ASSIGNMENT 3

Subject : Word Sense Disambiguation using Naive Bayes
Handed Out : 04.04.2018
Due date : 20.04.2018

Please submit your solution (code and a README file) by 17:00 pm on the due date. Please describe your code in detail in README file.

Introduction

Word Sense Disambiguation (WSD) is the task of identifying which sense of an ambiguous word given a context. Table 1 shows an example for the word "pen".

Naive Bayes (NB) classifiers is one of the best methods for supervised approach for WSD. We try to choose correct sense of a word (e.g. "pen") in this assignmen by using Naive Bayes Classifier. **(You must implement the Naïve Bayes Classifier)**

Table 1: Possible definitions for the sense tags for **pen**

Sense	Definition
pen	a writing implement with a point from which ink flows
pen	an enclosure for confining livestock
pen	a portable enclosure in which babies may be left to play
pen	a correctional institution for those convicted of major crimes
pen	female swan

0.1 Naive Bayes Classifier

Naive Bayes (NB) methods have been used for classification task. For word sense disambiguation, the context in which an ambiguous word occurs is represented by a vector of feature variables $F = (f_1, f_2, \cdots, f_n)$ and the sense of the ambiguous word is represented by classification variables $(s_1, s_2, \cdots, s_k)$. Choosing the right sense of the ambiguous word is finding the sense s_i that maximizes the conditional probability $P(w = s_i|F)$.

Suppose C is the context of the target word w, and $F = (f_1, f_2, \cdots, f_n)$ is the set of features extracted from context C, to find the right sense s' of w given context C, we have:

$$\begin{aligned} s' &= \underset{s_i}{\operatorname{argmax}} P(w = s_i|F) \\ &= \underset{s_i}{\operatorname{argmax}} \frac{P(F|w = s_i)}{P(F)} * P(s_i) \\ &= \underset{s_i}{\operatorname{argmax}} P(F|w = s_i) * P(s_i) \end{aligned} \tag{1}$$

The NB classifier works with the assumption that the features are conditional independent, so that we have

$$s' = \underset{s_i}{\operatorname{argmax}} \prod_{f_j \in C} P(f_j|w = s_i) * P(s_i) \tag{2}$$

The probability of sense s_i, $P(s_i)$, and the conditional probability of feature f_j with observation of sense s_i, $P(f_j|s_i)$, are computed via Maximum-Likelihood Estimation:

$$\begin{aligned} P(s_i) &= C(s_i)/N \\ P(f_j|w = s_i) &= C(f_j, s_i)/C(s_i) \end{aligned} \tag{3}$$

Where $C(f_j, s_i)$ is the number of occurrences of f_j in a context of sense s_i in the training corpus, $C(s_i)$ is the number of context words belonging to s_i in the training corpus, and N is the total number of words in the training corpus. To avoid the effects of zero counts when estimating the conditional probabilities of the model, when meeting a new feature f_j in a context of the test dataset, for each sense s_i we set $P(f_j|w = s_i)$ equal $1/N$.

Features [1]
Let w_i be the word at position i in the context of the ambiguous word w and p_i be the part-of-speech tag of w_i. Note that word w appears precisely at position 0 and i will be negative (positive) if w_i appears on the left (right) of w. We select the following features for the model:

$F1$ is a set of unordered words in the large context (You will use a window size ± 3),
$F1 = \cdots, w_{-3}, w_{-2}, w_{-1}, w_1, w_2, w_3, \cdots$
$F2$ is a set of part-of-speech tags assigned with their positions in the local context,
$F2 = \cdots, (p_{-3}, -3), (p_{-2}, -2), (p_{-1}, -1), (p_1, 1), (p,2), (p_3, 3) \cdots$

P.S. : Stop words and punctuations will not be included to feature vector. Use Porter Stemmer for the F1 features.

Once you find the right tags for the word tokens in the test set, they will be written in an output file (i.e. out.txt), such as

```
accident-n.700001 532675
accident-n.700002 532675
accident-n.700003 532675
accident-n.700004 532675
accident-n.700005 532675
accident-n.700006 532675
accident-n.700007 532675
accident-n.700008 538889
accident-n.700009 532675
accident-n.700010 538889
accident-n.700011 532675
```

---

which means the word in the instance 700001 is tagged with the sense id 532675 and the word in the instance 700008 is tagged with the sense id 538889.

Dataset

1. Each disambiguated word and its senses is preceded by <lexelt item= and followed by / >
2. Each sense of the disambiguated word is defined with instance id
3. Each word's PoS tag is given after the word preceded by <p= and followed by / >
4. The word to be disambiguated and its tag are preceded by <head >and followed by </ head >. You need to use regex to extract the words and their tags.

Example instance of the word "accident" to be tagged with one of the sense is given below:

```
<lexelt item="accident-n">

<instance id="accident-n.800001">

<answer instance="accident-n.800001" senseid="532675"/>

<context>

late <p="RB"/> on <p="IN"/> thursday <p="NNP"/> night <p="NN"/> it <p="PRP"/>
was <p="VBD"/> travelling <p="VBG"/> at <p="IN"/> about <p="IN"/>
three <p="CD"/> metres <p="VBZ"/> a <p="DT"/> second <p="NN"/>
 in <p="IN"/> wind <p="NN"/> blowing <p="VBG"/> at <p="IN"/>
20 <p="CD"/> to <p="TO"/> 25 <p="CD"/> knots <p="NNS"/> when <p="WRB"/>
 an <p="DT"/> empty <p="JJ"/> car <p="NN"/> ell <p="VBD"/> off <p="IN"/>
just <p="RB"/> as <p="IN"/> it <p="PRP"/> reached
 <p="VBD"/> the <p="DT"/> top <p="NN"/> . <p="."/>
```

```
the <p="DT"/> <head>accident <p="NN"/></head> appeared <p="VBD"/>
to <p="TO"/> have <p="VB"/> little <p="JJ"/> effect <p="NN"/>
 on <p="IN"/> the <p="DT"/> christmas <p="NNP"/> party <p="NN"/>
 , <p=","/> except <p="VB"/> to <p="TO"/> lengthen <p="VB"/>
 it <p="PRP"/> considerably <p="RB"/> . <p="."/>
</context>

</instance>
```

To download the dataset, Porter Stammer and stopWords.txt, please click the link :
Dataset : `https://piazza.com/class_profile/get_resource/jdiv1jqrgrp7kl/jfi5ojhceap3qu`

Notes

- Do not miss the submission deadline.
- Compile your code on *dev.cs.hacettepe.edu.tr* before submitting your work to make sure it compiles without any problems on our server.
- Save all your work until the assignment is graded.
- The assignment must be original, individual work. Duplicate, very similar assignments or code from Internet are going to be considered as cheating.
- You can ask your questions via Piazza and you are supposed to be aware of everything discussed on Piazza. You cannot share algorithms or source code. All work must be individual! Assignments will be checked for similarity, and there will be serious consequences if plagiarism is detected.
- You need to implement either in **Java** (Java 1.8) or **Python** (Python 3). Please submit your source codes and README file in the following submission format.
- You will be graded not only for the output, but also readibility, comment lines and README.md.
- I will run your programs from the command line as following. Any other command line format will not be accepted!
  Python

  python3 assignment3.py train-S1.pos test-S1.pos out.txt

  Java

  Java Main train-S1.pos test-S1.pos out.txt

  → <studend id>
  → code.zip
  → README.md

References

[1] Cuong Anh Le and Akira Shimazu. High wsd accuracy using naive bayesian classifier with rich features. In *Proceedings of the 18th Pacific Asia Conference on Language, Information and Computation*, pages 105–114, 2004.