Danmarks Tekniske Universitet

Analysis of Queen Margrethe II's and the Prime Ministers' New Year's speeches using a Convolutional Neural Network and sentiment analysis

(02461) Introduction to Intelligent Systems

Authors:
Bertram Kjærholm Foldberg Bendsen (s244829)
Christian Alexander Halberg (s245021)
Julie Thanh Thanh Nguyen (s245261)

Abstract:

The New Year's speeches given by the monarch and the Prime Minister of Denmark hold significant cultural importance, making the phrasing and language of these speeches particularly important in fulfilling their societal purposes. This study investigates whether it is possible to distinguish between the two speakers solely based on the language used in their speeches. We trained a convolutional neural network to classify whether sentences came from Queen Margrethe II's or the Prime Ministers' speeches, achieving a test accuracy of 71.58%. Additionally, we conducted a sentiment analysis for comparative data, revealing that the Queen's speeches more often had higher sentiment and arousal scores compared to the Prime Ministers' speeches, reflecting their roles in the Danish society. These results suggest that it is possible to discern between the two speakers with a high degree of accuracy only from the language. However, this also highlights potential risks and ethical concerns associated with training language models to emulate people's speech, such as impersonation and misinformation.

Responsible:

Abstract :	Bertram & Julie
Introduction :	Christian & Julie
Theory :	Bertram & Christian
Method :	Christian & Julie
Results :	Bertram & Christian
Discussion :	Bertram & Julie

24. January 2025

Indhold

Introduction

The New Year marks a new beginning. We look back and reflect on the previous year and look forward to the upcoming year with hope and determination. In Denmark, the New Year's speeches given by the monarch and the Prime Minister hold significant cultural importance. They address local as well as international trends, events and developments and how these affect Danish society at large. The speeches also have two distinctly different purposes, as the speeches given by the monarch address national self-awareness, history and culture, whereas the Prime Minister's speeches take a political and often broader international view of Denmark. In that context, the phrasing and language of these speeches are particularly important in both gauging and influencing the overall Danish societal sentiment. The motivation for our study then stems from our interest in determining the origin of a sentence by analyzing the choice of words in that sentence.

The focus of this study is to investigate whether a convolutional neural network can be trained to classify if a sentence comes from Queen Margrethe II or the Prime Minister of Denmark's New Year's speeches by analyzing the language used. Whether or not the model is able to discern between them, we investigate if there is a notable difference in the sentiment of the speeches. This raises important questions about the ethical implications if language models could be leveraged and exploited to imitate people, such as misinformation and scamming, and the potential loss of anonymity of people based on their writings.

We have focused on writing most of the code ourselves, without using in-built functions from libraries such as scikit-learn, to facilitate our learning.

Theory

To be able to conduct this study, our group has used a variety of different theories to create both the sentiment analysis and our text classification model. Certain methods used in our project proved to be more straightforward than others, which is why the following theory is aimed to help clarify more complex aspects of our project.

Latent Semantic Analysis (LSA) and Singular Value Decomposition (SVD):

Latent Semantic Analysis (LSA) analyzes the latent relationship between terms in a corpus. [1] The documents in the corpus can be vectorized using a Bag-of-Words (BoW) representation, which measures the count of the terms from a vocabulary. Normally, BoW is an unordered collection of words, but by using n-grams, which are sequences of n adjacent words, information in the word order is also captured. [2] These BoW-vectors are collected in a document-term matrix. Instead of the counts, the TF-IDF scores of the terms can be used, since the documents typically have different lengths.

The TF-IDF score is used to quantify the importance of a specific term in a specific document relative across all documents [3]:

$$\text{TF-IDF} = \frac{n_{td}}{n_d} \cdot \log(\frac{N}{n_t})$$

where n_{td} is the number of occurrences of the term in the document, n_d is the number of terms in the documents, n_t is the number of documents with the term and N is the total number of documents.

By using the TF-IDF score instead of the count, terms that are more important in distinguishing between different documents are emphasized.

When working with a big vocabulary, the document-term matrix can end up having a high dimensionality. A technique called Singular Value Decomposition (SVD) is then used to break down the matrix into three further matrices. The formula for SVD is as following:

$$M = U\Sigma V^T$$

To calculate these matrices, the eigenvalues and the eigenvectors for the given document-term matrix M are found. The squared eigenvalues will form the diagonal values of the Σ matrix, and the eigenvectors will form the V^T matrix. To find the U matrix, the Σ and V^T matrices are simply inserted, and then U is isolated [4].

