Contextual Word Spotting in Historical Manuscripts using Markov Logic Networks

David Fernández
Computer Vision Center
Dept. Ciències de la Computació
Universitat Autònoma de Barcelona
Barcelona, Spain
dfernandez@cvc.uab.cat

Simone Marinai
Dipartimento di Ingegneria dell'Informazione
University of Florence
Florence, Italy
simone.marinai@unifi.it

Josep Lladós
Computer Vision Center
Dept. Ciències de la Computació
Universitat Autònoma de Barcelona
Barcelona, Spain
josep@cvc.uab.cat

Alicia Fornés
Computer Vision Center
Dept. Ciències de la Computació
Universitat Autònoma de Barcelona
Barcelona, Spain
afornes@cvc.uab.cat

ABSTRACT

Text extraction in historical handwritten documents has been addressed by several works. Among various techniques, word spotting is one of the more frequently used. This task consists in detecting occurrences of handwritten word images given one or more sample words. In most cases these approaches use the shapes of the word images to measure the word similarity, but do not take into account the structure of the phrase during the recognition. On the other hand, natural languages can often be modelled by suitable grammars whose knowledge can improve the word spotting results. This structural information is even more useful when dealing with information that is intrinsically described as one collection of records.

In this paper we present one approach to word spotting which uses the structural information of records to improve the results. The method relies on Markov Logic Networks to probabilistically model the relational organization of handwritten records. The performance has been evaluated on the Barcelona Marriages Dataset that contains structured handwritten records that summarize marriage information.

Categories and Subject Descriptors

I.7 [**DOCUMENT AND TEXT PROCESSING**]: Document Capture; I.2.6 [**ARTIFICIAL INTELLIGENCE**]: Learning

Keywords

Handwritten documents, Document image processing, Historical document analysis, Word-Spotting, Markov Logic Networks

1. INTRODUCTION

Word spotting has become a popular and efficient strategy in the recognition of historical handwritten document. Due to the quality of physical preservation, the writing styles, and the obsolete languages, the full transcription of such documents is extremely difficult. In many applications, once the documents are digitized for preservation purposes, search contents-wise is the main purpose. Here is when the use of object retrieval approaches using visual features gains relevance.

The use of context can significantly improve the recognition of individual objects. In computer vision, it is an emerging trend [1]. Usually word spotting is built based solely on the statistics of local terms. The use of correlative semantic labels between codewords adds more discriminability in the process. Three levels of context can be defined in a word spotting scenario. First, the joint occurrence of words in a given image segment. Second, the geometric context involving a language model regarding to the relative 1D or 2D position of objects. Third, the semantic context defined by the topic of the document. A number of document collections convey an underlying structure. This structure is natural in records describing demographic events such as census, birth, marriage, or death records. This structure is characterized by a page arranged in records (paragraphs) or tables. In a finer level, each unit (record) uses to follow a syntactic structure. The analysis of the contents in such documents can not be solved by raw transcription, but word spotting is a good alternative for record linkage (linking names for genealogical analysis) or search of people, places and events.

In this paper, we show how the use of the context improves the performance of word spotting. In historical demographic documents, some words have high probability of co-ocurrence. For example, if we have genealogic linkage, we can learn joint probabilities between family names, some common words in the record like "married to" determine the position of the searched ones, migration movements from geographic areas also generate clusters of family names that can be linked to city names, etc. We particularly focus in the syntactic context intra-sentences. The use of dictionaries is a common approach to model this context [2]. However, there is the drawback that lexicons constructed generically from a language do not work properly in historical documents where the contents are very specific in terms of topic and time period. The use of closed dictionaries is corpus-specific, and practically unfeasible. We therefore focus on the syntactical structure of the text lines. The main idea of our approach is that given a query word image and its semantic category (e.g. family name, city name, date, etc.), the detection can be reinforced by the likelihood of this category to appear within a context, according to syntactic rules.

We propose the use of Markov Logic Networks (MLN) [3] to improve the results of word spotting according to the stated hypothesis. MLN is a very powerful statistical relational learning model that provides a very rich representation. The use of MLN to model a grammatical structure offers more flexibility in the definition of the rules, incremental and simple learning,with respect to traditional language models used in handwriting recognition. As experimental setup, a database of handwritten marriage licenses of the Barcelona Cathedral Archive has been used. The documents are semi-structured in records (paragraphs). Each record contains the information of a marriage using a regular structure, but with some variations from one period to another, or from one social status to another.

2. RELATED WORK

Some historical documents contain information that follows one rigid structure. Related information always appears in the same position or order. This structural information could be used to improve the results obtained by searching methods, such as word-spotting approaches [4, 5, 6].

Markov Logic Networks can be used to learn the probability that different fields appear in a certain order, and integrate this information with the output of the Word-Spotting retrieval. Fabian et al. [7] proposes a Markov Logic Model which incorporates the contextual information in the form of expectations of a dialogue system to perform semantic processing in a spoken Dialogue System.

