

RECURSOS LÉXICOS Y GRAMÁTICAS PARA RECUPERACIÓN DE INFORMACIÓN

Tecnologías Pregunta-Respuesta



Roberto Maestre Oscar Marín Miró



@rmaestrem @paradigmabols @oscarmarinmiro

INDICE

web 4.0

consorcio

UPM, Paradigma Tecnológico, Gesfor, Yell, ...

objetivos

crear una “caja” que entienda una pregunta en lenguaje natural en varios dominios y devuelva una respuesta avanzada

en produccion ...

INDICE

Pasos básicos del Text Mining

Raw text

NER

Gramáticas

Information extraction

Es completa? Establacer diálogo

INDICE

Pasos básicos del Text Mining

Unitex: <http://igm.univ-mlv.fr/~unitex/>

Python: nltk urllib json


NAME ENTITY RECOGNITION

*Reconocer y extraer menciones de entidades relevantes
en el texto*

“President Obama on Monday will call for a new minimum tax rate for individuals making more than \$1 million a year to ensure that they pay at least the same percentage of their earnings as other taxpayers, according to administration officials.”

NAME ENTITY RECOGNITION

Recurso: DBpedia Spotlight



Confidence:

Contextual score:

Prominence (support):

No 'common words'
Default Disambiguation
Show best candidate

President Obama on Monday will call for a new minimum tax rate for individuals making more than \$1 million a year to ensure that they pay at least the same percentage of their earnings as other taxpayers, according to administration officials.

NAME ENTITY RECOGNITION

basado en diccionarios

basado en reglas

machine learning

NAME ENTITY RECOGNITION

Enfoque basado en diccionarios

El objetivo es hacer matching de entidades relevantes utilizando diferentes recursos: ontologías, thesauros, ...

Este enfoque es muy sencillo, pero es sensible a errores de ortografía, master/MSc , etc ...

Dependiendo de la cobertura deseada puede ser un problema crear los diccionarios.

NAME ENTITY RECOGNITION

Enfoque basado en diccionarios

¿Como crear un diccionario para detectar los nombres de compañías de telefonía que operan en España?

¿Como crear un diccionario para detectar actores de television que tenga una cobertura lo mas grande posible?

NAME ENTITY RECOGNITION

Recurso léxico para crear diccionarios

The screenshot displays the Freebase Query Editor web application. The browser address bar shows `www.freebase.com/queryeditor`. The interface includes a top navigation bar with the Freebase logo, a "Query Editor" title, and links for "about", "refresh cache", "mqlread", "permanent link", and "Tinyurl". A yellow notification box prompts the user: "Don't know where to start? Watch this screencast here." with a "Close" button.

The main workspace is divided into several sections:

- Examples...**: Links to "MQL Tutorial" and "MQL Reference".
- Query Editor**: A text area containing a query:

```
[{
  "type": "/tv/tv_actor",
  "id": []
}]
```
- Query Assist**: A button to assist with query construction.
- Clean Up** and **Run**: Buttons to manage the query.
- Tools**: A tabbed interface with options: "Variables", "Envelope", "Custom Envelope", and "Settings".
- Results of tools**: A section showing the output of the selected tool.

The **Tools** section includes the following options:

- Qualify All Properties**: Expand all properties to their full IDs (e.g., "id" → "/type/object/id").
- Turn Inside Out**: Take the inner most query node {...} that contains the text cursor and make it the outermost query node.
- Generate Acre Template**: Generate an ACRE template that can render the result of this query.
- One-Liner**: Reformat query into a one-liner.

The **Results of tools** section shows the output of the "Qualify All Properties" tool, displaying a tree structure of the query results:

```
code: "/api/status/ok"
result: [ expand all • collapse all • 1-by-1
  0: {
    id: [
      0: "/en/dawn_wells"xp
      type: "/tv/tv_actor"xp
    ]
  }
  1: {
    id: [
      0: "/en/tina_louise"xp
      type: "/tv/tv_actor"xp
    ]
  }
  2: {
    id: [
      0: "/en/natalie_schafer"xp
      type: "/tv/tv_actor"xp
    ]
  }
  3: {
    id: [
      0: "/en/jim_backus"xp
      type: "/tv/tv_actor"xp
    ]
  }
  4: {
    id: [
      0: "/en/russell_johnson"xp
      type: "/tv/tv_actor"xp
    ]
  }
  5: {
    id: [
      0: "/en/bob_denver"xp
      type: "/tv/tv_actor"xp
    ]
  }
  6: {
    id: [
      0: "/en/mathew_st_patrick"xp
      type: "/tv/tv_actor"xp
    ]
  }
]
```


NAME ENTITY RECOGNITION

Enfoque basado en diccionarios

[https://api.freebase.com/api/service/mqlread?query={"query": "%20{"type": %20"/tv/tv_actor", %20"id": %20\[\]}"}](https://api.freebase.com/api/service/mqlread?query={)

```
import urllib
import json

# Query parameters
query = [{"type": "/tv/tv_actor", "id": []}]
query_envelope = {'query': query}
# Service url
service_url = 'http://api.freebase.com/api/service/mqlread'
url = service_url + '?query=' + json.dumps(query_envelope)
# Perform request
response = json.loads(urllib.urlopen(url).read())
# Read results
for actor in response['result']:
    print actor['id']
```


NAME ENTITY RECOGNITION

Enfoque basado en diccionarios

Para resolver el problema de los errores ortográficos podemos utilizar algún enfoque basado en distancias de palabras, e.g.: *Probabilistic Term Variant Generator for Biomedical Terms**

<http://www.mendeley.com/research/a-comparison-of-string-metrics-for-matching-names-and-records/>