The U matrix contains the documents and their context, the Σ matrix contains the singular values which captures the significance of the contexts and the V^T matrix captures the terms across different contexts [1]. Notably, this is done on a document-term matrix instead of a term-document matrix, contrary to the approach described in the cited website. To reduce the dimensionality of this matrix, only the top k dimensions are retained, thus only capturing the most significant latent relationships. Another way of reducing the dimensionality of a matrix is the Principal Component Analysis (PCA), which uses an orthogonal transformation that converts a set of correlated variables to a set of uncorrelated variables [5]. Since this study focuses on the SVD method, the in depths of the PCA method will not be explored further.

At last, the product of every BoW-vector and the embedding matrix V^T is calculated to do the LSA sentence embedding, thus reducing the dimension of the BoW-vectors to $1 \times k$.

Convolutional Neural Network (CNN):

The BoW-vectors can be used as features for training a CNN text classifier to distinguish different documents. The reason for using a CNN instead of a more regular recurrent neural network, is due to the fact, that CNNs are better to find the contextual relationships between n-grams [6]. A CNN allows the classification model to more easily determine which patterns are better indicators for a certain class. The CNN finds these indicators by striding over a n-size window, which is continuously done over the entire input matrix in smaller and overlapping convolutions, thereby capturing small, but important dependencies between n-grams [7]

In the first convolutional block, all of the extracted features are passed through a ReLU activation function. Then, lesser significant indicators are removed from the network with a dropout-layer, which randomly deactivates neurons in order to prevent overfitting [8]. Lastly, a one-dimensional max-pooling layer is used to discard insignificant features, which is done by retaining only the maximum value from a given window, further reducing the dimensionality [7]. The feature map is then flattened and passed through fully connected linear layers which include ReLU activation. When dealing with multiple outputs and text classification, cross entropy is the obvious choice of loss function, as it measures how well the predicted probability distribution of choosing a correct label matches the actual distribution of the true class labels. [9]

Method

Our group began collecting all available data, i.e. both Queen Margrethe II's and the PM's New Year's Eve speeches from 1972 until 2023. Most of the speeches were already transcribed; however, we had to transcribe a few of the older PM speeches from audio files into text. For this, we used OpenAI's 'Whisper' model.

Preprocessing:

Before creating both our CNN text classification model and sentiment analysis, the different speeches were preprocessed. Firstly, punctuation and non-Unicode characters were removed and the speeches were converted to lowercase. This was done to make the sentences consistent. Then, all the speeches were split into individual sentences using NLTK's sentence-tokenizer and labeled (One Hot Encoding: 0 for the Queen and 1 for Prime Ministers).

For our CNN, some sentences from the PM were deleted to balance the dataset. Then, the sentences were shuffled whilst keeping their label and then split into training- and testing sets. 30% of the sentences were set aside for testing (1643 sentences) while the rest were used for training (4929 sentences).

For the sentiment analysis, no sentences were removed; instead all 'stopwords' were removed. These stopwords were found online in a Danish stopwords list [10]. The speeches were tokenized down to singular words, and the tokens were stemmed using a danish stemmer from the NLTK library, as described in [11].

Sentence vectorization and LSA embedding:

In our project, the corpus is the sentences in the speeches while the terms are n-grams. Firstly, we found all unique unigrams, bigrams and trigrams in the training sentences to form a n-gram vocabulary. N-grams that only occurred once were excluded to reduce runtime, resulting in 16986 n-grams. All sentences were vectorized using Bag-of-Words (BoW) and the TF-IDF score was calculated. It was important to process the test sentences using the IDF values from the training sentences to avoid data leakage. The resulting vectors were collected in a document-term matrix, which resulted in 2 matrices: 1 matrix for the training sentences and 1 matrix for the test sentences, where rows are sentences and columns are n-grams. SVD was applied to the training document-term matrix using an in-built function. For the purpose of this project, only the V^T matrix was utilized, because it contains the n-grams and their term-topic associations. The top 200, 300 and 500 dimensions from the found embedding V^T matrix was retained to LSA sentence embed all of our BoW-vectors.

Analysis using CNN:

This project involved building a CNN to classify whether a sentence came from the Queen's or the PMs' New Year's speeches. Training and test data loaders were created with the LSA-embedded sentence vectors and their class labels to make our binary text classification CNN model.

