Assignment 3

1. The IOB format categorizes tagged tokens as I, O and B.

Why are three tags necessary? **[1 Mark]**

What problem would be caused if we used I and O tags exclusively? **[1 Mark]**

1. Write a tag pattern to match noun phrases containing plural head nouns, e.g. "many/JJ researchers/NNS", "two/CD weeks/NNS", "both/DT new/JJ positions/NNS". Try to do this by generalizing the tag pattern that handled singular noun phrases. **[2 Marks]**
2. Pick one of the three chunk types in the CoNLL corpus. Inspect the CoNLL corpus and try to observe any patterns in the POS tag sequences that make up this kind of chunk. Develop a simple chunker using the regular expression chunker nltk.RegexpParser. Discuss any tag sequences that are difficult to chunk reliably. **[2 Marks]**
3. The Senseval 2 Corpus contains data intended to train word-sense disambiguation classifiers. It contains data for four words: hard, interest, line, and serve. Choose one of these four words, and load the corresponding data:

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **>>> from nltk.corpus import senseval**  **>>> instances = senseval.instances('hard.pos')**  **>>> size = int(len(instances) \* 0.1)**  **>>> train\_set, test\_set = instances[size:], instances[:size]** | |

Using this dataset, **build a classifier that predicts the correct sense tag** for a given instance. See the corpus HOWTO at http://nltk.org/howto for information on using the instance objects returned by the Senseval 2 Corpus. **[3 Marks]**

1. ☼ Using the movie review document classifier discussed in **Chapter 6.**[**Learning to Classify Text**](https://www.nltk.org/book/ch06.html), generate a list of the 30 features that the classifier finds to be most informative. Can you explain why these particular features are informative? Do you find any of them surprising? **[3 Marks]**
2. Consider the following sentence: “SpaceX Starship test flight cancelled minutes before blast-off”. Kindly parse this sentence admitted by the following grammar: **[3 Marks]**

|  |  |  |
| --- | --- | --- |
| |  |  | | --- | --- | |  | **grammar1 = nltk.CFG.fromstring("""**  **S -> NP VP**  **VP -> V NP | V NP PP**  **PP -> P NP**  **V -> "saw" | "ate" | "walked"**  **NP -> "John" | "Mary" | "Bob" | Det N | Det N PP**  **Det -> "a" | "an" | "the" | "my"**  **N -> "man" | "dog" | "cat" | "telescope" | "park"**  **P -> "in" | "on" | "by" | "with"**  **""")** | |
|  |