2.1 Word spotting

Word spotting has been applied to localize instances of words in handwritten historical documents. Depending on how the input is specified, these approaches can be categorized in two groups: query-by-text and query-by-example. In query-by-text, the input is a text string. Character models are learned off-line and at runtime the character models are combined to form words and the probability of each word is evaluated [8, 9, 10]. In query-by-example the input is an image of the word to search, and the output is a set of the most representative (sub)images in the database containing a similar word shape [11, 12, 13].

The query-by-text has the advantage of flexibility to search any kind of keyword. However, labelled datasets are required in order to train the recognition engine. At the other hand, query-by-example methods can achieve sufficient accuracy to be useful in a practical scenario. As Manmatha et al. discuss in their work [11], these methods are mostly based on image matching. These methods are worth of attention when labeled training data are not available or would be too expensive to collect.

The two main components of the query-by-example methods are the representation (features describing the text) and the matching (measure of similarity). Among various types of features, some approaches describe the image with global representations, e.g., gradient, structural, and convexity features (e.g. [14]) or features based on moments of binary images [15]. Global representations usually have a fixed-size description.

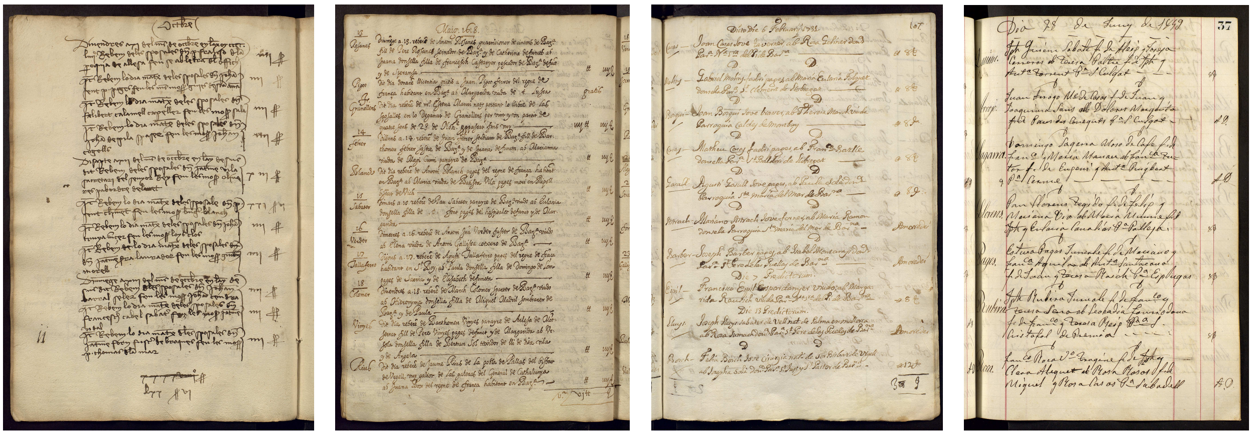
On the opposite, some widely used techniques compare features with variable sizes by using suitable matching methods such as Dynamic Time Warping (DTW) [12] or Hidden Markow Models (HMM) [16]. In doing so, the resulting keyword spotting approach is more flexible dealing with variations of style and word length [17, 18, 19, 20]. Typically, a sliding-window is used to scan the image from left to right and a feature vector is extracted at each position. The methods that extract the features in this way are categorized as segmentation-free, because they do not need to explicitly segment the words in the documents [6, 21]. The main problems of segmentation-free methods are the difficulty of learning with sequences and the computational time needed to compute the distance between words that is usually rather high.

At the other hand, we have the non-segmentation-free methods, which first need to segment the words from the documents [5]. A learning-based approach at word level was presented in [22]. Based on local gradient features, posterior probabilities of keyword HHMs are used for keyword spotting in conjunction with universal vocabularies for score normalization. A similar approach was presented in [10] for non-symmetric half plane HMMs.

2.2 Markov Logic Networks

In artificial intelligence, one of the open questions is concerned with techniques for combining expressive knowledge representation formalisms (such as relational and first-order logic) with principled probabilistic and statistical approaches used to learn and infer. Probabilistic and statistical methods refer to the use of probabilistic representations and reasoning mechanism grounded in probability theory, such as Bayesian networks, Hidden Markov Models and probabilistic grammars and the use of statistical learning and inference techniques.

One stochastic context-free grammar (SCFG, or probabilistic context-free grammar, PCFG) is one context-free grammar where one probability is associated to each production rule. In SCFG, the probability of one derivation is the product of the probabilities of the productions. SCFG has been



(a) 1481: volume 2 (b) 1618: volume 69 (c) 1729: volume 127 (d) 1860: volume 200

Figure 1: Examples of marriage licenses from different centuries.

applied to different applications, such as Natural Language Processing. In this kind of application SCFG are modelled as grammars, typically specified in syntaxes where the rules are absolute. Some speech recognition systems use SCFGs to improve their probability estimate and thereby their performance [23].