<http://labs.paradigmatecnologico.com/2011/07/06/information-propagation-in-tweeters-network/>

$$DL(\text{barack}, \text{barak}) = 1$$

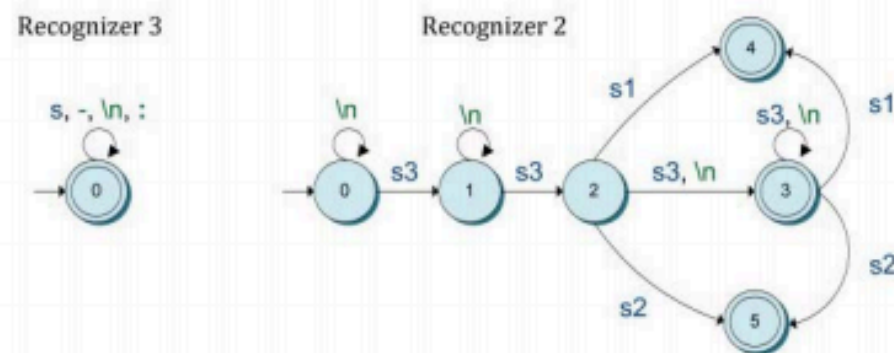
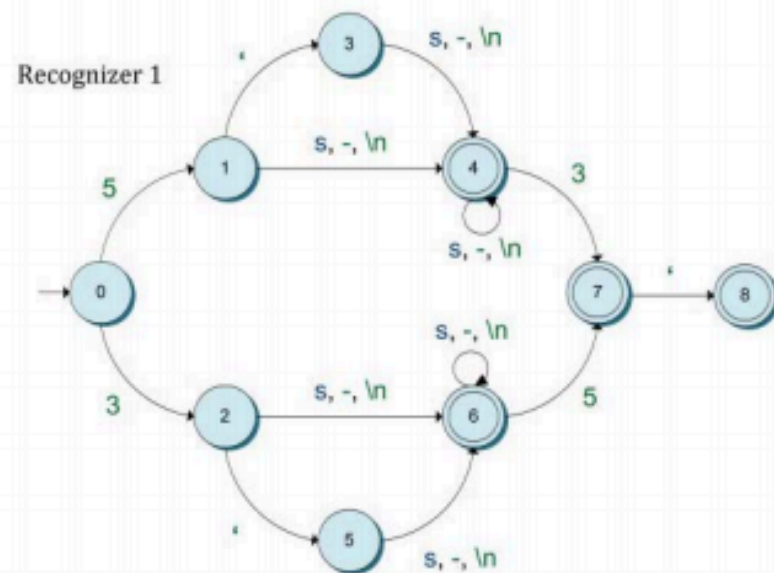
$$DL(\text{barack}, \text{varaka}) = 3$$

$$DL \leq 1$$

* Yoshimasa Tsuruoka and Jun'ichi Tsujii. 2003. Probabilistic term variant generator for biomedical terms. In *Proceedings of the 26th annual international ACM SIGIR conference on Research and development in information retrieval (SIGIR '03)*. ACM, New York, NY, USA, 167-173. DOI=10.1145/860435.860467 <http://doi.acm.org/10.1145/860435.860467>

