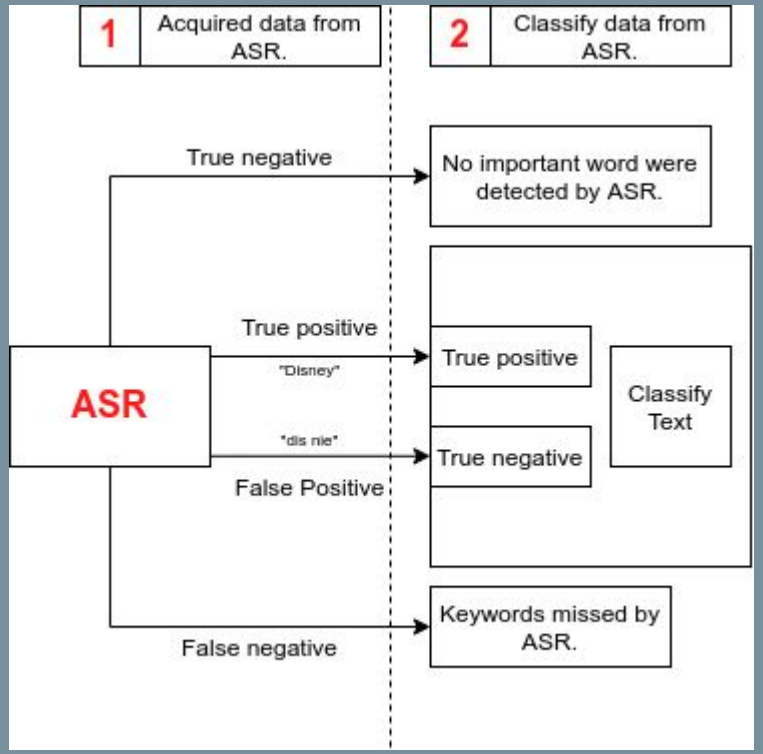
Classifying recognised speech with deep neural networks

Rhyno A. Strydom and Etienne Barnard
Multilingual Speech Technologies, NWU; and
Centre for Artificial Intelligence Research

What the project is about

- We acquired data from an automatic speech recognition (ASR) system.
- Analysed text obtained from the ASR system to determine whether or not certain keywords of interest are mentioned falsely in broadcast news.



Main contributions

1. We show that text obtained from ASR provides an interesting and important use case for NLP techniques.

2. We demonstrate the relative capabilities of various NLP solutions, including embeddings and more traditional methods on such a task in the multilingual South African context.

Investigation

- Investigated textual representations: TFIDF, Word2Vec, fastText and Doc2Vec.
- We use both linear and non-linear classification methods that explore the use of logistic regression and a DNN classifier.
- We also explored alternative methods of classification, using only word embedding and a CatBoost classifier trained on both the meta data and output of DNN trained on the textual features.

How to convert text vectors

Previously, methods included:

- TFIDF (Term frequency-inverse document frequency): a numerical static used to express the importance of a word as a whole when considering the entire corpus.

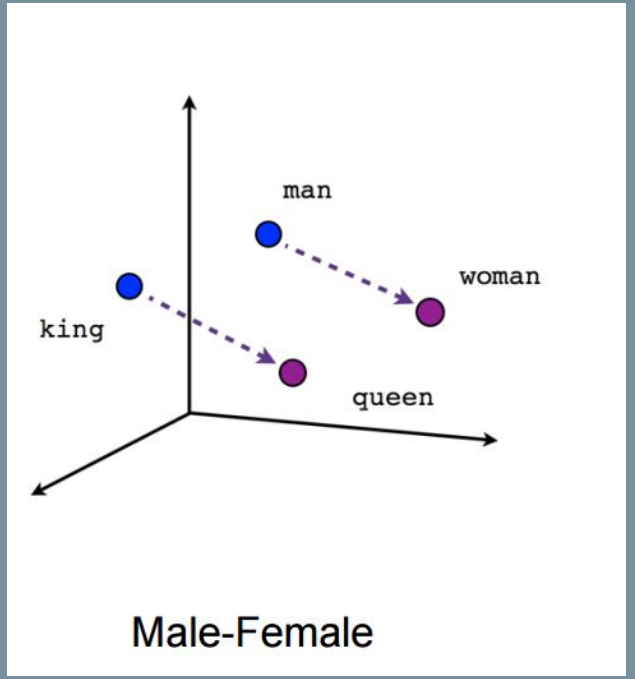
The problem with this type of modeling?

- These models fail to capture regularities such as the linguistic structure within languages.
- Last few years have shown promising performance relating to success of deep learning methods and word embeddings.

