**Activity**

Use the below text content, write python code for the following task

1. Convert the paragraph into sentences

2. Append the start and end tags

3. Display bigrams, trigrams, 4-grams

4. Display the probability for bigrams, trigrams, 4-grams

Note:

* Use ‘empty space’ as the delimiter for tokenization
* The below text content is taken from *https://ai.googleblog.com/2006/08/all-our-n-gram-are-belong-to-you.html*

Content:

Here at Google Research we have been using word [n-gram models](http://en.wikipedia.org/wiki/N-gram) for a variety of R&D projects, such as [statistical machine translation](http://googleresearch.blogspot.com/2006/04/statistical-machine-translation-live.html), speech recognition, [spelling correction](https://www.google.com/jobs/britney.html), entity detection, information extraction, and others. While such models have usually been estimated from training corpora containing at most a few billion words, we have been harnessing the vast power of Google's datacenters and distributed processing [infrastructure](http://labs.google.com/papers/mapreduce.html) to process larger and larger training corpora. We found that there's no data like more data, and scaled up the size of our data by one order of magnitude, and then another, and then one more - resulting in a training corpus of *one trillion words* from public Web pages. We believe that the entire research community can benefit from access to such massive amounts of data. It will advance the state of the art, it will focus research in the promising direction of large-scale, data-driven approaches, and it will allow all research groups, no matter how large or small their computing resources, to play together. That's why we decided to share this enormous dataset with everyone. We processed 1,024,908,267,229 words of running text and are publishing the counts for all 1,176,470,663 five-word sequences that appear at least 40 times. There are 13,588,391 unique words, after discarding words that appear less than 200 times.