Assignment 4: Named Entity Recognition

CSCI-5832 Natural Language Processing

Dan Prendergast

1. Introduction

Named Entity Recognition (NER) is an essential function of automated information extraction from a body of text. The NER process supports further function in the natural language processing pipeline such as coreference resolution or establishing links in a relational database. A popular approach to this problem is IOB tagging for each of the tokens in a sentence. This process can be accomplished using several algorithms – neural nets, Hidden Markov Models (HMMs), and even rule-based parsing. In this project, I implemented a HMM algorithm for performing NER and evaluated it using a body of text from biomedical journals which included named genes.

2. Problem Statement

In this assignment, we identified named genes in the text of a collection of biomedical journals. At a minimum, the task included calculating transition and observation probability models for IOB tags and implementing a Viterbi decoder to predict the most likely IOB tag sequence for finding the named entities in a sentence.

We used a corpus of biomedical journal articles for training and initial evaluation of our system. This data set included tokenized sentences with IOB tags for each token. The final test set was similarly tokenized; however, the test set did not include the IOB tags. Our task was to implement the IOB tagger, assign most likely tags to each token in the test set, and then save the results to a file matching the format of the training corpus.

3. Implementation

a. IOB Transition Probabilities

The transition probability matrix provides the probability of moving to IOB tag t_i given the previous tag t_j, or $p(t_i|t_j)$. The initial matrix was calculated by performing counts of each IOB tag given the preceding tag. One important addition to this procedure was the addition of a sentence break tag, *<blank>*. I believe the sentence break provides critical context to the presence or absence of named entities that start a sentence. That is, the beginning, "B" tag, may appear more frequently at the start of a sentence than others, since many sentences may begin with identifying a gene as the subject. During training using the corpus, I included the sentence break tags between each sentence to develop transition probabilities from sentence breaks to the next IOB, $p(t_i| < blank >)$.

To allow for the occurrence of unseen transitions in the test set, I smoothed the counts using a simple *add-1* smoothing approach, with the exception of <blank>→I and O→I transitions.

b. Observation Probabilities

In a typically HMM, the observation probabilities matrix indicates the probability of one observation given a tag. In NER, the word shape has been found to improve results (Jurafsky and Martin, 2018). In this project, I implemented two observation matrices – one for type (i.e., word) and the other for word shape.

Type Observation Probability Matrix

The type observation probabilities matrix encodes the probability that we observe a word type, w_i, given the tag, t_i, or $p(w_i|t_i)$. Again, a simple counting of occurrences of each word for each IOB tag provided the baseline data for this matrix. To account for the possibility of unseen words in test set, I added the *<unknown>* type for the list of possible types for each POS.

Word Shape Observation Probability Matrix

The shape probability matrix encodes the probability that we observe a word shape, s_i, given the tag, t_i. In this implementation we used a simple word shape where adjacent duplicate shapes were deleted from the word shape. For example, the words "Ribonucleic" and "DNA" would be represented as "Xx" and "X", as opposed to "Xxxxxxxxxx" and "XXX".

Smoothing

Simple add-1 smoothing was performed for both the Type and the Word Shape probability matrices.

Adjusting for Unknown Words and Shapes

Named entities commonly occur as words never seen before. For this reason any model for NER needs to handle unknown words and shapes appropriately. To address this issue, I performed one partial pass of the training data based on a percentage, p, to find the set of known words and word shapes. For the remainer of the training data, for any word or shape that was not in the known set, I counted the occurrences of each IOB tag. I then found the percentages of each tag that made up the total set of unknown words and shapes. I used these percentages to smooth the <unknown> probabilities in the Type and Word Shape probability matrices.

c. Viterbi Decoder

The implementation of the Viterbi decoder was in accordance with the algorithm provided in the course text (Jurafsky and Martin, 2018). My implementation calculated the IOB sequence one sentence at a time. The sequences for each sentence were then concatenated with blank spaces in between to create the sequence for the entire test set.

4. Assessment

Training was performed using a 90/10 training-to-test data split. Several iterations were made to vary the value of p, the percentage of the partial pass for identifying known words and shapes (described in "Adjusting for Unknown Words and Shapes" above), and determine its effect. The

table below shows the results from varying this percentage. Ultimately, the system performed best with $p = 0.95$ and resulted in an F1 score of 51.13%.

p	Precision (%)	Recall (%)	F1 (%)
0.001	51.81	41.75	46.23
0.3	51.53	49.79	50.65
0.6	48.79	52.19	50.43
0.95	49.15	53.27	51.13
0.999	48.51	52.14	50.26

Table 1: Results as a function of percentage training data scanned to find known words and shapes.

5. Discussion

It is difficult to predict the tagger's performance on the untagged test set without knowing the origin of that corpus. If the untagged test set is a randomly selected subset of the same biomedical corpus, I imagine the systems performance will be comparable to the results of the evaluation described above. If, however, the untagged test set comes from a different corpus, our ability to predict the tagger's performance is limited.

References

Jurafsky, Dan, and James H. Martin. *Speech and language processing*, 3rd ed., draft. https://web.stanford.edu/~jurafsky/slp3/