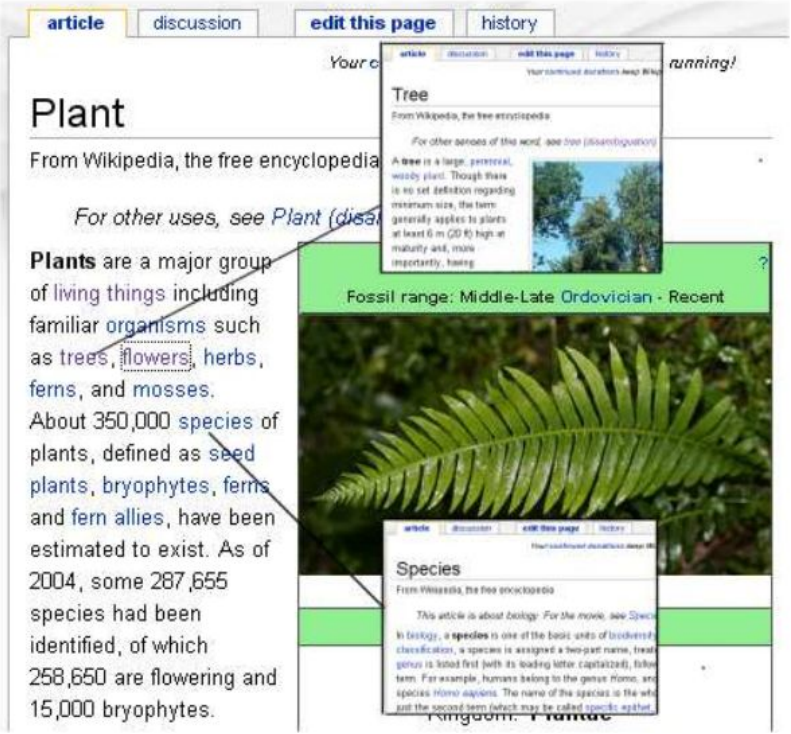
Wikify! Linking Documents to Encyclopedic Knowledge

PLN 2015 - FaMAF, UNC
Pablo Pastore

Introducción

- Wikipedia se ha convertido en la fuente más grande de conocimiento enciclopédico (+5M artículos!)
- Extenso vocabulario
- Se puede usar como corpus para extracción de *keywords* y *word sense disambiguation*
- Nos enfocaremos en estudiar un sistema que automatice la tarea de *text wikification*



Introducción

- Cada artículo en Wikipedia posee un *id* de referencia

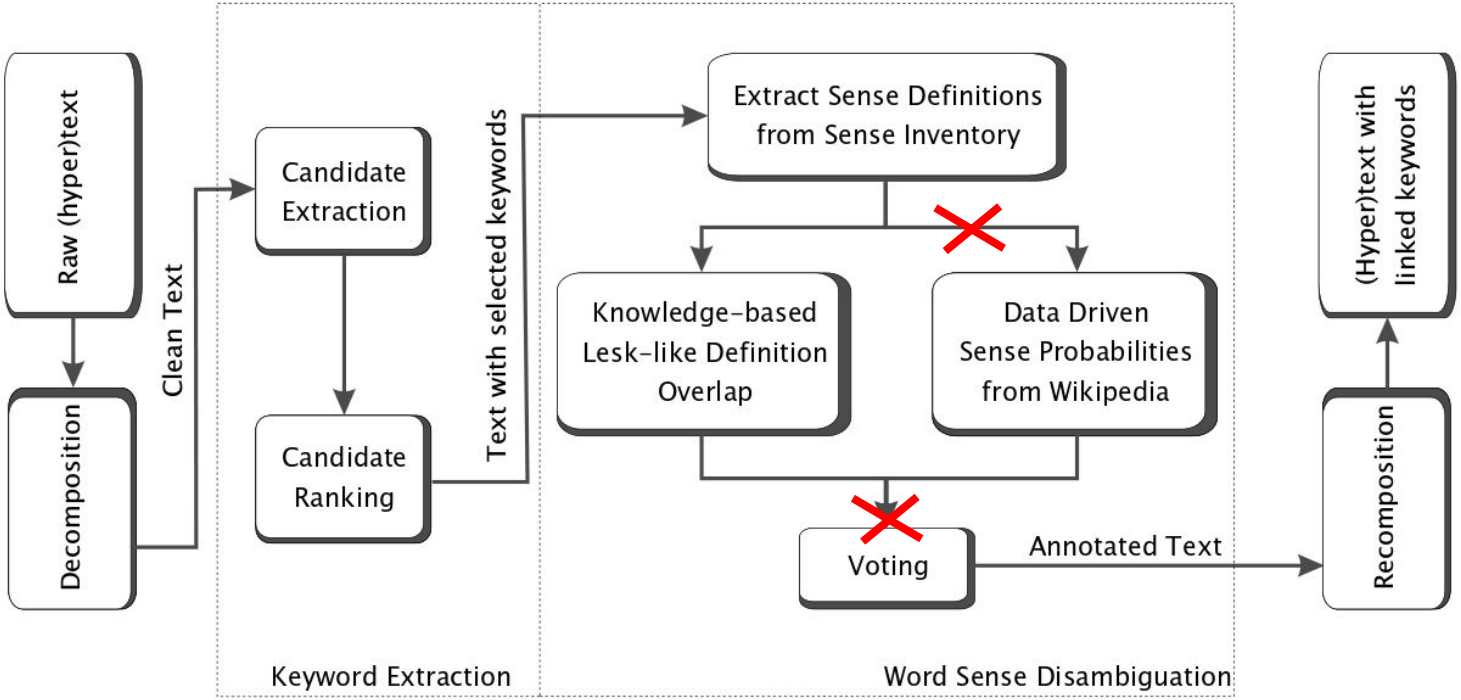
- Los *hyperlinks* dentro de Wikipedia son creados usando este *id*

