http://en.wikipedia.org/wiki/Word_sense_disambiguation

Word Sense Disambiguation

Word sense disambiguation (WSD) is an open problem of natural language processing, which comprises the process of identifying which sense of a word (i.e. meaning) is used in any given sentence, when the word has a number of distinct senses (polysemy).

WSD

- disambiguating word senses has the potential to improve many natural language processing tasks, such as machine translation, question-answering, information retrieval, and text classification.
- in their most basic form, WSD algorithms take as input a word in context along with a fixed inventory of potential word senses, and return as output the correct word sense for that use.

what is WSD

WordNet Sense	Spanish Translation	Roget Category	Target Word in Context
bass^4	lubina	FISH/INSECT	...fish as Pacific salmon and striped **bass** and...
bass^4	lubina	FISH/INSECT	...produce filets of smoked **bass** or sturgeon...
bass^7	bajo	MUSIC	...exciting jazz **bass** player since Ray Brown...
bass^7	bajo	MUSIC	...play **bass** because he doesn't have to solo...

Extracting Feature Vectors

> If one examines the words in a book, one at a time as through an opaque mask with a hole in it one word wide, then it is obviously impossible to determine, one at a time, the meaning of the words. [. . .] But if one lengthens the slit in the opaque mask, until one can see not only the central word in question but also say *N* words on either side, then if *N* is large enough one can unambiguously decide the meaning of the central word. [. . .]
>
> The practical question is : "*What minimum value of N will, at least in a tolerable fraction of cases, lead to the correct choice of meaning for the central word?*"

non è ancora stata trovata una risposta

Ide and Véronis (1998)

feature vectors

- to extract useful features from such a window, a minimal amount of processing is first performed on the sentence containing the window.
  - this processing typically includes part-of-speech (POS) tagging, lemmatization or stemming, and in some cases syntactic parsing to reveal information such as head words and dependency relations.
  - context features relevant to the target word can then be extracted from this enriched input.
- a feature vector consisting of numeric or nominal values is used to encode this linguistic information as an input to most machine learning algorithms.

collocational *vs.* bag-of-words

- two classes of features are generally extracted: collocational features and bag-of-words features.
- a collocation is a word or phrase in a position-specific relationship to a target word (i.e., *exactly one word to the right*, or *exactly 4 words to the left*, and so on).

preserva l'info. sulla sequenza
non c'è info. sulla struttura interna

collocational features

- let us consider a case where we have to disambiguate the word *bass* in the following WSJ sentence:
  - *An electric guitar and bass player stand off to one side, not really part of the scene…*

collocational features

*An electric guitar and **bass** player stand off to one side, not really part of the scene…*

- example of a collocational feature-vector, extracted from a window of two words to the right and left of the target word, made up of the words themselves and their respective parts-of-speech, i.e.,
  - $[w_{i-2}, POS_{i-2}, w_{i-1}, POS_{i-1}, w_{i+1}, POS_{i+1}, w_{i+2}, POS_{i+2}]$
- would yield the following vector:

  [guitar, NN, and, CC, player, NN, stand, VB]

bag-of-words approaches

- a bag-of-words means an unordered set of words, ignoring their exact position.
  - the simplest bag-of-words approach represents the context of a target word by a vector of features, each binary feature indicating whether a vocabulary word *w* does or doesn't occur in the context.

bag-of-words approaches

An electric guitar and bass player stand off to one side, not really part of the scene…

- for example a bag-of-words vector consisting of the 12 most frequent content words from a collection of bass sentences drawn from the WSJ corpus would have the following ordered word feature set:

  [fishing, big, sound, player, fly, rod, pound, double, runs, playing, guitar, band]

- using these word features with a window size of 10, in the example would be represented by the following binary vector:

  [0,0,0,1,0,0,0,0,0,0,1,0]

↑ invece che solo 0 o 1, potrebbe essere usato il conteggio delle occorrenze

the Lesk Algorithm

- by far the most well-studied dictionary-based algorithm for sense disambiguation is the Lesk algorithm.

ancora usato come baseline in competiz. internazionali

the Lesk Algorithm

```
1  function SimplifiedLesk(word,sentence)
2  returns best sense of word
3  best-sense ← most frequent sense for word
4  max-overlap ← 0
5  context ← set of words in sentence
6  for all senses of word do
7     signature ← set of words in the gloss and examples of sense
8     overlap ← ComputeOverlap(signature,context)
9     if overlap > max-overlap then
10       max-overlap ← overlap
11       best-sense ← sense
12    end if
13 end for
14 return best-sense
```

euristica, funziona molto bene

] inizializz.

← individuo il senso che massimizza l'intersezione

the Lesk Algorithm

- as an example of the Lesk algorithm at work, consider disambiguating the word bank in the following context:

  the bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

the Lesk Algorithm

the bank can guarantee deposits will eventually cover future tuition costs because it invests in adjustable-rate mortgage securities.

- given the following two WordNet senses:

***bank*^{1}**	Gloss:	a financial institution that accepts deposits and channels the money into lending activities
	Examples:	"he cashed a check at the bank", "that bank holds the mortgage on my home"
***bank*^{2}**	Gloss:	sloping land (especially the slope beside a body of water)
	Examples:	"they pulled the canoe up on the bank", ...

example problem

- let us consider the three senses of the noun *ash* in WordNet, along with their definition.
  - $sense_1$: *the residue that remains when something is burned* ;
  - $sense_2$: *any of various deciduous pinnate-leaved ornamental or timber trees of the genus Fraxinus*;
  - $sense_3$: *strong elastic wood of any of various ash trees; used for furniture and tool handles and sporting goods such as baseball bats*.

example problem

- let us suppose we want to disambiguate the term *ash* occurring in the two contexts:
  - $context_1$: *The house was burnt to ashes while the owner returned.*;
  - $context_2$: *This table is made of ash wood.*

example problem

- $context_1$: *The house was burnt to ashes while the owner returned*;
- $context_2$: *This table is made of ash wood.*

- using the number of words that the contexts have in common with the sense definitions:

	s_1	s_2	s_3
c_1	1	0	1
c_2	1	0	2

tools

- Find APIs and interfaces to WordNet at the URL

  https://wordnet.princeton.edu/related-projects

nltk.org/howto/wordnet.html

Consegna

- Implementare l'algoritmo di Lesk (!= usare implementazione esistente, e.g., in nltk…).

1. Estrarre 50 frasi dal corpus SemCor (corpus annotato con i synset di WN) e disambiguare (almeno) un sostantivo per frase. Calcolare l'accuratezza del sistema implementato sulla base dei sensi annotati in SemCor.
   - SemCor è disponibile all'URL

     http://web.eecs.umich.edu/~mihalcea/downloads.html

2. Randomizzare la selezione delle 50 frasi e la selezione del termine da disambiguare, e restituire l'accuratezza media su (per esempio) 10 esecuzioni del programma.

è giusto se è ≅ 65%

di.unito.it
Dipartimento di Informatica