# Assignment 7

CSE 447 and 517: Natural Language Processing - University of Washington

Winter 2022

Please consult the course website for current information on the due date, the late policy, and any data you need to download for this assignment. This assignment is designed to give you more practical hands-on experience with sequence labeling.

★ **problems are for CSE 517 students only.** Other problems should be completed by everyone.

**Submit:** You will submit your writeup (a pdf) and your code (do not include data) via Gradescope. Instructions can be found here. Note that you will make two submissions: one for the pdf, one for the code.

$\underbrace{\texttt{Love}}_{\texttt{NP}}\ \underbrace{\texttt{loves}}_{\texttt{VP}}\ \texttt{to}\ \underbrace{\texttt{love}}_{\texttt{VP}}\ \underbrace{\texttt{love}}_{\texttt{NP}}\texttt{.}$

JAMES JOYCE
*Ulysses*

## 1 Whitespace Tokenizer for English – based on Eisenstein 8.7 (p. 180–1)

1. Using the NLTK library, download the complete text to the novel *Alice in Wonderland* by Lewis Carroll. Hold out the final 1000 words as a test set.

2. Let $\boldsymbol{x} = \langle x_1, x_2, \ldots, x_M \rangle$ be the sequence of non-whitespace characters in the training set. Each character $x_m$ will receive a binary label $y_m$, taking the value 1 if $x_m$ is the final character of a whitespace-separated token, and 0 otherwise. You can now represent your training data as two $M$-length strings, $\boldsymbol{x}$ (with no whitespace) and $\boldsymbol{y} \in \{0, 1\}^M$. Do the same with the test data.

3. Train a logistic regression classifier to predict $y_m$, using the eleven characters $\langle x_{m-5}, \ldots, x_m, \ldots, x_{m+5} \rangle$ as features. (You have some flexibility in precisely how you create the features.) After training the classifier, run it on the test set, using the predicted segmentation points to retokenize the text.

4. Compute the per-character segmentation accuracy on the test set. You should be able to get at least 88% accuracy.

5. Print out a sample of segmented text from the test set, e.g.

```
Thereareno mice in the air , I ’ m afraid , but y oumight cat
chabat , and that ’ svery like a mouse , youknow . But
docatseat bats , I wonder ?’
```

6. Per-character segmentation accuracy is easy to compute, but it is not a very intuitive way to talk about the system’s errors. A better way is to partition the tokens that you see in either the ground truth segmentation or the system’s segmentation as follows:

| | ground truth segmentation | not in ground truth segmentation |
|---|---|---|
| system’s segmentation | true positives | false positives |
| not in system’s segmentation | false negatives | true negatives |

This is one kind of *confusion matrix*. Calculate the numbers of (a) true positives, (b) false positives, and (c) false negatives. Use them to calculate the token-level precision, recall, and $F_1$ score of your system. (Hint: to match predicted and ground truth tokens, you can represent a token as a pair of integers: a start position and a length. If you represent the predicted output as a *set* of such integer-pairs, and do the same with the ground truth data, then the number of true positives is the size of these two sets’ intersections.)

## 2 ★ Extensions to the Tokenizer – based on Eisenstein 8.8 (p. 181)

1. Train a conditional random field sequence labeler, by incorporating the tag bigrams $(y_{m-1}, y_m)$ as additional features. You may also incorporate features that consider the tag bigrams as well as information from the input sequence. You may use a structured prediction library such as CRFSuite, or you may want to implement Viterbi yourself. Compare the accuracy to that of your classification-based approach.

2. Calculate the token-level precision, recall, and $F_1$ score of your CRF system.

## 3 Ungrammarian – based on Eisenstein 9.8 (p. 213)

Construct three examples—a noun phrase, a verb phrase, and a sentence—which can be derived from the Penn Treebank grammar fragment in §9.2.3 (pp. 204–8), yet are not grammatical in your dialect of English. Avoid reusing examples from the text. Propose corrections to the grammar to avoid generating these cases.

## 4 Extra Credit: Artistic Differences – based on Jurafsky & Martin 8.9

Names of works of art (books, movies, video games, etc.) are quite different from the kinds of named entities we've discussed in this class. Collect a list of names of works of art. You may use a Web-based source (e.g., gutenberg.org, amazon.com, imdb.com, etc.), but you are welcome to create the list from your own cultural experience. Do you think the techniques learned in class might fail to learn to recognize the kinds of entities in your list? Explain how and why.