NAME ENTITY RECOGNITION

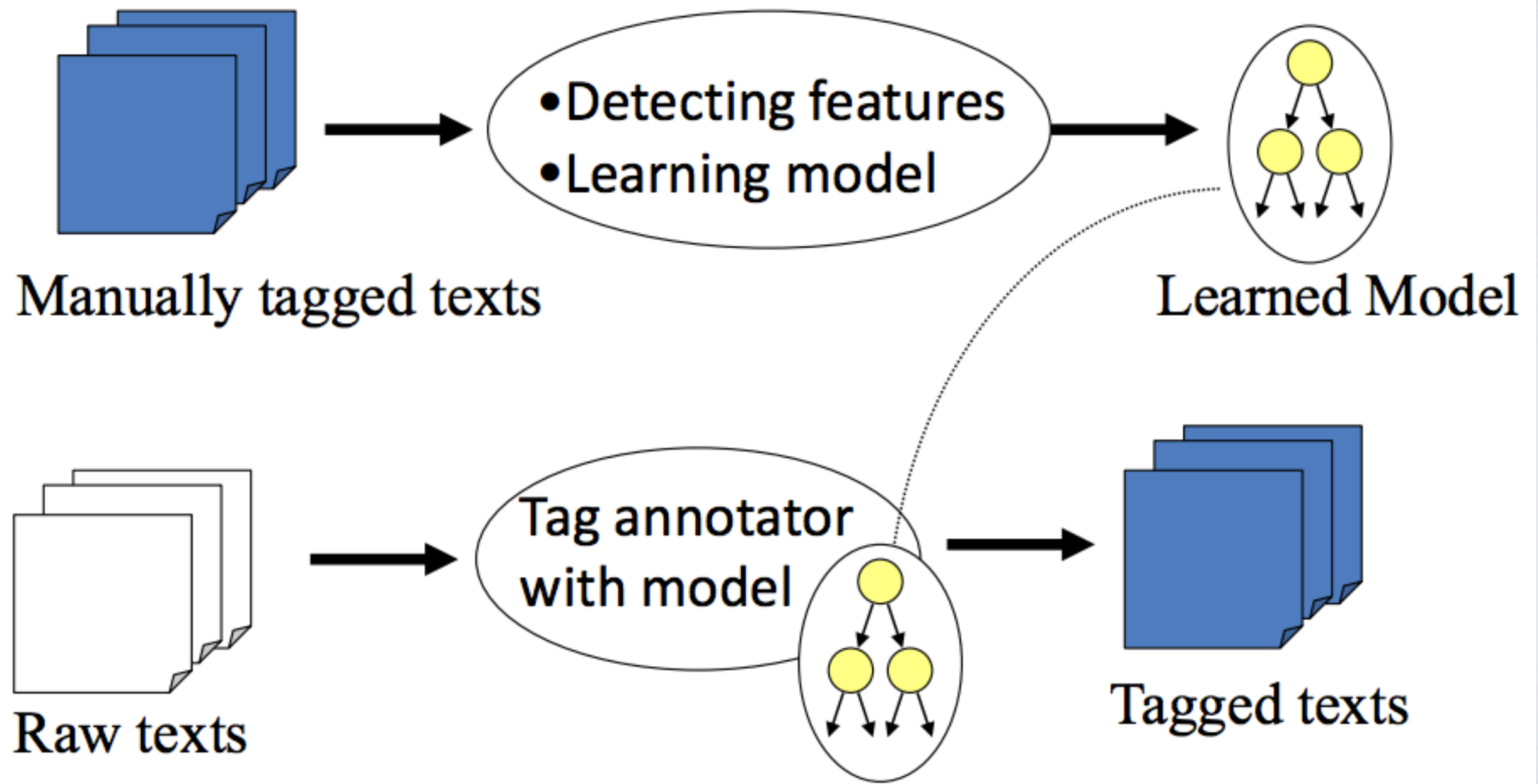
Rule based



Detector	PMID	Text String	List of Tokens
1	19781080	...primers AA247 (5'-TGCCATTGCCAAAGAGAC-3') and pLQ510-rp1...	{ "TGCCATTGCCAAAGAGAC" }
1	19664269	...mecA gene, mecAR (5'-TTACTCATGCCATACATAAATGGATA-\nGACG-3') and mecAF...	{ "TTACTCATGCCATACATAAATGGATA", "GACG" }
1	19379498	...specific primer pair traD-F (5'-caatgcttgatctatttgtag-3') and traD-R...	{ "caatgcttgatctatttgtag" }
1	19758438	...MY 09, 5-CGT CCM\n ARR GGA WAC TGA TC-3; where M = A/C, W = A/T...	{ "CGT", "CCM", "ARR", "GGA", "WAC", "TGA", "TC" }
2	19799780	B-globin outside R @ CTC AAG TTC TCA GGA TCC A @ 1st round PCR primer for Human Beta globin DNA	{ "CTC", "AAG", "TTC", "TCA", "GGA", "TCC", "A" }
2	18847469	btherm @ GAT GTG CCG GGC TCC TGC ATG @ This study	{ "GAT", "GTG", "CCG", "GGC", "TCC", "TGC", "ATG" }
2	18154687	Stx1 @ GTA CGT CTT TAC TGA TGA TTG ATA GTG GCA CAG GG @ 35 @ 73.5	{ "GTA", "CGT", "CTT", "TAC", "TGA", "TGA", "TTG", "ATA", "GTG", "GCA", "CAG", "GG" }
2	19558693	...are listed below.\n EP1- F ATG GTG GGC CAG CTT GTC\n EP1- R...	{ "ATG", "GTG", "GGC", "CAG", "CTT", "GTC" }
3	19754958	...with primer N309 (ACATGCGGATCCCTCGAGCCTTTGAA-\nGATGACTAACTCCCA) and N297...	{ "ACATGCGGATCCCTCGAGCCTTTGAA", "GATGACTAACTCCCA" }
3	19737401	...and 3' AAGCT TGGTA CCTCA CTGCA\nGCAGA GCGCT GAGGC CCAGC AGCAC. The resulting PCR...	{ "AAGCT", "TGGTA", "CCTCA", "CTGCA", "GCAGA", "GCGCT", "GAGGC", "CCAGC", "AGCAC" }
3	19149882	1 @ XAC0340 @ 432 @ gATACCCCATATgAATgCgAT	{ "gATACCCCATATgAATgCgAT" }
3	19775435	20 @ F:GAGATGGATTAACCAGATGTCTTAAAACTATCGTAAC	{ ":", "GAGATGGATTAACCAGATGTCTTAAAACTATCGTAAC" }

