Representation of Name Sequences in Spanish using Context Free Grammar

Noé Alejandro Castro Sánchez, José Ángel Vera Félix,
Igor A. Bolshakov, Grigori Sidorov

Center for Computing Research (CIC)
National Politechnic Institute (IPN)
México DF, México
{ncastro, javera}@sagitario.cic.ipn.mx
{igor, sidorov}@cic.ipn.mx

Abstract. Proper names identification and classification is a problem neatly related with information retrieval and text tagging. One of the most complex situations in identification of the named entities is the great diversity of elements and organization forms that they present. In this paper we study all conditions that influence the constitution of names and surnames of Hispanic background, denominated Hispanic name sequences (NS). This study proposes a generative grammar approach that is based on certain rules that establish all possible elements that play a fundamental role in the formation of the named sequences. The application of this grammar allows for the development of the algorithm and the program that is able to detect and categorize NS in texts.

1 Introduction

Named entity recognition is an aspect that has been intensively investigated in many areas of NLP such as machine translation systems, information retrieval systems or natural language understanding systems. The "named entity" concept is used to refer the terms that refer to elements such as proper names, names of cities, news papers, books, rivers, songs, etc. Named Entity Recognition process would be easy if the set of named entities were small, but this is not so. For that reason we cannot use a simple list of names to resolve all the types of named entities recognition. Another aspect that makes difficult the use of lists of names is the ambiguity that is generated when some elements of the list have two or more meanings; for example, the word "*blanco*" can indicate a color: *carro blanco* (*white car*), or can indicate a named entity: *Cuahutémoc Blanco* (proper name).

In this work we focus on proper names that we denominated Hispanic Name Sequences (NS). Proper names identification is a subtask of Named Entity Recognition (NER) task at the Message Understanding Conference (MUC). The problem has been studied in the field of Information Extraction [7] for diverse purposes. For example, [1] employed proper names for an automatic newspaper articles classification. Information Extraction requires the robust handling of proper names for successful performance in diverse tasks such as pattern filling with correct entities that correspond to semantic roles [4], etc.

The NER contest in MUC was dedicated to the English language and different tools and huge resources were developed. For example, in [5] four modules were used for name identification: List Lookup, Part of speech tagger, Name parsing, and Name matching. The system in [6] recognizes proper names by matching the input against pre-stored lists of proper names, etc. Some NER systems also use lists of surnames, for example, [3].

In this paper, we propose a generative grammar that processes all elements of the Hispanic names. The grammar is based on certain rules that guide the NS construction and allow for the development of an algorithm and a program liable to distinguish and categorize NS in texts.

2 Named Sequences Formation

The Hispanic system of Name Sequences presents two sections of elements with different nature: **Name Sequences** and **Surname Sequences**. However, the last one, although it emerged and evolved at the same time in systems of other European languages presents some differences that could be confusing for someone not accustomed to the Hispanic system.

We will call **Complete Name Sequence** the structure that has two mentioned sequences. Let us describe in a more detailed form the structure and formation of the Complete Name Sequence.

2.1 Name Sequences

Name sequences can have one or more elements. Actually, they are limited to a maximum three elements because in practice the greater numbers does not appear in texts, at least, in Mexico. In the sequences, we distinguish three types of different entities:

- **Single names**: They are single word elements (*Victor, Jazmín, Alina,* etc.). They can be Male Single Name (*Juan, Edgar, Alberto,* etc.) or Female Single Name (*Gloria, Ana, María,* etc.).
- **Combined names**: They are formed with a union of two single names that can be of four different types, i.e., Male-Female, Male-Male, Female-Male, Female-Female. The sex of a name has the value of the first Single Name sex, e.g., *María José, María Jesús, José María*, etc. These combinations are relatively rare.
- **Composed names**: They are formed with a Single name, preposition *de*, possibly an article, and a second Single name, e.g., *de Jesús, de la Luz, del Carmen*, etc. Due to the fact that single names can have different sex, the sex of the first single name determines the composed name sex.

