CS 325 Quiz 1

Qingyuan Zhang

P(w[0],…,w[6]) is a conditional probability. In order to estimate its value, we have to use the Markov assumption in order to simplify the process. Due to the Markov assumption, we can approximate estimate that P(w[0],…,w[6]) = P(w[0]) \* P(w[1]|w[0])\* P(w[2]|w[1]) \* P(w[3]|w[2]) \* P(w[4]|w[3]) \* P(w[5]|w[4]) \* P(w[6]|w[5])

The first P(w[0]) is the probability that only “he” appears. As a result P(“he”) can be generate by our unigram class. P(w[1]|w[0]) to P(w[6]|w[5]) are bigram, since each of them are sequence of two words. So the probability which is the maximum likelihood could be gain by bigram class.

After we get each value of P(w[0]), P(w[1]|w[0]), P(w[2]|w[1]) ,P(w[3]|w[2]), P(w[4]|w[3]), P(w[5]|w[4]), P(w[6]|w[5]). We simply plug them into equation and multiply them together to get the final estimation of this whole words sequence.

**package** main.ngrams;

**import** java.io.FileInputStream;

**import** java.io.FileNotFoundException;

**import** main.ngrams.smoothing.NoSmoothing;

**import** main.utils.IOUtils;;

**public** **class** **Quiz1** {

**public** **static** **void** **main**(**String**[] args) **throws** **FileNotFoundException** {

**Unigram** **unigram** = **new** Unigram(**new** NoSmoothing());

**Bigram** **bigram** = **new** Bigram(**new** NoSmoothing());

**IOUtils**.*readUnigrams*(unigram, **new** FileInputStream("C:\\Users\\Qingyuan\\workspace\\cs325\\src\\main\\ngrams\\1grams.txt"));

**IOUtils**.*readBigrams* (bigram , **new** FileInputStream("C:\\Users\\Qingyuan\\workspace\\cs325\\src\\main\\ngrams\\2grams.txt"));

**String**[] **w** = {"he","came","to","my","school","to","study"};

**double** **p** = 1;

**double** [] **prob** = **new** **double** [7];

prob [0] = unigram.getLikelihood(w[0]);

**for** (**int** **i** = 1; i <=6; i++){

prob[i] = bigram.getLikelihood(w[i-1],w[i]);

}

**for**(**int** **j** = 0; j<=6;j++){

p = p \* prob[j];

}

**System**.***out***.println(p);

}