IEPY: Una plataforma para Extracción de Información en Python

Franco M. Luque

Grupo de Procesamiento de Lenguaje Natural
Universidad Nacional de Córdoba & CONICET
Córdoba, Argentina

XII Jornadas de Ciencias de la Computación
Rosario, 16 de Octubre de 2014

Resumen de la Charla

Resumen de la Charla

Introducción: Procesamiento de Lenguaje Natural

- El campo de las Ciencias de la Computación que estudia el procesamiento automático del lenguaje humano.
- Tareas relevantes:
  - Segmentación de palabras (tokenización) y oraciones,
  - etiquetado de tipos de palabras (categorías léxicas o part-of-speech),
  - análisis sintáctico,
  - extracción de información,
  - etc.
- En esta charla nos concentraremos en la tarea de Extracción de Información o *Information Extraction (IE)*.

Resumen de la Charla

Extracción de Información

- Trata el problema del análisis de texto no estructurado para encontrar y estructurar determinada información de interés.
- Sub-tareas:
  - Reconocimiento de Entidades Nombradas (*Named Entity Recognition* o NER): Etiquetado de menciones de entidades de diferentes tipos (personas, lugares, fechas, etc.) en textos de lenguaje natural.
  - Extracción de Relaciones (*Relationship Extraction* o RE): Identificación de menciones de relaciones entre entidades en textos de lenguaje natural (presencia de persona en un lugar, vínculos entre personas, etc.).

Resumen de la Charla

Reconocimiento de Entidades Nombradas

- Etiquetado de menciones de entidades de diferentes tipos (personas, lugares, fechas, etc.) en textos de lenguaje natural.
- Problemas relacionados:
  - Desambiguación/clasificación de entidades: Mapeo de menciones de entidades a entidades/objetos en un dominio semántico (una base de datos, la Wikipedia, etc.).
  - Resolución de co-referencias: Etiquetado de a qué entidades se refieren los pronombres (y otras referencias) en un texto.

Ejemplo

```
<person>Jim</person> bought 300 shares of
<organization>Acme Corp.</organization> in <date>2006</date> .
```

Reconocimiento de Entidades Nombradas

- Métodos:
  1. Basados en reglas (e.g. *gazetteers*, expresiones regulares).
  2. Estadísticos.
- Codificación:
  1. Etiquetado a nivel de tokens.
  2. Etiquetado a nivel de segmentos.
- Modelos:
  1. Clasificación ordenada.
  2. Markov Models de diferentes sabores (HMMs, MEMMs y CMMs).
  3. Conditional Random Fields (estado del arte).
  4. Otros...

Reconocimiento de Entidades Nombradas

- Métodos:
  1. Basados en reglas (e.g. *gazetteers*, expresiones regulares).
  2. Estadísticos.
- Codificación:
  1. Etiquetado a nivel de tokens.
  2. Etiquetado a nivel de segmentos.
- Modelos:
  1. Clasificación ordenada.
  2. Markov Models de diferentes sabores (HMMs, MEMMs y CMMs).
  3. Conditional Random Fields (estado del arte).
  4. Otros...
- Esta charla: 2, 1, 1.

Métodos Estadísticos

- Modelos y algoritmos de Aprendizaje por Computadora (*Machine Learning*).
- Basados en datos (corpus):
  - Datos anotados: aprendizaje supervisado.
  - Datos no anotados: aprendizaje no-supervisado.
  - Combinación de los anteriores: aprendizaje semi-supervisado.
  - Oráculo humano: aprendizaje interactivo.
- Más flexibles que los métodos basados en reglas:
  - Toleran ruido en los datos.
  - Toleran situaciones no visas o contempladas.
  - Son independientes del idioma o dominio temático.
- Evaluación experimental:
  - Comparación con corpus estándar de referencia (*gold standard*).
  - Métricas: precision, recall y F1.

Etiquetado a nivel de tokens

- Se codifica el problema de reconocimiento de entidades como un problema de etiquetado de tokens.
- Codificación BIO:
  - `O`: no es parte de una entidad.
  - `B-<E>`: comienzo de entidad de tipo `<E>`.
  - `I-<E>`: interior de entidad de tipo `<E>`.

Ejemplo

```
Here is my review of Fermat  's  last theorem  by S.     Singh
O    O  O  O      O  B-PER   O   O    O        O  B-PER  I-PER
```

Clasificación ordenada

- Se etiquetan los tokens de cada oración de izquierda a derecha.
- Se utiliza un clasificador para decidir la etiqueta de cada token.
- El clasificador toma como input la información conocida hasta el momento.

Ejemplo: input para clasificar "Fermat"

```
Here is my review of Fermat  's  last theorem  by S.  Singh
O    O  O  O      O  ?       ?   ?    ?        ?  ?   ?
```

Clasificación ordenada: Características (*Features*)

- Para etiquetar cada token, se obtiene primero un vector de características relevantes para la elección de la etiqueta.
- Features típicos:
  - La palabra en mínusculas,
  - el POS tag de la palabra,
  - si la palabra empieza en mayúsculas, si es todo mayúsculas,
  - si la palabra pertenece a un diccionario dado,
  - los mismos features sobre palabras anteriores o siguientes,
  - las etiquetas de las palabras anteriores,
  - etc.

Clasificación ordenada: Características (*Features*)

Ejemplo

```
Here is my review of Fermat 's  last theorem by S.     Singh
O    O  O  O      O  B-PER  O   O    O       O  B-PER  I-PER
```

Vector de características para el token `Fermat`:

```
{'word=fermat': 1.0,
 'pos=NNP': 1.0,
 'capitalized': 1.0,
 'all_caps': 0.0,
 'prev_word=of': 1.0,
 'prev_label=O': 1.0,
 'is_surname': 1.0,
 ...
}
```

(obs: son vectores potencialmente muy grandes!)

Clasificadores

- Un clasificador es una función de vectores de features en un conjunto finito de clases.
- La función se aprende a través de un algoritmo que toma como input vectores etiquetados con su clase.
- Pipeline de preprocesamiento de los vectores:
  - Escalado y ajuste de media.
  - Selección de features relevantes.
  - Reducción de dimensionalidad.
- Algunos clasificadores:
  - Árboles de decisión.
  - Naive y Multinomial Bayes.
  - Support Vector Machines (SVMs).
  - Métodos combinados.

Clasificadores: Árboles de decisión

- Un árbol cuyos nodos internos son condiciones sobre los features y cuyas hojas son etiquetas de salida.
- Ventajas:
  - Se pueden escribir manualmente o usar algoritmos de entrenamiento.
  - Soporta naturalmente clasificación multiclase.
  - Es fácilmente entendible e interpretable.
  - La clasificación provee una explicación.
- Desventajas:
  - Optimización NP-completa. Sólo algoritmos greedy.
  - Expresividad limitada.
  - Riesgo de baja generalización (*overfitting*).
  - La clasficación no provee un valor probabilístico.

Clasificadores: Árboles de decisión

Ejemplo

```
Here is my review of Fermat 's  last theorem by S.     Singh
O    O  O  O      O  B-PER  O   O    O        O  B-PER  ?
```

Árbol de decisión:

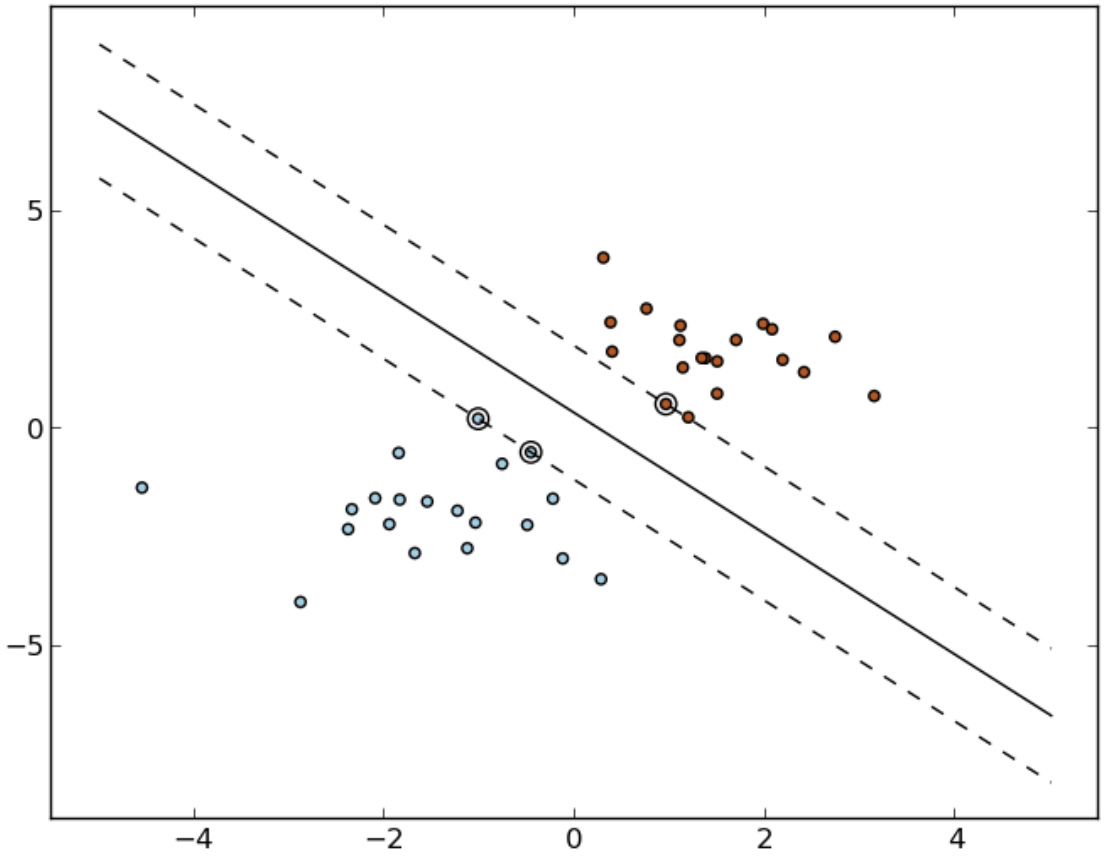
```
if ('prev_label=O'=1.0)
    if ('is_surname'=1.0)
        return 'B-PER'
    else
        return 'O'
else if ('prev_label=B-PER'=1.0 or 'prev_label=I-PER'=1.0)
    if ('capitalized'=1.0 or 'word=von'=1.0)
        return 'I-PER'
    else
        return 'O'
```

Clasificadores: Support Vector Machines (SVMs)

- Una división en dos partes del espacio de vectores de features utilizando un hiperplano (clasificador binario).
- Clasificación multiclase simulada: 1-vs-1 ó 1-vs-el-resto.
- Ventajas:
  - Alta expresividad: división no-lineal del espacio a través de kernels.
  - Alta generalización: para versión lineal y kernels no muy raros.
  - Optimización tratable: problema convexo.
  - Muy buenos resultados experimentales.
- Desventajas:
  - Difícil de interpretar.
  - La clasficación no provee un valor probabilístico.

Clasificadores: Support Vector Machines (SVMs)

- Algoritmo de aprendizaje: maximizar el tamaño de la franja entre los dos grupos de puntos (vectores) de entrenamiento.
- Variante soft-margin: tolera ruido y errores.



Resumen de la Charla

Extracción de Relaciones

- Identificación de menciones de relaciones entre entidades en textos de lenguaje natural (presencia de persona en un lugar, vínculos entre personas, etc.).
- Variantes:
  - Cantidad de entidades relacionadas: binarias, n-arias.
  - Tipos de relaciones: fijos o libres.
  - Dominio: abierto o cerrado.

Ejemplo

```
ADVISOR_OF('Donald Knuth', 'Broder'):
  <person>Broder</person> completed his doctoral thesis
  under the supervision of <person>Donald Knuth</person>
```

Extracción de Relaciones

- Tratamos el problema de relaciones binarias entre dos tipos fijos de entidades en un dominio cerrado.
- Simplificación adicional: se asume que toda la información relativa a una relación se encuentra dentro de una misma oración.
- Cada par de entidades del tipo apropiado mencionado en una misma oración es una posible evidencia de relación.
- Se codifica el problema de encontrar las relaciones como un problema de clasificación binaria de todas las posibles evidencias.
- Se pueden usar todas las técnicas de clasificación conocidas.

