SNLP 2015
Exercise 03

Submission date: 29.05.2015, 23:59

Word Sense Disambiguation

Dictionary-based WSD methods rely on the definition of senses in dictionaries and thesauri. In this exercise you will implement a dictionary-based algorithm which compares the context of the word to be disambiguated and the dictionary definitions of the different senses of this word, and selects the most similar sense.

1. (10 points) Implement a *simplified* version of the algorithm proposed by Lesk (Lesk, 1986). Here we will treat the contexts and definitions as bags-of-words, and will estimate the overlap between the two *sets* of words using Dice coefficient:

$$overlap(A, B) = \frac{2 \times |A \cap B|}{|A| + |B|} \tag{1}$$

   The evaluation data[1] for this exercise consist of six sense-annotated English words. Each ambiguous word can have one of two senses (for instance, the word *crane* has the senses *crane%machine* and *crane%bird*). The evaluation data is divided into six files, one per word, and contains text snippets which serve as contexts for disambiguation. Each text snippet is labeled with the correct sense (to be used for evaluation). The definitions of each of the two senses of the word we want to disambiguate are provided in a separate file.

   (a) (1 point) Discuss the difference between this WSD algorithm and the original algorithm proposed by Lesk.

   (b) (3 points) Tokenize and normalize the data. You can reuse your implementation from a previous exercise. Make sure to:

   - remove stop words
   - lowercase
   - remove punctuation
   - perform stemming or lemmatization with a tool of your choice

   (c) (4 points) Implement a function which takes as input the normalized tokens of a text snippet and the normalized tokens in each of the two sense definitions, and returns the sense with the most overlap. In case of ties, select the first sense definition, as the senses have been sorted by their frequency according to WordNet.

   (d) (2 points) Report the accuracy of your WSD implementation for each of the six words.

Bonus

2. (2 points) Describe the two constraints proposed by Yarowski (Yarowski, 1995) in relation to the task of word sense disambiguation. Provide short examples for each to illustrate your answer.

[1] Consult the README provided in the data archive for more information about the structure of the files.

Submission instructions: read carefully

- You should form groups of 3 people.
- Submit only 1 archive file in the **ZIP** format with name containing the MN of all the team members, e.g.:

  Exercise_01_MatriculationNumber1_MatriculationNumber2_MatriculationNumber3.zip

- Provide in the archive:
  - your code, accompanied with sufficient comments,
  - a **PDF** report with answers, solutions, plots and brief instructions on executing your code,
  - a **README** file with the group member names, matriculation numbers and emails,
  - **Data** necessary to reproduce your results[2].
- The **subject** of your submission mail must contain the string "[SNLP]" (*including* the braces) and explicitly denoting that it is an exercise submission, eg:

  [SNLP] Exercise Submission 01

- Depending on your tutorial group, send your assignment to the corresponding tutor:
  - `kalofoli@ceid.upatras.gr`
  - `pmandros@mpi-inf.mpg.de`
  - `ilianas@coli.uni-saarland.de`.

General information

- In your mails to us regarding the tutorial please add the tag "[SNLP]" in the subject accompanied by an appropriate subject briefly describing the contents.
- Feel free to use any programming language of your liking. However we strongly advise in favour of *Python*, due to the abundance of available tools (also note that *Python3* comes with an excellent native support of *UTF8* strings).
- Avoid using libraries that solve what we ask you to do (unless otherwise noted).
- Avoid building complex systems. The exercises are simple enough.
- Do not include any executables in your submission, as this may cause the e-mail server to reject it.
- **In case of copying, all the participants (including the original solution) will get 0 points for the whole assignment. Note: it is rather easy to identify a copied solution. Plagiarism is also not tolerated.**
- **Missing the deadline even for a few minutes, will result in 50% point reduction. Submission past the next tutorial, is not corrected, as the solutions will already be discussed.**
- **Please submit in your solutions necessary to support your claims. Failure to do so, might results in reduction of points in the relevant questions.**
- Each assignment has 10 points and perhaps some bonus points (usually 2 or 3). In order to qualify for the exams, you need to have 2/3 of the total points. For example, in case there are 12 assignments, you need to collect at least 80 out of the 120 points to be eligible for the exams. A person that gets 10 plus 2 bonus points in every exercise, needs to deliver only 7 assignments in order to be eligible for the exams, since 7*12=84.

[2] If you feel that these files are beyond reasonable size for an email submission and also reasonably convenient, please provide a means for us to access them online

- Attending the tutorial gives a 30% points increase, disregarding bonus points. For example, if a team scores a grade of 8 plus 2 bonus points, the total grade is $8 + 2 = 10$. Each student of the team, upon attending the corresponding tutorial, is attributed a final grade of $8 \cdot 1.3 + 2 = 12.4$ points.
- Exercise points (including any bonuses) guarantee only the admittance to the exam, however have no further effect on the final exam grade.