NAME ENTITY RECOGNITION

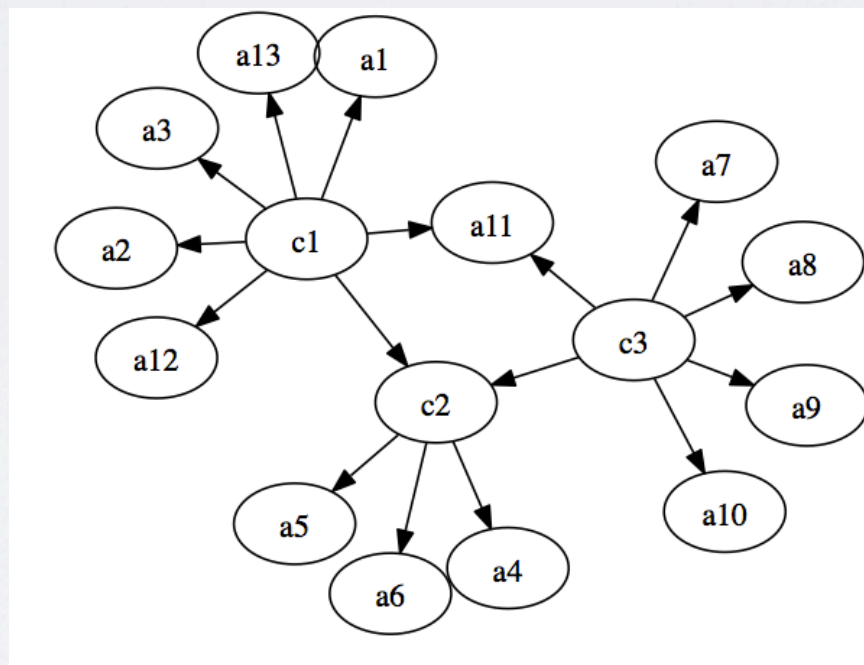
Machine learning based



NAME ENTITY RECOGNITION

Paradigma labs. Wikipedia NER

Usar las relaciones entre categorías, subcategorías y artículos



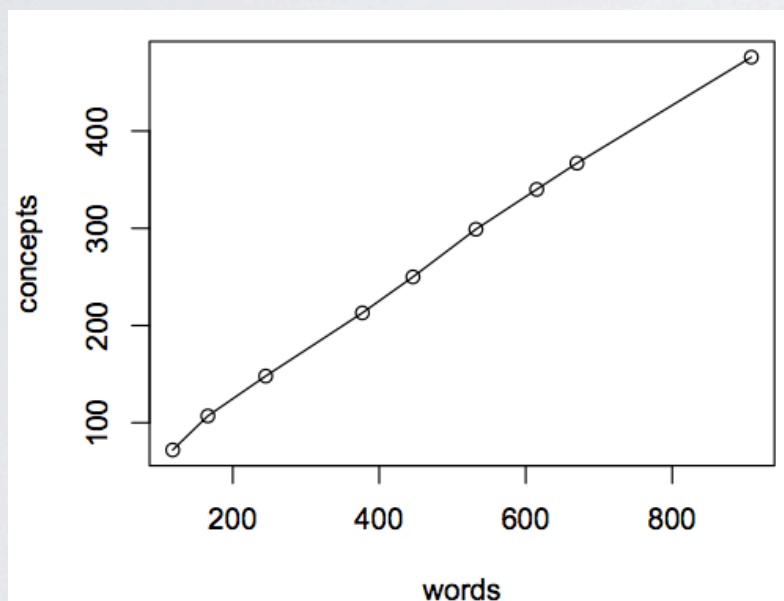
<https://twitter.com/#!/rmaestrem/status/162656378555596801/photo/1>

Networkx (20%) + Kyoto cabinet (80%)

NAME ENTITY RECOGNITION

Paradigma labs. Wikipedia NER

Podemos detectar 25.000.000 de categorías



Podemos utilizar los enlaces entre categorías

```
{
  - clusters: {
    - mercado financiero: {
      mariano rajoy: 7,
      jose luis rodriguez zapatero: 7,
      politicos: 6,
      crisis economica: 6
    },
    - actimel: {
      yogur: 3,
      danone: 2,
      lanjaron: 4
    }
  },
  sse_error: 3
}
```


NAME ENTITY RECOGNITION

Paradigma labs. Wikipedia NER

- La desambiguation la hacemos en base al “surface text”

$$E(w) = - \sum_{i \in R_s} P(r_i|w) \log P(r_i|w)$$

Zapatero va a arreglar mis zapatos

Zapatero

El término **zapatero** puede referirse a:

- cualquier cosa perteneciente o relativa a los **zapatos**;
- la **profesión de zapatero**, consistente en fabricar, arreglar o vender zapatos;
- el **mueble zapatero**, un mueble para guardar zapatos.

Personas con el apellido **Zapatero**:

- José Luis Rodríguez Zapatero (1960 -), presidente del Gobierno de España (2004 -);
- Virgilio Zapatero Gómez (1946 -), ministro español de Relaciones con las Cortes (1986 - 1993);
- Luis Arroyo Zapatero (1951 -), rector español de la Universidad de Castilla-La Mancha (1988 - 2003);
- Ismael Piñera Zapatero (1977 -), futbolista español;
- Carlos Arroyo Zapatero (1964 -), arquitecto español;
- Gonzalo Ruiz Zapatero (1954 -), catedrático español en Prehistoria.

Personas conocidas con el sobrenombre de Zapatero:

- Teódoto el Zapatero (finales del s. II), escritor cristiano de Bizancio;
- Simón el Zapatero (s. IV - s. III a.C.), discípulo directo de Sócrates.

```
<result name="response" numFound="20" start="0" maxScore="0.8154754">
  <doc>
    <float name="score"> 0.8154754 </float>
    <str name="from"> zapatero </str>
    <str name="texto"> profesion consistente fabricar arreglar vender zapatos </str>
    <str name="to"> profesion de zapatero </str>
  </doc>
  <doc>
    <float name="score"> 0.63237333 </float>
    <str name="from"> zapatero </str>
    <str name="texto"> mueble guardar zapatos </str>
    <str name="to"> mueble zapatero </str>
  </doc>
  <doc>
    <float name="score"> 0.31618667 </float>
    <str name="from"> zapatero </str>
    <str name="texto"> zapato cualquier cosa perteneciente relativa s </str>
    <str name="to"> zapato </str>
  </doc>
  <doc>
    <float name="score"> 0.0 </float>
    <str name="from"> zapatero </str>
    <str name="texto"> jose luis rodriguez 1960 presidente gobierno españa 2004 </str>
    <str name="to"> jose luis rodriguez zapatero </str>
  </doc>
```

NAME ENTITY RECOGNITION

Paradigma labs. Wikipedia NER

NER

[clear](#)

Inlinks threshold:

1

Text:

zapatero como presidente del gobierno lo hizo bien

Analyze

Analyzing "zapatero como presidente del gobierno lo hizo bien"

Entities:

zapatero

presidente del gobierno

bien

Disambiguations:

jose luis rodriguez zapatero (zapatero)

Wikipedia Distances

[clear](#)

From:

informática

To:

telecomunicaciones

Analyze

Analyzing "informática - telecomunicaciones"

Distance:

3.0

Wikipedia Distances

[clear](#)

From:

danone

To:

oro

Analyze

Analyzing "danone - oro"

Distance:

7.0

POSTAGGING

NLTK

<http://www.nltk.org/>

```
import nltk

###
#   Simple example of tokenize and POS tag with NLTK
###
text = "All that is gold does not glitter. not all those that wander
are lost."

sentences = nltk.sent_tokenize(text)

for sent in sentences:
    tokens = nltk.word_tokenize(sent)
    pos_tags = nltk.pos_tag(tokens)
    print pos_tags
```