Extracción de Relaciones: Características (*Features*)

- Se obtiene primero un vector de características relevantes para la clasificación de una posible evidencia.
- Features típicos:
  - La distancia en tokens entre ambas entidades,
  - el orden en que las entidades aparecen,
  - si existen otras entidades entre las entidades en cuestión,
  - el conjunto de palabras (*bag-of-words*) entre ambos tokens,
  - el conjunto de POS (*bag-of-pos*) entre ambos tokens,
  - features sobre tokens del contexto,
  - distancia sintáctica entre las entidades,
  - etc.

Extracción de Relaciones: Características (*Features*)

Ejemplo positivo

```
<person>Broder</person> completed his doctoral thesis
under the supervision of <person>Donald Knuth</person>
```

Vector de características para la posible evidencia de `ADVISOR_OF('Donald Knuth', 'Broder')`:

```
{'distance': 9.0,
 'reverse_order': 1.0,
 'other_entities': 0.0,
 'between_word=completed': 1.0, 'between_word=his': 1.0, ...
 ...
}
```

(obs: los valores no son siempre binarios)

Extracción de Relaciones: Características (*Features*)

Ejemplo negativo

```
The invited speakers for the conference are
<person>A. Broder</person> , <person>R. Karp</person> ,
sand <person>D. Knuth</person> .
```

Vector de características para la posible evidencia de `ADVISOR_OF('D. Knuth', 'A. Broder')`:

```
{'distance': 6.0,
 'reverse_order': 1.0,
 'other_entities': 1.0,
 'between_word=,': 1.0, 'between_word=R.': 1.0, ...
 ...
}
```

(obs: los valores no son siempre binarios)

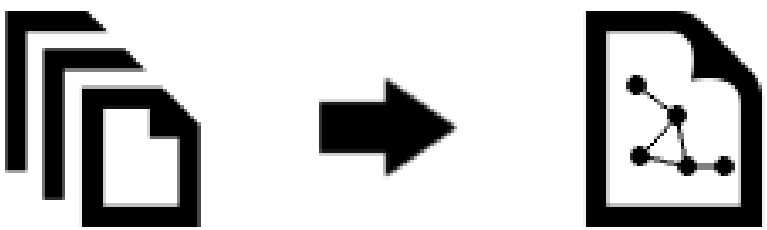
Resumen de la Charla

IEPY

- Plataforma para Extracción de Información en Python.
- Proyecto de código abierto:

  `http://iepy.machinalis.com/`
  `http://github.com/machinalis/iepy`
- Creado por la empresa cordobesa Machinalis en colaboración con el Grupo de PLN de la UNC:

  `http://www.machinalis.com/`
  `http://www.pln.famaf.unc.edu.ar/`
- Estado primario de desarrollo: API inestable.



IEPY: Características

- Pipeline de preprocesamiento configurable:
  - Tokenización y segmentación.
  - Etiquetado POS.
  - Resolución de co-referencias.
  - Análisis sintáctico.
- Reconocimiento de entidades:
  - Stanford (inglés y castellano).
  - Gazetteers (Freebase o propios).
- Extracción de relaciones:
  - Basados en reglas.
  - Aprendizaje interactivo con clasificador configurable.
- Interfaz web y por línea de comandos.
- Plataforma de experimentación automática.
- Documentación y ejemplos.

IEPY: Tecnologías

- Plataforma:
  - Python 3 (antes, Python 2).
- PLN:
  - NLTK
  - Stanford CoreNLP.
- Machine learning:
  - Scikit-learn.
- Modelo de datos:
  - Django ORM (antes: MongoDB).
- Interfaz web:
  - Django.

