Prominence detection in spoken Dutch using prosodic features and machine learning

MARK KOK, University of Twente, The Netherlands

Research into various aspects of speech is increasingly making use of more advanced machine learning. These advancements are beneficial to the development of better, more realistic conversational agents. If the use of prosodic features can be extended further, agents could use these to understand implicit contextual clues in human speech. We use several types of machine learning models to test and compare how well word prominence can be classified. The machine learning models tested are a support vector machine, a random forest classifier, and a multi-layer perceptron. The openSMILE Python library is used to extract prosodic features as described by the GeMAPS feature set. We also use two data preprocessing methods, standardization and feature selection, and tested how these affect the results. The support vector machine with both data preprocessing methods performed best with an F_1-score of 0.698. Conversely, the multi-layer perceptron performed the worst with F_1-scores ranging from 0.542 on unprocessed data to 0.692 on standardized data.

Additional Key Words and Phrases: Prominence detection, machine learning, support vector machine, random forest, multi-layer perceptron, openSMILE, GeMAPS, Corpus Gesproken Nederlands

1 INTRODUCTION

The fields of conversational agents and speech recognition have made significant progress in recent years. One of many topics in the field of speech recognition is the use of prosodic features such as frequency and intensity to determine prosodic/lexical stress[1], or prominence[2], in a sentence. This is useful for conversational agents to disambiguate context in sentences where prominence on different words conveys different implications.

Take for example a simple sentence, such as "the car is red". At face value, the information conveyed by this sentence is straightforward. However, prominence on certain words can convey implicit information. For instance, if prominence is placed on the word "car", it could imply that there are multiple other objects that are not red. When only looking at the words themselves, this implicit information is missed.

In languages where the subject-object order in a sentence is not fixed, such as Dutch, situations can occur where the grammatical case of a word is ambiguous. Monique Lamers and Peter de Swart give an example of such an ambiguous sentence used by the Dutch postal services, namely *"Een echte vriend stuurt u een echte kaart"*[3], where the words *"vriend"* and *"u"* can both be either the subject or the object of the sentence. Prominence plays a considerable role in disambiguating such sentences.

Prominence detection in our everyday conversations is something that comes intuitively to most people. However, there are many variables in play when it comes to determining word prominence and there is a decent amount of research dedicated to using machine learning to detect it, though research using the Dutch language seems to be uncommon. Better prominence detection will allow for the development of better conversational agents that exhibit more natural responses.

This research will test various classifiers on their ability to detect prominence in spoken Dutch using a set of 62 audio features. The classifiers that will be tested are a support vector machine, a random (decision) forest, and a multi-layer perceptron. Furthermore, we will compare the effects of different preprocessing methods to the input of these classifiers, such as standardization to express features as their deviation from the mean, and selecting the most significant audio features to use.

The illustrated situation leads to the following research questions that this research will aim to answer:

(1) How accurately can different types of classifiers identify words as having prominence within a sentence?
(2) How can we improve the performance of these classifiers through preprocessing of the input data?

2 RELATED WORK

2.1 Prominence

Prominence is defined by Monique Lamers and Peter de Swart as "an element's ranking on a hierarchy of semantic features", and that "prominence is concerned with semantic/pragmatic features of arguments such as their animacy, definiteness, person and semantic role".[3] They also list prosody as one of several factors that determine prominence.

Julia Hirschberg states in her research that prominent words (which she refers to as accented words) can be identified by maxima and minima in the fundamental frequency (f0), and that prominent words are also often perceived as louder and more stretched out. Conversely, words without prominence may have the vowel in their stressed syllable reduced, and may have less well-defined word boundaries.[4]

2.2 Prominence detection

A 2016 paper by Milos Cernak et al.[5] utilized a deep belief network in emphasis detection, with the assumption that different prosodic events have unique sound patterns which allow the neural network to identify them. They used both English and French speech data. Their method of using sound pattern matching resulted in a 96.8% accuracy in English and a 90.3% accuracy in French, compared to a 71.0% accuracy in both languages using an empirical model.

A 2009 research by Ozlem Kalinli and Shrikanth Narayanan[6] performs prominence detection using a basic neural network. This neural network accepts 5 words or syllables as input, using not only the target word/syllable but also the adjacent ones to determine

38th Twente Student Conference on IT, February 3rd, 2023, Enschede, the Netherlands

prominence. Additionally, they used lexical and syntactic information to improve classification. For prosodic features, they list intensity, frequency contrast, temporal contrast, orientations, and pitch as features that were used. Tested on the Boston University Radio News Corpus, they achieved a 85.71% accuracy at the word level, and a 88.33% accuracy at the syllable level. When only looking at the prosodic information the accuracy was 83.11% and 85.45% for word-level and syllable-level respectively.