What is a word embedding?

A way of representing words as continuous vectors, able to capture semantic relationship between words in relatively low dimensional space.

$$\vec{\omega}_{king} - \vec{\omega}_{man} + \vec{\omega}_{woman} \approx \vec{\omega}_{queen}$$

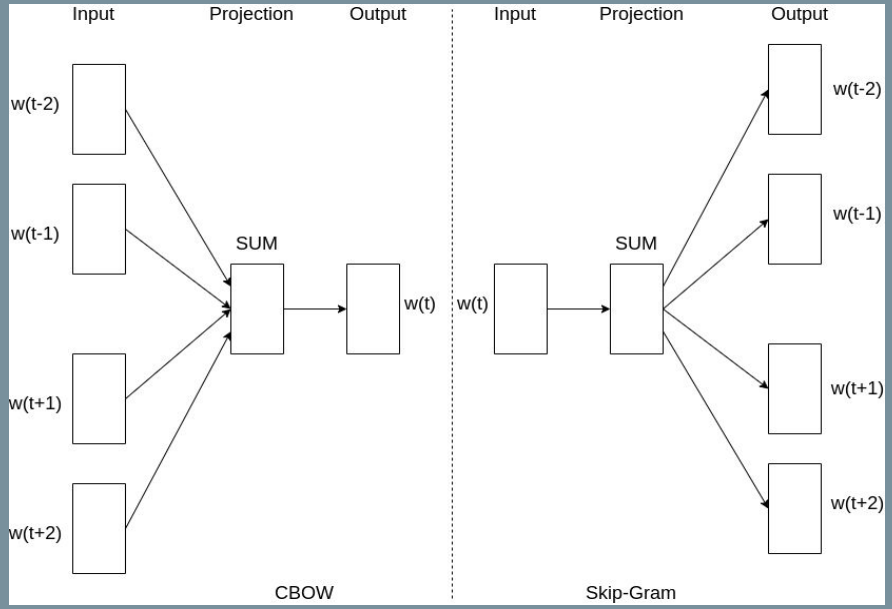


How word embeddings are trained

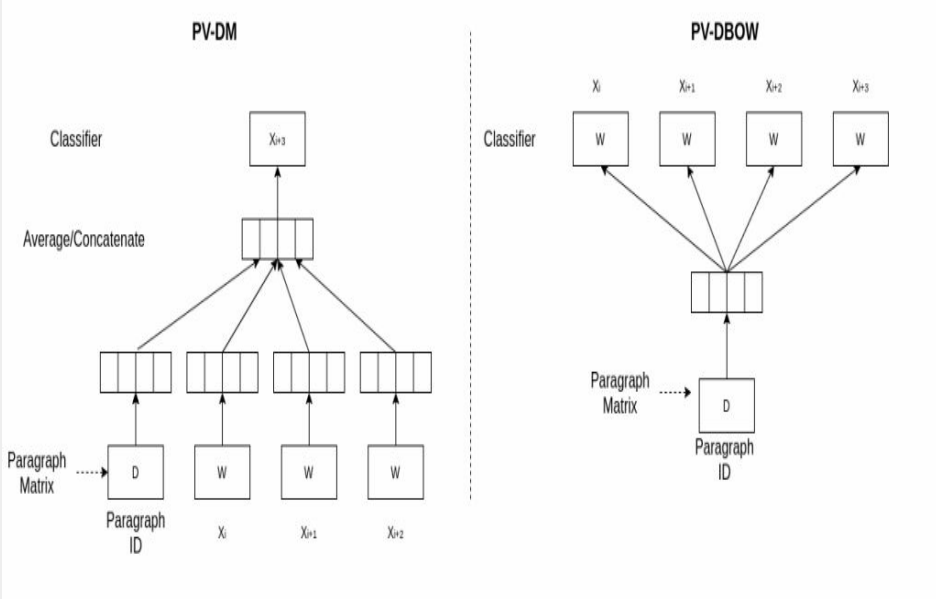
- Word embeddings are trained based on the distributional hypothesis: “words that appear in the same context share semantic meaning with each other” - basic premise of Word2Vec.
- fastText works the same, additionally enrinching embeddings with the words orthographic N-gram (i.e. “dizzy” = <di, diz, izz, zzy, zy>).
- Doc2Vec works in a similar fashion to training word vectors. Instead of training vectors for individual words it trains a fixed-length paragraph representation for the document as a whole by trying to predict words.

