ADVANCED DATA MANAGEMENT

Gianluca Oldani – Tutoring 5

What will be discussed today

- Distributed Job Scheduling (actual implementation)
- Genaral introduction to NLP

What is NLP

NLP is an interdisciplinary field concerned with the interactions between computers and natural human languages (e.g. English) — speech or text

NLP – The classics

The first way to approach the problem, is to have a system to evaluate how much similar two strings are. It can be done with two methods:

- Syntactic
- Semantic

NLP – The classics → Syntactic

- Hamming distance: it is evaluated on two strings of equal length, it is the number of characters with the same index which differ
- Levenshtein distance: it is evaluated between two strings, it is the number of edits required to change one sequence to another. Operations are: insertion, deletion and substitution
- Problem: position is significative

NLP – The classics → Syntactic

- Hamming distance: very simple algorithm, efficient (O(N)), but can be employed in a limited number of cases
- Levenshtein distance: more complex than hamming, can be used on every string, complexity is (O(N^2)) with dynamic programming

NLP – The classics → Syntactic

Algorithm Edit distance

Input: $\alpha = \alpha_1 \dots \alpha_n$ and $\beta = \beta_1 \dots \beta_m$

```
for i ← 0 to n do
    D_{i,0} ← i;
end for
for j ← 0 to m do
    D_{0,j} ← j;
end for
for i ← 1 to n do
    for j ← 1 to m do
        t ← (α_i = β_j)? 0 : 1;
        D_{i,j} ← min{D_{i−1,j−1} + t, D_{i,j−1} + 1, D_{i−1,j} + 1};
    end for
end for
return D_{n,m}
```

NLP – Phonetic → Syntactic

- This class of algorithms tries to capture the pronunciation similarities of words
- Soundex: very old algorithm, used in some system for indexing strings
- Metaphone: improvement on Soundex and less old (1990 → 2009 last version). It has several variants. Shows better results than its predecessor
- Main limitation: each algorithm is designed to work well for a single language

NLP – Phonetic → Syntactic

function SOUNDEX(*word*)

  $result := \text{upperCase}(word_1)$;

  for $i \in \{2, \ldots, \text{length}(word)\}$ **do**

$$code := \begin{cases} 1 & \textbf{if } word_i \in \{b, f, p, v\}, \\ 2 & \textbf{if } word_i \in \{c, g, j, k, q, s, x, z\}, \\ 3 & \textbf{if } word_i \in \{d, t\}, \\ 4 & \textbf{if } word_i \in \{l\}, \\ 5 & \textbf{if } word_i \in \{m, n\}, \\ 6 & \textbf{if } word_i \in \{r\}, \\ \varepsilon & \textbf{otherwise}; \end{cases}$$

    if $result_{\text{length}(result)} \neq code$ **then**

      $result := result \circ code$;

  while $\text{length}(result) < 3$ **do**

    $result := result \circ 0$;

  return *result*;

end.

NLP – Token based → Syntactic

- This class of algorithms tries is an adaptation of set similarity algorithms, thus they are express in mathematical terms
- $$J(A,B) = \frac{|A \cap B|}{|A \cup B|} = \frac{|A \cap B|}{|A| + |B| - |A \cap B|}.$$
  Jaccard index
- $$DSC = \frac{2|X \cap Y|}{|X| + |Y|}$$
  Sorensen-Dice formulae
- Problem 1: what is an element of a set?
- Problem 2: repeated tokens do not matter

NLP – Syntactic techniques recap

- Classic techniques: good for matching a small string against a large corpus of text (e.g., spell checker). Very bad when the order of words is not meaningful
- Phonetic: good for indexing single terms (compact representation). They need to be developed for each language to function
- Token based: good for comparing sentences, specifically when ordering is not meaningful. They let the programmer identify what is a token(letter, triplet, n-gram, word). Do not take into account token frequencies

NLP – Information Retrieval (IR)

- Extensions of the token based methods in order to be able to reasion about what is contained in a text
- Bag of Words (BoW): the most simple application of IR. It transforms text in a frequency set of the words found in it.
- Term Frequency – Inverse Document Frequency (TF-IDF): technique used to measure the importance of a term in a specific corpus of text which is part of a set of multiple documents

NLP – BoW

	about	bird	heard	is	the	word	you
About the bird, the bird, bird bird bird	1	5	0	0	2	0	0
You heard about the bird	1	1	1	0	1	0	1
The bird is the word	0	1	0	1	2	1	0

NLP – BoW (useful preprocessing)

- Remove punctuation → while useful to the semantic of a sentence, it serves no purpose in the syntax of it
- Remove stop words → very frequent and not meaningful words (e.g., “and” “or” “of”). They depend on the analyzed language
- Stemming → same word but with some derivation returned to its original form (e.g., playing → play)

NLP – TF-IDF

- Two element of the formula:
  - $TF(i,j) = n(i,j) / |dj|$ → number of occurrences of term i in document j over the overall number of words in j
  - $IDF(i) = \log_{10}(|D| / |\{d: i \text{ in } d\}|)$ → logarithm with base 10 of the total number of documents over the number of documents that contains the term i
  - $TF\text{-}IDF(i,j) = TF(i,j) * IDF(i)$

NLP – TF-IDF - intuition

- TF: the more frequent a term is in a document, the more relevant it is. The divisor is used to avoid to favor longer documents over smaller ones
- IDF: it was proposed as first as an heuristic to measure the specificity of a term. If a lot of documents contains it, it is an indication of how much a term could be similar to a “stop-word” and thus not being relevant

NLP – BoW and TF-IDF problems

- While in some cases the order of the words is not meaningful, when trying understand “what” a document is about, this information is crucial
- All the techniques explained until now share one common problem: they have no way to actual understand the semantics of a text, since it has no information about the context of the retrieved terms