NLP 1 - Assignment 2

Selene Baez Santamaria

November 22, 2016

Exercise 1. Spelling corrector. Correct usages of *there* and *their*

(a) Unigram model

- Write the general equation for sentence probability under the unigram model.

$$P(w_1^n) = P(w_1) * P(w_2) * P(w_3) * ... * P(w_n)$$

- Does this seem like a good solution to the their vs. there problem? Justify your answer.

  This is not a good solution because it assumes words are independent and context free. Therefore, the joint probability directly depends on the count of *their/there*. Thus, if the training data has more of *their* it will prioritize this usage over *there*, and viceversa, regardless of which is the correct word.

(b) Bigram model

- Write the general equation for sentence probability under the bi-gram model.

$$P(w_1^n) = P(w_1) * P(w_2|w_1) * P(w_3|w_2) * ... * P(w_n|w_{n-1})$$

- Why might this model be better than the model in the previous question?

  By using a bigram model we include a certain extent of context in our language model. As such, we can record the words that *their* is usually surrounded by and note that they are not the same words that *there* is surrounded by. For example, *there* is usually followed by *is, are, have, had*, for example, while *their* could be followed by *cat, house, child, car*. Thus, in a bigram model the count for appearances is more meaningful, and the contribution of the words before and after *their/there* aids in correcting the grammar.

Exercise 2. Second order Markov assumption. A trigram model

(a) Give three examples in English where English grammar indicates that this independence assumption is very clearly violated.
In order to find examples where a trigram model is not sufficient, we need to find sentences that need four or more words to assemble together in particular ways to create a grammatically correct sentence. For example:

- **The bag of chips I have in my hands are delicious.**: Subject verb agreement between *bag of chips* and *are* is incorrect. In a trigram model, *bag* can only look as far as *of chips*, thus missing the verb completely.
- **My mom, as well as all the females in my family, are always hungry.** Subject verb agreement between *mom* and *are* is incorrect.
- **The cheese, which in my opinion is the best discovery of mankind, have expired.**: Subject verb agreement between *cheese* and *have* is incorrect.

Exercise 3. HMM Tagger. Name entity recognition.

(a) Transition probability matrix

In order to build the transition matrix we make the following assumptions:

- Dots are marked as $<OTH>$
- After a dot finalizes the sentence, we add the "end of sentence" tag $</s>$
- The last tag $</s>$ does not transition into anything else, and is not counted in the table.

$$\mathbf{P} = \begin{array}{r|rrrrr|} & <s> & <PER> & <ORG> & <OTH> & </s> \\ <s> & 0/5 & 2/5 & 0/5 & 3/5 & 0/5 \\ <PER> & 0/6 & 3/6 & 0/6 & 3/6 & 0/6 \\ <ORG> & 0/16 & 0/16 & 9/16 & 7/16 & 0/16 \\ <OTH> & 0/90 & 1/90 & 7/90 & 77/90 & 5/90 \\ </s> & 4/5 & 0/5 & 0/5 & 0/5 & 0/5 \end{array}$$

(b) Transition probability matrix with add-one smoothing

To smooth the matrix, we add one to all the cells in the matrix, except for the transition between $<s> \Rightarrow </s>$ since we have an explicit assumption that there are no empty sentences.

$$\mathbf{P} = \begin{array}{r||rrrrr||}
 & <s> & <PER> & <ORG> & <OTH> & </s> \\
<s> & 1/9 & 3/9 & 1/9 & 4/9 & 0/9 \\
<PER> & 1/11 & 4/11 & 1/11 & 4/11 & 1/11 \\
<ORG> & 1/21 & 1/21 & 10/21 & 8/21 & 1/21 \\
<OTH> & 1/95 & 2/95 & 8/95 & 78/95 & 6/95 \\
</s> & 5/9 & 1/9 & 1/9 & 1/9 & 1/9
\end{array}$$

We note that adding ones in the last row, columns 2-5, introduces cases where the end of a sentence may followed by anything other than the start of a new sentence (e.g. Obama/PER was a community organizer in Chicago before earning his law degree . civil rights attorney ...). This means that sentences could start ”in the middle ” which is undesirable. Hence we remove these cases.
Furthermore, adding ones in the first column, rows 1-5, implies that the start of a sentence could be preceded by anything other than the end of the previous sentence. This would be a mistake in our tagging, thus we remove these cases too.

Similarly, adding ones in the last column, rows 2-3, implies that any tag could precede end of a sentence, and not necessarily a *./OTH*. Although this will form grammatically incorrect sentences, it is not impossible to find sentences as such in the real life, thus we keep these cases.

