4 | 6 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 |
| 7 | 3 | 3 | 37 | 4 | 5 | 4 | 90 | 3 | 12 | 10 | 14 | 12 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 8 | 3 | 10 | 5 | 5 | 5 | 15 | 5 | 88 | 5 | 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 |
| 9 | 2 | 2 | 2 | 2 | 2 | 3 | 13 | 2 | 78 | 12 | 12 | 15 | 2 | 2 | 2 | 2 | 2 | 2 | 2 | 2 |
| 10 | 3 | 3 | 3 | 3 | 3 | 3 | 9 | 4 | 9 | 91 | 9 | 10 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 11 | 3 | 3 | 3 | 3 | 3 | 4 | 14 | 3 | 11 | 10 | 85 | 10 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 12 | 3 | 3 | 3 | 3 | 3 | 4 | 9 | 3 | 12 | 9 | 8 | 79 | 3 | 3 | 3 | 3 | 3 | 3 | 3 | 3 |
| 13 | 5 | 6 | 5 | 6 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 75 | 6 | 11 | 12 | 11 | 6 | 5 | 6 |
| 14 | 4 | 5 | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 76 | 26 | 5 | 4 | 4 | 4 | 5 |
| 15 | 5 | 6 | 5 | 6 | 5 | 5 | 5 | 55 | 5 | 5 | 5 | 5 | 5 | 30 | 90 | 6 | 5 | 5 | 5 | 5 |
| 16 | 4 | 5 | 4 | 5 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 4 | 5 | 9 | 4 | 74 | 4 | 4 | 4 | 5 |
| 17 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 87 | 6 | 6 | 6 |
| 18 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 5 | 91 | 5 | 5 |
| 19 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 6 | 81 | 6 |
| 20 | 0 | 2 | 0 | 1 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 1 | 1 | 1 | 1 | 0 | 0 | 0 | 69 |

**Figure 4. Matching comparison**

tures are good candidate to perform comparisons among documents.

### 5.2 Matching evaluation

The second set of experiments aims to compare each summary to the whole document collection in order to evaluate its similarity grade 3.

Figure 4 shows the results concerning the 20 documents and 20 summaries used in the first experiment. The comparison is performed among a given summary (x axis) and the 20 notary documents (y axis); in this way, each line reports the similarity grade between each couple of documents (i.e. summary and notary document). The marked cells contains the higher similarity value for a given row and as clearly shown in the figure 4, the main diagonal contains the higher score, meaning that, as is and as we expected, the i-th document and the i-th summary share the same information.

On the average, the similarity value is very low each time a summary and a document are compared and they don't share the same information. Some anomalies are reported when exactly the same people buy-sell an apartment and found a company (see the darker cell). Even in this case the similarity value is not very high to justify a false positive.

Moreover as proofed the summary could be used as a query to retrieve the original document.

## 6 Conclusion and future works

In this paper we have presented simple algorithms that are able to extract relevant information from notary documents and we have defined a measure capable of comparing the original document and its related summary. The experimental section has shown very encouraging results. Future works will be relying on a complete textual information processing and retrieval system for application in the legal domain.

## References


[1] A. K. D. Bruninghaus St. Finding factors: Learning to classify case opinions under abstract fact categories. *in Proc ICAIL'97*, pages 123–131, 1997.

[2] F. Y. Y. CHOI. Advances in domain independent linear text segmentation. *In Proceedings of NAACL 00*, 2000.

[3] H. J. R. et al. Sri international: Description of the fastus system used for muc-4. *Fourth Message Understanding Conference, Morgan Kaufmann*, pages 143–147, 1992.

[4] M. S. et all. A full-text retrieval system with a dynamic abstract generation function. *in Proc SIGIR 94*, pages 152–161, 1994.

[5] C. M. Firmin T. An evaluation of automatic text summarization systems. *In Advances in Automatic Text Summarization, I. Mani and M. T. Maybury, Eds. MIT Press,*, page 325336, 1999.

[6] R. L. F. Jacobs P S. Scisor: Extracting information from on-line news. *Comm ACM*, 33(11):88–97, 1990.

[7] D. MCDONALD and H. CHEN. Using sentence-selection heuristics to rank text segments in txtractor. *In Proceedings of the Second ACM/IEEE-CS JCDL (Portland, OR).*, 2002.

[8] F. M. Z. Roberto Basili. Parsing engineering and empirical robustness. *Journal of Natural Language Engineering*, 2-3(8), June 2002.

[9] A. Roventini. Italwordnet: Building a large semantic database for the automatic treatment of the italian language. *In Zampolli, A., Calzolari, N., Cignoni, L. (eds.), Computational Linguistics in Pisa, Special Issue of Linguistica Computazionale*, Vol. XVIII-XIX, 2003.

[10] D. Tiscornia. Some ontological tools to support legal regulatory compliance, with a case study. *Workshop on Regulatory Ontologies and the Modeling of Complaint Regulations (WORM CoRe 2003) Springer LNCS*, November 2003.

[11] E. Zanchetta and M. Baroni. Morph-it! a free corpus-based morphological resource for the italian language. *Proceedings of Corpus Linguistics 2005*, pages 23–32, 2005.