Methodology

Word2Vec & fastText



Doc2Vec

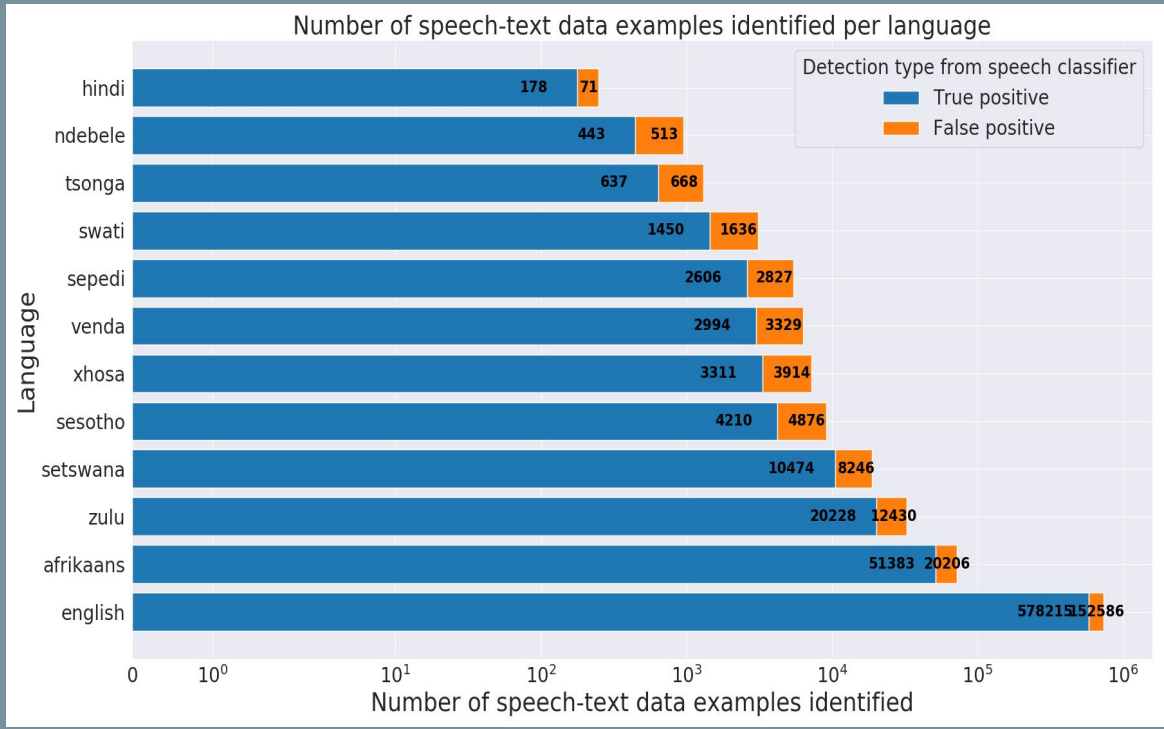


Data analysis and exploration

- Speech-text samples: 885,930
  - True positives: 675,425

    76.24%
  - True negatives: 210,505

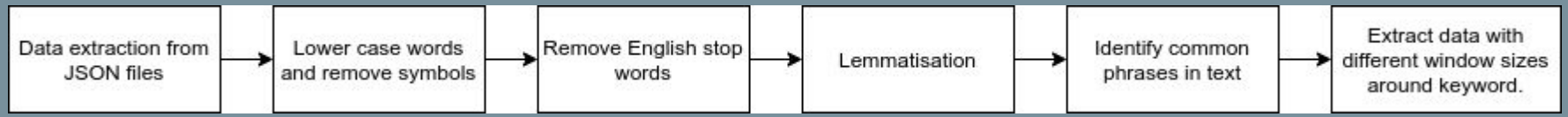
    23.76%



Data analysis and exploration

Text preprocessing pipeline:

- Before the text is converted to a numerical representation for the various machine learning algorithms, preprocessing is done on the corpus:

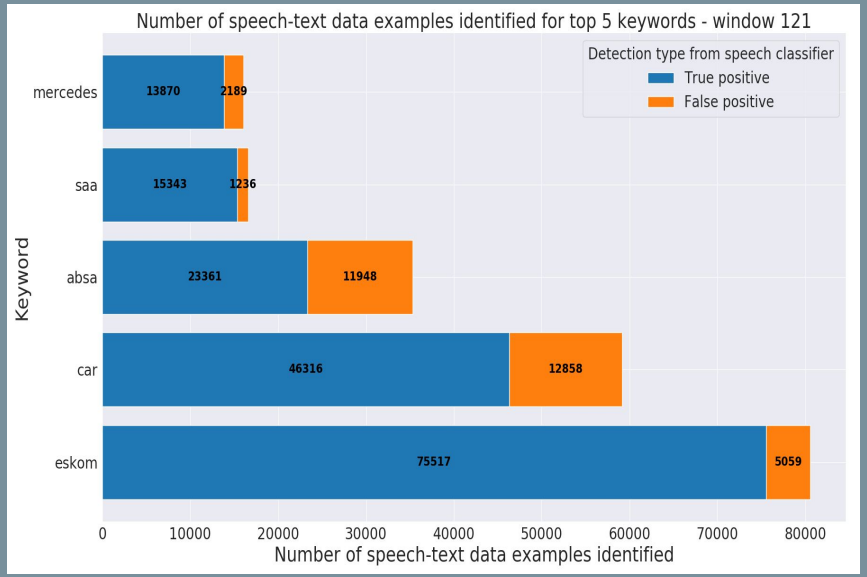


Corpus information

- The data is split into a train, validation and test set by randomly splitting the data in according to how frequently a specific keyword appears in the corpus.
- Training: 80% | Validation: 10% | Test: 10%

Window size around keyword	Number of words in corpus	Vocabulary size
window 5	7,562,980	**105,685**
window 11	15,521,514	**150,951**
window 21	27,244,705	**184,033**
window 41	46,840,613	**208,273**
window 61	63,174,235	**217,431**
window 121	99,466,572	**226,408**

Example:

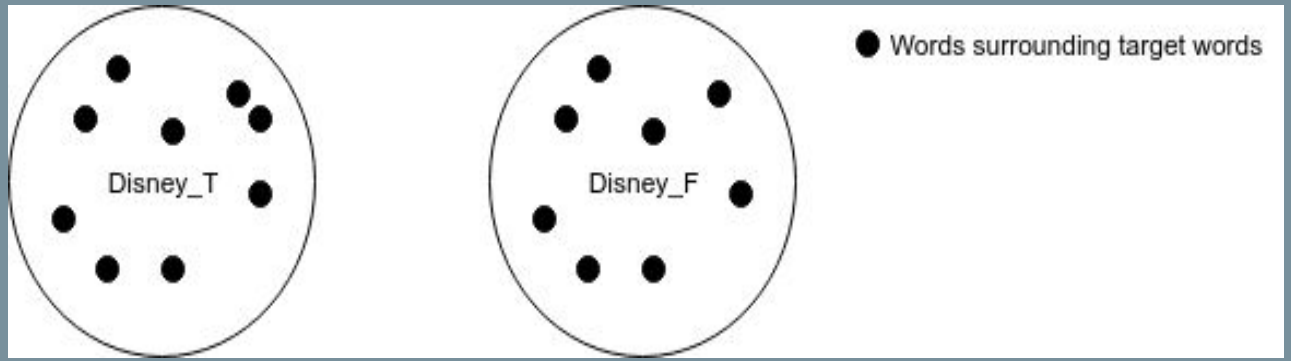


Logistic regression, DNN and CatBoost classifier

- Doc2Vec, Word2Vec, fastText and TFIDF are investigated using a logistic regression (linear classifier) and a one-layer feedforward neural network classifier (non-linear classifier), with varying capacity.
- Metadata associated with text is also used to train a CatBoost classifier.
- Features used include: broadcast-station name, keyword of interest, language and time of day. - Output of DNN is also used as additional input feature.
- Class weighted loss used to mitigate the effect of having an unbalanced data set.

Classifiers, optimisation and metrics

- Average cosine similarity method: keywords that are identified as false positives will be surrounded by similarly incorrect words that have nothing to do with the actual keyword.
- We add a flag and average cosine similarity between examples to distinguish between both cases of misclassified keywords.



Metrics used for evaluation

- Cohen Kappa statistic is used to evaluate and compare the various classifiers - measures the inter-rater agreement between categorical items.
- The CatBoost classifier SHAP values are used to evaluate feature importance.
- This is calculated by comparing what the model predicts with and without a specific categorical feature, and this is then used to determine which features are relevant and which are not.