Since there is no established theory on the optimal number of convolutional or hidden layers in CNNs, nor specific guidance on the most suitable dimension for the embedding matrix, we automated a program to test various configurations. Specifically, we experimented with three embedding matrix dimensions: 200, 300, and 500. Optimally, we would have tested on even more matrices with different dimensions, however we chose these three matrices, as larger matrices would have been too computationally expensive and smaller matrices might lose critical information. For each matrix, we tested models with one to five hidden layers. Additionally, for each configuration of hidden layers, we experimented with one and two convolutional layers. Further increasing the number of hidden or convolutional layers was avoided, as this would risk overfitting as well as increase computational demand.

Through this systematic approach, we identified the optimal setup: a 500-dimensional embedding matrix, a single convolutional layer and three hidden layers with a decreasing number of neurons in each layer, resulting in 2 neurons in the output layer. We also added two drop-out layers to decrease overfitting and one pooling layer.

Cross entropy was used as the loss function, specifically the function "CrossEntropyLoss"which applies the softmax activation function. Softmax was applied to our CNN's 2 outputs, transforming the logits into probabilities that the sentence belongs to each of the classes (0 for Queen or 1 for PM). The sum of the probabilities add up to 1 [12]. The model's prediction was selected as the output with the highest probability. We also used 'Adam' as an optimizer, 'CosineAnnealing' as the learning rate scheduler, and balanced the loss function with even class weights. The reasoning for using CosineAnnealing and even class weights is described in appendix 2.

Training and testing was done simultaneously for a number of epochs, meaning that the model's learned weight after every epoch was tested on test data. This was done with a specified batch size, learning rate and weight decay rate. For each epoch, the training & test loss and accuracy rate were calculated. The number of epochs chosen was 33, as future epochs beyond this level would drastically overfit the model resulting in overall lower test accuracy. That is also the reason for the chosen learning rate and weight decay, as these values seemingly hit the sweet spot in relation to the number of epochs. This reasoning also applies to the batch size and dropout rates. In some cases, tuning the hyper-parameters in different ways resulted in overall lower train loss. However, this often resulted in poorer generalization and test accuracy due to excessive training. Graphs showing the training & test loss and accuracy per epoch were created. This made it easily visible whether the model overfit on the training data. By experimenting with the batch size, learning rate, weight decay rate and drop-out rates, the goal was to optimize the test accuracy.

Sentiment analysis:

To compare the results of our CNN analysis with other data, we conducted a sentiment analysis. This analysis aimed to determine whether differences in sentiment and arousal exist between the Queen's speeches and the PMs' speeches. For this purpose, we utilized the Danish version of the AFINN lexicon [13], which includes 3551 words scored based on their sentiment. These scores range from -5 to 5, where negative scores indicate negative sentiment and positive scores indicate positive sentiment. The lexicon was stemmed.

The sentiment and arousal scores were calculated for both Queen Margrethe and the PMs' speeches. We also counted positive- and negative words. The scores and counts of positive & negative words were normalized by dividing by the number of tokens in the given speech. Normalizing ensured fairer comparison between the speeches, since they have different lengths. Then, histograms for the sentiment score, arousal score and count of positive words & negative words for the Queen's and the PMs' speeches were created. The data was displayed in the same plots to make it easier to compare the speakers' speeches.

The code for the implementation of the CNN and the sentiment analysis can be found in our GitHub repository, which is linked in appendix 1.

Results

CNN:

After experimenting with tuning the parameters, we ended up with these that maximized the test accuracy:

Number of epochs	Learning rate	Weight decay	Batch size	Dropout rates
33	0.0006	0.0001	128	Both layers: 0.3

Table 1: Parameters used in our CNN

The overall test accuracy of our text classification model is 71.58 ± 2.18%, where the label specific test accuracies are:

Queen (0) accuracy: 70.07 ± 3.11% and PM (1) accuracy: 73.12 ± 3.05%.

The value of the overall test accuracy is quite promising, particularly because both of the label specific test accuracies are close to this value. These test accuracies also prove that the model does not just guess, but is in fact learning to recognize significant indicators which contain some deeper contextual meaning. As the percentages are above 50%, which was the split between the actual amount of Queen and PM sentences, the model possesses the capabilities to discern between the two speakers.

Graphs with training & test loss and accuracy per epoch are shown in appendix 2.

Sentiment analysis:

Results from the sentiment analysis:

	Queen	Prime Minister
Mean of SS	0.18	0.11
Variance of SS	0.004	0.006
Mean of AS	0.50	0.45
Variance of AS	0.005	0.002

Table 2: Mean and variance of the sentiment score (SS) and arousal score (AS)

The histograms in figure 1 and the data in table 2 show that the Queen's speeches on average have a higher arousal score compared to the PMs' speeches. The Queen also uses more positive words, which explains why she generally has a higher sentiment score.

