Word Sense Disambiguation
Lab Session - Solutions

Diego Frassinelli

December 13 2019

The main goal of this lab session is to make you familiarize with the concept of *Word Sense Disambiguation*. You will use the simplified version of the Lesk algorithm (Kilgarriff & Rosensweig (2000)) to perform word sense disambiguation. In this handout there are 4 tasks to perform.

Try to work in pairs, write the code by yourself but discuss with the others the results you obtain. This would make this session more interesting and definitely more fruitful. When you are asked to write your own code, feel free to chose the language you feel more comfortable with. We will discuss the results of this handout in the next lab session. In this tutorial you will find a combination of questions that are useful to recap some key topics of the previous lectures (Review) and some practical exercises that make you familiarise with real data.

Task 1

In class we talked about the Lesk Algorithm (simplified version). **Review:** what are the main steps of this algorithm? Which kind of information does it use? If you don't remember exactly how it works, please have a look at pages 680-81 from J&M.

1. Take the context words
2. Take the glosses (and examples) for the target word
3. Compute the number of overlapping words (without including the target!)

1. Using the online version of WordNet, extract the glosses and the examples for each sense of the noun "bank".
2. Generate a "bank-senses.txt" file that contains the glosses and examples for all the senses of the noun "bank". Make this information easy to process, you will use this data in the following task.
3. Look at the sentence:

After withdrawing all the money, he went to the bank to watch the river for the last time.

4. Manually use the Lesk Algorithm to disambiguate the noun “bank”.
5. Which sense of the noun “bank” is the most probable in this context?
   Sense#1
6. What are the steps that you have to perform in order to compare context words and glosses words? (hint: do we pre-process?)
   (a) Lemmatisation
   (b) Accept only content words and discard function words and punctuation
   (c) Decide if we want to count types or tokens

Task 2

As you have probably realised, the success of this technique is strongly affected by the number of words in the glosses. A possible way to include the size of the glosses into the computation is the use of the **Dice Coefficient**. This coefficient returns the similarity between two strings ranging from 0 to 1.

$$\mathrm{D} = \frac{2\mathrm{M}}{\mathrm{C} + \mathrm{I}}$$

- **M** is the number of matching words
- **C** is the number of words in the context (after pre-processing)
- **I** is the number of words in the glosses (after pre-processing). These words are also called “indicators”.

1. Compute the Dice Coefficient for the data in the previous Task.

1. **0.22**
2. 0.08
3. 0.00
4. 0.00
5. 0.00
6. 0.00
7. 0.00
8. 0.11
9. 0.00
10. 0.12

Context: ['withdraw', 'money', 'go', 'watch', 'river', 'last', 'time']

Indicators: ['sloping', 'land', 'slope', 'body', 'water', 'pull', 'canoe', 'sit', 'river', 'watch', 'current']

Task 3

1. Write a program that applies the Lesk algorithm to your data.
2. You will need to pre-process your data before counting overlapping words.
3. Use the senses information from "bank-senses.txt".
4. Return the Dice Coefficient for each of the senses in that file.

Task 4

1. Read the paper "Automatic sense disambiguation using machine readable dictionaries: how to tell a pine cone from an ice cream cone" (Lesk, 1987) available on ILIAS. This paper describes the original version of this algorithm as proposed by Michael Lesk.
2. What are the major similarities and differences between the original and the simplified version of the Lesk Algorithm?

- It compares the signature (words in the glosses and examples) of the target word with the signature of each context word and not directly with the context words.
- More indirect than the simplified one.
- Overall, the simplified version works better.