Problem Statement:

Create an algorithm that can shift through ngrams to predict which word will most likely be typed next.

Objective:

Take a word and predict the next word using bi-gram. Take the word as 'the' from the corpus to predict next word.

Corpus:

"the cat is red the cat is green the cat is blue the dog is brown

In [1]:
```
from collections import defaultdict
```

In [2]:
```
corpus = "the cat is red the cat is green the cat is blue the dog is brown"
```

In [3]:
```
tokens = corpus.split()
```

Build tokens dictionary

with next word list for each token

In [4]:
```
previous_word = ""
token_dict    = defaultdict(list)
```

In [5]:
```
for current_word in tokens:
    if previous_word != "":
        token_dict[previous_word].append(current_word)

    previous_word = current_word
```

In [6]:
```
token_dict
```

Out[6]:
```
defaultdict(list,
            {'the': ['cat', 'cat', 'cat', 'dog'],
             'cat': ['is', 'is', 'is'],
             'is': ['red', 'green', 'blue', 'brown'],
             'red': ['the'],
             'green': ['the'],
             'blue': ['the'],
             'dog': ['is']})
```

Compute the probability of each observed next word for each word in the dictionary.

In [7]:
```
for key in token_dict.keys():

    next_words   = token_dict[key]

    unique_words = set(next_words) # removes duplicates
```

```
    nb_words     = len(next_words)

    probabilities_token = {}

    for unique_word in unique_words:
        probabilities_token[unique_word] = float(next_words.count(unique_word)) / n

    token_dict[key] = probabilities_token
```

In [8]:
```
token_dict
```

Out[8]:
```
defaultdict(list,
            {'the': {'cat': 0.75, 'dog': 0.25},
             'cat': {'is': 1.0},
             'is': {'green': 0.25, 'red': 0.25, 'brown': 0.25, 'blue': 0.25},
             'red': {'the': 1.0},
             'green': {'the': 1.0},
             'blue': {'the': 1.0},
             'dog': {'is': 1.0}})
```

Predicting next word

In [9]:
```
token_ask = 'the'
```

In [10]:
```
if token_ask in token_dict:
    next_words_prob = token_dict[token_ask]
    print(next_words_prob)
    print({k: v for k, v in sorted(next_words_prob.items(), key=lambda item: item[1
```

```
{'cat': 0.75, 'dog': 0.25}
{'cat': 0.75, 'dog': 0.25}
```

In [ ]:

In [ ]: