CNN text classification model using Word2Vec

GALDO SEARA Luis
GONZALEZ HUESCA Juan Manuel

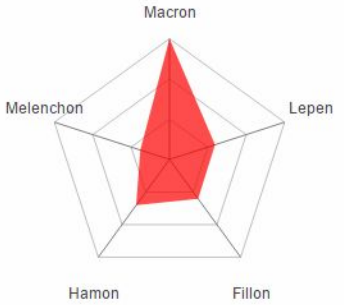
Supervisor: Professor PRECIOSO Frederic

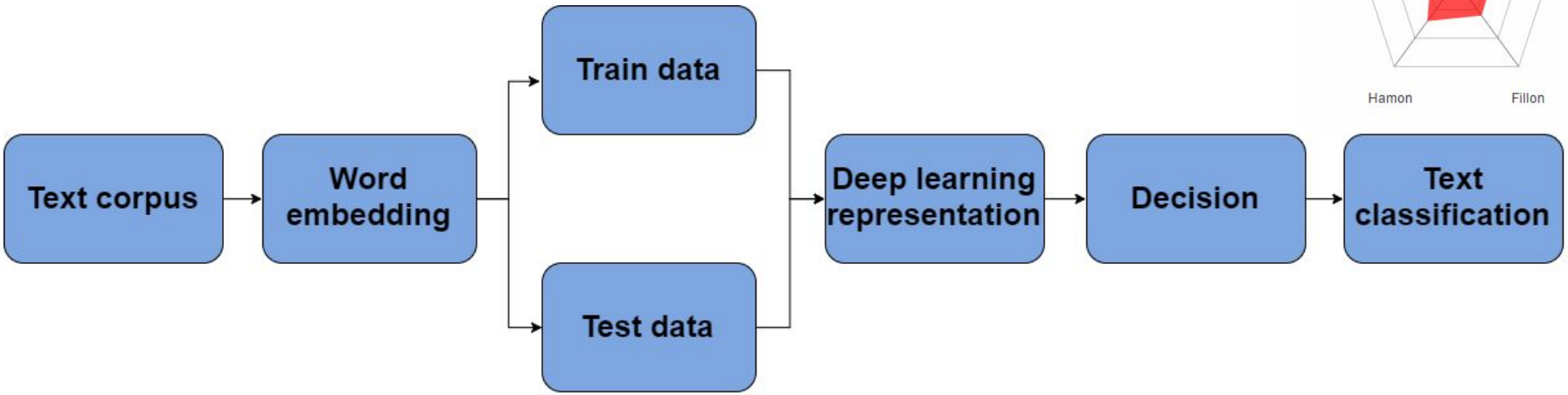
Agenda

- Project description
- Introduction to Neural Networks
- Analysis and Design
- Implementation
- Testing
- Results
- Conclusion, Evaluation and Further Work

Project Description

- Word2Vec implementation on presidential campaign speeches
- Find semantic relationships between different candidates' speeches
- Classification of political speeches
- Usage of Python deep learning libraries

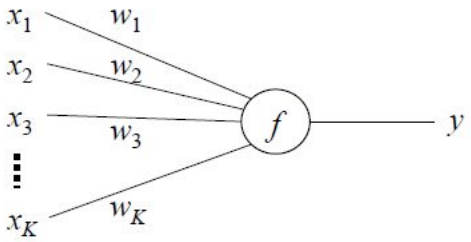




http://mesure-du-discours.unice.fr/

Introduction to Neural Networks - Neuron

- An Artificial Neuron



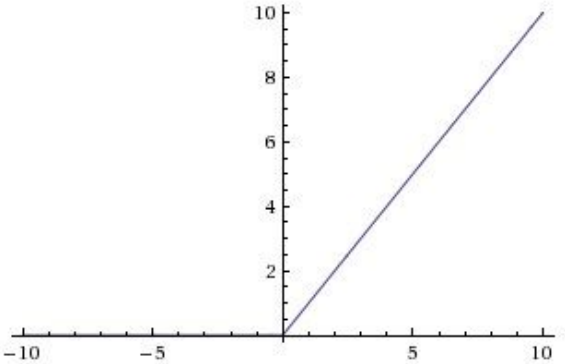
$$y = f(u)$$

$$u = \sum_{i=0}^{K} w_i x_i$$

$$f(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{otherwise} \end{cases}$$

$$\mathbf{w}^{(\text{new})} = \mathbf{w}^{(\text{old})} - \eta \cdot (y - t) \cdot \mathbf{x}$$

- Activation function
  - Step function $f(u) = \begin{cases} 1 & \text{if } u > 0 \\ 0 & \text{otherwise} \end{cases}$
  - Sigmoid function $\sigma(u) = \dfrac{1}{1 + e^{-u}}$
  - ReLU $RELU(x) = \begin{cases} 0 \text{ if } x < 0 \\ x \text{ if } x >= 0 \end{cases}$