Uncertainty and complex relational structure characterize the major part of real-world application domains. Statistical learning is related to uncertainty while relational learning deals with relational information. Statistical relational learning (SLR) [24] attempts to combine the best of both. SRL is a combination of statistical learning which addresses uncertainly in data and relational learning which deals with complex relational structures. There is an increasing interest to develop SLR approaches such as stochastic logic programs [25], probabilistic relational models [26], relational Markov models [27], structural logistic regression [28], and others.

Markov Logic Networks (MLN) is one of the most well-known methods proposed for SLR [29, 30]. Syntactically MLNs extend first-order logic and associate one weight to each formula. Semantically, they can represent a probability distribution over possible worlds using formulas and their corresponding weights. Several applications are developed using MLN as a basis to infer some knowledge of the world. In [31] the application of MLN as a language for learning classifiers is investigated. In [32] is presented a goal recognition framework based on MLN.

A *first-order knowledge base* (KB) is a set of sentences or formulas in first order logic. Formulas are built using four types of symbols: *constants*, *logical variables* ranging over objects of a domain on interest, *functions* representing mappings from tuples of objects to objects, and *predicates* representing relations among objects in the domain or attributes of objects. If a world violates even one formula, it has probability zero. A KB can thus be interpreted as a set of hard constraints on the set of possible worlds. Markov logic networks soften these constraints so that when a world violates one formula in the KB it becomes less probable, but not impossible. The fewer formulas a world violates, the more probable it is.

A Markov logic network L is a set of pairs (F_i, w_i), where F_i is a formula in first-order logic and w_i is a real number. Together with a finite set of constants $C = \{c_1, c_2, \ldots, c_{|C|}\}$, it defines a Markov network $M_{L,C}$ (Equation 1) as follows:

- $M_{L,C}$ contains one binary node for each possible grounding of each predicate appearing in L. The value of the node is 1 if the ground predicate is true, and 0 otherwise.
- $M_{L,C}$ contains one feature for each possible grounding of each formula F_i in L. The value of this feature is 1 if the ground formula is true, and 0 otherwise. The weight of the feature is the w_i associated with F_i in L

$$P\left(X = x\right) = \frac{1}{Z} exp\left(\sum_j w_j f_j\left(x\right)\right) \qquad (1)$$

A world is an assignment of truth values to all possible ground atoms. Each state of the Markov network presents a possible world. The probability distribution over possible worlds x specified by the ground network is calculated by Formula 1, where $f_j\left(x\right)$ is the number of true groundings for F_i in x and Z is the partition function that is used to make the summation of all possible groundings adds up to one.

Inference has two main phases in MLNs. In the first phase, a minimal subset of the ground Markov network is selected. Many predicates that are independent of the query predicates may be filtered in this phase. As a result, the inference can be carried out over a smaller Markov network. In the second phase the inference is performed on the Markov network using Gibbs sampling [33] where the evidence nodes are observed and are set to their values.

3. METHOD

The results of the word spotting are improved using the structural information of the documents and MLN. In this section the database used in the experiments is illustrated,

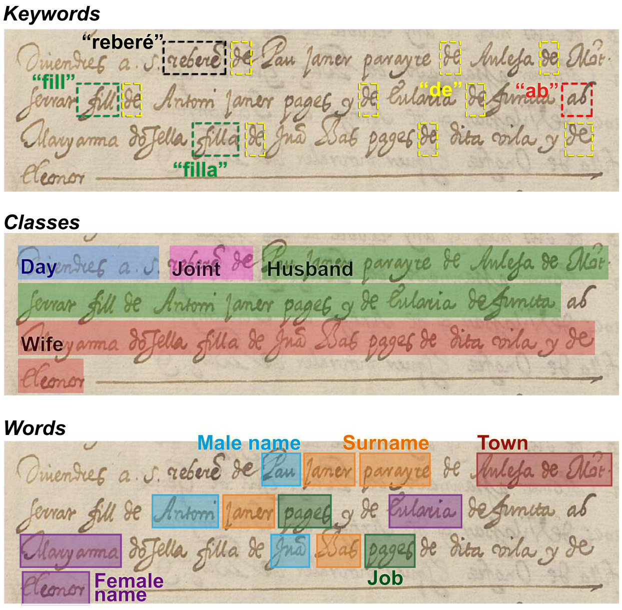


Figure 2: Grammar structure of two records.

the Word Spotting approach used in the first step is presented, and the rules used to learn and infer the structural information of the documents are discussed.

