VUThiHaiYen
haiyen96.hp@gmail.com

Lab session # 1
ALTEGRAD 2019

10/13/19

1 Question 1

We obtained the graph density for different values of window size as follows:

w	density
2	0.106
3	0.212
4	0.318
5	0.417
6	0.523
7	0.583
8	0.629
9	0.667

We see that as the window size increases, the density increases also. This could be explained by the fact that as we increase the window size w, the number of vertices in the graph $|V|$ remains unchanged while the number of edges $|E|$ increases due to the fact that one word can point to other words in the larger window. As a consequence, the density $\frac{|E|}{|V|(|V|-1)}$ increases.

2 Question 2

We compute the complexity of the Algorithm by first following line by line, we suppose that $|V| = n$:

```
$p \leftarrow \{v : degree(v)\} \forall v \in V$ # Take $|V|$ time complexity.
while $|V| > 0$ do # Loop $|V|$ times.
    $v \leftarrow$ element of $p$ with lowest value # Take $len(p)$ complexity.
    $c[v] \leftarrow p[v]$ # Take $1$ time complexity.
    $neighbors \leftarrow \mathcal{N}(v, V)$ # Take $1$ time complexity if the neighbors of $v$ is stored in a list, $|V|$ otherwise.
    $V \leftarrow V \backslash \{v\}$ # Take $|V|$ time complexity.
    $E \leftarrow E \backslash \{(u, v) | v \in V\}$ # Take $degree(v)$ time complexity.
    for $u \in$ neighbors do # Loop $degree(v)$ times.
        $p[u] \leftarrow \max(c[v], degree(u))$ # Take $1$ time complexity.
    end for
end while
```

By taking all the steps, we find a complexity of $O(|V|^2 + |V||E|)$.

3 Question 3

Running *keyword_extraction.py*, we obtain the following results:

	unweighted k-core	weighted k-core	pagerank	tfidf
precision	51.86	**63.86**	60.18	59.21
recall	**62.56**	48.64	38.3	38.5
F1-score	**51.55**	46.52	44.96	44.85

We see clearly that the two k-core methods outperform the other two methods. While the PageRank and TF-IDF approaches obtain very similar results in terms of precision, recall and F1-score, the two k-cores methods give better score in all the metrics.

4 Question 4

The two seem to give different results. The unweighted k-core gives a much better recall score, indicating that it achieves to get the true keywords better than other methods. The weighted k-core, in the other hand, give better precision. This is due to the fact that the main core of the weighted k-core algorithm contains more elements than the unweighted one, resulting in better precision score. In general, the unweighted k-core gives the best performance with a F1-score significantly higher than the other ones.

5 Question 5

We tried to take the union and intersection sets of the keywords extracted by the two k-core methods and obtain the following results:

	unweighted k-core	weighted k-core	union k-cores	intersection k-cores
precision	51.86	63.86	51.45	**64.36**
recall	62.56	48.64	**64.67**	46.53
F1-score	51.55	46.52	**52.52**	45.51

We see that by taking the union of the two methods, we obtain a better result than using just one of the two.