TLN-LAB

utilizzo di risorse lessicografiche per la concept similarity e la WSD

Daniele Radicioni

credits

the following slides have been mostly built on materials from:

M. Lesk. Automatic Sense Disambiguation using Machine Readable Dictionaries: How to Tell a Pine Cone from an Ice Cream Cone. In *Proceedings of the 5th International Conference on Systems Documentation*, 1986.

Tanveer Siddiqui and U.S. Tiwary, *Natural Language Processing and Information Retrieval*, Oxford University, 2008.

di.unito.it
DIPARTIMENTO DI INFORMATICA

conceptual similarity with WordNet

credits

- il nucleo originario di questa esercitazione è stato ideato dai dott. Davide Colla e Enrico Mensa.

conceptual similarity with WN

- dati in input due termini, il task di conceptual similarity consiste nel fornire un punteggio numerico di similarità che ne indichi la vicinanza semantica.
  - ad esempio, la similarità fra i concetti *car* e *bus* potrebbe essere *0.8* in una scala *[0,1]*, in cui *0* significa che i sensi sono completamente dissimili, mentre *1* significa identità.
- per risolvere il task di conceptual similarity è possibile sfruttare la struttura ad albero di WordNet.

input

- l'input per questa esercitazione è costituito da coppie di termini contenute nel file *WordSim353* (disponibile nei formati *.tsv* e *.csv*)
  - Il file contiene *353* coppie di termini utilizzati come testset in varie competizioni internazionali,
  - A ciascuna coppia è attribuito un valore numerico *[0,10]*, che rappresenta la similarità fra gli elementi della coppia.

consegna

- l'esercitazione consiste nell'implementare tre misure di similarità basate su WordNet.
- per ciascuna di tali misure di similarità, calcolare gli indici di correlazione di Spearman and gli indici di correlazione di Pearson fra i risultati ottenuti e quelli 'target' presenti nel file annotato.

Article Talk

Spearman's rank correlation coefficient

From Wikipedia, the free encyclopedia

Definition and calculation [edit]

The Spearman correlation coefficient is defined as the Pearson correlation coefficient between the rank variables.[3]

For a sample of size *n*, the *n* raw scores X_i, Y_i are converted to ranks $\operatorname{rg} X_i, \operatorname{rg} Y_i$, and r_s is computed from:

$$r_s = \rho_{\operatorname{rg}_X, \operatorname{rg}_Y} = \frac{\operatorname{cov}(\operatorname{rg}_X, \operatorname{rg}_Y)}{\sigma_{\operatorname{rg}_X} \sigma_{\operatorname{rg}_Y}}$$

where

- ρ denotes the usual Pearson correlation coefficient, but applied to the rank variables.
- $\operatorname{cov}(\operatorname{rg}_X, \operatorname{rg}_Y)$ is the covariance of the rank variables.
- $\sigma_{\operatorname{rg}_X}$ and $\sigma_{\operatorname{rg}_Y}$ are the standard deviations of the rank variables.

WIKIPEDIA
The Free Encyclopedia

Article Talk

Pearson correlation coefficient

From Wikipedia, the free encyclopedia

Definition [edit]

Pearson's correlation coefficient is the covariance of the two variables divided by the product of their standard deviations. The form of the definition involves a "product moment", that is, the mean (the first moment about the origin) of the product of the mean-adjusted random variables; hence the modifier *product-moment* in the name.

For a population [edit]

Pearson's correlation coefficient when applied to a population is commonly represented by the Greek letter ρ (rho) and may be referred to as the *population correlation coefficient* or the *population Pearson correlation coefficient*. Given a pair of random variables (X, Y), the formula for ρ[7] is:

$$\rho_{X,Y} = \frac{\operatorname{cov}(X,Y)}{\sigma_X \sigma_Y} \quad \textbf{(Eq.1)}$$

where:

- cov is the covariance
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

Wu & Palmer

$$\text{cs}(s_1, s_2) = \frac{2 \cdot depth(\text{LCS})}{depth(s_1) + depth(s_2)}$$

- la misura di similarity di Wu & Palmer si basa sulla struttura di WordNet
- *LCS* è il primo antenato comune (Lowest Common Subsumer) fra i sensi s_1 e s_2; e *depth*(*x*) è una funzione che misura la distanza fra la radice di WordNet e il synset *x*.