3.1 Dataset

We have applied our word spotting approach to the Marriage Licenses Books conserved at the Archives of the Cathedral of Barcelona. These manuscripts, called *Llibre d'Esposalles* [34], consist of 244 books written between 1451 and 1905, and include information of approximately 550.000 marriages celebrated in over 250 parishes (Fig. 1). Each marriage record contains information about the couple, such as their names and surnames, occupations, geographical origin, parents information, as well as the corresponding marriage fee that was paid (this amount depends on the social status of the family). Each book contains a list of individual marriage license records (analogous to an account book) of two years, and was written by a different writer. Information extraction from these manuscripts is of key relevance for scholars in social sciences to study the demographical changes over five centuries.

The marriage license records present a fixed structure in all of them (See Fig. 2 - *Classes*). The first part of the record specifies the day that the marriage took place. The next words are *Reberé de* (which means *Received from* in Catalan). Following these two words, information related to the husband is found. And finally, information related to the wife is showed.

There are some keywords in the records (See Fig. 2 - *Keywords*) that always appear in all of them: *reberé*, *de*, *ab*, *fill* and *filla* (*receive*, *the*, *with*, *son* and *daughter* in old Catalan). These keywords always appear at the same position in the registers, and they usually indicate that certain kind of information is going to be written. For example, the day of the marriage always appears before the key-word *reberé*, and after that, the information of the husband is written. Information about the husband and his parents comprises until the key-word *ab*. This word indicates the beginning of the wife's information. There are some key-words which indicate some specific information, for instance, after the word *fill*, the husband's father name appears, and after the word *filla*, there is the wife's father name.

The rest of the words can be classified in different categories (See Fig. 2 - *Words*), and usually appear at the same position inside the record.

In this work we have used 50 pages of the *volume 69* for the experiments. We have used the two first classes of each record: *Day* and *Joint*. As future work, we are planning to use the rest of the classes.

3.2 Word spotting approach

The Word Spotting approach [5] used in this work follows a query-by-example strategy. Thus given a query image, it locates all the instances of the same word class into the documents, which have been previously indexed. Shape matching techniques are used in the holistic approach. The descriptor used is inspired in Loci characteristics [35], aggregating pseudo-structural information.

The spotting strategy can be separated into two major modules (Fig. 3): the indexing and the retrieval stage. First, word images are indexed considering a feature space considering shape features. Second, word images are used as queries and similar instances from the database in terms of shape similarity are retrieved.

The quality of the documents can be affected by their lifetime and degradations. A pre-processing step that includes binarization and noise removal is used to improve the quality of the documents for the subsequent processing. The words are then segmented using projections analysis techniques in combination with Anisotropic Gaussian Filters to smooth the projection function.

Once the words are segmented, one feature vector is computed for each word and stored in one suitable hash structure. The descriptor is an adaptation to word images of the descriptor devised by Glucksman [35]. A characteristic Loci feature is composed by counting the number of intersections along eight directions (up, down, right, left, and the four diagonals). For each background pixel in the binary image and for each direction, we count the number of intersections (black/white transitions between pixels). Hence, each key-point generates a codeword (called *Locu number*) which corresponds to a position inside the features vector. Each generated position increments the count in that position of the feature vector. The feature vector can be seen as a histogram of *Locu numbers*.

Basically, the retrieval process consists in organizing the feature codewords in a look up table *M*, whereas the classification process consists in searching the best matching of the query with all the words of *M*.

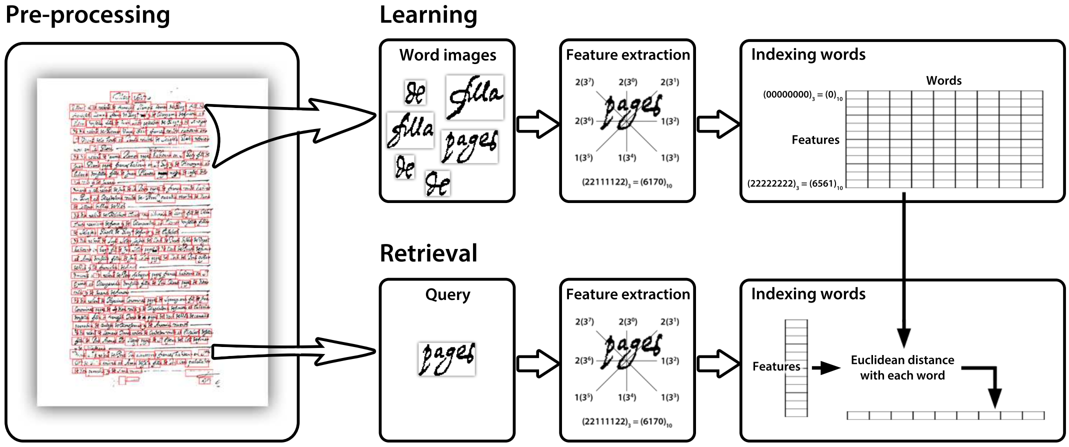


Figure 3: Outline of the Word-Spotting approach used.

