L'esercitazione prevede di ridurre le dimensioni dei documenti in input del 10, 20 e 30% utilizzando un semplice algoritmo estrattivo:

1. Individuazione dell'argomento (topic) del testo da riassumere. L'argomento può essere indicato come un insieme di vettori Nasari.

$$Vt_{\mathbf{1}} = \{term_{\mathbf{1}}\_score, term_2\_score, \dots, term_{10}\_score\ \}$$
$$Vt_{\mathbf{2}} = \{term_{\mathbf{1}}\_score, term_2\_score, \dots, term_{10}\_score\ \}$$

2. Creazione del contesto, utilizzando i vettori Nasari del topic;
3. Mantenere i paragrafi che contengono i termini più importanti, identificati in base alla **Weighted Overlap**, una misura di similarità tra vettori Nasari. In questa esercitazione per determinare il topic verrà utilizzato il criterio del titolo (vengono utilizzate le parole all'interno del titolo per ricavare un insieme di vettori Nasari).

Vengono forniti due file Nasari:

- *dd-nasari.txt* , un sottoinsieme di NASARI (ottenuto troncando i vettori a 10 caratteristiche). 3.587.754 vettori, ~ 600 MB (https://goo.gl/85BubW);
- *dd-small-nasari-15.txt* , un sottoinsieme di NASARI. È stato applicato lo stesso filtro di *dd-nasari.txt*, con 15 caratteristiche più l'intersezione con i lemmi 60K nel Corpus of Contemporary American English:
  13.084 vettori, 2MB di archiviazione (in questo file molte entità sono state rimosse).

Il secondo file è stato estratto per iniziare la nostra sperimentazione, mentre il primo ha lo scopo di esplorare la risorsa in maniera più approfondita.
I documenti da riassumere sono:

- *Andy-Warhol.txt*
- *Ebola-virus-disease.txt*
- *Life-indoors.txt*
- *Napoleon-wiki.txt*

Effettuare delle sperimentazioni con diversi livelli di compressione (10%, 20% *e* 30%).

1.2 Svolgimento

Inizialmente viene importato il file contenente i vettori Nasari e viene processato perché ogni vettore è nella forma *[babelnetid, word, correlated term1_weight, …correlated term10_weight]* e viene eliminato il valore *weight*, perché ai fini della computazione il numero non serve, ma basta l'ordine dei termini (i vettori sono ordinati dal termine più correlato a quello meno correlato). Vengono poi importati i documenti da riassumere nella forma di liste di paragrafi. Per ogni documento viene utilizzata la funzione `summarize(document, nasari, percentage)` con percentage a 10, 20 e 30.

Summarization
Viene calcolato l'insieme dei vettori nasari del topic (tramite `findNasari(word, nasari)`). Per ogni paragrafo del documento viene calcolata la misura di similarità di ogni parola del paragrafo con il topic, tramite la suddetta metrica **Weighted Overlap**, con la funzione `calculateWO(paragraph, topi, nasari)`). La rilevanza di un paragrafo è calcolata come la somma degli score di weighted overlap per ogni parola nel paragrafo.
La sovrapposizione pesata tra due vettori Nasari si calcola determinando inizialmente l'insieme di termini in comune ai due. Nel caso in cui ci siano delle chiavi comuni, la Weighted Overlap si calcola come

$$WO(v1, v2) = \frac{\sum_{q \in O}(rank(q, v1) + rank(q, v2))^{-1}}{\sum_{i=1}^{|O|}(2i)^{-1}}$$

O: insieme delle chiavi comuni ai due vettori
q: chiave contenuta nell'insieme delle chiavi comuni ai due vettori
$rank(q, v_{\mathbf{i}})$: determina la posizione della chiave q nel vettore Nasari, in modo da determinare la sua rilevanza in quel vettore. Questo è possibile poichè i vettori Nasari sono ordinati in maniera decrescente in base alla rilevanza.

In base al tasso di compressione, viene calcolato il numero di paragrafi che è necessario rimuovere. I paragrafi che ottengono una rilevanza minore vengono rimossi e il testo riassunto viene salvato su file.