Wu & Palmer, 1994
https://arxiv.org/pdf/cmp-lg/9406033.pdf

Shortest Path

$$\text{sim}_{path}(s_1, s_2) = 2 \cdot \text{depthMax} - len(s_1, s_2)$$

- for a specific version of WordNet, *depthMax* is a fixed value.
- the similarity between two senses (s_1,s_2) is the function of the shortest path *len*(s_1,s_2) from s_1 to s_2.
  - if *len*(s_1,s_2) is *0*, $sim_{path}(s1,s2)$ gets the maximum value of 2* *depthMax*.
  - if *len*(s_1,s_2) is 2* *depthMax*, $sim_{path}(s1,s2)$ gets the minimum value of *0*.
  - thus, the values of $sim_{path}(s1,s2)$ are between *0* and 2* *depthMax*.

Leakcock & Chodorow

$$\mathrm{sim}_{LC}(s_1, s_2) = -\log \frac{len(s_1, s_2)}{2 \cdot \mathrm{depthMax}}$$

- when s_1 and s_2 have the same sense, $len(s_1,s_2)=0$. in practice, we add *1* to both $len(s_1,s_2)$ and $2*depthMax$ to avoid $log(0)$.
  - thus the values of $sim_{LC}(s_1,s_2)$ are in the interval $(0,log(2*depthMax + 1)]$

termini *vs.* sensi

- attenzione: l'input è costituito da coppie di *termini*, mentre la formula utilizza *sensi*.
- per calcolare la similarity fra 2 termini immaginiamo di prendere la massima similarity fra tutti i sensi del primo termine e tutti i sensi del secondo termine.
  - l'ipotesi è cioè che i due termini funzionino come contesto di disambiguazione l'uno per l'altro.
  - nella formula *c* sono i concetti che appartengono ai synset associati ai termini w_1 e w_2.

$$\text{sim}(w_1, w_2) = \max_{c_1 \in s(w_1), c_2 \in s(w_2)} [\text{sim}(c_1, c_2)]$$

di.unito.it
Dipartimento di Informatica

WSD

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?*""

<u>Ide and Véronis (1998)</u>

di.unito.it
DIPARTIMENTO DI INFORMATICA

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).

di.unito.it
DIPARTIMENTO DI INFORMATICA

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.

di.unito.it
DIPARTIMENTO DI INFORMATICA

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]

the Lesk Algorithm

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

the Lesk Algorithm

```
function SimplifiedLesk(word,sentence)
returns best sense of word
best-sense ← most frequent sense for word
max-overlap ← 0
context ← set of words in sentence
for all senses of word do
    signature ← set of words in the gloss and examples of sense
    overlap ← ComputeOverlap(signature,context)
    if overlap > max-overlap then
        max-overlap ← overlap
        best-sense ← sense
    end if
end for
return best-sense
```

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 <u>deposits</u> will eventually cover future tuition costs because it invests in adjustable-rate <u>mortgage</u> securities.

- given the following two WordNet senses:

bank¹	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"
bank²	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.*

dı.unıto.ıt
DIPARTIMENTO DI INFORMATICA

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

Consegna

- Implementare l'algoritmo di Lesk (!= usare implementazione esistente, e.g., in nltk…).
- Disambiguare i termini polisemici all'interno delle frasi del file 'sentences.txt'; oltre a restituire i synset ID del senso (appropriato per il contesto), il programma deve riscrivere ciascuna frase in input sostituendo il termine polisemico con l'elenco dei sinonimi eventualmente presenti nel synset.
- 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