3.3 Markov Logic Networks for Marriages

In the proposed approach we use the *Alchemy* [36] software package that provides a series of algorithms for statistical relational learning and probabilistic logic inference, based on the Markov logic representation.

A Markov Logic Network [37] is a probabilistic logic which applies the ideas of a Markov network to the first-order logic, enabling uncertain inference. The MLN can be considered as a collection of first-order logic rules to each of which it is assigned one real number, the weight. Each rule represents a rule in the domain, while the weights indicate the strength of the rule.

Since marriage records have one regular but not fixed structure it is possible to model this structure with statistical parsing. The latter allows to identify the most probable parse of a sentence given a probabilistic context-free grammar (CFG). This grammar is then translated into an MLN as described in the following.

To use the MLN framework in our application we mapped the structure of the records in the marriage dataset in a weighted Context-Free grammar (CFG) in Chomsky normal form.
Non-terminal symbols are: R that corresponds to the entire record, D that is for the part of the record that represents the day of the wedding, P is the joint-words (*Rebere de*).
Terminals symbols define all the tokens (in our case handwritten words) that appear in the document. The terminal *de* represents the word *de*, *rebere* represents the word *rebere* and *nom* is one class which represents all the other words.

The CFG grammar is therefore defined by the following production rules:

$$\begin{array}{lll} R & \rightarrow & D\ P \\ D & \rightarrow & nom\ nom \\ D & \rightarrow & nom\ nom\ nom \\ P & \rightarrow & nom\ de \\ P & \rightarrow & nom\ rebere \end{array}$$

To translate this in an MLN we encode each production rule as a clause, for instance $R \rightarrow D\ P$ becomes $D \wedge P \Rightarrow R$. The next step is to denote the position of the words or phrases in the record. To this purpose each terminal or non-terminal is described as a predicate with two arguments that denote the beginning and the end in the record. These positions denote the beginning and end of a record or phrase as well as positions between words. Therefore one record with n words has $n + 1$ positions. The MLN formulation is the following:

```
// Definition of R
D(a,b)^P(b,c) => R(a,c)

// Definition of D
nom(a,b)^nom(b,c)^nom(c,d) => D(a,d)
nom(a,b)^nom(b,c) => D(a,c)

// Definition of P
nom(a,b)^de(b,c) => P(a,c)
nom(a,b)^rebere(b,c) => P(a,c)
```

Here, `a` and `b` indicate the positions between the words. To encode the sequential nature of the records we shall consider one predicate `Succ(j,i)` that states that the position `i` follows position `j`.

We should then match the ideal record structure with the noisy output generated by keyword spotting on the handwritten registers. To this purpose we define one `WordSpot (hword,pos)` predicate that assigns the class `hword` to the word at position `pos`. Possible classes are the occurrences of the keywords *"DE"* and *"REBERE"* as well as non-recognized words that are labeled as *"SHORT"*, *"MEDIUM"*, or *"LONG"* according to their length. Obviously, in the handwriting recognition there could be false positives and false negatives and this should be reflected by suitable production rules that link the non-terminals with the output of the keyword spotting:

```
// de
WordSpot("DE",i)^Succ(j,i) => de(i,j)
WordSpot("SHORT",i)^Succ(j,i) => de(i,j)

// rebere
```

```
WordSpot("REBERE",i)^Succ(j,i) => rebere(i,j)
WordSpot("MEDIUM",i)^Succ(j,i) => rebere(i,j)

// nom
WordSpot("LONG",i)^Succ(j,i) => nom(i,j)
WordSpot("MEDIUM",i)^Succ(j,i) => nom(i,j)
WordSpot("SHORT",i)^Succ(j,i) => nom(i,j)
```

The above rules take care of possible errors in the recognition. For instance, *de* can correspond either to one word recognized as *de* or to a short word.

If there are homonyms belonging to different parts of the record, such as `"MEDIUM"` (*rebere* or *nom*), then we have to make sure that only one of these parts is assigned. The ambiguities in the lexicon are solved making mutual exclusion rules for each pair of parts as described in the following where the numbers before each rule denote the corresponding weight (in this case very high, meaning certitude).

```
// Mutual exclusion rules
999 !de(i,j) v !rebere(i,j)
999 !de(i,j) v !nom(i,j)
999 !rebere(i,j) v !nom(i,j)

999 !D(i,j) v !P(i,j)
999 !D(i,j) v !R(i,j)
999 !P(i,j) v !R(i,j)

999 D(a,b) ^ P(b,c)

999 !D(a,a)
999 !P(a,a)
999 !R(a,a)
```

The last step for using MLN is the training of weights associated to rules. The weights are learned taking into account labeled training data. The training data are the records recognized with word spotting integrated with information from the ground-truth. Rules that are most often true will get higher weights while rules that are sometimes violated (for instance due to errors in the word spotting approach) will get lower weights.