POSTAGGING

<http://nlp.isi.upc.edu/freeling/demo/demo.php>

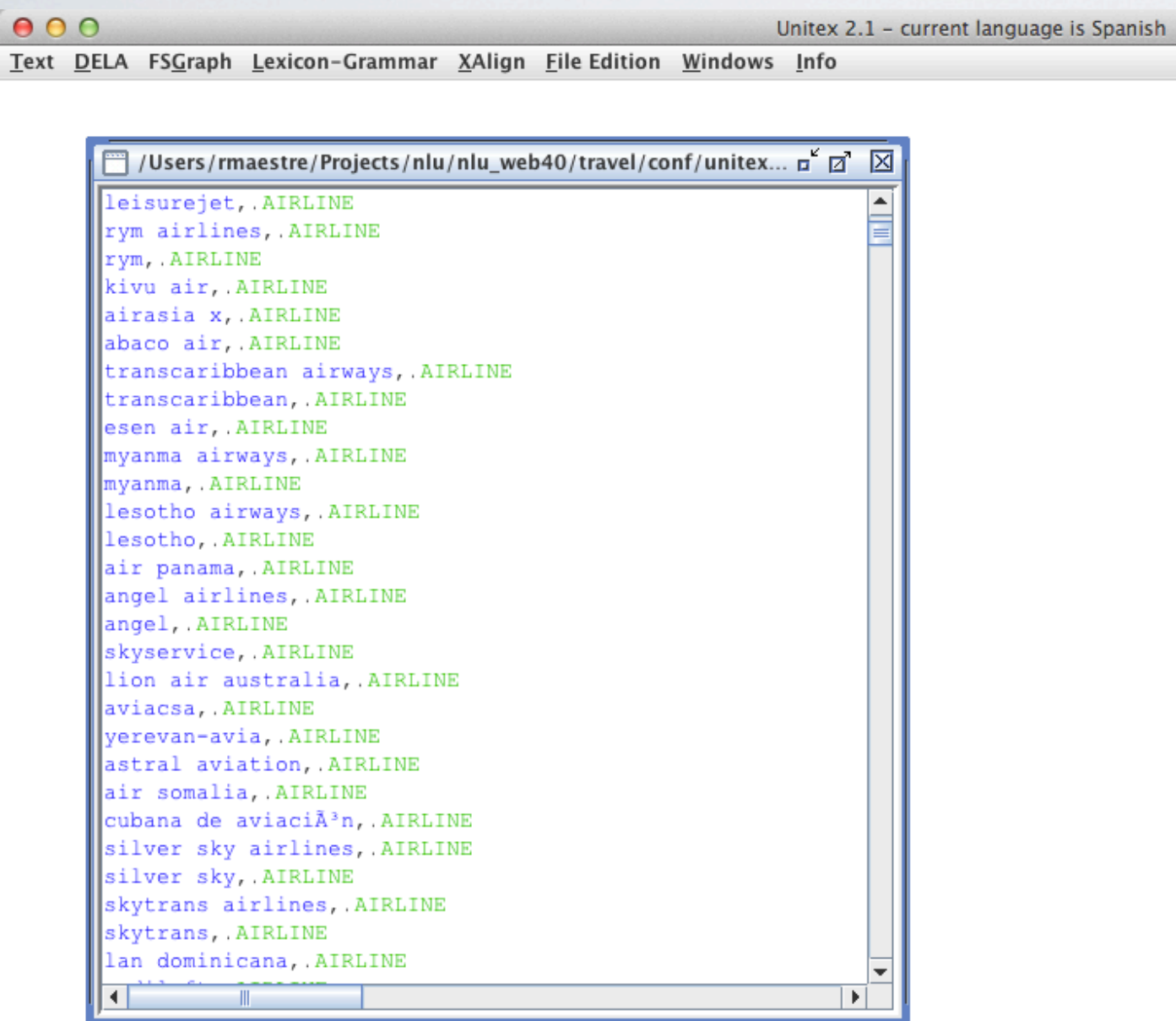
El gato come pescado y bebe agua.

El	gato	come	pescado	y	bebe	agua	.
<i>el</i>	<i>gato</i>	<i>comer</i>	<i>pescado</i>	<i>y</i>	<i>beber</i>	<i>agua</i>	<i>.</i>
DA0MS0	NCMS000	VMIP3S0	NCMS000	CC	VMIP3S0	NCCS000	Fp
1	1	0.75	0.833333	0.999812	0.994868	0.973333	1
		<i>comer</i>	<i>pescar</i>	<i>y</i>	<i>beber</i>	<i>aguar</i>	
		VMM02S0	VMP00SM	NCFS000	VMM02S0	VMIP3S0	
		0.25	0.166667	0.000188324	0.00513196	0.0133333	
						<i>aguar</i>	
						VMM02S0	
						0.0133333	

UNITEX

<http://igm.univ-mlv.fr/~unitex/UnitexManual2.1.pdf>

Diccionarios.



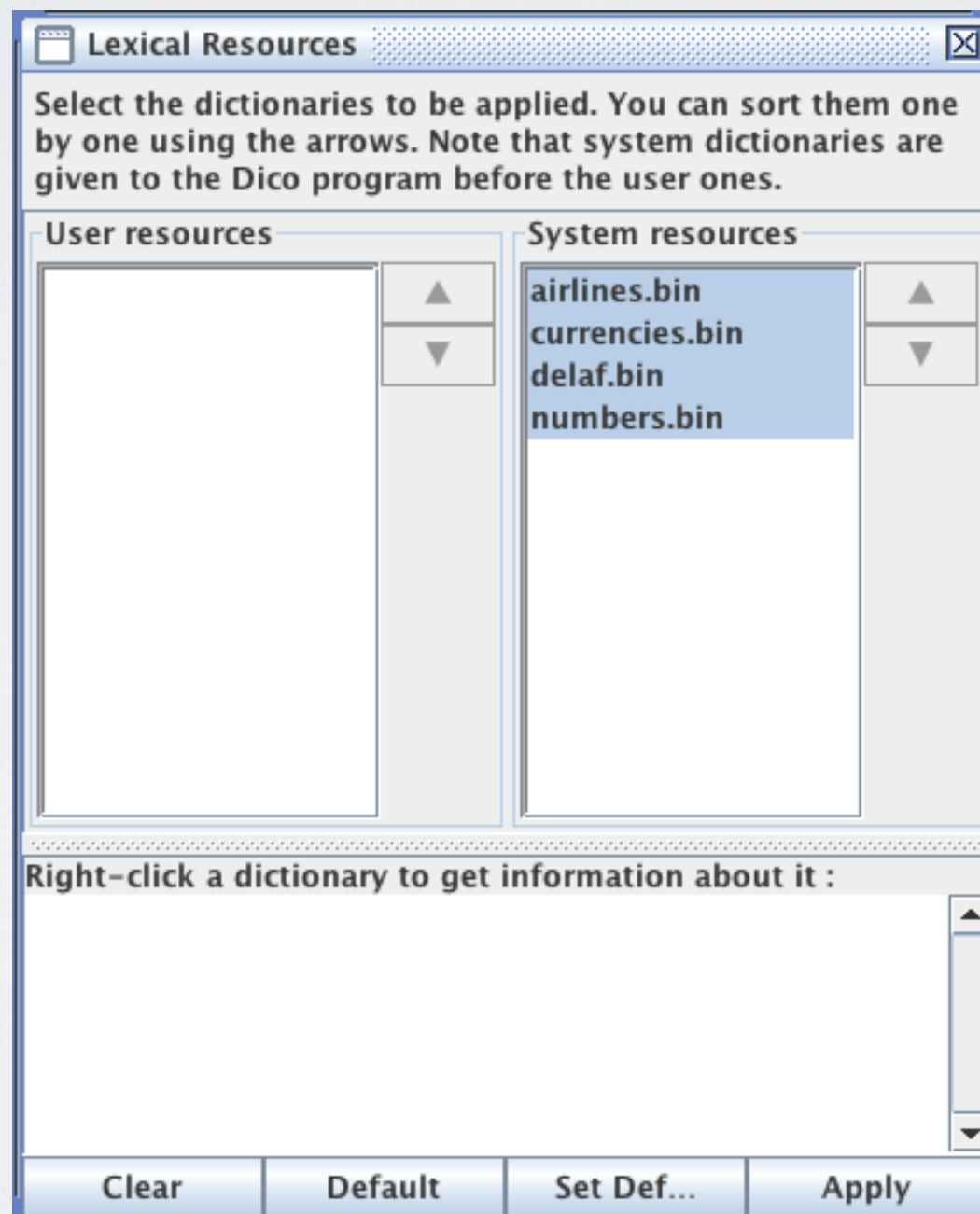
Forma conjugada , lemma • informacion semantica

