Similarity graph

Collaborators: * Mate Poljak * Mateo Martinjak * Mateja Pejić * Mihaela Zima

Algorithm

1. Given data

dico.txt	index.txt
`1` 2546	`ocean` 1
`1` 3640	`sunset` 2
...	`school` 3
`2` 14	`horse` 4
`2` 225	`pancake` 5
...	...

2. New data for word `WORD`

```
Goal: we want to create smaller dictionary and pairs list that is only
related to our fixed word WORD. Other data is not our concern so we try to
eliminate all unnecessary data.
```

2.a Find index of `WORD` → `i`

In later text, it will just say `WORD(i)` `it is FIXED during all steps`

2.b Find a set of all words that are connected in dictionary to `i`

```
Goal: we want a set of all unique words that are related to our fixed word
```

		new set
`i` k1	→	k_1
k_1 `i`	(already in set)	/
k_2 `i`	→	k_2
`i` k_3	→	k_3
`i` ...	→	k...
... `i`	→	k...

`k` $_i$ now represents all indexes of words that are related to `i`

2.c Once we have this set,we generate a `mapping`

```
Mapping can be really simple, we can just map words in our
set with their indexes (indexes are given in no particular order, Python
has handy tool for that,we just convert it into a list with: list(set)
```

index	new set
1.	k1
2.	k2
3.	k3
...	...

2.d We then select `pairs` of which BOTH indexes are in our set:

```
 1. Our fixed WORD(i) IS NOT in our set since set holds only indexes
       of words to which i is related (i in not related to itself)
2. pair i,j is SELECTED if:
           1. i is related to j
           2. i,j in SET (i,j related to WORD(i))
2. Algorithmically:
    for each pair i,j in dictionary:
        if (i IS IN set & j IS IN set) → add it to selected pairs list
```

selected pairs
k_5 k_1
k_1 k_3
k_2 k_{752}
$k_{...}$ $k_{...}$

2.e Now we use `mapping` we created to generate new `pairs` and new `index`

pairs	mapping	new pairs
k_5 k_1	→	5 1
k_1 k_3	→	1 3
k_2 k_{752}	→	2 752
$k_{...}$ $k_{...}$	→	...

our set	dict	mapping	new index
k_1	`house` k_1	→	1. `house`
k_2	`lake` k_2	→	2. `lake`
k_3	`tea` k_3	→	3. `tea`
...		→	

3. This is our input for the algorithm: `new pairs` and new `new index`

```
 new index: all words that are related to our fixed word WORD(i)
new pairs: all pairs i,j that are related,and also that WORD(i)
                is related to both
```