Each record in the training set is described by assigning the appropriate values to the previous predicates. One example is as follows that corresponds to one record where the text `dit dia rebere de` has been recognized as `"SHORT"`, `"SHORT"`, `"REBERE"`, `"SHORT"` (in this case the `de` keyword was not properly recognized).

```
WordSpot("SHORT",0)
WordSpot("SHORT",1)
WordSpot("REBERE",2)
WordSpot("SHORT",3)

R(0,4)
D(0,2)
P(2,4)

nom(0,1)
nom(1,2)
rebere(2,3)
de(3,4)

Succ(1,0)
Succ(2,1)
Succ(3,2)
Succ(4,3)
```

Here, in the first part of the training file, we define the words in the record that are recognized by word spotting, then the position and order of each non-terminal. At the end we define the order of the terminals. Likewise, the test data are generated from the output of the word-spotting approach without considering the ground truth information. For each record the following information is generated:

```
WordSpot("REBERE",0)
WordSpot("SHORT",1)
WordSpot("MEDIUM",2)
WordSpot("SHORT",3)

R(0,4)

Succ(1,0)
Succ(2,1)
Succ(3,2)
Succ(4,3)
```

The structure of training and test files is similar, but the non-terminals are not defined. The position of the non-terminals, and therefore the labeling of parts of the record according to the two main classes (D and P) is obtained by running the MLN inference on test records.

4. EXPERIMENTS AND RESULTS

We have used the Word-spotting approach proposed in [5]. This work was evaluated using different characteristic pixels (background and foreground pixels of the word image), using different mask sizes (size of regions of interest to compute the number of intersections for each key-point) and using different distance measures. In order to test the proposed approach, we have used the parameters that obtain the best results in the word-spotting approach: the mask size is fixed to 100; the key-points used are the background of the word image; and the comparison measure used is the Euclidean distance.

The experiments have been performed using 50 documents of the volume 69 of the *Llibres de Esposalles*. The keywords searched are *reberé* and *de*, and the grammar classes used are the two first ones: *Day* and *Joint*. We have 200 records from the documents and we have performed four different experiments (Table 1). In all of them, we have trained using 50 records and we have used different number of registers. In the first experiment, we train and test using the 50 same records (from the 50 documents explained before). In the second experiment, we train with 50 records, and test with 188 records. For the third experiment, we have removed the 50 records used in the training step. Some records present big distortions in the output of the word-spotting due to a bad word segmentation, e.g.- there are some cases with over-segmentation or under-segmentation, producing a non-well-formatted structure. These records have been removed in the experiment explained above. In the last experiments, we have introduced the non-well-formatted-records.

Some examples of the weighted rules obtained after training by using 50 records are shown below.

```
-2.407 !D(a1,a2) v !P(a2,a3) v R(a4,a3)
0.4005 D(a1,a2) v !nom(a1,a3) v ...
-1.198 D(a1,a2) v !nom(a1,a3) v ...
0.3854 P(a1,a2) v !de(a3,a2) v ...
-0.140 P(a1,a2) v !rebere(a3,a2) v ...
-304.7 de(a1,a2) v !WordSpot("DE",a1) v ...
-3.402 de(a1,a2) v !WordSpot("TSHORT",a1) ...
152.15 rebere(a1,a2)v!WordSpot("REBERE",a1)...
-0.055 rebere(a1,a2)v!WordSpot("TMEDIUM",a1)...
0      nom(a1,a2) v !WordSpot("TLONG",a1) ...
76.945 nom(a1,a2)v!WordSpot("TMEDIUM",a1)...
4.4305 nom(a1,a2) v !WordSpot("TSHORT",a1)...
545.56 de(a1,a2) v !WordSpot("DE",a1)...
155.84 rebere(a1,a2) v !WordSpot("REBERE",a1)...
0      !de(a1,a2) v !WordSpot("DE",a1) v ...
0      !de(a1,a2) v !WordSpot("DE",a3) v ...
0      rebere(a1,a2) v !WordSpot("TSHORT",a2)...
0      de(a1,a2) v !WordSpot("TSHORT",a1) v ...
1.7022 !rebere(a1,a2) v !WordSpot("REBERE",a1)...
851.67 !de(a1,a2) v !rebere(a1,a2)
872.73 !de(a1,a2) v !nom(a1,a2)
878.00 !rebere(a1,a2) v !nom(a1,a2)
893.99 !D(a1,a2) v !P(a1,a2)
974.20 !D(a1,a2) v !R(a1,a2)
899.24 !P(a1,a2) v !R(a1,a2)
9.8652 !D(a1,a2) v !P(a2,a3)
1046.2 !D(a1,a1)
970.24 !P(a1,a1)
733.35 !R(a1,a1)
-3.212 D(a1,a2)
-4.065 P(a1,a2)
-2.424 R(a1,a2)
-243.0 de(a1,a2)
-156.6 rebere(a1,a2)
-3.401 nom(a1,a2)
0      WordSpot(a1,a2)
0      Succ(a1,a2)
```

We can observe that each rule has a weight indicating the importance of that rule in our grammar. For example, the rule: `851.671 !de(a1,a2) v !rebere(a1,a2)` has a high weight because it means that a word cannot be both *"DE"* and *"REBERE"*.

