Mini-project 3: Naive Bayes for WSD

Implement a Naive Bayes model from scratch, then train it to perform supervised word sense disambiguation on the Senseval2 annotated corpus (see Laborator 6 for access to the dataset).

You will implement the computations for the parameters of Naive Bayes without using the NLTK/sklearn library (or any other implementation).
You can esti

$$P(c_i \mid s_k) = \prod_{v_j \; in \; c_i} P(v_j \mid s_k) = \prod_{j=1}^{J} \left(P(v_j \mid s_k)\right)^{|v_j \; in \; c_i|},$$

where by $|v_j \; in \; c_i|$ we denote the number of occurrences of feature v_j in context c_i. Then, the likelihood of the corpus $\mathscr{C}$ is

$$P(\mathscr{C}) = \prod_{i=1}^{I} \sum_{k=1}^{K} P(s_k) \prod_{j=1}^{J} \left(P(v_j \mid s_k)\right)^{|v_j \; in \; c_i|}$$

The parameters of the probability model with independent features are

$$\left\{P(s_k), k = 1, ..., K \text{ and } P(v_j \mid s_k), \quad j = 1, ..., J, k = 1, ..., K\right\}$$

Notation:

- $P(s_k) = \alpha_k$, $k = 1, ..., K$, $\alpha_k \geq 0$ for all k, $\sum_{k=1}^{K} \alpha_k = 1$
- $P(v_j \mid s_k) = \theta_{kj}$, $k = 1, ..., K$, $j = 1, ..., J$, $\theta_{kj} \geq 0$ for all k and j, $\sum_{j=1}^{J} \theta_{kj} = 1$ for all $k = 1, ..., K$

mate the probabilities of words occurring in the context based on their frequencies in the training data.

Naive Bayes model preliminaries

The Naive Bayes model in the context of word sense disambiguation involves the probabilities of senses of words given their contexts (where the context is encoded as a set of features):

With this notation, the likelihood of the corpus $\mathscr{C}$ can be written as

$$P(\mathscr{C}) = \prod_{i=1}^{I} \sum_{k=1}^{K} \alpha_k \prod_{j=1}^{J} (\theta_{kj})^{|v_j \ in\ c_i|}$$

The well known Bayes classifier involves the a posteriori probabilities of the senses, calculated by the Bayes formula for a specified context c,

$$P(s_k \mid c) = \frac{P(s_k) \cdot P(c \mid s_k)}{\sum_{k=1}^{K} P(s_k) \cdot P(c \mid s_k)} = \frac{P(s_k) \cdot P(c \mid s_k)}{P(c)},$$

with the denominator independent of senses.

The Bayes classifier chooses the sense s' for which the a posteriori probability is maximal (sometimes called the Maximum A Posteriori classifier)

$$s' = \arg\max_{k=1,\ldots,K} P(s_k \mid c)$$

$$s' = \arg\max_{k=1,\ldots,K} (\log P(s_k) + \log P(c \mid s_k))$$

Parameter estimation

For *supervised disambiguation*, where an annotated training corpus is available, the parameters are simply estimated by the corresponding frequencies:

$$\widehat{\theta}_{kj} = \frac{|occurrences\ of\ v_j\ \ in\ a\ context\ of\ sense\ s_k|}{\sum_{j=1}^{J} |occurrences\ of\ v_j\ \ in\ a\ context\ of\ sense\ s_k|},$$

$$k = 1, \ldots, K;\ j = 1, \ldots, J$$

$$\widehat{\alpha}_k = \frac{|occurrences\ of\ sense\ s_k\ in\ \mathscr{C}|}{|occurrences\ of\ w\ in\ \mathscr{C}|}, \quad k = 1, \ldots, K$$

Create a function that trains a Naive Bayes model, where features are words in the target word's context. Customizable parameters:

- window size
- vocabulary size

Split the data into train set / validation set / test set. Train the implemeted model on the training set, and choose the optimal window/vocabulary size which maximizes accuracy on the validation set.

Evaluate the model on the test set using:

- accuracy
- confusion matrix

There will be one model for each of the 4 target words in the Senseval2 corpus (“line” / “hard” / “serve” / “interest”), each with its own training and optimization process.

Optional: add Laplacian smoothing (add +1 to each probability term) in order to avoid zero-valued probabilities: https://courses.cs.washington.edu/courses/cse446/20wi/Section7/naive-bayes.pdf

Deliverables (to send before deadline)

- code used to implement the model
- statistics and results, for each of the 4 target words:
    - report obtained scores in terms of the metrics above
    - a few selected examples of errors

You can send the deliverables either as a Google Colab file or .py and .txt files. If you choose the latter, please upload the files in a cloud storage of your own (e.g. Google Drive) and send the links to me by email (rather than email attachments).

Deadline: 3 weeks (06.06.2022)
This mini-project accounts for **2p.**

Bonus points (+0.5p): Experiment with different features, such as including the POS of the words in the context as features as well, measure and compare the performance.