Optimisation

- Both CatBoost and DNN use early stopping based on the Area Under the Curve (AUC) value to counter over-fitting.
- The DNN uses the Adaptive Moment Estimation (Adam) .
- Batch size: 1024
- Learning rate: 0.001

Cohen Kappa

Cohen kappa	Agreement
<0.00	Less than chance agreement
0.01-0.20	Slight agreement
0.21-0.40	Fair agreement
0.41-0.60	Moderate agreement
0.61-0.80	Substantial agreement
0.81-0.99	Almost perfect agreement

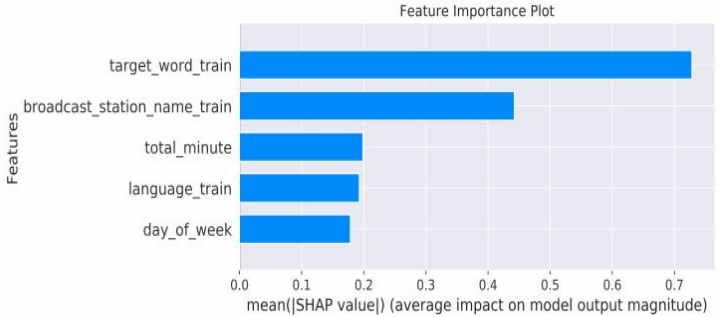
Results

Validation set

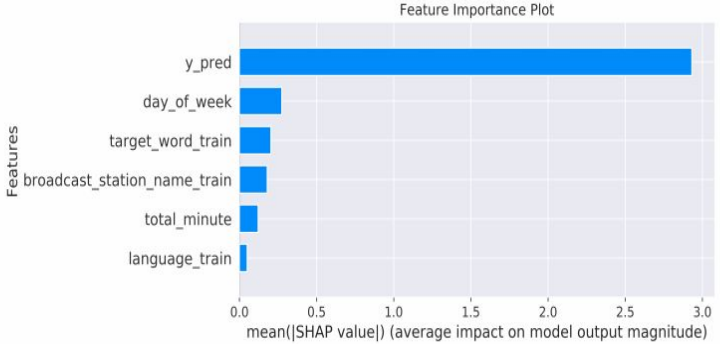
Embedding method	Classifiers					
	LR	DNN-100	DNN-200	DNN-400	Avg Cosine similarity method	CatBoost
Window size 11						
fastText SG	-	0.38563	0.39127	0.39540	-	-
Window size 21						
fastText SG	-	-	-	-	0.36370	-
Word2Vec SG	-	0.38288	0.39344	0.39876	-	-
Window size 41						
TFIDF	0.43271	-	-	-	-	-
fastText CBOW	-	-	-	-	0.22921	-
Window size 61						
fastText SG	0.32488	-	-	-	-	-
Word2Vec SG	0.32436	-	-	-	0.32971	-
Doc2Vec DBOW	-	0.37808	0.38126	0.38343	-	-
Window size 121						
No embedding (Baseline)	-	-	-	-	-	**0.49639**
TFIDF	-	0.45667	0.47026	0.47310	-	**0.46679**
Word2Vec CBOW	-	-	-	-	0.30704	-
Word2Vec SG	-	-	-	0.36289	-	-
Doc2Vec DBOW	0.30610	-	-	-	-	-

Test set

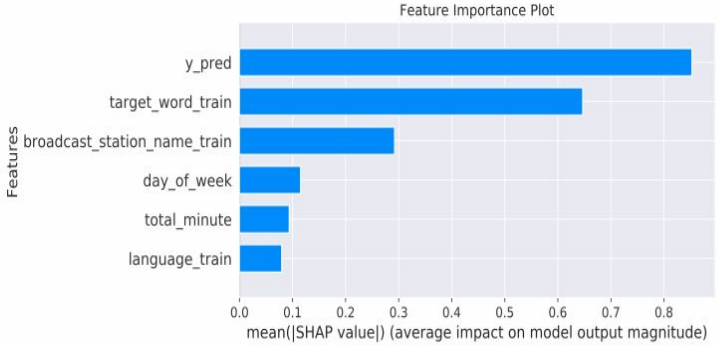
Embedding method	Classifiers					
	LR	DNN-100	DNN-200	DNN-400	Avg Cosine similarity method	CatBoost
Window size 121						
No embedding (Baseline)	-	-	-	-	-	**0.49481**
TFIDF	-	-	-	0.47209	-	**0.46465**
Word2Vec SG	-	-	-	0.36128	-	**0.51399**



(a) SHAP values for baseline Catboost classifier.



(b) SHAP values for Catboost classifier with added output prediction from 400 node feedforward neural network using TFIDF.



(c) SHAP values for Catboost classifier with added output prediction from 400 node feedforward neural network using Word2Vec.

Conclusion

- Applying TFIDF as the textual representation with a neural network, outperformed all other embedding based methods.
- Considering the output of the DNN using TFIDF and CatBoost classifier as an additional feature to the CatBoost, caused classifier to over-fit on this feature.
- Word2Vec slightly enhanced the model's performance compared to the baseline CatBoost classifier relying on the metadata associated with the speech-text data.

Questions ?