ANISHA VIJAYAN (UIN: 662618335)
SALONI KATARIA (UIN: 662519005)
JAHNAVI MUTHYALA (UIN:667960987)
NAINI NARAMA LNU (UIN:679008394)

IDS566: Advanced Text Analytics
Mini-Project #2

(a) Explain all WSD systems that you have built. Ideally two systems: the Simple Lesk and the Corpus Lesk for ontological WSD. Another two systems: add-1 and add-lambda smoothing Naive-Bayes for supervised WSD.
Ans.

❖ SUPERVISED WORD SENSE DISAMBIGUATION (Naïve Bayes Model):

We have used co-occurrence feature extraction method with window size as 1. We took the train.data and test.data files as input. For every distinct target word in the test data file, we have generated the train model, by first fetching the feature vectors of the target word from the train data based on the target word's index and the window size specified, then we have calculated the count of different sense ids for the target word, based on this we computed the prior probabilities for the target word. Then we counted the occurrences of feature words within the context of target word and calculated the feature probability for each sense. If smoothing was enabled, then we computed the smoothed feature probability for each sense of the target word. After generating the train model for the target word, for every sentence in the test data for the target word, we checked if the feature words from the sentence exists in the train model that we obtained in the previous step. If it exists then we collected the individual feature probabilities, took the product of it and multiplied it with the prior probability to get the final probability of the sense of the target word. We did this for all different sense ids. Out of all the probabilities, we took the sense id with the max probability as the predicted sense id. This was done for all the target words from test data and extracted into a csv file.

- **Add-1 smoothing:** We used the lambda value as 1 and got an accuracy of 54%
- **Add-lambda smoothing:** We experimented with different lambda values like 0.01, 0.001 and also with different wind sizes as 1, 2.

Lambda	Window Size	Accuracy
1	1	53.59%
0.001	**1**	**81.78%**
0.01	1	81.14%
0.01	2	74.17%

❖ ONTOLOGICAL WORD SENSE DISAMBIGUATION

To implement ontological word sense disambiguation, we have chosen to apply three models based on: Simple Lesk, Original Lesk and Corpus Lesk Algorithms. We calculate accuracy for each of them and choose the one which gives us the maximum accuracy to predict the senses for the test data.
For each model, we use the data and it is preprocessed: Dictionary.xml, Train.data, Validation.data, Test.data

Data Preprocessing: Split the data, Data cleansing which involved removing punctuations and digits, lowercasing the sentence, lemmatizing and retrieving the POS tags of each word from the test sentence using WordNet, removing stop words.
To increase the overlap matching, we have used ***lemmatization***.
For example, 'He was running' is lemmatized to 'He be run'

Simple Lesk Algorithm:

Principle: The Lesk algorithm is based on the principle that words in a given text will have a similar meaning. In the Simple Lesk Algorithm, the meaning of each word context is generated by getting the sense which overlaps the most among the given context and its dictionary meaning.
Model Building: After data split and preprocessing, for each sense id of the target word, we calculate the count of overlapping words between the target word's gloss + examples from the dictionary provided, and the lemmatized words from the given test sentence. This count is taken as the score for each sense id.
The sense ID with maximum overlap score is taken as the predicted sense ID for the target word.
Accuracy achieved: Validation Data: **49.088** ; Train Data: **47.650**

Original Lesk Algorithm:

ANISHA VIJAYAN (UIN: 662618335)
SALONI KATARIA (UIN: 662519005)
JAHNAVI MUTHYALA (UIN:667960987)
NAINI NARAMA LNU (UIN:679008394)

Principle: The Original Lesk algorithm is based on the principle that words in a given text will have a similar meaning. In the Original Lesk Algorithm, the meaning of each word context is generated by getting the sense which overlaps the sense of context words in the sentence and the sense of target word.

Model Building: After data split and preprocessing, to build model on Original Lesk algorithm, we derive the following:

- *Target Sense*: senses of the target words from the dictionary given, lemmatize them
- *Context Words*: these are all words from given sentence apart from stopwords and the target word itself, lemmatize them
- *Context Sense*: Sense of each lemmatized context word is generated, lemmatize them
- *Single word score*: Overlapping score of target sense and context sense
- *Consecutive word score*: Bigrams for the target and context sense are generated and added twice to the single word score to give higher weightage

In this model, we derived a metric in an attempt to increase the overlap between senses: BIGRAMS
We tried calculating the score of overlaps by adding the intersection score of the (target + context) senses calculated before and twice the score of bigram context score.
The sense ID with maximum overlap score is taken as the predicted sense ID for the target word.
Accuracy achieved: Validation Data: **41.37** ; Train Data: **41.69**