A 2022 paper by A. Reddy and V. Vijayarajan[7] similarly makes use of prosodic features, except in emotion detection rather than prominence detection. They make use of a convolution neural network known as AlexNet[8] for classification, creating frequency-amplitude spectrograms as images to train the convolutional neural network with. This resulted in an overall accuracy of 92.9%. While the thing being classified for is different, this still shows the potential of using an image-based neural network in classifying audio.

In terms of which prosodic features to use for this purpose, research has been done on exhaustive and compact feature sets. For example, the ComParE feature set is extensive with 6373 distinct prosodic features[9]. On the other hand, the GeMAPS feature set has 62 carefully curated prosodic features[10].

3 METHODOLOGY

3.1 Data sets

For data sets, we use the Corpus Gesproken Nederlands ("Spoken Dutch Corpus") which contains audio recordings of spoken Dutch and Flemish. Specifically the C component, which consists of recorded phone conversations[1]. In this component, the different speakers are mostly on separate audio channels. Although there is some bleeding over into the opposite audio channel when both speakers speak at the same time, this means they can be separated so less interference will occur in the data. However, the acoustic data will not be entirely pure.

Of the different annotation sets, we use the *pro2* set for prosodic annotations. This is because, upon cursory inspection of the data, the other available prosodic set (*pro1*) had instances where syllable prominence markings were used too loosely for our goal of finding prominence within a sentence. We also use the *wrd* set to obtain audio timestamps for each individual word.

The Praat[11] application is used to view and verify the data and identify which speaker occupies which audio channel.

To ensure that the classifiers are not trained on the details of the voice of any one speaker to improve the test score, we divide the data set by files. The Spoken Dutch Corpus provides 10 files of prosodic annotations, each corresponding to a different conversation with different speakers. The first five files comprise the training data, and the last five files comprise the test data.

Additionally, we apply data augmentation on the training data by reversing the audio, and stretching the audio by adding an interpolated audio sample between each pair of samples. To ensure that the training data is balanced between words with and without prominence, we remove some training samples without prominence. This results in a training set with 5817 words for each class. The test data comprises of 6974 words without prominence and 1549 words with prominence.

3.2 Extracting prosodic features

The prosodic features that we extract are according to the GeMAPS feature set.[10] This set contains 62 distinct features, including data on the fundamental frequency, frequency and amplitude data up to the 3rd harmonic, loudness data, jitter data, and data on the length of (un)voiced segments. We decided to use this feature set due to its compactness while still providing a wide range of prosodic features.

The Python bindings for openSMILE[12] are used to extract these features per word using the built-in GeMAPSv01b set at the functionals level. The values for these features are then used as training data for the classifiers.

3.3 Classifiers

For the classifiers, we have decided to use a support vector machine, a random forest classifier, and a multi-layer perceptron. For each classifier, we use a grid search algorithm with cross validation in order to find the best parameters for the training data, ranked by F_1-score. For the implementation of the classifiers we use the scikit-learn Python library.

In performing the grid search, we found that for the support vector machine, results for the radial basis function kernel and the polynomial kernel were very close. As such, we have decided to run both kernels on the test data to see if either kernel performs better on unfamiliar data.

The following classifiers and parameters are used:

(1) A support vector machine with a 2nd-degree polynomial kernel, a constant coefficient of 0.3, and a C-value of 1.0
(2) A support vector machine with a radial basis function (RBF) kernel, with a C-value of 1.0
(3) A random forest with entropy function and 200 estimators
(4) A multi-layer perceptron neural network with 1 hidden layer, consisting of 3 neurons

3.4 Data preprocessing

To test the effect of processing the data, we apply two preprocessing methods on the data and use each combination on the classifiers. The preprocessing methods used are standardization and feature selection. For the test where both methods are applied, the standardization is applied first, then the feature selection.

Standardization normalizes data around the mean and divides by the standard deviation, causing the data to assume a normal distribution. This is done through the function $f(x) = (x - \mu)/\sigma$, where μ is the mean and σ is the standard deviation.

Feature selection makes use of principal component analysis to select the most significant features from the data. The number of features to keep is determined using Minka's algorithm.[13]

3.5 Experiment setup

For the experiment, since the Spoken Dutch Corpus prosody annotations only annotate prominence on certain syllables, a word is

[1] https://lands.let.ru.nl/cgn/doc_English/topics/version_1.0/annot/prosody/info.htm#data

counted as having prominence if it contains a syllable with prominence.