The final transition matrix is:

$$\mathbf{P} = \begin{array}{r||rrrrr||}
 & <s> & <PER> & <ORG> & <OTH> & </s> \\
<s> & 0/8 & 3/8 & 1/8 & 4/8 & 0/8 \\
<PER> & 0/10 & 4/10 & 1/10 & 4/10 & 1/10 \\
<ORG> & 0/20 & 1/20 & 10/20 & 8/20 & 1/20 \\
<OTH> & 0/94 & 2/94 & 8/94 & 78/94 & 6/94 \\
</s> & 5/5 & 0/5 & 0/5 & 0/5 & 0/5
\end{array}$$

(c) What are the estimates for $P\left(Obama|PER\right)$ and $P\left(Obama|ORG\right)$

Original emission estimates:

$$P(Obama|PER) = \frac{count(Obama)}{count(PER)} = \frac{3}{6}$$
$$P(Obama|ORG) = \frac{count(Obama)}{count(ORG)} = \frac{0}{16}$$

Add-one smoothing, taking a vocabulary size of 69:

$$P(Obama|PER) = \frac{3+1}{6+V} = \frac{3+1}{6+69} = \frac{4}{75}$$
$$P(Obama|ORG) = \frac{0+1}{16+V} = \frac{0+1}{16+69} = \frac{1}{85}$$

(d) Suppose we used four tags for this task: the three already mentioned, plus a LOC tag for locations. In a general text, will context always be able to disambiguate between the LOC and ORG tags?

The difference between locations and organizations can be ambiguous in cases like *United States*, *Columbia* or *Chicago*. The latter two can be disseminated by context, for example the *University* after *Columbia*, implies we are referring to an organization and not a location. However, cases like *United States* cannot be disseminated by context since the difference is semantic distinguishing between a political economical state, and a piece of land.

(e) Can you think of any sources of information that might help an automatic NER system perform better, but which are not used by an HMM tagger? Back up your answer with examples from the text here, or give examples that could occur in another text.

The semantic web is a valuable resource for an automatic NER system. The usage of ontologies like FOAF and DBpedia can help to recognize individuals and locations.

For example, Barack Obama can be linked to this DBpedia resource: `http://dbpedia.org/page/Barack_Obama`

Exercise 4. Bigram tagging. Tag the sentence: *The healthy man the lifeboats*

(a) Use the tags DT, N, V, ADj
As instructed, we assume that *the* can only be tagged as DT, and *lifeboat* can only be tagged as N. Thus, we know that the beginning of our sentence has a DT tag, while the end of the unknown sequence (*healthy man*) finishes with a DT tag as well. This lead to:

The/DT *healthy*/UNKNOWN *man*/UNKNOWN *the*/DT *lifeboats*/N.

In order to determine the two unknown tags we look at the frequency data.

- healthy: Since the previous word was tagged as DT, we need only to look at the first column of its frequency table:

$$\begin{aligned}
P(healthy = N|DT) &= \frac{8}{42}\\
P(healthy = V|DT) &= \frac{0}{42}\\
P(healthy = Adj|DT) &= \frac{34}{42}
\end{aligned}$$

- man: Next, we look at the third and fourth column, since they correspond to *man* succeeding a N or Adj tag, which are the only valid option for *healthy*:

$$\begin{aligned}
P(man = N|healthy = N) &= \frac{45}{56} * \frac{8}{42} = 0.1531\\
P(man = V|healthy" = N) &= \frac{11}{56} * \frac{8}{42} = 0.0374\\
P(man = Adj|healthy" = N) &= \frac{0}{56} * \frac{8}{42} = 0
\end{aligned}$$

$$\begin{aligned}
P(man = N|healthy = Adj) &= \frac{86}{90} * \frac{34}{42} = 0.7735\\
P(man = V|healthy" = Adj) &= \frac{4}{90} * \frac{34}{42} = 0.0356\\
P(man = Adj|healthy" = Adj) &= \frac{0}{56} * \frac{34}{42} = 0
\end{aligned}$$

The sequence that maximizes the likelihood is $(man = N|healthy = Adj)$, hence we tag the sentence as:

The/DT *healthy*/Adj *man*/N *the*/DT *lifeboats*/N.

Exercise 5. Viterbi. Tag the sentence: *The healthy man the lifeboats*

(a) Hand simulate the Viterbi algorithm using the given transition and emission probabilities

We name the transition matrix as A, with entries $a_{i,j}$ corresponding to the transition from $i \Rightarrow j$. We name the emission matrix as B, with entries $b_j(word)$ corresponding to a tag j emitting *word*.