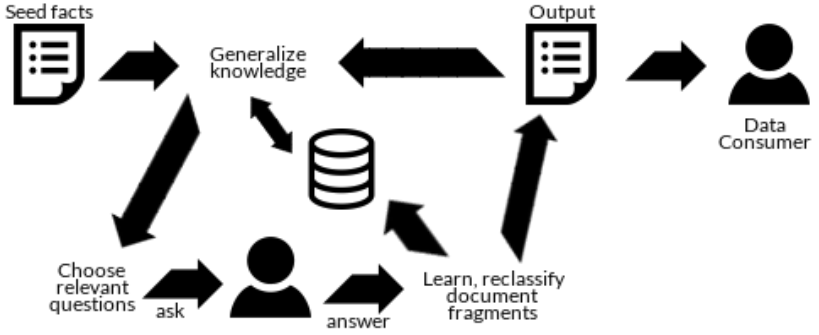
(demo)

IEPY: Extracción de Relaciones

- Input:
  - Documentos preprocesados (NER incluído).
  - La relación a extraer, y su tipo.
  - Un conjunto de instancias semilla (*seed facts*) para la relación.
- Configuración del algoritmo:
  - Conjunto de features.
  - Algoritmos y parámetros de preprocesamiento del clasificador (scaling, selección de features, etc.).
  - Algoritmo de clasificación y sus parámetros.
  - Valores de confidencia para la aceptación de nuevas instancias.
- Output:
  - El conjunto de instancias de la relación aprendidas.

IEPY: Algoritmo de Aprendizaje interactivo

- Primera iteración:
  1. IEPY genera preguntas a partir de las semillas.
- Ciclo principal:
  1. El usuario responde preguntas hasta que se cansa.
  2. IEPY incorpora las respuestas y (re)entrena el clasificador.
  3. IEPY (re)clasifica todas las posibles evidencias.
  4. De acuerdo a la confidencia, IEPY acepta nuevas instancias y genera nuevas preguntas para el usuario.



Resumen de la Charla

Aplicaciones: Archivo de la Memoria

- Convenio con el Archivo de la Memoria de la Provincia de Córdoba:

  `http://www.apm.gov.ar/`

- Búsqueda de información relevante para los juicios por delitos de lesa humanidad.
- Características:
  - Documentos: De origen militar (texto semi-estructurado).
  - Entidades: Personas, lugares y fechas.
  - Relación: Traslados de personas a lugares en determinadas fechas (3-aria).
- Métodos:
  - Segmentación de los documentos: Estadístico (clasificadores).
  - NER y RE: Basados en reglas (expresiones regulares sobre objetos).
- Por ahora no se usa el núcleo de IEPY.

Aplicaciones: Resoluciones UNC

- Proyecto de código abierto iniciado en el HackatONG+Program.AR Córdoba 2014:

  `http://hackatong-programar.github.io/`
  `http://github.com/HackatONG-ProgramAR/resoluciones-unc`
- Sistematización de información de planta docente para hacer diagnósticos (crecimiento, concursos, etc.).
- Características:
  - Documentos: Resoluciones universitarias (texto semi-estructurado).
  - Entidades: Personas, cargos y fechas.
  - Relaciones: eventos como designaciones interinas y por concurso, licencias, renuncias, etc.
- Métodos:
  - Sólo hay NER basado en reglas (expresiones regulares).
- Estado de avance embrionario.

Conclusiones

- La Extracción de Información es una de las tareas más complejas que aborda el Procesamiento de Lenguaje Natural.
- IEPY tiene potencial para varios propósitos:
  - Puede ser usado por investigadores como plataforma de experimentación.
  - Puede ser usado por programadores con pocos conocimientos sobre PLN y Machine Learning para desarrollar aplicaciones.
- Ya existen aplicaciones exitosas que utilizan la plataforma IEPY.

Trabajo Futuro

- IEPY:
  - Desambiguación de entidades.
  - NER estadístico propio (en part. usando métodos espectrales).
  - RE estadístico para relaciones n-arias.
  - Aprendizaje interactivo con etiquetado de features.
  - etc., etc., etc.
- Aplicaciones:
  - Archivo de la Memoria: Incorporar métodos estadísticos e interacción con el usuario.
  - Resoluciones UNC: Empezar!

¡Gracias! ¿Preguntas?