# Assignment-1 Discussion (Word Vectors and Analogy Test)

Mahesh Ashok Abnave, 203059010
Akshay Eknath Mali, 213079006
Joshi Meet Anilkumar, 21319R001
Sanjna Mohan, 20305R006

30 April 2023

# Problem Statement (1/2)

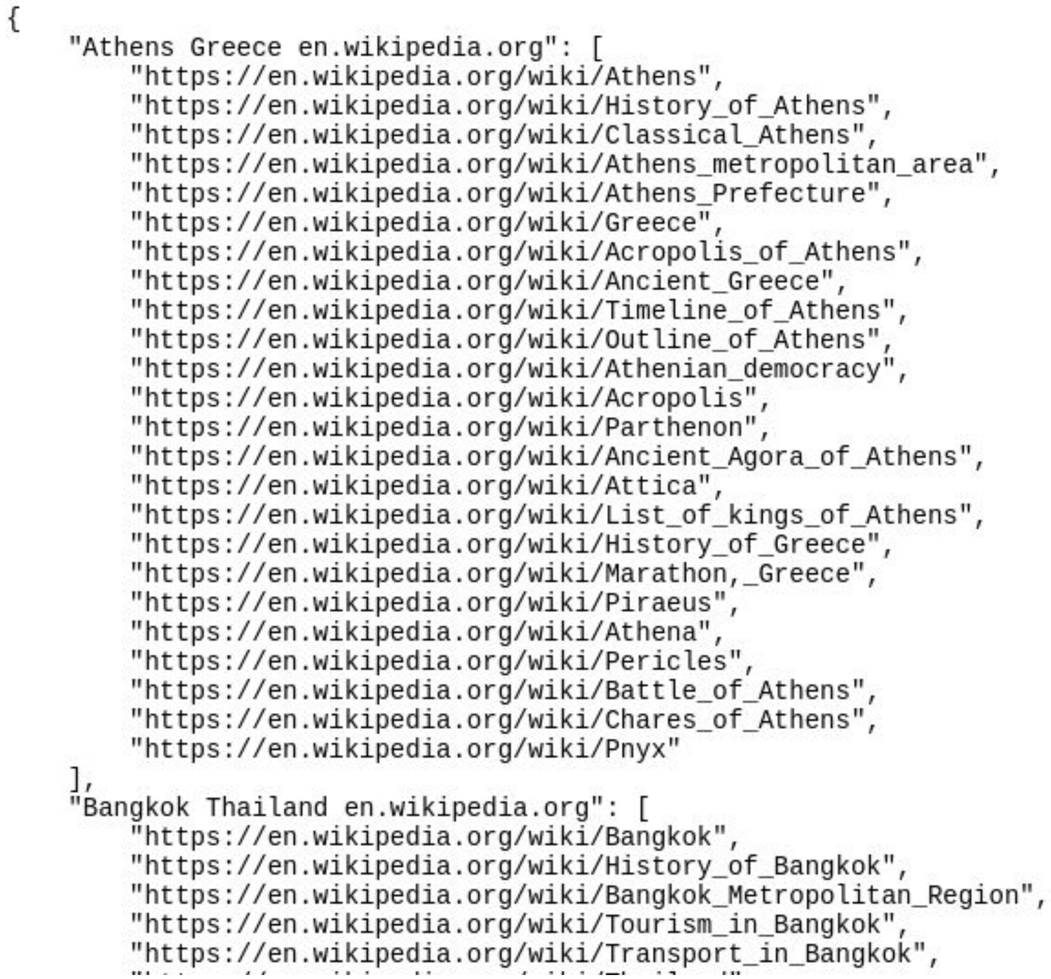
- In this assignment, you will have to implement the backpropagation algorithm from scratch. After implementing backpropagation ab-initio, train CBow and Skip-gram with backpropagation. (This link might help as a quick refresher for Skip-gram and CBoW.)
- The task is to compare the performance of CBoW and Skip-gram embeddings on the word analogy task.
  - Analogy task: Given an analogy, find a word by correctly determining its relationship with another word. For example,

# Problem Statement (2/2)

- man:woman :: king:_____
- ( man is to woman, what king is to ____ )
- The blank should be filled with “queen”.
- **Input**: An analogy pair with one blank. For e.g.,
  - Delhi:India :: Paris:_____
- **Output**: The correct word to satisfy the analogy given in the input.
- You will have to report on the validation data:
  - Accuracy
  - Compare the performance of CBoW and Skip-gram models.
  - Perform detailed error analysis