Crear 3 diccionarios

Numbers
Currency
Airlines

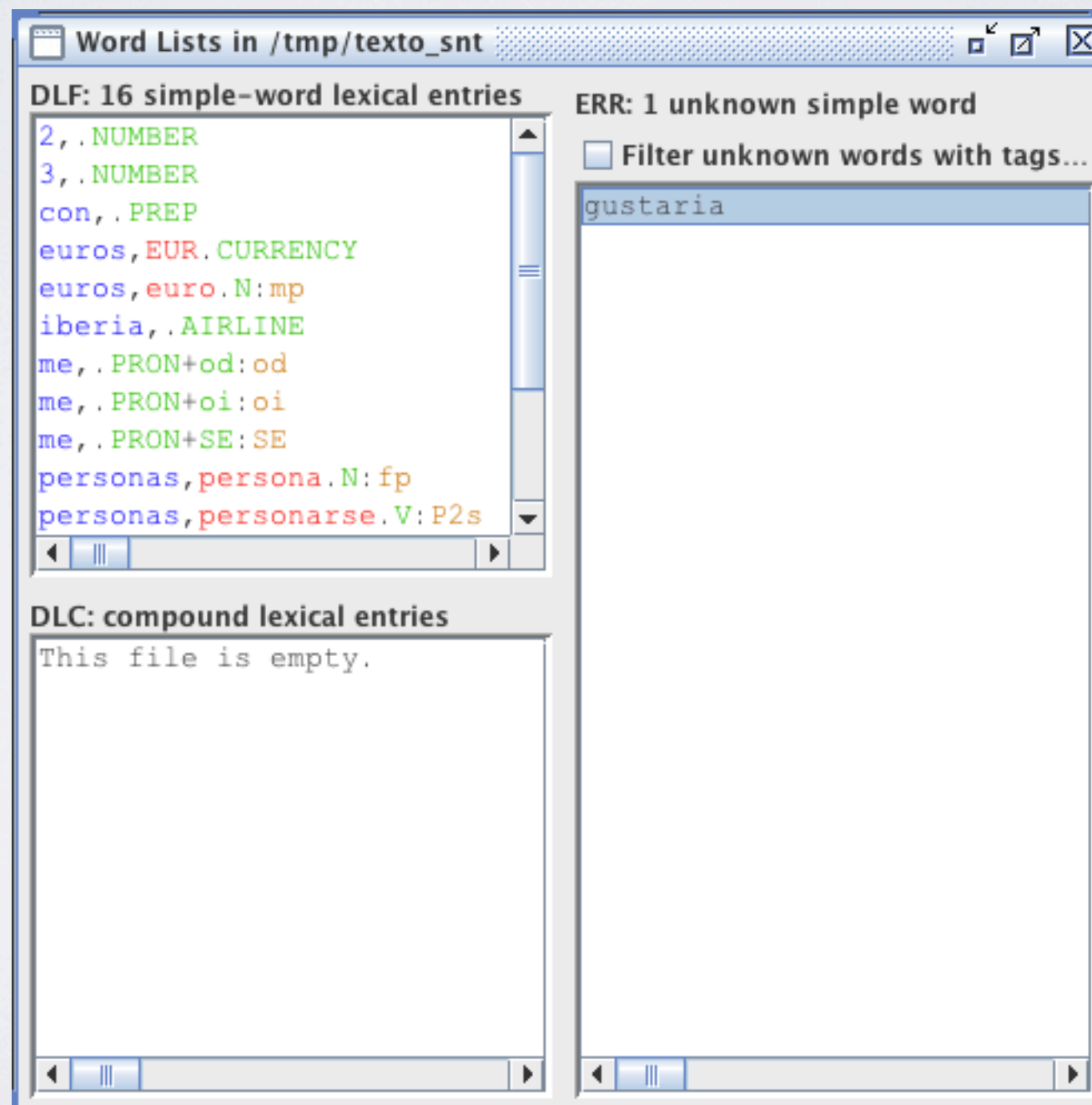
UNITEX

Aplicar recursos.