First we extract the audio and annotations from the data set, match them together, and split them into each individual word. Each word is labeled with whether or not it has prominence, the number of the file it originated from, and the corresponding audio fragment. This data is then fed to a script that extracts the audio features using openSMILE and splits the data into a training and test set based on the file numbers. The training and test sets are then saved to a CSV file.

Each classifier then reads the data from the aforementioned CSV files. This is where the preprocessing methods are optionally applied using the Pipeline class in scikit-learn. The classifier is then trained and used to predict the classes for the test set. The predicted classes are compared to the actual classes and the results are then made into a contingency table. From the contingency tables, the precision and recall are calculated. These are then used to calculate the F_1-score. As the proportion of words with and without prominence in our test set is somewhat unbalanced, we have decided to refrain from using accuracy as a measure of performance.

Additionally, each classifier is run with both preprocessed and unprocessed data sequentially. The preprocessed results are then placed against the unprocessed results in a contingency table, on which we apply A. Edwards' corrected version of McNemar's test for statistical significance.[14] The formula for this corrected McNemar's test is as follows, where e_a represents the amount of words classified correctly in the unprocessed data but incorrectly in the preprocessed data, and e_b represents the amount classified incorrectly in the unprocessed data but correctly in the preprocessed data:

$$\chi^2 = \frac{(|e_a - e_b| - 1)^2}{e_a + e_b} \tag{1}$$

This results in a χ^2-distributed value with 1 degree of freedom, which is then looked up in the p-score table for this distribution[2]. If the p-score for the preprocessing method is below 0.05, it is conventionally considered statistically significant.

4 RESULTS

Figures 1 through 4 show the results of the classifier tests we performed. Each classifier is scored on precision and recall, as well as the F_1 score and the χ^2 value from the McNemar test, comparing the data preprocessing methods to the results with unprocessed data.

Of note is that the feature selection using Minka's algorithm only reduced the amount of features from 62 to 61.

It should be noted that higher χ^2-values lead to a lower p-score. Referencing a p-score table for the χ^2 distribution with 1 degree of freedom shows that a χ^2-value of 12.116 corresponds to a p-score of 0.0005. As such, any χ^2-values higher than this will automatically be assumed to be statistically significant due to their low p-score.

Classifier	Precision	Recall	F_1
SVM (poly)	60.74%	67.12%	0.638
SVM (RBF)	60.61%	67.81%	0.640
Random forest	66.07%	73.38%	0.695
MLP	59.10%	50.07%	0.542

Fig. 1. Results on unmodified data

Classifier	Precision	Recall	F_1	χ^2
SVM (poly)	65.74%	73.72%	0.695	163.3
SVM (RBF)	65.91%	73.88%	0.697	552.0
Random forest	66.15%	73.70%	0.697	2.33
MLP	65.80%	72.94%	0.692	2589.1

Fig. 2. Results on standardized data

Classifier	Precision	Recall	F_1	χ^2
SVM (poly)	60.86%	68.00%	0.642	90.7
SVM (RBF)	60.94%	68.19%	0.644	109.1
Random forest	65.69%	73.15%	0.692	4.52
MLP	65.14%	73.26%	0.690	2212.2

Fig. 3. Results on feature-selected data

Classifier	Precision	Recall	F_1	χ^2
SVM (poly)	65.74%	73.72%	0.695	163.3
SVM (RBF)	65.81%	74.35%	0.698	552.0
Random forest	65.92%	73.74%	0.696	5.33
MLP	65.19%	73.20%	0.690	2599.3

Fig. 4. Results on standardized and feature-selected data

Judging by the χ^2-values of most classifiers, the data preprocessing methods appear to have a significant effect, though it should be noted that feature selection appears to adversely affect the results. It also seems to bring down the results when both methods are applied, evidenced by the lower F_1-scores for most classifiers compared to when only standardization is applied.

The χ^2-values of the random forest classifier are below the threshold of 12.116, so for these we have also determined their p-scores as seen in Figure 5. These p-scores suggest that standardization does not have a statistically significant effect on the random forest classifier. Feature selection has a slight significance if the common threshold of $p < 0.05$ is used, but the classifier results are affected adversely by this method as seen by the 0.003 decrease in its F_1-score compared to the random forest classifier using unprocessed data.

Finally, the F_1-scores of each classifier are collated into Figure 6, along with the average F_1 score across all preprocessing configurations, for easier comparison. The highest F_1-score has also been highlighted for convenience.

