In [7]:

```
import nltk
from nltk.corpus import brown
```

In [8]:

```
def train_hmm_tagger () :
    tagged_sentence = brown.tagged_sents(categories = 'news')
    size = int(len(tagged_sentence)*0.9)
    trained_sents = tagged_sentence[:size]
    test_sents = tagged_sentence[size:]
    symbols = set([word for sentence in tagged_sentence for word,_ in sentence ])
    states = set([tag for sentence in tagged_sentence for _,tag in sentence])
    trainer = nltk.tag.hmm.HiddenMarkovModelTrainer(states = states , symbols = symbols)
    hmm_tagger = trainer.train_supervised(tagged_sentence)
    return hmm_tagger
```

In [9]:

```
def pos_tag_sentence(sentence, hmm_tagger):
    tokens = nltk.word_tokenize(sentence)
    tagged_tokens = hmm_tagger.tag(tokens)
    return tagged_tokens
```

In [10]:

```
hmm_tagger = train_hmm_tagger()
sentence = input("Enter the sentence to be tagged ? ")
tagged = pos_tag_sentence(sentence, hmm_tagger)
print(tagged)
```

```
Enter the sentence to be tagged ? i am feeling sleepy
[('i', 'DT-HL'), ('am', 'DT-HL'), ('feeling', 'DT-HL'), ('sleepy', 'DT-HL')]
```

In [11]:

```
train_hmm_tagger()
```

Out[11]:

```
<HiddenMarkovModelTagger 218 states and 14394 output symbols>
```

In [ ]:

```
```