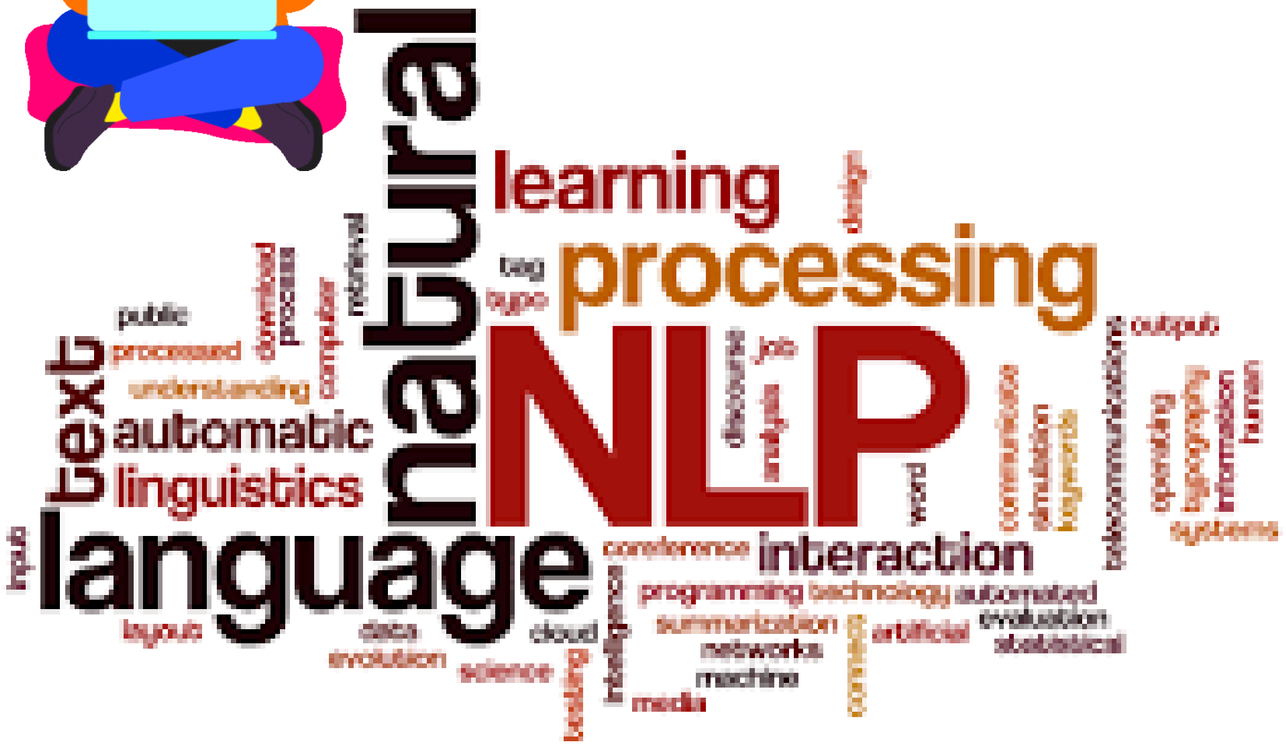


Group No.: 15

Query Formation

Professor:
Prof. Swati Agarwal

Team members:

- Digvijaysinh Mane
- Nupur Funkwal
- Nirzari Shah
- Prayas Kumar
- Mitali Doshi

innovate achieve lead

Project Description

Part 1: Automatically correcting the queries by processing syntactic and semantic features of the words present in the query. Improving this correction by validating the search results.

Part 2: Automatically complete queries using n-gram linguistic models. Compare the quality of completion across various models.

innovate achieve lead

Part 1: Query Correction

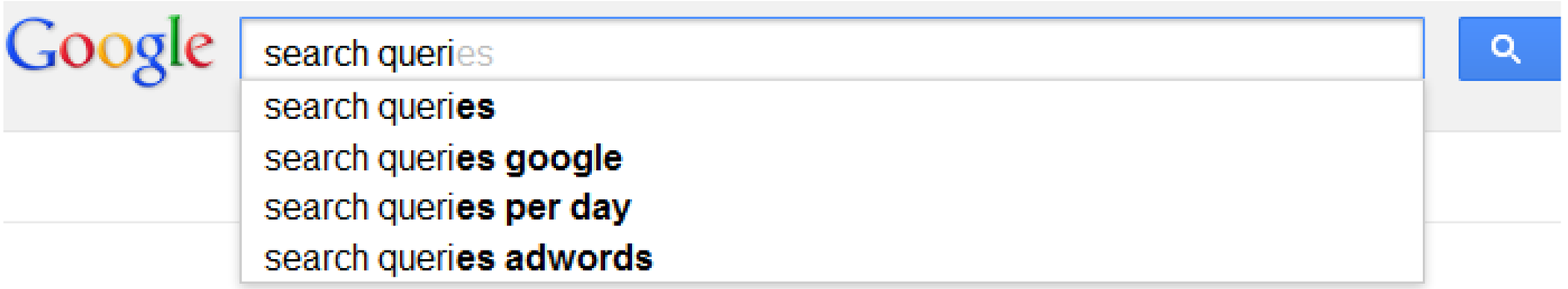
Automatically correcting the queries by processing syntactic and semantic features of the words present in the query. Improving this correction by validating the search results.