[2] There are many instances of the p-score table for the χ^2 distribution online, for example here: https://www.statology.org/chi-square-distribution-table/

Preprocessing	χ^2	p
Standardization	2.33	0.127
Feature selection	4.52	0.033
Both	5.33	0.021

Fig. 5. p-scores for the McNemar 1 degree of freedom χ^2-values of the random forest classifier

5 DISCUSSION

5.1 Limitations

The available time for this research was 9 weeks, including a research proposal phase of 2 weeks. This means the scope of the research had to be carefully considered in order to make deadlines in time. This resulted in having to use less complex classifiers, as well as having to use less different classifiers. This does offer opportunities for future research with more advanced classifiers and more carefully selected parameters.

The classifiers were trained and tested on a system with an 8-core Intel i7 processor and 32 GB of physical memory. Training the classifiers took between (approximately) 10 seconds and 30 minutes depending on the classifier. This caused performing grid searches to take an extensive amount of time, and limited the number of searches that could be performed despite being run on all processor cores.

Better hardware performance and/or more allocated time would allow for more grid searches and would likely have resulted in some of the resulting F_1-scores being higher. This is particularly true for the multi-layer perceptron classifier, whose best found parameters of 1 hidden layer with 3 neurons are unlikely to be ideal.

Additionally, a deep learning framework that utilises CUDA[3] could be used to improve training times.

Another limitation was the availability of prosody-annotated Dutch data sets. The Dutch Spoken Corpus used in this research was the only data set we were able to find that had prosodic annotations. However, this data set has some limitations that may have affected our results. For one, the conversations in the C component had some instances of speakers speaking at the same time. While speakers were mostly separated by audio channel in this component, they could still be softly heard on the other audio channel. Furthermore, we have noticed anecdotally that the timings in the word annotations, which were used to determine the start- and endpoints of the audio data for each word, may not have been accurate in all instances. This was discovered when pairing up the prosodic annotations and word annotations, which failed on one word whose timing did not match any prosodic annotations. These factors may have lead to a reduction of the quality of the available training and test data.

5.2 Conclusion

Based on the results, we have concluded that the support vector machine with the radial basis function kernel seems to have the best overall performance as indicated by its F_1 score of 0.698 when the input data is standardized and feature-selected.

The random forest classifier works best when neither of the two data preprocessing methods we tested are applied, and is also the least improved by these methods. The apparent advantage of having a relatively strong F_1-score of 0.695 without either of the data preprocessing methods is offset by the longer classification time from using multiple decision trees. While classification time was not a consideration in this research, usage of this classifier in conversational agents may result in unacceptably long response times.

For the multi-layer perceptron, we have to conclude that such a basic type of neural network may not be suited for this task, or at the very least requires a significant amount of parameter tuning, as it is outperformed by the other classifiers in spite of existing research using more advanced neural networks to far greater effect, such as the 96.8% accuracy achieved by Cernak et al.[5]

The effects of the data preprocessing methods depend on the type of classifier used, but in general, standardization has a greater and more positive effect than feature selection. The adverse effect of feature selection, along with only one feature being removed, suggests that most of the features described by the GeMAPS feature set are significant in determining word prominence.

All in all, there is a notable difference between our results and those of existing research using machine learning in prominence detection. As stated before, Cernak et al. saw a 96.8% accuracy using a deep belief network. At face value this may imply that using a more advanced neural network results in better results. However, Kalini and Narayanan managed to get a 85.71% accuracy using a basic neural network, so other differences may need to be taken into consideration.

The main features of this research that set it apart from the others we looked at were the use of the Spoken Dutch Corpus data set, and the use of the GeMAPS prosodic feature set. As discussed in section 5.1, we believe that the data set may have contributed to the somewhat moderate results of this research. Perhaps the prosodic features in the GeMAPS feature set were too diverse, or not diverse enough, and also lead to worse results. These are factors that could be looked into by future research.

However, the previous research findings (using neural networks) are mostly compared to our research on grounds of our research containing the basic multi-layer perceptron neural network. Due to time and hardware limitations, the parameters we selected for the multi-layer perceptron may have lead to it severely under-performing. This means that there is a possibility that the data set and feature set may not have been the issue. At this point, it is difficult to say for certain where the problem lies.