Ejemplo:

“Henry Barnard, American educationalist, was born in Hartford, Connecticut”

“Henry Barnard, [[United States|American]] [[educationalist]], was born in [[Hartford, Connecticut]]”

Arquitectura Del Sistema



Keyword Extraction

Extracción:

- Extraer todas las keywords sin el id: *[[United States|American]], [[educationalist]]*
- Borrar las que aparecen menos de 5 veces
- Recolectar hasta 3-gramas
- Descartar números y nonwords (i, me, my, myself, we, our ...)

Ranking (Keyphraseness):

- $P(\text{keyword} \mid W) \approx \text{count}(D_{\text{keyword}}) / \text{count}(D_W)$

Keyword Extraction

Evaluación:

- Cómo seleccionar la cantidad adecuada de keywords?
  - Calcular el radio promedio entre el número de keywords por artículo y la cantidad de palabras en este = **4%**
- Entrenar y evaluar: Separar aleatoriamente el corpus en train y test
  - De test solo usar artículos cuyo radio de keywords esté entre 3% y 5%

	Train: 1,5M artículos Test: 85 artículos Vocabulario: 1,9M palabras	Train: 4M artículos Test: 300K artículos Vocabulario: 2,2M palabras
Precision	53.3%	65.32%
Recall	55.90%	60.33%
F-measure	54.63%	62.72%

Word Sense Disambiguation

- **Lesk**
  - Asume que las palabras en cierta sección del texto comparten el mismo tópico
  - Implementación simple: dada una palabra para desambiguar, comparar su contexto con el de cada una de las posibles definiciones
  - Pros: Rápido, fácil de usar y con muchas implementaciones (también en nltk)
  - Cons: Pocas definiciones comparado con el vocabulario de wikipedia (es inutil para millones de palabras)
- **Usando modelos de machine learning supervisado**
  - Para cada palabra ambigua usar las siguientes features: la palabra mencionada, palabras que la rodean (con POS tags) y 5 palabras mas comunes en el texto. Usar como etiqueta la palabra desambiguada
  - Pros: Usando Naïve Bayes se puede obtener alta precisión
  - Cons: No es trivial implementarlo, el principal problema está en la alta dimensionalidad
    - se podría usar word2vec

Word Sense Disambiguation

Evaluación:

- Actualmente en wikipedia existe una gran cantidad de keywords que referencian a artículos cuyo nombre ha cambiado, wikipedia maneja estos casos automáticamente, pero en este modelo no y esto claramente afecta la performance

	Train: 1,5M artículos Test: 85 artículos Vocabulario: 1,9M palabras	Train: 4M artículos Test: 1M artículos Vocabulario: 2,2M palabras
Method	Precision	
Most frequent sense	87.03%	80.17%
Lesk	80.63%	12.12%
Lesk + Most frequent sense	-	76.82%

Dificultades adicionales

- Poca documentación acerca del formato del corpus
- El corpus contiene muchas otras páginas de wikipedia además de los artículos
- Demasiadas palabras o caracteres raros en el texto
- Dado el tamaño del corpus hay que recorrerlo iterativamente (con mucho cuidado en la forma de limpiar nodos ya leídos)