2.2 Surname Sequences

In the Hispanic surname system people have two surnames: father's surname (paternal surname) and mother's surname (maternal surname). Each one of these is inherited from father and mother paternal surnames correspondingly.

We define the following surname types:

- **Single surnames**: They are formed with a single word and it is the most common case in the Hispanic system (*García, Pérez, López,* etc.).
- **Bivalent surnames**: This group contains surnames that can be used as single names as well (*Santiago, Félix, Santos*, etc.).
- **Deado surnames**: They are formations which consist of single surnames or single names preceded by the preposition *de,* with or without an article (*Del Valle, De la Barrera, De León, De los Santos*, etc.).
- **Complex surnames**: They are formed with a single word (which not exist as a single surname) preceded by an article (*La Chica, La Moneda*, etc.) or with a group of words, where one of them can exists as a single surname (*Montes de Oca, Cabeza de Vaca*, etc.).
- **Composed surnames**: This is surname group that only contains paternal or maternal surnames (*Contreras y Molina, Lara-Aguayo, De Castro Muñoz*, etc.).

3 Generative Grammar

We define the following symbols as terminal symbols:

1. Conjunction '*y*'.
2. Preposition '*de'*.
3. Hyphens.
4. Single names and one-word surnames in the dictionary.

Other symbols are non terminal.

3.1 New Terms types

Our grammar is defined with a more compact terms group that we show in Figure 1:

Figure 1. Term definition

NmlSeq	Named Sequence	Snam	Surname
CmpteNmlSeq	Complete Named Sequence	PatSn	Father's Surname
NamSeq	Names Sequence	MatSn	Mother's Surname
SnamSeq	Surnames Sequence	SgleSnam	Single Surname
SgleNam	Single Name	DeadoSn	Deado Surname
CpsedN	Composed Name	CplxSn	Complex Surname
CmbN	Combined Name	BvltSn	Bivalent Surname
CplmtyN	Complementary Name	CpsedSn	Composed Surname
Prgtor	Progenitor	Sgmt(j)	Segment *j*
Pat	Paternal	Mal	Male
Mat	Maternal	Fem	Female

3.2 Rules Definition

In Figure 2 the rules are shown that define all possible variants and elements that participate in the Hispanic name sequences formation:

Figure 2. Grammar rules

1. NmlSeq → NamSeq (Sex) | SnamSeq | NamSeq (Sex) SnamSeq
2. NamSeq (Sex) → SgleNam (1, Sex)
   | SgleNam (1, Sex) SgleNam (2, Sex)
   | SgleNam (1, Sex) SgleNam (2, Sex) SgleNam (3, Sex)
   | CpsedN (1, Sex) | SgleNam (1, Sex) CpsedN (2, Sex)
   | SgleNam (1, Sex) CpsedN (2, Sex^{-1})
   | CmbN (Sex) | SgleNam (1, Sex) CmbN (Sex)
3. CpsedN (*j*,Sex) → SgleNam (*j*, Sex) CplmtyN
4. SnamSeq → Snam(pat) | Snam(pat) Snam(mat)
5. Snam(Prgtor) → SgleSnam | DeadoSnam | CplxSn | BvltSn | CpsedSn
6. CpsedSn → Sgmt(1) Sgmt(2) | Sgmt(1) *y* Sgmt(2) | Sgmt(1)–Sgmt(2)
   | Sgmt(1)–*y*–Sgmt(2)
7. Sgmt(*j*) → SgleSnam | DeadoSn | CplxSn | BvltSn

Note 1. Each SgleNam (1, Sex), SgleNam (2, Sex) and SgleNam (3, Sex) element are different, i.e., SgleNam (1, Sex) ≠ SgleNam (2, Sex), SgleNam (2, Sex) ≠ SgleNam (3, Sex) and SgleNam (1, Sex) ≠ SgleNam (3, Sex)

Note 2. Sex^{-1} is the Sex complement (opposite value).

Note 3. In the second rule, due SgleNam (1, Sex) CpsedN (2, Sex^{-1}) have different sex, the fist element of the Composed Name must form a Combined Name with SgleNam (1, Sex).