Using these weighted rules, we have computed the results showed in the Table 1. It can be seen that we have outperformed the original word spotting method. In all the experiments done, we have reduced the number of False Positives and we have increased the True Negatives samples. In addition to this, the Precision is increased in all the cases, as shown by the F_1 score.

5. CONCLUSIONS AND FUTURE WORK

The objective of this work is to demonstrate that context information improves the performance of a Word Spotting approach. We have proved that, using MLN, we reduce the number of False Positives and increase the True Negatives. Accordingly, we have shown that, using the spatial information, which relates the words of the documents, the results of the word-spotting approaches can be improved.

This work has been tested with a small number of classes and keywords. As future work, we plan to use all the classes of the records, and all the keywords searched in the work [5].

Acknowledgement

The authors thank the *CED-UAB* and the Cathedral of Barcelona for providing the images. D. Fernandez, J. Llados and A. Fornes are partially supported by the Spanish projects TIN2011-24631, TIN2009-14633-C03-03 and CSD-2007-00018, by the EU project ERC-2010-AdG-20100407-269796 and by a research grant of the UAB (471-01-8/09). S. Marinai is partially supported by the Italian PRIN project *Statistical Relational Learning: Algorithms and Applications*.

6. REFERENCES

[1] M. J. Choi, A. Torralba, and A. S. Willsky, "Context models and out-of-context objects," *Pattern Recognition Letters*, vol. 33, no. 7, pp. 853–862, 2012.

[2] R. Grishman, "Information extraction," *The Handbook of Computational Linguistics and Natural Language Processing*, pp. 515–530, 2003.

[3] M. Richardson and P. Domingos, "Markov logic networks," *Machine learning*, vol. 62, no. 1-2, pp. 107–136, 2006.

[4] J. Almazán, A. Gordo, F. A., and E. Valveny, "Efficient exemplar word spotting," in *BMVC*, 2012, pp. 1–11.

[5] D. Fernández, J. Lladós, and A. Fornés, "Handwritten word spotting in old manuscript images using a pseudo-structural descriptor organized in a hash structure," in *IbPRIA*, 2011, pp. 628–635.

[6] M. Rusinol, D. Aldavert, R. Toledo, and J. Llados, "Browsing heterogeneous document collections by a segmentation-free word spotting method," in *ICDAR*, 2011, pp. 63–67.

[7] A. Fabian, M. Hernandez, L. Pineda, and I. Meza, "Contextual semantic processing for a spanish dialogue system using markov logic," in *10th Mexican international conference on Advances in Artificial Intelligence*, ser. MICAI'11, 2011, pp. 258–266.

[8] J. Chan, C. Ziftci, and D. Forsyth, "Searching off-line arabic documents," in *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, vol. 2, 2006, pp. 1455–1462.

[9] J. Edwards, Y. Teh, R. Bock, M. Maire, V. G., and D. Forsyth, "Making latin manuscripts searchable using ghmm's," in *Advances in Neural Information Processing Systems 17*. Cambridge, MA: MIT Press, 2004, pp. 385–392.

[10] C. Choisy, "Dynamic handwritten keyword spotting based on the nshp-hmm," in *ICDAR*, vol. 1, 2007, pp. 242–246.

[11] R. Manmatha, C. Han, and E. M. Riseman, "Word spotting: a new approach to indexing handwriting," in *CVPR*, 1996, pp. 631–637.

Experiment	Method	TP	FP	TN	FN	Precision	Recall	F1
50 samples	MLN	63	**24**	**75**	13	**0.72**	0.82	**0.77**
	Word Spotting	63	51	48	13	0.55	0.82	0.66
188 samples	MLN	278	**98**	**270**	31	**0.73**	0.89	**0.81**
	Word Spotting	278	195	173	31	0.58	0.89	0.71
138 samples	MLN	200	**69**	**198**	35	**0.74**	0.85	**0.79**
	Word Spotting	215	144	125	18	0.59	0.92	0.72
150 samples	MLN	215	**78**	**244**	23	**0.73**	0.90	**0.80**
	Word Spotting	215	157	166	22	0.57	0.90	0.70

Table 1: Accuracy for the different experiments.

