# Viterbi-Exercise

Mohamed Elbadrashiny

# Steps

1. **Create lm object:**

   ```
   bin/lmplz -o 2 </path/to/training/data.txt  >/path/to/output/lm.arpa
   from LanguageModel import LanguageModel
   lm = LanguageModel(‘/path/to/output/lm.arpa’, 'TEXT')
   ```

2. **Create mapping object:**

   ```
   from LMDisambigMapBuilder import LMDisambigMapBuilder
   from LMDisambigMap import LMDisambigMap
   LMDisambigMapBuilder.build(‘/path/to/training/corpus.txt’, ‘/path/to/output/map.txt')
   mapping = LMDisambigMap(‘/path/to/output/map.txt', 'TEXT', lm)
   ```

3. For each input sentence, build a Lattice object:

   ```
   from SearchLattice import SearchLattice
   lattice = SearchLattice(sequence, mapping)
   ```

   **Note:** the lattice will have the <s> and </s>

4. Write Viterbi algorithm

# LM and Mapping helper Functions

1. **mapping.get_w_given_a(q, w)** —> returns $P(q \mid w)$
2. **mapping.get_possibilities(q)** —> returns a list of the alternative diacritized words for the the input undiacritized word
3. **lm.get_cond_prob(seq, len(sequence) - 1, len(sequence) - 1)**—> returns the conditional probability of the given sequence of words. In our case, sequence is a list of 2 words because we are using bi grams

5.

# Lattice helper Functions

1. **lattice.get_columns_number()** —> returns the number of columns in the Lattice (T+2); where T is the number of words in the input sequence. There is extra 2 because of the <s> and </s>
2. **lattice.columns[i].get_input_word()** —> returns the undiacritized word of column number i
3. **lattice.columns[i].get_possibilities_number()** —>returns the number of alternatives diacritized words (N) in column number i
4. **lattice.columns[i].rows.get_possibility(j)** —> returns the diacritization alternative number j for the undiacritzed word in column number i
5. **lattice.columns[i].rows.set_score(j, score)** —> set the score of the diacritization alternative number j for the undiacritized word in column number i
6. **lattice.columns[i].rows.get_score(j)** —> returns the score of the diacritization alternative number j for the undiacritized word in column number i
7. **lattice.columns[i].rows.set_previous_possibility(j, best_previous_idx, i - 1)** —> create a pointer from the alternative number j in column i to the index of the best previous alternative in column (i-1)
8. **lattice.columns[i].rows.get_previous_possibility_index(j)** —> get the index of the best previous alternative for word number j in column number i