# Working with Data

- Data Scraping to facilitate analogy:
    - *urls.json*



```
{
    "Athens Greece en.wikipedia.org": [
        "https://en.wikipedia.org/wiki/Athens",
        "https://en.wikipedia.org/wiki/History_of_Athens",
        "https://en.wikipedia.org/wiki/Classical_Athens",
        "https://en.wikipedia.org/wiki/Athens_metropolitan_area",
        "https://en.wikipedia.org/wiki/Athens_Prefecture",
        "https://en.wikipedia.org/wiki/Greece",
        "https://en.wikipedia.org/wiki/Acropolis_of_Athens",
        "https://en.wikipedia.org/wiki/Ancient_Greece",
        "https://en.wikipedia.org/wiki/Timeline_of_Athens",
        "https://en.wikipedia.org/wiki/Outline_of_Athens",
        "https://en.wikipedia.org/wiki/Athenian_democracy",
        "https://en.wikipedia.org/wiki/Acropolis",
        "https://en.wikipedia.org/wiki/Parthenon",
        "https://en.wikipedia.org/wiki/Ancient_Agora_of_Athens",
        "https://en.wikipedia.org/wiki/Attica",
        "https://en.wikipedia.org/wiki/List_of_kings_of_Athens",
        "https://en.wikipedia.org/wiki/History_of_Greece",
        "https://en.wikipedia.org/wiki/Marathon,_Greece",
        "https://en.wikipedia.org/wiki/Piraeus",
        "https://en.wikipedia.org/wiki/Athena",
        "https://en.wikipedia.org/wiki/Pericles",
        "https://en.wikipedia.org/wiki/Battle_of_Athens",
        "https://en.wikipedia.org/wiki/Chares_of_Athens",
        "https://en.wikipedia.org/wiki/Pnyx"
    ],
    "Bangkok Thailand en.wikipedia.org": [
        "https://en.wikipedia.org/wiki/Bangkok",
        "https://en.wikipedia.org/wiki/History_of_Bangkok",
        "https://en.wikipedia.org/wiki/Bangkok_Metropolitan_Region",
        "https://en.wikipedia.org/wiki/Tourism_in_Bangkok",
        "https://en.wikipedia.org/wiki/Transport_in_Bangkok",
        "https://en.wikipedia.org/wiki/Thailand"
```