Corpus Lesk Algorithm:

Principle: The Corpus Lesk algorithm is based on the same principle as the Simple Lesk but it uses an augmented dictionary which is generated using the training data.
Model Building: For each sense of the target word, we calculate the count of overlapping words between the target word's gloss + examples from the new dictionary, and lemmatized words from the given test sentence for each sense of the target word. The count is taken as the score for each sense id.
The sense ID with maximum overlap score is taken as the predicted sense ID for the target word.
Accuracy achieved: Validation Data: **83.39** ; Train Data: **98.70**

(b) Try various scoring functions and different feature engineering. Pick the best one for each WSD system, providing several intuitive real examples chosen from the training data that can justify your design decisions.

Ans. For feature engineering, in case of Supervised WSD model, while generating the train model we limited our dataset to 100 rows for each target word and generated the model based on that data. In case of scoring function, we have trained a classifier for each word in the training data and took the sense with the max probability based on the Naïve Bayes Model computation. Also we experimented with different lambda values and window sizes, N.
For example, in case of **lambda = 0.01 and N = 1** for the target word '**say.v**' in the context '*Cadillac posted a 3.2 % increase despite new competition from Lexus , the fledging luxury-car division of Toyota Motor Corp . Lexus sales weren't available ; the cars are imported and Toyota reports their sales only at month-end . The sales drop for the No. 1 car maker may have been caused in part by the end in September of dealer incentives that GM offered in addition to consumer rebates and low-interest financing , a company spokesman %% said %% . Last year , GM had a different program in place that continued rewarding dealers until all the 1989 models had been sold . Aside from GM , other car makers posted generally mixed results*.', we get the predicted sense id same as the actual sense id, **1**.
But in case of **lambda = 0.01 and N=2**, the predicted sense is **4** based on more overlapping terms from the context and target sense.
And we got better overall accuracy in case of window size =1, therefore we stuck to lambda = 0.01 and window size = 1 for predicting the test data.

For scoring functions in ontological WSD, we have taken the score as the count of overlapping words between the target sense and the given context in case of Simple Lesk, count of overlapping words between the target sense and the given context sense in case of Original Lesk. We also took the count of overlapping words appearing consecutively in both target sense and the given context sense in case of Original Lesk and added twice of this count to the single word overlaps. For the Corpus Lesk, we have taken the count of overlapping words between the target sense and the given context.
For example, in case of the word '**affect.v'** in the context '*Some U.S. allies are complaining that President Bush is pushing conventional-arms talks too quickly , creating a risk that negotiators will make errors that could %% affect %% the security of Western Europe for years . Concerns about the pace of the Vienna talks -- which are aimed at the destruction of some 100,000 weapons , as well as major reductions and realignments of troops in central Europe -- also are being registered at the Pentagon . Mr. Bush has called for an agreement by next September at the latest .*' , Simple Lesk algorithm incorrectly predicts the sense id as 2, Original Lesk with the consecutive overlapping incorrectly predicts the sense as 3, whereas Corpus Lesk algorithm correctly predicts the sense as 1.

ANISHA VIJAYAN (UIN: 662618335)
SALONI KATARIA (UIN: 662519005)
JAHNAVI MUTHYALA (UIN:667960987)
NAINI NARAMA LNU (UIN:679008394)

The same is the case with word '**allow.v**' In the context '*Lloyd 's of London said it plans to clamp down on the ability of underwriting syndicates to leave their annual accounts open beyond the customary three years . Underwriting syndicates at Lloyd 's , the world 's largest insurance market , generally do n't close their accounts for three years , to %% allow %% for the filing of claims and litigation . When such claims and litigation extend beyond the period , the syndicates can extend their accounting deadlines . Lloyd 's said there are currently 115 open account years involving 68 of the market 's roughly 360 syndicates .*' Simple Lesk and Original Lesk predicted the sense as 1 whereas only Corpus Lesk predicted it as 2 correctly. Also, generally we obtained a higher accuracy for Corpus Lesk algorithm since it includes more amount of data in the dictionary by augmenting it with the training data.

