Token Clouds for Taxonomy

AHMET FURKAN EMREHAN

matahmet@gmail.com

Pebbles

INTRODUCTION

As time progresses, technology gets more complicated. Demands, supplies, objects and necessities get more sophisticated due to this progression. This fact over-diversifies the names or titles of these concepts. Over-diversified naming creates a taxonomy problem difficulty for handling (or managing) concepts.

Pebbles

MOTIVATION

My approach is based on «names» (or «titles» in some concepts) and «descriptions». If we want to reduce the high number of overdiversified titles to a manageable level, we need a classification algorithm. Obviously, «descriptions» have more information than «names» have. I use a token set obtained from «description» to reduce the number of the names.

Pebbles

STEP 1

First of all, for all objects we can focus two features, «titles» (or «name») and «descriptions»

$t_i = title\ (or\ name)\ of\ object\ i$

$d_i = description\ of\ object\ i$

$token_i = set\ of\ tokens\ obtained\ from\ d_i$

$i = 1,2,3, \ldots n\ (index\ of\ objects)$

$n; number\ of\ the\ objects$

$n_u; number\ of\ unique\ titles\ (or\ names)$

Step 2

We can define the taxonomy problem in the light of Step 1. n_u can be very high and then is not manageable. Our goal is to create main categories to be easily managed. We can choose a number m and focus on "top m" objects and their descriptions.

$j = 1,2,3, \ldots m\ (index\ of\ titles\ in\ top\ m\ list)$

$c_j; member\ of\ title\ j\ in\ titles\ of\ top\ m\ list$

Now we can token clouds

$cloud_j := \bigcup_{(t_i == c_j)} token_i$

Pebbles

Step 3

Now we can give new taxonomy "NT" for object with i index

$$NT_i = c_{j^*} \; where \; j^* := \arg max_j \, \#(cloud_j \cap token_i)$$

$\#(set) \; is \; number \; of \; elements \; in \; the \; set$

$\forall j = 1,2,3, \ldots m, \;\; \#(cloud_j \cap token_i)$=0 then $NT_i =$ "NA"

$or \; NT_i = t_i \; (in \; this \; case, category \; numbers \; obviously \; increase.)$

Pebbles

Remarks

1) This model failed its initial experience. That is why some clouds tend to be very large, then the bias unavoidably occurs. To overcome this problem, emaciated clouds can be employed instead of $cloud_j$ alternatively.

$$cloud_j^{em} := cloud_j \backslash \left(\cup_{(k \neq j)} cloud_k \right)$$

2) The clouds can be created by using domain knowledge if we have it.

3) Data structures of Python are suitable for the model.

Pebbles

The End

Thanks for your attention

Pebbles