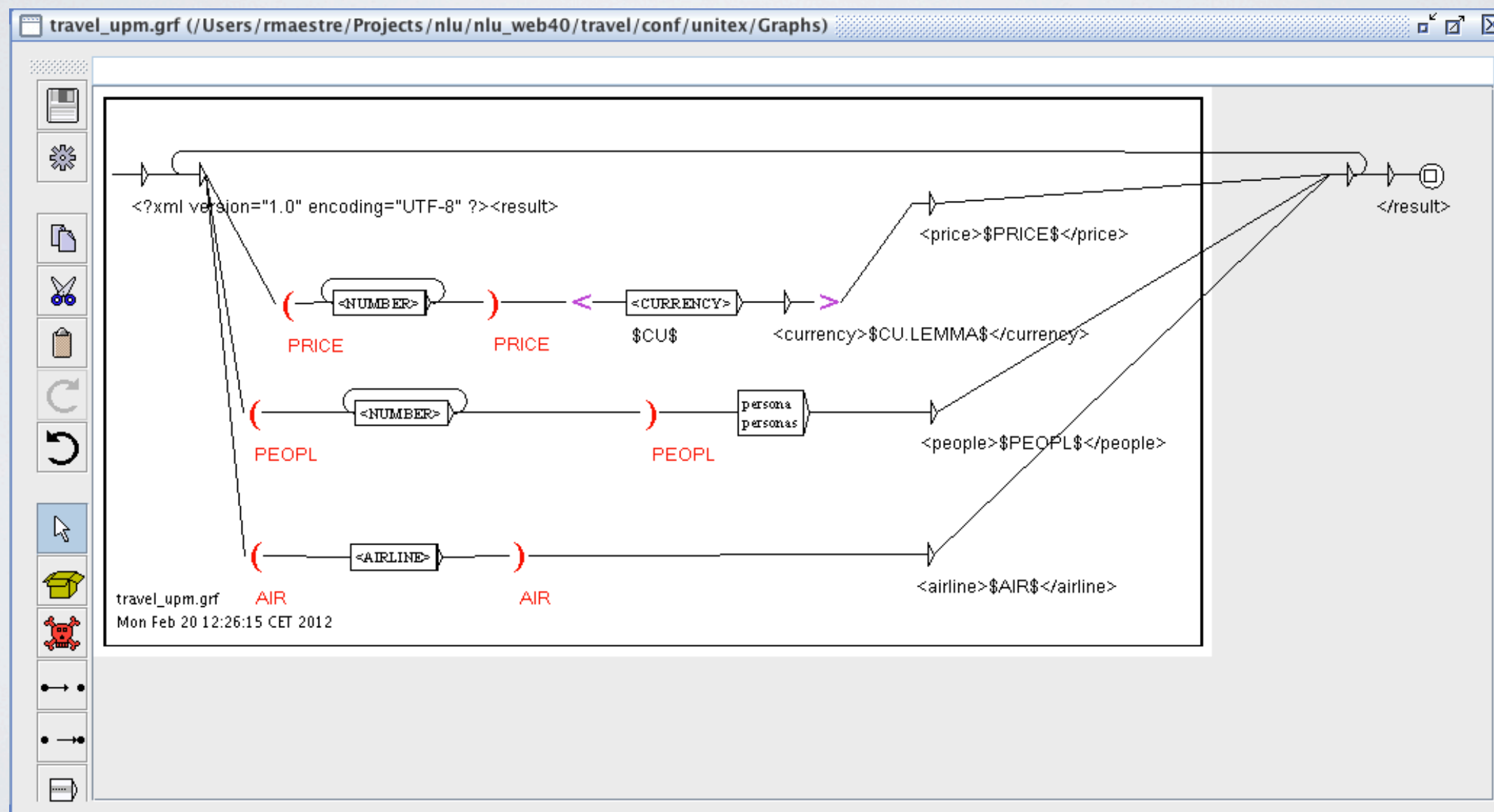
UNITEX

Resultado.



