# Assignment 1

**Table of Contents**

**- [Tokenization]**

**- [N-gram]**

**- [Smoothing and Interpolation]**

**- [Generation]**

**Tokenization**

1. Tokenization:

*- The sentence has been tokenized with respect to the following:*

*- Sentence Tokenizer: Divides text into sentences.*

*- Word Tokenizer: Splits sentences into individual words.*

*- Numbers: Identifies numerical values.*

*- Mail IDs: Recognizes email addresses.*

*- Punctuation: Detects punctuation marks.*

*- URLs: Identifies website links.*

*- Hashtags (#omg): Recognizes social media hashtags.*

*- Mentions (@john): Recognizes social media mentions.*

*- For the following cases, tokens with appropriate placeholders are used:*

*- URLS: `<URL>`*

*- Hashtags: `<HASHTAG>`*

*- Mentions: `<MENTION>`*

*- Numbers: `<NUM>`*

*- Mail IDs: `<MAILID>`*

***Example****:*

*Input: `./test.py test.py <has My name is Radhika Garg.my rollno is 2023201030 .my favourite websites are www.takeyouforward.com and https://www.hi.com .my friend is @vv. friendship.>`*

*Output:*

*```*

*[['My', 'name', 'is', 'radhika', 'garg', '.', 'my', 'rollno', 'is', '<NUM>', '.', 'my', 'favourite', 'websites', 'are', '<URL>', 'and', '<URL>', '.', 'my', '<MENTION>', '.'], ['<HASHTAG>', '.']]*

*```*

2.N-gram

1. N-gram Model Generation:

- This file takes a value of `N` and the `<corpus\_path>` and generates an N-sized N-gram model from the given corpus.

- Example Usage:

- Input:

- Enter N for N-gram: `2`

- Enter Filepath (corpus Path): `./<path> @radhsh 264574554 #gsdhgs www.hdghd.com http://www.fgdfg.com`

- Output:

```

O/p: [('<s>', '<mention>'), ('<mention>', '<number>'), ('<number>', '<hashtag>'), ('<hashtag>', '<url>'), ('<url>', '<url>'), ('<url>', '</s>')]

```

Smoothing and Interpolation

*1. Language Models:*

*- Language models created with the following parameters:*

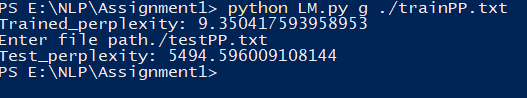
- **On "Pride and Prejudice" corpus:**

- LM 1: Tokenization + 3-gram LM + Good-Turing Smoothing

- Perplexity:

- Train: 9.3

- Test: 5494

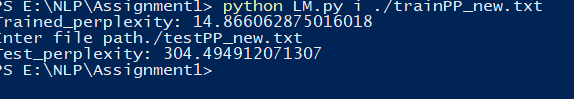


LM 2: Tokenization + 3-gram LM + Linear Interpolation

- Perplexity:

- Train: 14.86

- Test: 304



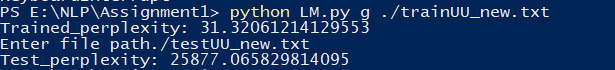
- On "Ulysses" corpus:

LM 3: Tokenization + 3-gram LM + Good-Turing Smoothing

- Perplexity:

- Train: 31

- Test: 25877

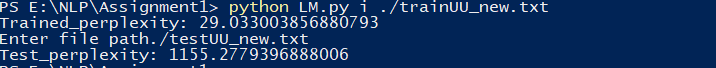


-LM 4: Tokenization + 3-gram LM + Linear Interpolation

- Perplexity:

- Train: 29.86

- Test: 1155.77



Generation

1. Text Generation:

- Generate text from a given input using Pride and Prejudice and Ulysses models developed by linear interpolation model.

- \*\*Example Input:\*\* `Hey do not`

- \*\*Output (k=8, corpus=Ulysses):\*\*

```

N=3:

like 0.16

know 0.12

agree 0.08

deny 0.08

charge 0.08

solicit 0.08

copy 0.04

to 0.04

N=2

to 0.06923076923076923

a 0.05934065934065934

? 0.02197802197802198

the 0.020879120879120878

in 0.020879120879120878

so 0.01978021978021978

, 0.01868131868131868

. 0.01868131868131868

N=1

Every word generating as next will have the equal probability cause it does not see the previous n word.

```

- For Out of Vocabulary (OOV): `khjfhd jj, N=2`

```

<s> 0

the 0

project 0

gutenberg 0

ebook 0

of 0

ulysses 0

, 0

```

- LM2 Generation for "Hey do not":

```

know 0.053807423399488866

be 0.032587205880767536

, 0.026541705296852198

to 0.02652751496889085

. 0.026063035892206012

think 0.02208691019309448

the 0.0175993733841363

make 0.01687191629230169

- LM4 Generation for "Hey do not":

```

like 0.06585859959666134

know 0.046508725076636726

to 0.04136314000171877

agree 0.030696374242924963

charge 0.030355154551686758

deny 0.030343585926400837

solicit 0.03034262187429368

. 0.027205583686320013

```