3.3 Non terminal symbol definitions

Examples of the rules for the non terminal symbols are shown in Figure 3.

Figure 3. Non terminal symbols definitions

j → 1 | 2
Sex → Male | Female
SgleNam (*i*, *mal*) → *Carlos* | *Antonio* | *Rubén* | *Humberto* |...
SgleNam (*i*, *fem*) → *Luz* | *María* | *Ana* | *Jimena* | ...
CplmtyN → *de la Luz* | *del Carmen* | *del Refugio* | *de Jesús* | …
CmbN (*mal*) → *José Trinidad* | *José María* | …
CmbN (*fem*) → *María José* | *María Jesús* | …
SgleSnam → *Bautista* | *Castrejón* | *Torres* | *Páramo* | ...
DeadoSn → *Del Valle* | *De los Cobos* | *De los Santos* | *De León* | …
CplxSn → *La Moneda* | *La Chica* | *Montes de Oca* | *Cabeza de Vaca*
BvltSn → *Jorge* | *Santiago* | *Félix* | *Camilo* | *Alonso* | ...

The variable *i* can have values 1, 2 or 3.

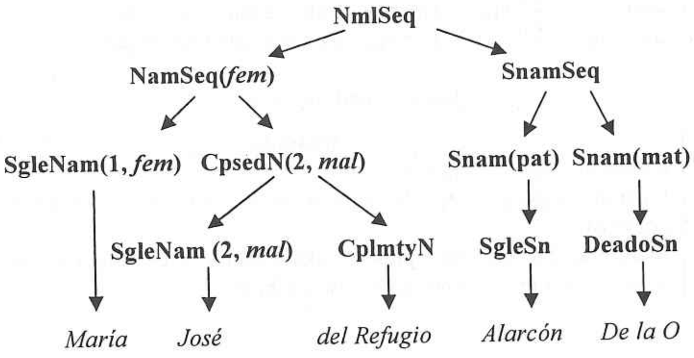
4 Grammatical Analyzer

Now we have the possibility to develop a program that applies our grammar rules.

We are going to develop the ascendant analysis scheme taking into account the simplest elements. We show this at the following syntactic analysis tree (Figure 4).

From NS *María José del Refugio Alarcón De la O* we obtain:

Figure 4. Non terminal symbols definitions



4.1 Program Description

The program is implemented using C++ and is based on dictionaries (data bases) of names and surnames. It can receive as input a text or independent sentences. The parsing process is defined in the following two points:

- The program starts examining each word in the text and comparing it with the dictionaries. If the word is not recognized as an NS element, it is ignored and placed into unrecognized list. On the other hand, if it belongs to the NS data base, it is saved into an array variable.
- If a word is not recognized and the array is not empty, then it is accomplished a tagging process where each word of the array is assigned with its respective tag in accordance with its place into the NS extracted. After the tagging process is finished, the result is put into the output text control.

4.2 Used Tags

Tags are placed after each element and they group the related elements. In the following tables we define tags giving their description and the values that they can be assigned.

Table 1. Terms meaning

Variable	Meaning
TotSeq	Total sequence.
NamSeq	Name sequence.
sex	Sex variable which can have the male (M) or female (F) values.
com1, com2	Final comments about the tagged element.

Table 2. Tags format

Tag	Meaning
(TotSeq, sex, com1)	Tag placed after a NS.
(NamSeq, sex)	Type 1: Tag placed after a name sequence.
(NamSeq, sex, com2)	Type 2: Tag placed after a name sequence.

Table 3. com1 meaning

Value	Meaning
Cmpl	Complete [Named sequence].
InCmpl	Incomplete [Sequence]. The name sequence or the surname sequence does not exist.
Indfnt	Indefinite. The sequence presents elements that are recognized as nominal elements but they are not correctly combined.

Table 4. com2 meaning

Value	Meaning
Indfnt	Indefinite. The sequence presents elements that are recognized as nominal elements but they are not correctly combined.
OpstSex	Opposite sex. Name sequence with different sex elements that are not recognized as combined names.
rep	Repeated. The name sequence present repeated elements.