The Viterbi simulation has the following steps:

1) Create a matrix of J by T, where J is the number of POS tags, and T is the number of words in the sentence.

v(j,t)	The	healthy	man	the	lifeboats
DT	v(1,1)	v(1,2)	v(1,3)	v(1,4)	v(1,5)
N	v(2,1)	v(2,2)	v(2,3)	v(2,4)	v(2,5)
V	v(3,1)	v(3,2)	v(3,3)	v(3,4)	v(3,5)
Adj	v(4,1)	v(4,2)	v(4,3)	v(4,4)	v(4,5)

2) Initialize the matrix by filling in the leftmost column as:

$$v(j, t = 1) = a_{i=<s>,j} * b_j(word = the_{t=1})$$

v(j,t)	The	healthy	man	the	lifeboats
DT	(0.4 * 0.5) = 0.2	v(1,2)	v(1,3)	v(1,4)	v(1,5)
N	(0.3 * 0) = 0	v(2,2)	v(2,3)	v(2,4)	v(2,5)
V	(0.1 * 0) = 0	v(3,2)	v(3,3)	v(3,4)	v(3,5)
Adj	(0.2 * 0) = 0	v(4,2)	v(4,3)	v(4,4)	v(4,5)

At this point we note that many entries will be filled with 0, since the emission matrix is sparse. From this point forward, before performing any computations, we check if the corresponding $b_j(word_t)$ term is zero, and directly assign a 0 to the computation if it is.

3) Continue to fill in columns going from left to right[1]. Use the following formula:

$$v(j, t) = \max_{i=1}^{J} \left(v(i, t-1) * a_{i,j} * b_j(word_t)\right)$$

$$V(1, 2) = max \left\{ \begin{array}{l} v(i = 1, t = 1) * a_{i=1,t=1} * b_{j=1}(word = healthy_{t=1}) = 0 \\ v(i = 2, t = 1) * a_{i=2,t=1} * b_{j=1}(word = healthy_{t=1}) = 0 \\ v(i = 3, t = 1) * a_{i=3,t=1} * b_{j=1}(word = healthy_{t=1}) = 0 \\ v(i = 4, t = 1) * a_{i=4,t=1} * b_{j=1}(word = healthy_{t=1}) = 0 \end{array} \right\}$$

$$= 0$$

$$V(2, 2) = max \left\{ \begin{array}{l} v(1, 1) * a_{1,2} * b_2(healthy) = 0.2 * 0.6 * 0.2 = 0.024 \\ v(2, 1) * a_{2,2} * b_2(healthy) = 0 \\ v(3, 1) * a_{3,2} * b_2(healthy) = 0 \\ v(4, 1) * a_{4,2} * b_2(healthy) = 0 \end{array} \right\} = 0.024$$

[1] For illustration purposes, we perform the steps for the second column, and assume the next columns follow the same algorithm.

$$V(3,2) = max \left\{ \begin{array}{l} v(1,1) * a_{1,2} * b_3(healthy) = 0 \\ v(2,1) * a_{2,2} * b_3(healthy) = 0 \\ v(3,1) * a_{3,2} * b_3(healthy) = 0 \\ v(4,1) * a_{4,2} * b_3(healthy) = 0 \end{array} \right\} = 0$$

$$V(4,2) = max \left\{ \begin{array}{l} v(1,1) * a_{1,4} * b_4(healthy) = 0.2 * 0.4 * 0.4 = 0.032 \\ v(2,1) * a_{2,4} * b_4(healthy) = 0 \\ v(3,1) * a_{3,4} * b_4(healthy) = 0 \\ v(4,1) * a_{4,4} * b_4(healthy) = 0 \end{array} \right\} = 0.032$$

Leading to the following partial result:

v(j,t)	The	healthy	man	the	lifeboats
DT	0.2	0	v(1,3)	v(1,4)	v(1,5)
N	0	0.024	v(2,3)	v(2,4)	v(2,5)
V	0	0	v(3,3)	v(3,4)	v(3,5)
Adj	0	0.032	v(4,3)	v(4,4)	v(4,5)

The complete matrix is shown here:

v(j,t)	The	healthy	man	the	lifeboats
DT	0.2	0	0	0.000192	0
N	0	0.024	0.0048	0	0.00002304
V	0	0	0.00096	0	0
Adj	0	0.032	0	0	0

4) Select most likely path: *The*/DT *healthy*/Adj *man*/N *the*/DT *lifeboats*/N.

(b) Give the joint probability

The joint probability is 0.00002304