Language, Speech and Dialogue Processing
Lab 3

Ellen Bogaards, Femke Witteveen, Sabijn Perdijk
Roos Vervelde

March 2020

Pitch and Focus Marking

1 Introduction

We have learned in the past weeks that pitch difference determines a lot about the focus of a sentence. Therefor, it could be useful for a program to determine the focus condition of a sentence to better understand the discourse. This results in the following research question for this report: "How accurately can we predict the focus condition using a supervised classification algorithm?" To find an answer to this research question, we looked at a dataset with pitch files extracted from recorded Dutch sentences. The sentences all had similar phrases at the end, that all consisted of a preposition *in* or *uit*, either followed by a noun or as part of a verb (e.g. "*inpakken*", "*in pakken doen*"). There can be three distinct kinds of focus on these type of phrases: high focus (HI focus), neutral focus (NEU focus), or low focus (LO focus). The focus is considered as HI if it is contrastive on the preposition, NEU if it is not contrastive but the focus is on the whole last phrase, and LO if the last phrase does not have focus because the focus was on another part of the sentence. We will test if we can train a classifier to correctly predict if the phrases in the audios have HI focus, NEU focus or LO focus.

2 Method

2.1 The SVM algorithm

To test our research question, we used a support vector machine algorithm (SVM). SVM is a linear model that can be used for classification problems. The algorithm tries to find an optimal line or hyperplane in multidimensional space that divides the given data (input) into two or more different classes. To do this, you give the algorithm training data existing of the information for each point and the labels of the classes for these points. Based on this training data, a hyperplane is created to separate the different classes. The algorithm can then look at new data and make predictions about the classes these data points fit in. Comparing the predictions with the actual classes, the accuracy of the algorithm for the classification can be measured.

In the case of our research problem, we have to keep in mind that there are three different classes, which are equally distributed. Therefore, an accuracy of 33.3 percent or lower would mean that the algorithm can not help in making correct predictions, because it equals (or is worse than) chance. If we want to conclude that an SVM algorithm can be helpful in predicting the focus condition in phrases, we will have to find an accuracy that is (preferably a lot) higher than 33.3 percent.

The advantages of SVM are the accuracy and the relatively low use of memory (because for the decisions, only subsets of the training points are used). Disadvantages can be that SVM is less suitable for very large data sets (because the training time is longer) and for data sets with overlapping classes, but since this is not the case for this current data set, SVM is a suitable classifier to test our research question with.

We used K-fold cross validation to estimate the accuracy of the predictions the SVM can make about our data. K is the number of groups that the data sample is split into. This means it also allows to measure the accuracy K times, because every group will appear once as the test data using all the other groups as the training data. This way, cross validation ensures that every sample in the dataset is used in the training set and in the test set. This takes out randomness and generally results in a less biased estimate of the accuracy than for example a simple train/test split.

2.2 Extracting the data

To analyse the pitch files from the folder PITCH2, we adjusted the given Praat program. Our used program can be found on the last page under the name "Data preprocessing in Praat". Running this program results in a text file with on every line for every file all the pitch vectors per frame.

We loaded this file into a pandas dataframe, and split the Filename at the last underscore to create a column with the focus markings NEU, LO and HI. This was necessary because these are the labels we want our algorithm to classify between. The file we extracted from Praat contained a lot of undefined data points. These were excluded from the data set. To be able to work in a pandas dataframe, we had to have the same amount of pitch vecotrs per file. After excluding the undefined datapoints, the file with the smallest amount of leftover pitch vectors had 43 left. We changed the other datafiles to contain the same amount by excluding random datapoints till 43 were left and created a pandas dataframe containing all these datapoints.

We realised that we lost a lot of data when we applied the method of taking a maximum of 43 datapoints per file. Therefor we decided to delete some of the smallest files. This sadly also results in losing data. We had to find an optimum

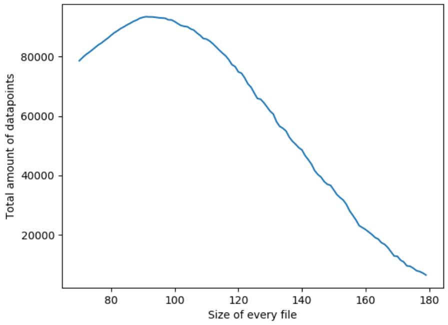

Figure 1: Optimal Dimensions

between the amount of columns and rows of our set to create the biggest amount of datapoints to work with. We created a function that returns the amount of datapoints when we changed the amount of files and the length of every file (see Figure 1). This showed an optimum that contained 93457 datapoints (column row). The optimal amount of datapoints per file is 90. All files with fewer datapoints were excluded.