[12] T. M. Rath and R. Manmatha, "Word image matching using dynamic time warping." in *CVPR (2)*. IEEE Computer Society, 2003, pp. 521–527.

[13] K. Terasawa and Y. Tanaka, "Locality sensitive pseudo-code for document images," in *ICDAR 2007*, vol. 1, 2007, pp. 73–77.

[14] B. Zhang, S. Srihari, and C. Huang, "Word image retrieval using binary features," in *Document Recognition and Retrieval XI*, vol. 5296, 2004, pp. 45–53.

[15] A. Bhardwaj, D. Jose, and V. Govindaraju, "Script independent word spotting in multilingual documents," in *IJCNLP*, 2008, pp. 48–54.

[16] A. Fischer, A. Keller, V. Frinken, and H. Bunke, "Hmm-based word spotting in handwritten documents using subword models," in *ICPR*, 2010, pp. 3416–3419.

[17] V. Frinken, A. Fischer, R. Manmatha, and H. Bunke, "A novel word spotting method based on recurrent neural networks," *PAMI*, vol. 34, no. 2, pp. 211–224, 2012.

[18] T. Rath and R. Manmatha, "Word spotting for historical documents," *IJDAR*, pp. 139–152, 2007.

[19] J. A. Rodriguez and F. Perronnin, "Local gradient histogram features for word spotting in unconstrained handwritten documents," in *ICFHR*, 2008.

[20] J. Rodriguez and F. Perronnin, "A model-based sequence similarity with application to handwritten word spotting," *PAMI*, vol. 34, no. 11, pp. 2108–2120, 2012.

[21] B. Gatos and I. Pratikakis, "Segmentation-free word spotting in historical printed documents," in *ICDAR*, 2009, pp. 271–275.

[22] J. A. Rodriguez and F. Perronnin, "Handwritten word-spotting using hidden markov models and universal vocabularies," *Pattern Recognition*, vol. 42, no. 9, pp. 2106–2116, 2009.

[23] R. Beutler, T. Kaufmann, and B. Pfister, "Integrating a non-probabilistic grammar into large vocabulary continuous speech recognition," in *Automatic Speech Recognition and Understanding, 2005 IEEE Workshop on*, 2005, pp. 104–109.

[24] H. Khosravi and B. Bina, "A survey on statistical relational learning," in *Canadian conference on Advances in Artificial Intelligence*, 2010, pp. 256–268.

[25] J. Cussens, "Loglinear models for first-order probabilistic reasoning," in *15th Conference on Uncertainty in Artificial Intelligence*. Morgan Kaufmann, 1999, pp. 126–133.

[26] N. Friedman, L. Getoor, D. Koller, and A. Pfeffer, "Learning probabilistic relational models," in *IJCAI*. Springer-Verlag, 1999, pp. 1300–1309.

[27] C. R. Anderson, P. Domingos, and D. S. Weld, "Relational markov models and their application to adaptive web navigation," in *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, 2002, pp. 143–152.

[28] A. Popescul and L. H. Ungar, "Structural logistic regression for link analysis," in *2nd International Workshop on Multi-Relational Data Mining*, 2003, pp. 92–106.

[29] P. Domingos and M. Richardson, "Markov logic: A unifying framework for statistical relational learning," in *ICML workshop on stadistical relational learning and its connections to other fields*, 2004, pp. 49–54.

[30] P. Singla and P. Domingos, "Entity resolution with markov logic," in *6th International Conference on Data Mining*, 2006, pp. 572–582.

[31] V. A. Silva and F. G. Cozman, "Markov logic networks for supervised, unsupervised and semisupervised learning of classifiers," in *IV Workshop on MSc Dissertation and PhD Thesis in Artificial Intelligence (WTDIA)*, 2008.

[32] E. Ha, J. P. Rowe, B. W. Mott, and J. C. Lester, "Goal recognition with markov logic networks for player-adaptive games." in *AIIDE*. The AAAI Press, 2011.

[33] G. Casella and E. I. George, "Explaining the Gibbs Sampler," *The American Statistician*, vol. 46, no. 3, pp. 167–174, 1992.

[34] V. Romero, A. Fornés, N. Serrano, J. A. Sánchez, A. H. Toselli, V. Frinken, E. Vidal, and J. Lladós, "The {ESPOSALLES} database: An ancient marriage license corpus for off-line handwriting recognition," *Pattern Recognition*, vol. 46, no. 6, pp. 1658 – 1669, 2013.

[35] H. Glucksman, "Classification of mixed-font alphabets by characteristic loci," *Proc. IEEE Comput. Conf.*, pp. 138–141, Sep. 1967.

[36] P. Domingos, "Alchemy - open source ai," http://alchemy.cs.washington.edu/.

[37] J. Davis and P. Domingos, "Deep transfer: A markov logic approach," *AI Magazine*, vol. 32, no. 1, pp. 51–53, 2011.