**TEXT PREDICTION WITH N-GRAM MODEL**



**Submitted by**:

Jagendra Singh (14103136)

Prasant Kumar (14103153)

Rishabh Bisht (14103156)

## CONTENTS

PROJECT DESCRIPTION .............................................2

PROBLEM STATEMENT ..............................................2

TECHNOLOGY USED ..................................................3

N-GRAM MODEL .........................................................3

REFERENCES ................................................................4

PROJECT DESCRIPTION

Mobile devices have become indispensable everyday companions at home and work, to socialize, play and do business. But lacking a full-size keyboard, text entry on touch screen devices in particular can be cumbersome. Automated text prediction aims to solve this by using entered text to predict the next word.

PROBLEM STATEMENT

It may be expected that the accuracy of a predictive text model primarily depends on the number of unique words that are available in the original body of text. Complementary data sources may include dictionaries of profanity words (assuming that the prediction of such words is to be avoided), stop words, named entities (sports, cities, states, presidents), synonyms (WordNet database) and jargon dictionaries.

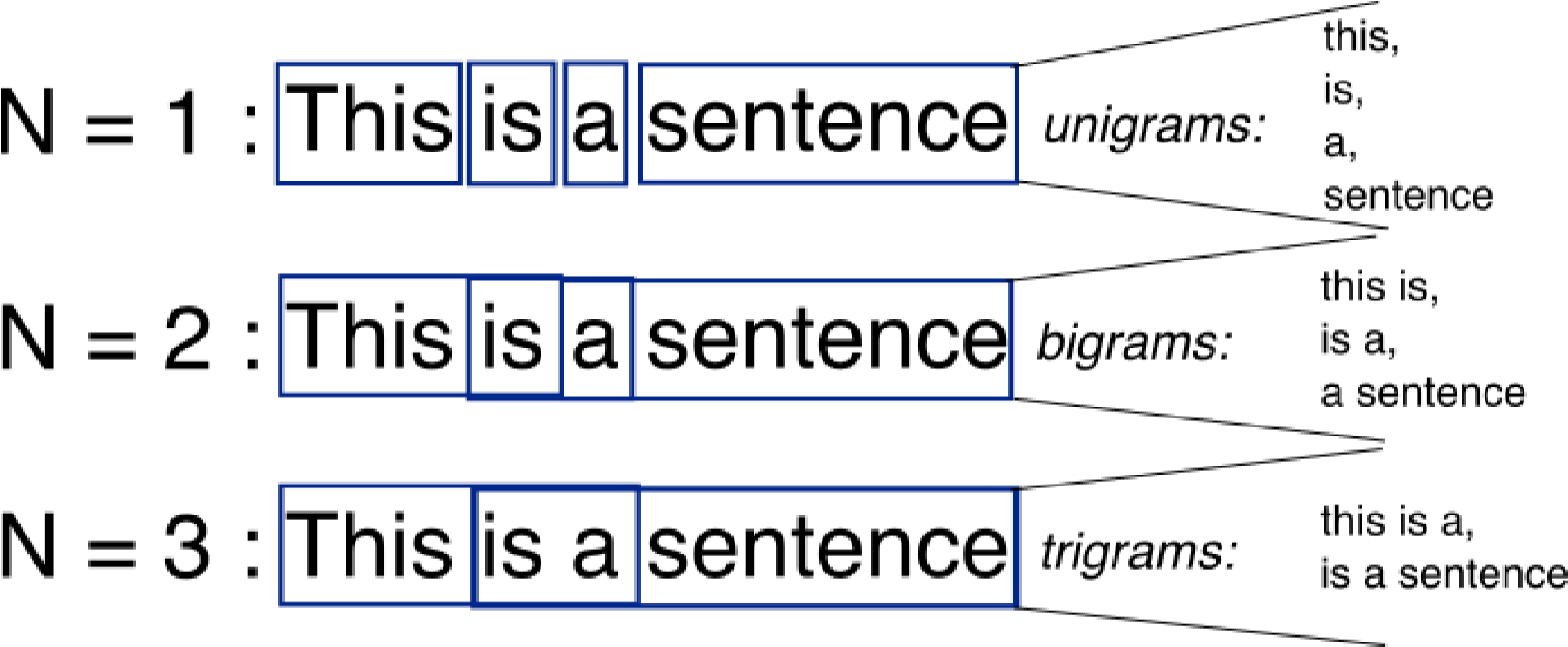
Common issues in the analysis of text data are the use of colloquial language and punctuation as well as occurrences of misspellings (bye vs by). Especially on social media people often use non-words and acronyms (lol) that are a language on their own.

TECHNOLOGY USED

1. python
2. Numpy

N-GRAM MODEL

One of the oldest methods used in trying to compute the probability that a given word is the next word in a sentence is employing n-gram models. N-gram models are attempts to guess the next word in a sentence based upon the (n - 1) previous words in the sentence. These models base their guesses on the probability of a given word without any context (i.e., the is a more common word than green and is thus more probable than green if context is ignored) and the probability of a word given the last (n – 1) words. For example, take the sentence beginning “The four leaf clover was the color...”. Using a bigram model, one would compute P(green | color) and P(the | color) to determine the more probable guess between these two words.



REFERENCES

1. [https://rstudio-pubstic.s3.amazonaws.com/96252\_bd61a0777ad44d04b619ce95ca44219c.html#building-the-ngram-tables](https://rstudio-pubs-tic.s3.amazonaws.com/96252_bd61a0777ad44d04b619ce95ca44219c.html#building-the-n-gram-tables)
2. http://cs.stanford.edu/people/eroberts/courses/soco/projects/2004-

05/nlp/techniques\_word.html