The pandas dataframe was then classified with an SVM learning algorithm, using the sklearn package from python. After testing this, we also tested how well it could classify the data if only two labels were taken into account. All the results are shown below.

3 Results

When selecting the optimal dataframe as mentioned in the method section, running this with the SVM learning algorithm showed an accuracy of 55 percent. It is also a lot higher than the 33.3 percent chance of a random guess.

We were curious which label was most difficult to classify by the algorithm. Therefor we ran the SVM algorithm with two instead of three labels. The results are shown in table 1. We see that the accuracy is also higher than chance, which in this case would be 50 percent. It is interesting to see that the SVM is a lot better at differentiating between HI & LO and LO & NEU, than NEU & HI.

4 Conclusion

We conclude that an SVM can be trained to make predictions about focus conditions, with an accuracy higher than would be expected with a random guess. However, the accuracy would need to be a lot higher if we want to be reasonably sure about the correctness of the predictions.

When testing the labels in pairs of two, it is interesting to see that HI & LO

labels	accuracy
HI & LO	73%
LO & NEU	69%
NEU & HI	59%

Table 1: Accuracy of SVM with two labels

and LO & NEU are easier to classify than NEU & HI. As we have learned in the introduction, NEU means that the focus is implied on the whole sentence. With HI and LO the focus lies on a particular part of the sentence, which explains why it is relatively easy for the classifier to differentiate between these two. For LO & NEU an explanation could be that with a LO focus, there is no focus on the last phrase of the sentence, while with HI there usually is. That could mean it resembles NEU more, because that also does not have a focus on the last phrase, because that has focus on the entire sentence.

To create a better SVM learning algorithm, there are a lot of improvements to consider. At first, it is unclear whether just having pitch files of the recordings really is enough to train the SVM. We did not use other information to test the algorithm, like for instance the intensity of the sound. Training the SVM on these parameters too could give a better result. It is known that different focus conditions can be created by changes in pitch and intensity. For our research we only looked at the first, but taking both in consideration might improve the accuracy of the predictions. At second, using all the pitch vectors of all the frames may create a lot of noise, because a large amount of the vectors might be irrelevant for the focus condition and might therefor not help in the classification process. This could be improved by applying different dimension reduction techniques. At third, it would be better if we did not have to apply such a drastic dimension reduction, because of the present NaN values. Having more samples would also improve the learning ability of a learning algorithm.

So we can conclude that investigating focus markings with an SVM algorithm is of a promising interest. Still, considering other learning algorithms as well might be interesting for future research.

Data preprocessing in Praat

```
form Read all files of the given type from the given directory
   sentence Source_directory ./PITCH2
   sentence File_name_or_initial_substring ADU
   sentence File_extension .Pitch
endform

Create Strings as file list... list 'source_directory$'/'
        file_name_or_initial_substring$'*'file_extension$'
head_words = selected("Strings")
file_count = Get number of strings

for current_file from 1 to file_count
   select Strings list
   filename$ = Get string... current_file
   Read from file... 'source_directory$'/'filename$'
   name$ = filename$ - file_extension$
   select Pitch 'name$'
   pitches# = List values in all frames: "Hertz"
   resultline$ = "'name$', pitches#'newline$'"
   fileappend "resultfile8$" 'resultline$'
   #for i from 1 to size (pitches#)
           #resultline = pitches#[i]
           #fileappend "resultfile7$" 'resultline'
           #resultline$ = " "
           #fileappend "resultfile7$" 'resultline$'
   #endfor

   # Save result to csv file

   #resultline$ = "'newline$'"
   #fileappend "resultfile7$" 'resultline$'

   # Remove temp objects from object's list
   select Pitch 'name$'
   Remove
   select Strings list
endfor
```