The 95%-confidence intervals for the mean of the sentiment scores (SS) and the arousal scores (AS):

	SS	AS
Queen	[0.16 : 0.20]	[0.48 : 0.52]
Prime Minister	[0.09 : 0.13]	[0.43 : 0.46]

Table 3: 95%-confidence intervals for the means

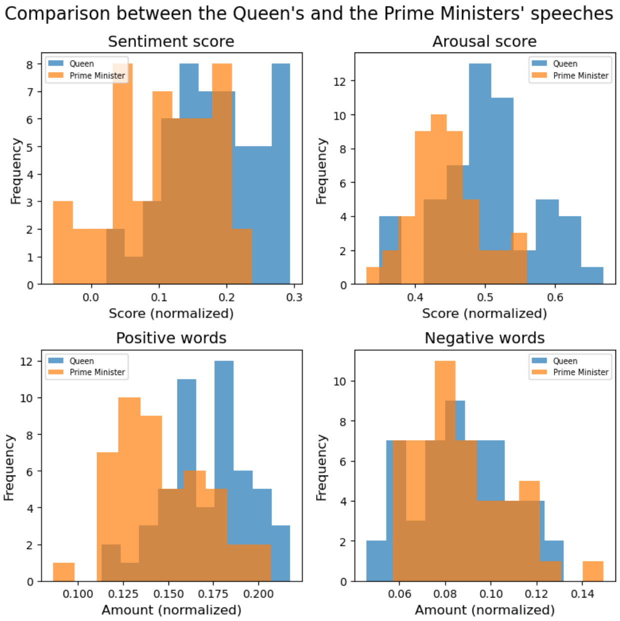

Fig. 1: Sentiment & arousal score and positive & negative words in the Queen's and the PM's speeches.

Discussion

What might the results imply?

The results section demonstrates that the calculated 95% confidence intervals for both the mean of the sentiment and arousal scores do not overlap, indicating that the difference in the sentiment of the language used in the speeches is statistically significant. This suggests a distinct divergence in emotional tones, reflecting their roles in the danish society. The PM tends to use a more neutral, formal tone, whilst describing political events, whereas the Queen would use a tone that has more dynamic sentiment and emotive arousal, for example to lift the spirits of the danish people after a difficult year.

The sentiment scores were calculated using the AFINN lexicon [13]. While it provides a useful framework for assigning sentiment values to words, one could argue that it's scoring system is subjective. The interpretation of words and their sentiment can vary between individuals and there might be a bias when using this lexicon, since all words have been scored by a single person. Another limitation with using the AFINN lexicon is, that it only analyses the sentiment of the particular words and not the whole sentence. We would analyze the sentence ”this is not good”as having a negative sentiment because of the negation, but AFINN would only recognize “good” as positive.

The results from the CNN model demonstrate that it is possible to determine whether a sentence was spoken by the Queen or the PMs with an accuracy of 71.58%, though there is an uncertainty in the accuracy percentage, since we have used speeches from one Queen but many different PMs. This is not a problem as it rather shows that the two functions, the monarch and the PM, use different language that is detectable. This is a pretty interesting result, as it highlights that with a sufficient dataset, it is possible to train a model to recognize distinct differences in the words and sentences used by each speaker. Expanding the dataset, thus acquiring a larger sample size, would likely enhance the model's performance, allowing it to better capture and differentiate the patterns and characteristics of the Queen and the PMs.

Ethical challenges & privacy and societal impact:

Our study has made it clear that it is possible to discern between two speakers based solely on the language used, with high accuracy. But what are the implications, that we are able to deduce who has said what and that we can emulate it?

A naive view on this could be, that we can create language models that learn to imitate the language used in the Queen's and the PMs' speeches to help us write more formally or write speeches with a higher impact. The ability to imitate speech patterns of people - especially public figures - should not be taken lightly, as it can be exploited for impersonation and spreading misinformation. Not only can this be used to target groups that are more susceptible, such as older people [14], but it will generally erode trust in all communication.

Being able to discern between individuals based on their writing style also imposes serious issues about the loss of anonymity. This could be used in a political setting to target people with dissenting opinions in oppressive regimes or whistle-blowers within organizations and companies [15]. Additionally, not being able to be truly anonymous could stifle free expression, as people could become afraid of their words being traced back to them leading to retaliation.