# Working with Data

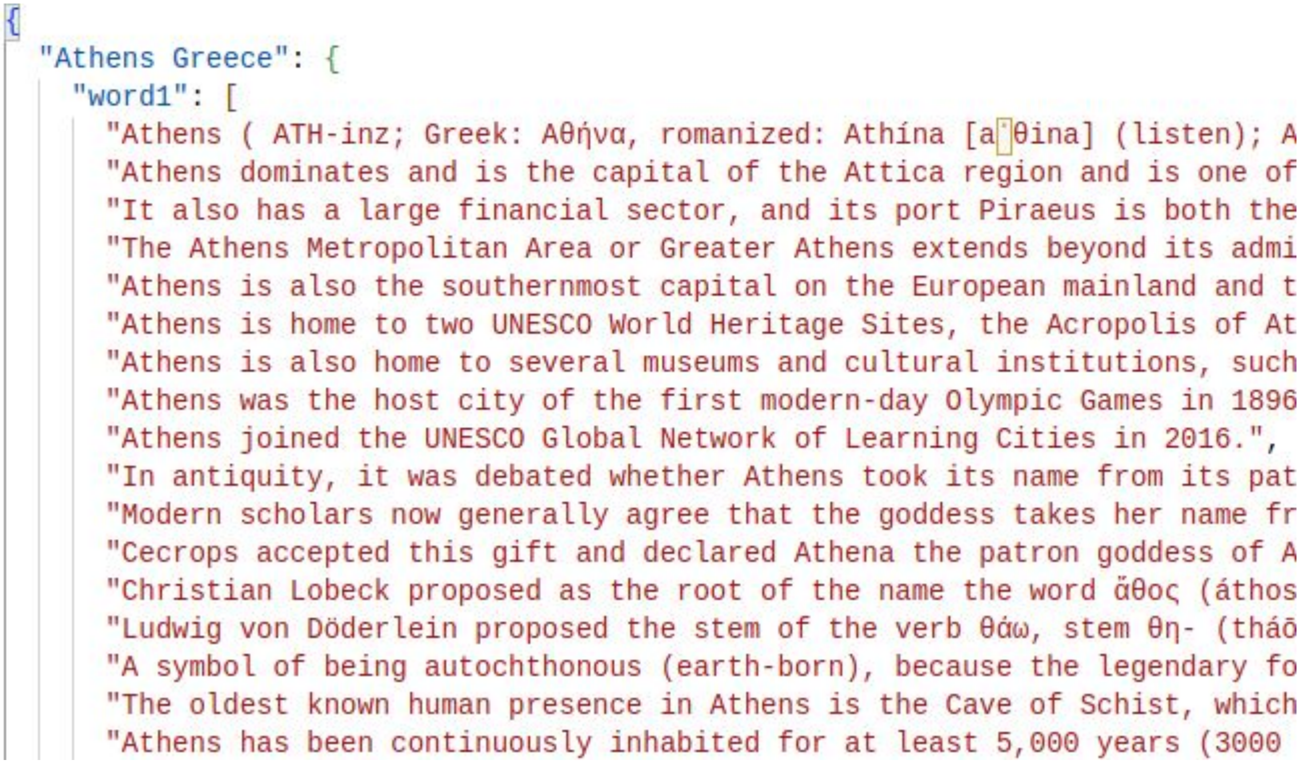
- Data Scraping to facilitate analogy:
  - Analogies — Google API → Relevant URLs
  - ***Google search**: “India rupees en.wikipedia.com”*
  - *urls.json — Wikipedia API → Sentences JSON*

```
{
   "Athens Greece": {
         "word1": [ ... word1-only sentence list ... ]
         "word2": [ ... word2-only sentence list ... ]
         "bothwords": [ ... both words sentence list ... ]
    },
    ... rest of the analogy pairs ...
}
```

  -

# Working with Data

- Data Scraping to facilitate analogy:
    - *sents.json*



```
{
  "Athens Greece": {
    "word1": [
      "Athens ( ATH-inz; Greek: Αθήνα, romanized: Athína [aˈθina] (listen); A
      "Athens dominates and is the capital of the Attica region and is one of
      "It also has a large financial sector, and its port Piraeus is both the
      "The Athens Metropolitan Area or Greater Athens extends beyond its admi
      "Athens is also the southernmost capital on the European mainland and t
      "Athens is home to two UNESCO World Heritage Sites, the Acropolis of At
      "Athens is also home to several museums and cultural institutions, such
      "Athens was the host city of the first modern-day Olympic Games in 1896
      "Athens joined the UNESCO Global Network of Learning Cities in 2016.",
      "In antiquity, it was debated whether Athens took its name from its pat
      "Modern scholars now generally agree that the goddess takes her name fr
      "Cecrops accepted this gift and declared Athena the patron goddess of A
      "Christian Lobeck proposed as the root of the name the word ἄθος (áthos
      "Ludwig von Döderlein proposed the stem of the verb θάω, stem θη- (tháō
      "A symbol of being autochthonous (earth-born), because the legendary fo
      "The oldest known human presence in Athens is the Cave of Schist, which
      "Athens has been continuously inhabited for at least 5,000 years (3000
```

# Working with Data

- Data Scraping to facilitate analogy:
  - Analogies → Google API → Relevant URLs
  - ***Google search**: “India rupees en.wikipedia.com”*
  - *urls.json Wikipedia API Sentences JSON*
  - *Stats*
    - *Avg word 1 sents: 601*
    - *Avg word 2 sents: 652*
    - *Avg both word sents: 128*
  - *Priority: “both words” sentences*

# Working with Data

- Pre-processing:
  - Stop words removal
  - Punctuation removal
  - Only alphabetical word (no numbers and other language characters) a-z and A-Z
  - Lower casing non Proper noun words
  - No lemmatization (may lose morphological / syntactic information, e.g. “run : runner” will become “run : run”)

