# AI4001 - Fundamentals of Natural Language Processing

**Due Date:** Friday, March 14th by 11:55pm on Google Classroom.

Assignments are to be done individually. No late assignments will be accepted.
**Submissions that do not comply with the specifications given in this document will not be marked and a zero grade will be assigned.**
Write your name and e-mail id in the first text block on top of each python notebook. You are required to submit a python notebook, and a single zip file containing any corpora and language models that you have used to run your code, on Google Classroom. You should name your notebook as i20-XXXX.ipynb where i20-XXXX represents your student id. You should document your algorithm/technique within the notebook using text cells.

## Poetry Generation in Roman Urdu

## 1 Introduction

After the first assignment you have developed a fairly good idea about the stucture and processing of urdu sentences. In this assignment, you will use $n$-gram language modeling to generate some poetry using the **spaCy** pipeline for text processing. For the purpose of this assignment a *ghazal* that will consist of seven stanzas each containing two verses where each verse consists of 6—8 words. For example, following is a ghazal alegedly written by Bahadur Shah Zafar during his exile.

```
lagta nahin hei ji meira ujarei dayar mein
kiski bani hei alame napayedar mein

bulbul ko pasban se na saiyyad se gila
qismet mein qaid likhi tthi fasle bahar mein

kaeh do in hassreton se kahin aur ja basein
itni jageh kahan hei dile daghdar mein

ik shakhegul pe baith ke bulbul hei shadman
kante bicha diye hein dile lalazar mein

umre daraz mang ke laye tthe char din
do arzu mein kat gayei do intezar mein

din zindagi ke khatm huei sham ho gayi
phaila ke paon soyenge kunje mazaar mein

kitna hei bad naseeb zafar dafn ke liye
do gaz zamin bhi na mili kueyar mein
```

The task is to print seven such stanzas with an empty line in between. The generational model can be trained on the provided poetry corpus containing poems from Faiz, Ghalib and Iqbal. You can either convert the corpora into Roman or you can train your model in Urdu and convert the generated version into Roman using your implementation in Assignment 1. You should augment the poetry corpus by scraping online sources (such as rekhta.org, hamariweb.com, etc.) to get a better representation of Urdu poetry. You will train bigram and trigram models using this corpus. These models will be used to generate poetry. Several online solutions are available for English that use the NLTK library. However, we will be using the spaCy pipeline to accomplish this task!

# 2 Assignment Task

The task is to generate a ghazal using different models. Your implementation will generate a ghazal verse by verse until all stanzas have been generated. The poetry generation problem can be solved using the following algorithm:

1. Load the Poetry Corpus
2. Tokenize the corpus in order to split it into a list of words
3. Generate $n$-gram models
4. For each of the stanzas
    - For each verse
        * Generate a random number in the range [4...5]
        * Select first word (intelligently)
        * Select subsequent words until end of verse
        * [**bonus**] Try to rhyme the last words of the stanza with the last word of the first stanza
        * Print verse
    - Print empty line after stanza

## 2.1 Implementation Challenges

Among the challenges of solving this assignment will be the selection of subsequent words once we have chosen the first word of the verse. But, to predict the next word what we want to compute is the most probable next word out of all of the possible next words. In other words, find the set of words that occur most frequently after the already selected word and pick the next word from that set. We can use a **Conditional Frequency Distribution (CFD) with interpolation** to figure that out! A CFD tells us: given a condition, what is likelihood of each possible outcome. [bonus] Rhyming the generated verses is also a challenge. You can build your dictionary for rhyming.The Urdu sentence is written from **right to left**, so makes your $n$-gram models according to this style.

## 2.2 Standard $n$-gram Models

We can develop our model using the Conditional Frequency Distribution method. First develop a Bigram model and then the Trigram model. Select the first word of each line randomly from starting words in the vocabulary and then use the bigram model to generate the next word until the verse is complete. Generate the next verse similarly. Follow the same steps for the trigram model and compare the results of the two n-gram models. [**bonus**] Can we make the sonnet rhyme? (Hint: Build a pronunciation dictionary using the most probable rhyming endings!)

## 2.3 Backward Bigram Model

Standard $n$-gram language models model the generation of text from left to right. However, in some cases, words might be better predicted from their right context rather than their left context. The next task is to produce a backward bigram model that models the generation of a sentence from right to left. Think of a very simple way of very quickly using BigramModel to produce a BackwardBigramModel that is identical except for the modeling direction. Compare the results of the backward bigram model with previous implementations.

## 2.4 Bidirectional Bigram Model

Next, build a BidirectionalBigramModel that combines the forward and backward model. Both the **BackwardBigramModel** and **BidirectionalBigramModel** should take the same input and produce the same style of output as BigramModel. Compare the output of the BidirectionalBigramModel with the Trigram model.

## Honor Policy

This assignment is a learning opportunity that will be evaluated based on your ability to think in a group setting, work through a problem in a logical manner and write a research report on your own. You may however discuss verbally or via email the assignment with your classmates or the course instructor, but you are to write the actual report for this assignment without copying or plagiarizing the work of others. You may use the Internet to do your research, but the written work should be your own. **Plagiarized reports or code will get a zero**. If in doubt, ask the course instructor.