These models all rely on lot of data about us, which is being scraped from the internet without explicit consent, resulting in a big violation of privacy. We have little control over what information is collected about us and what it is used for. With our data being widely available on the internet, there is a risk of others scraping our data and using it to train generative AI tools to mimic us and our writing style. This could be used to enable targeted scams, such as spearfishing [16], where scammers pose as friends and family, exploiting trust.

Litteratur

[1] geeksforgeeks. Latent semantic analysis.

[2] Wikipedia. n-gram.

[3] geeksforgeeks. Understanding tf-idf (term frequency-inverse document frequency).

[4] Roshan Joe Vincent. Singular value decomposition (svd) — working example.

[5] geeksforgeeks. Principal component analysis(pca).

[6] ARTiBA. Rnn vs. cnn: Understanding key differences in text classification.

[7] Wikipedia. Convolutional neural network.

[8] Jonathan Yeh Ali Khodabaksh Hesar, Chandra Kumar R Pillai. How can you use dropout to improve deep learning model robustness? tab.

[9] Nikita Malviya for 'Medium'. Demystifying cross-entropy: A key concept for understanding classification in machine learning.

[10] Bertel Torp. Dansk stopords liste / danish stopwords.

[11] Finn Årup Nielsen. Danish resources.

[12] geeksforgeeks. Softmax activation function in neural networks.

[13] F. Å. Nielsen. Afinn, mar 2011.

[14] Theis Stenh olt Engmann. Hvert ellevte ældre offer for it-kriminalitet har oplevet økonomiske tab.

[15] Sophie Luskin for corporatecomplianceinsights. Ai's potential chilling effect on corporate whistleblowing.

[16] Katharine Miller for Stanford University (Human-Centered Artificial Intelligence. Privacy in an ai era: How do we protect our personal information?

Appendix

Appendix 1: Link to GitHub

Link to our GitHub repository with all the files we have used and code we have written:

`https://github.com/ChristianDTU/Nyt-rstaler`

Appendix 2: CNN

Final data from epoch number 33:

Training loss: 0.4614, Training accuracy: 77.16%
Test loss: 0.5738, Test accuracy: 71.58%
Queen (0) accuracy: 70.07%
Prime Minister (1) accuracy: 73.12% .

Graphs with training & test loss and accuracy per epoch:

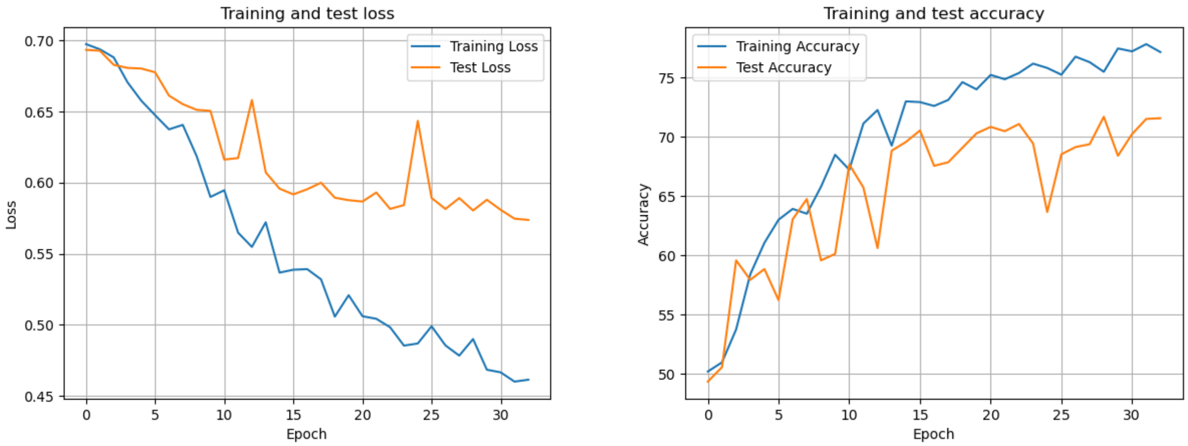

Fig. 2: Training & test loss and accuracy per epoch

For the training of the CNN, CosineAnnealing was used as a learning rate scheduler. It was primarily used, as it produced the highest test accuracy compared to other learning rate schedulers. Furthermore, the cross entropy loss function was given the two weights [1., 1.] to balance the dataset completely evenly, as there were still a few more PM sentences in the dataset after balancing the dataset.