5.3 Future work

A main point of focus for future research could be to expand the number of classifiers and data preprocessing methods used. The classifiers used here were basic, and more sophisticated neural networks may yield better results. In addition, more time could be spent on optimizing classifier parameters. There are also many ways in

[3] https://developer.nvidia.com/cuda-zone

Preprocessing	SVM (poly)	SVM (RBF)	Random forest	MLP	Average
None	0.638	0.640	0.695	0.542	0.629
Standardization	0.695	0.697	0.697	0.692	0.695
Feature selection	0.642	0.644	0.692	0.690	0.667
Both	0.695	0.698	0.696	0.690	0.695
Average	0.668	0.670	0.695	0.654	-

Fig. 6. F_1-scores for each classifier and preprocessing configuration, with the highest F_1-score highlighted in yellow

which data can be preprocessed, and perhaps a method or combination of methods that was not considered in this research could have beneficial effects.

This research has focused mainly on data from Dutch speakers. Future research could focus on different languages and comparing the results to see if Dutch is easier or harder to find word prominence for. Additionally, more research could be performed to look into the effect of the Spoken Dutch Corpus and the GeMAPS feature set which were used for this research.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Khiet P. Truong of the HMI department at the University of Twente, for her expertise and guidance during this research and for reviewing and providing feedback on this paper, as well as the other students of the Intelligent Interaction track for their insight and peer feedback.

REFERENCES

[1] L. Ferrer, H. Bratt, C. Richey, H. Franco, V. Abrash, and K. Precoda. 2015. Classification of lexical stress using spectral and prosodic features for computer-assisted language learning systems. *Speech Communication*, 69, 31–45. doi: 10.1016/j.specom.2015.02.002.

[2] B.M. Streefkerk. 2002. *Prominence: acoustic and lexical/syntactic correlates.* Ph.D. Dissertation. University of Amsterdam.

[3] Monique Lamers and Peter de Swart, (Eds.) 2012. *The interaction of case, word order and prominence: language production and comprehension in a cross-linguistic perspective. Case, Word Order and Prominence: Interacting Cues in Language Production and Comprehension.* Springer Netherlands, Dordrecht, 1–15. isbn: 978-94-007-1463-2. doi: 10.1007/978-94-007-1463-2_1.

[4] J. Hirschberg. 1993. Pitch accent in context predicting intonational prominence from text. *Artificial Intelligence*, 63, 1-2, 305–340. doi: 10.1016/0004-3702(93)90020-C.

[5] M. Cernak, A. Asaei, P.-E. Honnet, P.N. Garner, and H. Bourlard. 2016. Sound pattern matching for automatic prosodic event detection. In vol. 08-12-September-2016, 170–174. doi: 10.21437/Interspeech.2016-875.

[6] O. Kalinli and S.S. Narayanan. 2009. Prominence detection using auditory attention cues and task-dependent high level information. *Ieee Transactions on Audio, Speech, and Language Processing*, 17, 5. doi: 10.1109/TASL.2009.2014795.

[7] A.P. Reddy and V. Vijayarajan. 2022. Fusion based aer system using deep learning approach for amplitude and frequency analysis. *ACM Transactions on Asian and Low-Resource Language Information Processing*, 21, 3. doi: 10.1145/3488369.

[8] A. Krizhevsky, I. Sutskever, and G.E. Hinton. 2017. Imagenet classification with deep convolutional neural networks. *Communications of the ACM*, 60, 6, 84–90. doi: 10.1145/3065386.

[9] B. Schuller et al. 2016. The INTERSPEECH 2016 Computational Paralinguistics Challenge: Deception, Sincerity & Native Language. In *Proc. Interspeech 2016*, 2001–2005. doi: 10.21437/Interspeech.2016-129.

[10] F. Eyben et al. 2016. The geneva minimalistic acoustic parameter set (gemaps) for voice research and affective computing. *IEEE Transactions on Affective Computing*, 7, 2, 190–202. doi: 10.1109/TAFFC.2015.2457417.

[11] P. Boersma and D. Weenink. 1992. Praat: doing phonetics by computer [computer program]. Version 6.3.01, retrieved 23 November 2022 from https://www.praat.org. (1992).

[12] F. Eyben, M. Wöllmer, and B. Schuller. 2010. Opensmile - the munich versatile and fast open-source audio feature extractor. In 1459–1462. doi: 10.1145/1873951.1874246.

[13] T. Minka. 2000. Automatic choice of dimensionality for pca. In *Advances in Neural Information Processing Systems.* T. Leen, T. Dietterich, and V. Tresp, (Eds.) Vol. 13. MIT Press. https://proceedings.neurips.cc/paper/2000/file/7503cfacd12053d309b6bed5c89de212-Paper.pdf.

[14] A.L. Edwards. 1948. Note on the "correction for continuity" in testing the significance of the difference between correlated proportions. *Psychometrika*, 13, 185–187. doi: 10.1007/BF02289261.