Preprocessing & Laguage Model

- Tokenization of queries from corpus
- Vocabulary building with the help token frequency
- Model Corrects: Spelling, context based word correction
- It uses bigram model for word correction
- Uses Levenshtein edit distance of 2 to correct spelling mistakes.

Part 2: Query Completion

Automatically complete queries using n-gram linguistic models. Compare the quality of completion across various models.



Base Model

- Base Model is : based on trigram of characters
- We started with a trigram model. As a result we got very poor results on training dataset.
- Bigram model - randomness because of char model

innovate achieve lead

Word Gram Model

- N - gram model with back - off
- Worked well for training data
- Issue - too many unknown words in test data (less overlap with train data) - probability calculation from n-gram model not feasible
- Possible solution 1 - use lemmatization or stemming on test data
- But, there were still a considerable number of unknown words
- Possible solution 2 - replace unknown words with <unk> tokens and create an n-gram model
- Since there were too many <unk> tokens, results were random and there was no way of replacing <unk> tokens in resulting queries with actual words

innovate achieve lead

Character Gram Model

- Alternate solution : we started exploring character gram model (with back-off)
- Drawback - lesser context compared to word gram model

Combination of Word Gram and Character Gram Model

- First we try to get prediction from word gram
- If we are unable to get a prediction, the code switches to a character gram model

Comparison with Trigram base model

```
final_pred("new york")
```

```
[(3126.1157613395158, 'new york $'),
 (0.7488271331816343, 'new york city $'),
 (0.025382940594723832, 'new york times $'),
 (0.012588946325481435, 'new york court com $'),
 (0.0022654941867570628, 'new york colleges $'),
 (0.0005060279319011757, 'new york lottery $'),
 (4.151951169707596e-07, 'new york gas prices $'),
 (1.482450343405114e-09, 'new york homes for sale $'),
 (1.097175409514018e-11, 'new york city the villiage $'),
 (2.783598426101728e-13, 'new yorkwww jetblue comom $')]
```

```
char_prediction_trigram("new york",k_gram_model_char)
```

```
[(10718.111181735481, 'new york '),
 (67.48637670419278, 'new yorker '),
 (53.64162512769259, 'new yorking '),
 (7.179101370254715, 'new yorktv '),
 (6.7029658953663755, 'new yorkwww '),
 (2.733134907832855, 'new yorkzoo '),
 (1.428470584883758, 'new yorknobs '),
 (1.1038226994285691, 'new yorkbook '),
 (0.9948164121624311, 'new yorkout '),
 (0.22487724863822114, 'new yorkouts ')]
```

```
final_pred("uniq")
```

```
[(0.011935343850234863, 'unique marine $'),
 (0.010927045196449292, 'unique thongs $'),
 (1.2090403337904697e-11, 'unique vinyl siding $')]
```

```
[40] char_prediction_trigram("uniq",k_gram_model_char)
```

```
[(124010.21164447266, 'unique '),
 (41133.70569339591, 'uniq '),
 (776.4523615231295, 'uniqsex '),
 (0.002645138637323352, 'uniqiudgeration ')]
```

```
final_pred("quick l")
```

```
[(0.3926660720611863, 'quick lore'),
 (0.0007522204209171671, 'quick lachey'),
 (0.0001176610487794 5546, 'quick loretto'),
 (2.737083273639174e-05, 'quick lorenzo')]
```

Run cell (⌘/Ctrl+Enter)
cell executed since last change

executed by Digvijaysinh Mane
11:16 AM (0 minutes ago)
executed in 0.573s

```
prediction_trigram("quick l",k_gram_model_char)
```

```
[(1.3335191929842758, 'quick l '),
 (0.16668989912303447, 'quick l0 '),
 (0.07716917230227867, 'quick lyrics '),
 (0.07143852819558619, 'quick lcd '),
 (0.06650291229961101, 'quick lore '),
 (0.05825447182179312, 'quick like '),
 (0.03333797982460689, 'quick lg '),
 (0.026319457756268 6, 'quick lv '),
 (0.004102351521798 8174, 'quick luke '),
 (0.002645871414651 3408, 'quick lbm ')]
```

```
final_pred("chica")
```

```
[(137.58766710865444, 'chicago $'),
 (0.1063390065469008, 'chicagoland $'),
 (0.03518594106973896, 'chicago il $'),
 (2.9834014222435287e-07, 'chicagoland area $'),
 (2.799023193469347e-08, 'chicago tribune $'),
 (1.2477515970241637e-11, 'chicago suntimes sports $'),
 (2.480677341242974e-16, 'chicago suburban newspapers $')]
```

```
char_prediction_trigram("chica",k_gram_model_char)
```

```
[(136291.6965259011, 'chica '),
 (96144.90331011936, 'chical '),
 (46537.55981753027, 'chican '),
 (27380.08377933516, 'chicas '),
 (20490.075827426004, 'chicard '),
 (11069.536556014144, 'chicals '),
 (10361.396700909341, 'chicate '),
 (9839.07906111276, 'chicards '),
 (9004.455053447922, 'chicap '),
 (7613.839111407606, 'chicare ')]
```

innovate achieve lead

BITS Pilani
Pilani | Dubai | Goa | Hyderabad

Sincere Thanks

Prof. Swati Agarwal

Professor, Department of Computer Science
at Bits Pilani-Goa