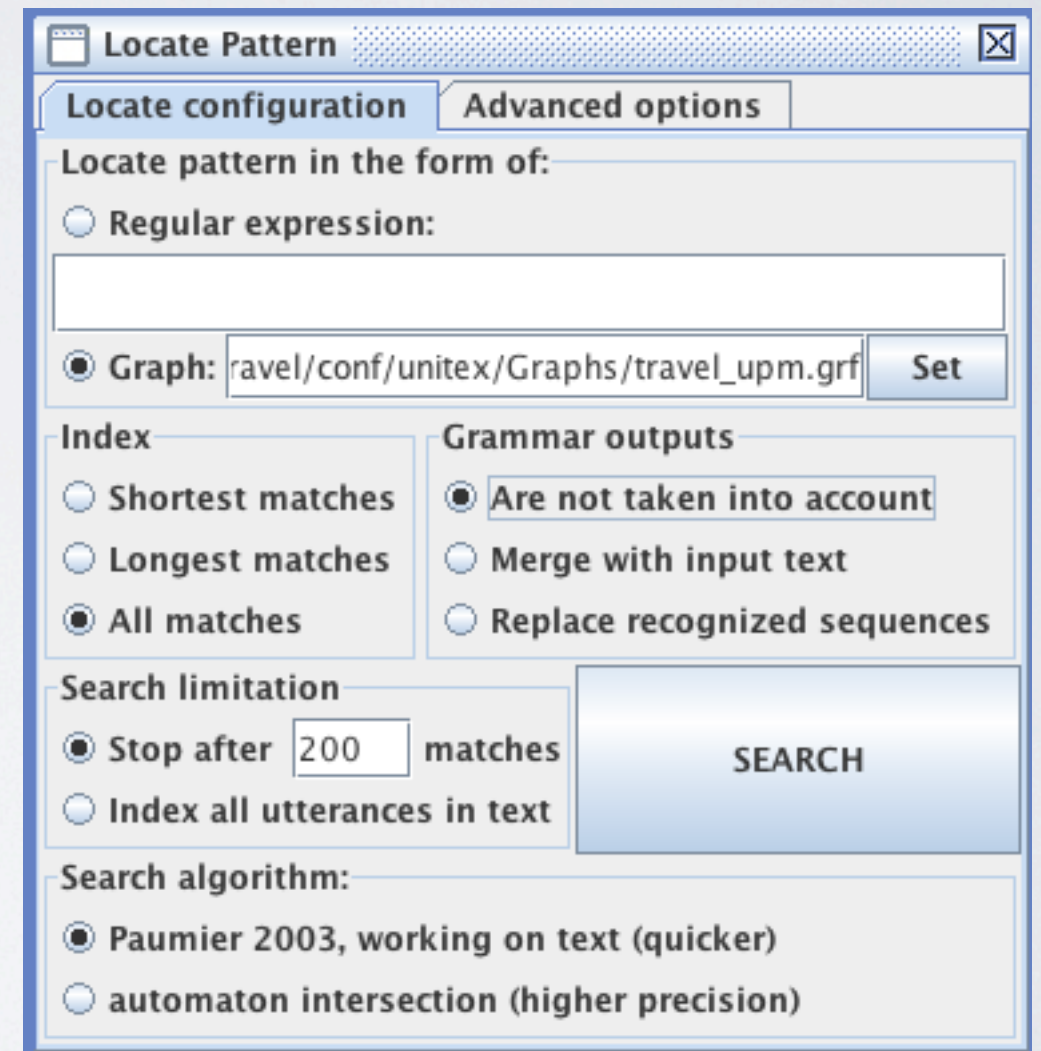
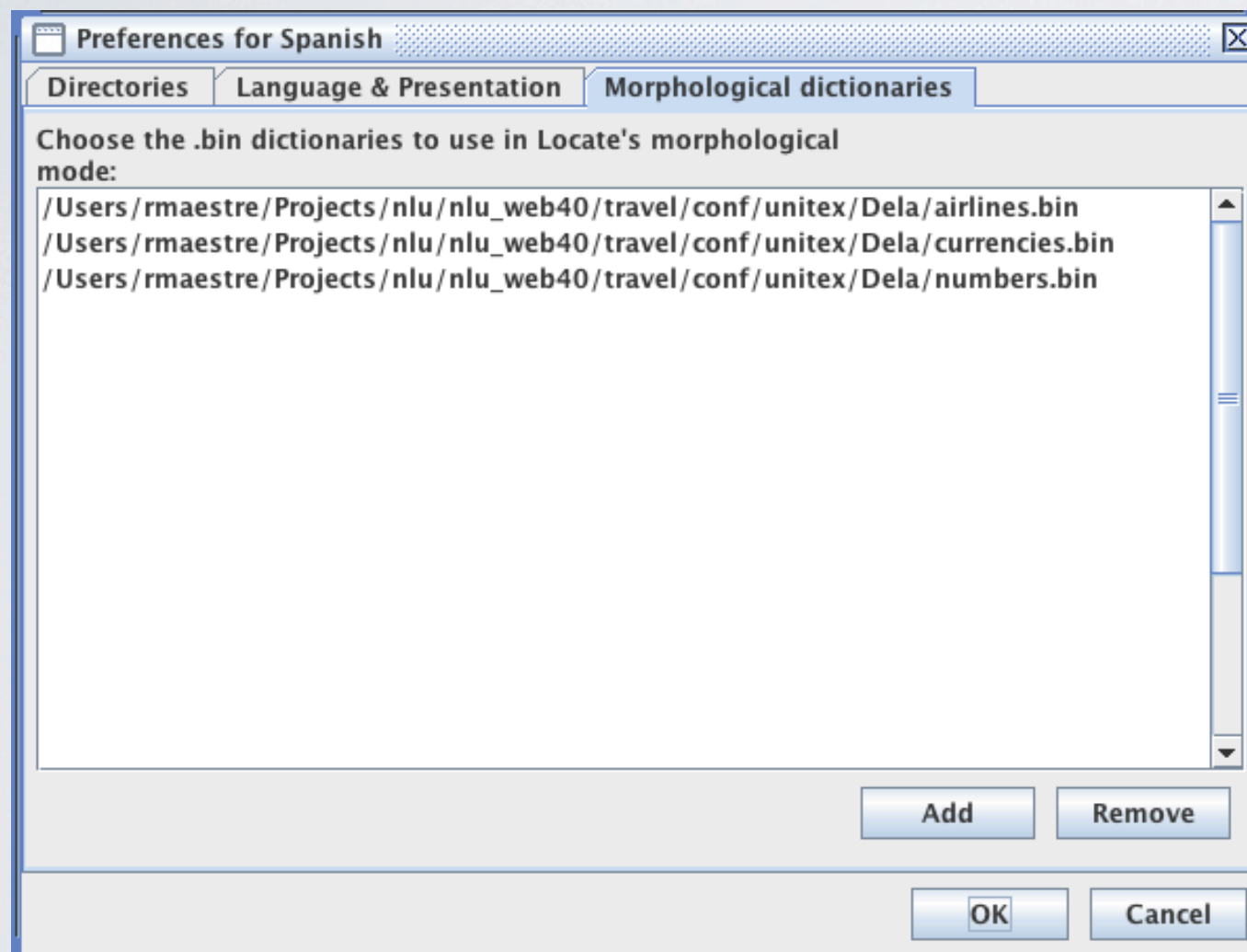
UNITEX

Diseño de la Gramática.



GRAMATICAS

Aplicar gramática



```
import pyunitex
u = pyunitex.Unitex()
u.Convert('-s', 'UTF8', 'file_name', '-o', 'file_name_converted')
```

<https://github.com/moliware/pyunitex>

GRAMATICAS

Ejercicio práctico

Sistema que detecte el grupo de música que quiere comprar un usuario y que detecte también el tipo de formato: cd, dvd, vinilo, mp3, etc ... Al final necesitaré saber si es DIGITAL o ANALÓGICO.

Puede ser que el usuario solo quiera ver un listado de grupos por estilo musical.

Quiero comprar el CD de Kings of lion
Quiero ver todos los grupos de dance

RECURSOS LÉXICOS Y GRAMÁTICAS PARA RECUPERACIÓN DE INFORMACIÓN

Tecnologías Pregunta-Respuesta

Muchas gracias



Roberto Maestre Oscar Marín Miró



@rmaestrem @paradigmallabs @oscarmarinmiro