(c) Report the comparative performance among your ontological WSD systems with table/-graphs by testing on the validation set. Report the comparative performance similarly among your supervised WSD systems. Note that there must be a baseline WSD system that always predicts to the most frequent sense. No comparisons with the baseline cannot justify performance of your systems. Finally compare the entire WSD systems, reporting clearly labeled tables/graphs with a written summary of the results.
Ans.
For ontological WSD:

Model	**Accuracy**
Simple Lesk	49.09%
Original Lesk without consecutive overlapping score	41.26%
Original Lesk with consecutive overlapping score	41.37%
Corpus Lesk	83.39%

For supervised WSD:

Model parameters	**Accuracy**
lambda = 1 and window size = 1	53.59%
lambda = 0.001 and window size = 1	81.78%
lambda = 0.01 and window size = 1	81.14%
lambda = 0.01 and window size = 2	74.17%
Baseline model	80.70%

Accuracies calculated based on different models:

Models	**Accuracy**
Baseline Model	80.70%
Supervised WSD (Naïve Bayes)	81.78%
Simple Lesk	49.09%
Original Lesk	41.26%
Corpus Lesk	**83.39%**

As we saw in the previous question as well, Corpus Lesk predicts more accurately than any other algorithm and the above table also tells us that maximum accuracy is attained by Corpus Lesk algorithm therefore it is better than others and we have used it to predict the senses for the test data.

(d) Include observations that you achieve during the experiment. One essential discussion is to analyze informative features based on the real examples. In addition, Discuss the difference between the supervised and the dictionary-based WSD systems. Which system is more appropriate for which cases based on the real examples chosen from the data.
Ans. The word '**president.n**' has three different meanings in our dictionary. In the context '*The best thing individual investors can do is `` just sit tight , '' says Marshall B. Front , executive vice %% president %% and head of investment counseling at Stein Roe & Farnham Inc. , a Chicago-based investment counseling firm that manages about $ 18 billion.*', it means the sense 1 which is 'chair'. Apart from the simple lesk algorithm, all other algorithms correctly identify its sense. This may be since simple lesk just searches for overlaps between the target word sense and the given context and not the senses of the context.
Similarly, in the context *'People close to the utility industry said Mr. Dingell 's proposal appears to guarantee only an estimated seven-million-ton cut in annual sulfur-dioxide emissions that lead to acid rain , though additional cuts could be ordered later . Mr. Bush 's legislative package promises to cut emissions by 10 million tons -- basically in half -- by the year 2000 . Although final details weren't*

available , sources said the Dingell plan would abandon the %% president %% 's proposal for a cap on utilities ' sulfur-dioxide emissions . That proposal had been hailed by environmentalists but despised by utilities because they feared it would limit their growth . It also would junk an innovative market-based system for trading emissions credits among polluters .' , because we are giving higher score to consecutive overlapping words, we get the correct prediction for the sense of the target word only in case of Original Lesk algorithm. All other algorithms have incorrect predictions for this.

Difference between supervised and dictionary based WSD systems(Lesk):

The Lesk algorithm assumes that words with similar surroundings will usually share a common topic. In other words, the contextual overlap between dictionary senses, is used as a measure to pick up the most likely sense for a given word. The Lesk algorithm is also classified as knowledge-based is because it acquires knowledge only from a set of dictionary entries where a separate entry is needed for each sense of every word, concentrating on the immediate context of the target word.

The idea behind supervised methods for WSD is to use the machine learning algorithm for classification. These methods automatically learn to make correct predictions if they are provided the possibility to have some observations in advance. Each of the new examples are categorized by using calculated probabilistic parameters in case of probabilistic methods. The probabilistic parameters include the probability distributions of the categories and the contexts that are being described by the features in the feature vectors. Naïve Bayes being one of the simplest representatives of probabilistic methods, supposes that the features are conditionally independent given the class label. The main idea is that an example is created by selecting the most probable sense for the instance and as well as for each of its features independently considering their individual distributions.

Supervised WSD is only as good as its training data. It is good for use in case of information retrieval applications where it is enough to know that a word is used in the same sense in the query and a retrieved document. When the precise meaning is required as in target word selection cases, we could use the dictionary based WSD algorithms.

(e) Report your additional findings if you decide to implement some of the extensions.

Ans. As an extension, we have implemented the Corpus Lesk algorithm for the ontological WSD. It is like the Simple Lesk algorithm but with an augmented new dictionary which has sentences from the training data added as examples to the senses in the existing dictionary.

Model Building: For each sense of the target word, we calculated the count of overlapping words between the target word's gloss + examples from the new dictionary, and lemmatized words from the given test sentence for each sense of the target word. The count is taken as the score for each sense id. The sense ID with maximum overlap score is taken as the predicted sense ID for the target word.

Accuracy achieved: Validation Data: **83.39** ; Train Data: **98.70**