# Experimental Setup

| Architecture | Embedding size | Window size | No of sentences per analogy word | Totals training samples |
|---|---|---|---|---|
| CBOW | 80 | 3 | 200 | 12.09 Lacs |
| SKIPGRAM | 80 | 3 | 200 | 60.76 Lacs |

**No. epochs: cbow - 64 , skipgram - 27**
**Learning rate: 0.0004 to 0.002**
**Vocabulary size: 49245,Embedding dim: 80**
**3 layers(input = V ,hidden =h , output =V size)**
**Context size = 3**

# Backpropagation

- 3 layers(input = V ,hidden =h , output =V size)
- Context size = 3
- Forward :
- Output of hidden layer (h): dot(W0.T,x)
- Output of last layer (Y_pred): softmax(dot(W1.T,h))
- CE loss =np.sum(Y_true*np.log(Y_pred))/batchsize
- Backpropagation :
- ***Change in weight dW=alpha * (t-o) * input***
- dW1 = np.dot(h,(y_pred - y_true).T)
- dW0 = np.dot(x,np.dot(W1 ,(y_pred - y_true)).T)
- W1 -= alpha * dW1
- W0 -= alpha * dW0

# Results

| Architecture | Accuracy ( in %) | | |
|---|---|---|---|
| | Top 1 | Top 10 | Top 50 |
| CBOW | 5.47 | 15.77 | 25.42 |
| SKIPGRAM | 2.737 | 8.474 | 15.90 |

## Tests:

King : queen :: son : ?? (result for top-10 closest)

**Skipgram:** {'dear': 0.65, 'sister': 0.63, 'father': 0.61, 'daughter': 0.60, 'mother': 0.59, 'brother': 0.59..}

**CBOW**: {'daughter': 0.52, 'sister': 0.50, 'victoria': 0.49, 'ishmael': 0.49, 'gardener': 0.48, 'rephaiah': 0.48, 'father': 0.48,..}

# Observations and Analysis

- **CBOW takes less epoch to train than Skipgram.** Since CBOW only has to predict one word, it requires less computation than Skipgram, which has to predict multiple words.
- **Better CBOW analogy accuracy than skipgram.** CBOW predicts a word based on the context, and skipgram that predicts the context based on the current word.

  Words is known by company it keeps.
- **Smaller embedding size performs better than larger.**

  More generalization of words similar to dimensionality reduction.

•

# Observations and Analysis

- **Word embedding learned are nicely identifying 4th word given 3 ( nouns,proper names (city,country))**

  **But in case of adjectives,adverb it gives 4th word semantic to true 4th word.**

  **Test:** Free : freely :: happy : ?? (happily not found in top 50)

  **Skipgram:** {'glad': 0.648, 'longed': 0.56, 'fill': 0.54, 'rejoice': 0.54, 'hope': 0.53, 'love': 0.53,, 'eternity': 0.533,, 'doth': 0.53, 'weary': 0.52, 'joy': 0.52..}

  **Cbow:** {'glad': 0.55, 'hope': 0.508, 'rejoice': 0.503, 'paradise': 0.491, 'joy': 0.487, 'love': 0.48, 'children': 0.48, 'taste': 0.47, 'fill': 0.47, 'exceedingly': 0.46, 'longed': 0.464, 'wished': 0.454, 'lover': 0.449}

  **Need more corpus size, train more time, vary context size**

# Demo

## Some output samples: tests on top 10 output

Make : made :: see : ?

**Skipgram:** {'like': 0.91, 'saw': 0.91, 'even': 0.90, 'first': 0.904, 'also': 0.898514589672766, 'time': 0.892, 'set': 0.89, 'two0.89 'seen':0.88}

**Cbow:** {'seen': 0.82, 'like': 0.81, 'saw': 0.81, 'even': 0.79, 'also': 0.78, 'ever': 0.788, 'great': 0.780, 'first': 0.779, 'one': 0.77,, 'much': 0.775,...}

DEMO : On notebook

# Marking (max 100)

- Data pre-processing: 10 (no pre-processing 0)
- Scraping: 20 (respectable size and good scraping strategy full marks)
- Good implementation of BP: 20
- Accuracy: >90: 20 marks; >70-90: 10 marks; >50-70: 5 marks; else 0
- Analysis: 10 marks
- Comparison of CBOW and Skip Gram: 10 marks
- Demo: 10 marks
- Topper of leader board: 5 marks bonus