Larg	Large. There are 4 or more elements in the name sequence.
Pat.surn.	Paternal surname
Mat.surn.	Maternal surname

4.3 Program Results

The following examples show the input NS that the program processes and the output it obtains:

Input: *Lorena Patricia del Carmen Muñoz Del Valle*
Output:

```
[ [[[Lorena(nom1,F)][Patricia del Carmen(nom2,F)]](NomSeq,F)]
 [Muñoz(pat.surn.)][Del Valle(mat.surn.)] ](TotSeq,F,Cmpl)
```

Input: *Miguel Ángel Ricardo Benítez*
Output:

```
[ [[[Miguel(nom1,M)] [Ángel(nom2,M)] [Ricardo(nom3,M)]]
 (NomSeq,M,larg)][Benítez(pat,surn)] ] (TotSeq,M,Cmpl)
```

Input: *María Carlota*
Output:

```
[ [[[María(nom1,F)][Carlota(nom2,F)]](NomSeq,F)] ](TotSeq,F,Incmpl)
```

5 Conclusions

The diversity of Hispanic Named Sequences makes difficult the creation of a model that allows for processing of all possible cases.

This paper examines the most common NS and it is adjusted according the most general rules, i.e., it does not take into account some cases that could be valid in some Spanish-speaking countries, though these cases are very rare in Mexico. For example, in Spain, the law prohibits the Name Sequences with more than 3 elements, while in Mexico this restriction does not exist. Nevertheless, we added this rule in our grammar because actually it is normal that the Name Sequences have only 2 or 3 elements. The other example is related with that composed surnames that have more than two elements, i.e., cases like *Sánchez de Anda y Martínez Salgado, Martí y Zayas-Bazán, Bancés y Fernández-Criado,* etc., where each of them is a paternal or maternal surname, can be identified by our grammar like two surnames: paternal surname and maternal surname.

The developed program depends of the data bases of names and surnames, so the entities that belong to NS but do not exist in our dictionaries will not be recognized. The solution can be the usage of a morphologic analysis [2], e.g., the *-ez* ending means 'son of': *Sánchez* means *son of Sancho*, *Fernández* means *son of Fernando*, and so on. This kind of analysis could allow identify the elements that are not in the dictionary.

Acknowledgments

This work was realized under the partial support of Mexican Government (CONACyT, SNI) and National Polytechnic Institute, Mexico (CIC, PIFI, CGPI, COFAA).

References

1. Friburger, N. and D. Maurel. *Textual Similarity Based on Proper Names.* Mathematical Formal Information Retrieval (MFIR'2002) 155-167.
2. Gelbukh A., and G. Sidorov. *Approach to construction of automatic morphological analysis systems for inflective languages with little effort.* In: Computational Linguistics and Intelligent Text Processing. Proc. CICLing-2003, 4th International Conference on Intelligent Text Processing and Computational Linguistics, February 15–22, 2003, Mexico City. LNCS, N 2588, Springer-Verlag, pp. 215–220.
3. Krupka, G. and K. Hausman. *Description of the NetOwl(TM) extractos system as used for MUC-7.* In Sixth Message Understanding Conference (MUC-7) 1998.
4. MUC: Proceedings of the Sixth Message Understanding Conference. (MUC-6). Morgan Kaufmann (1995).
5. Stevenson and Gaizauskas. *Using Corpus-derived Name List for name Entity Recognition.* In: Proc. Of ANLP", Seattle (2000).
6. Wakao, T., R. Gaizauskas & Y. Wilks.: *Evaluation of an Algorithm for the Recognition and Classification of Proper Names.* In Proccedings of the 16th International Conferencie on Computational Linguistics (COLING96), Copenhagen (1996) 418-423.
7. Wilks Y. *Information Extraction as a core language technology.* In M. T. Pazienza (ed.), Information Extraction, Springer-Verlag, Berlin (1997).