Xin Rong. word2vec Parameter Learning Explained [https://arxiv.org/abs/1411.2738]

Introduction to Neural Networks - Word2Vec

- CBOW (one target word given context words)

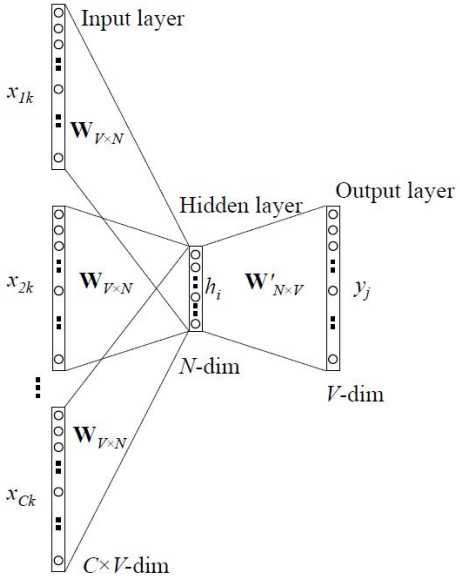


Figure 3. Continuous bag-of-words model

- Skip-gram (context words given a target word)

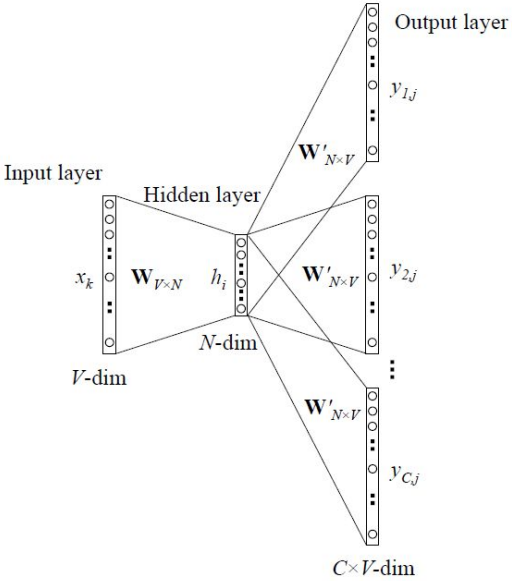


Figure 4. Skip-gram model

Xin Rong. word2vec Parameter Learning Explained [https://arxiv.org/abs/1411.2738]

Analysis and Design

1. Text cleaning
2. Word2Vec (word embedding)
3. Train and Test Data
4. Convolutional Neural Network
   a. Convolutional layers
   b. Pooling layer
   c. Activation function
5. Classification

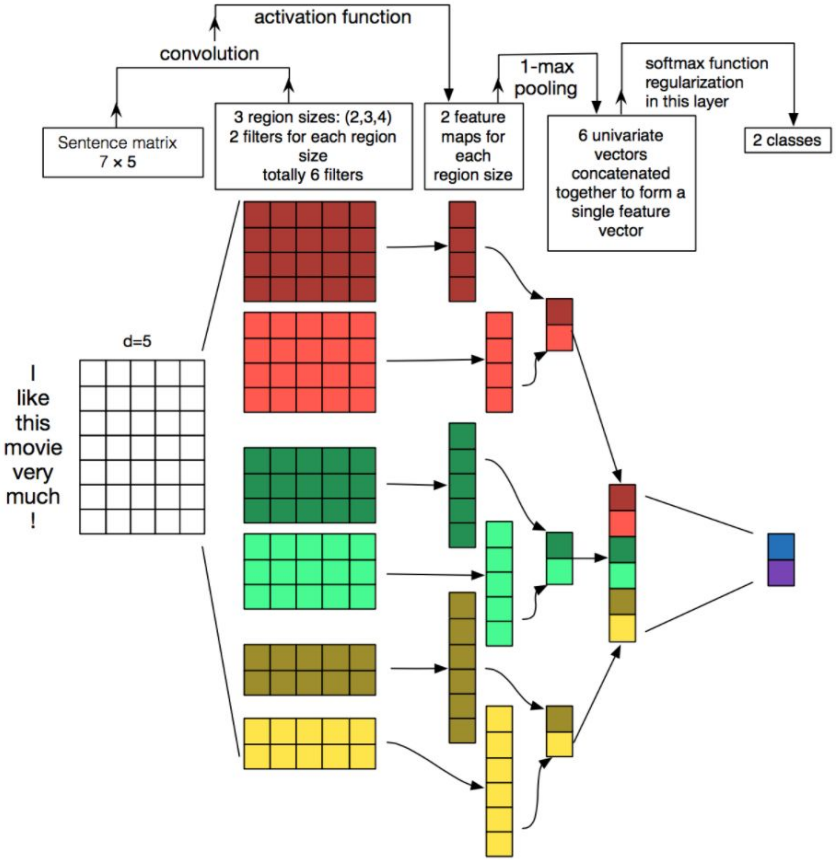
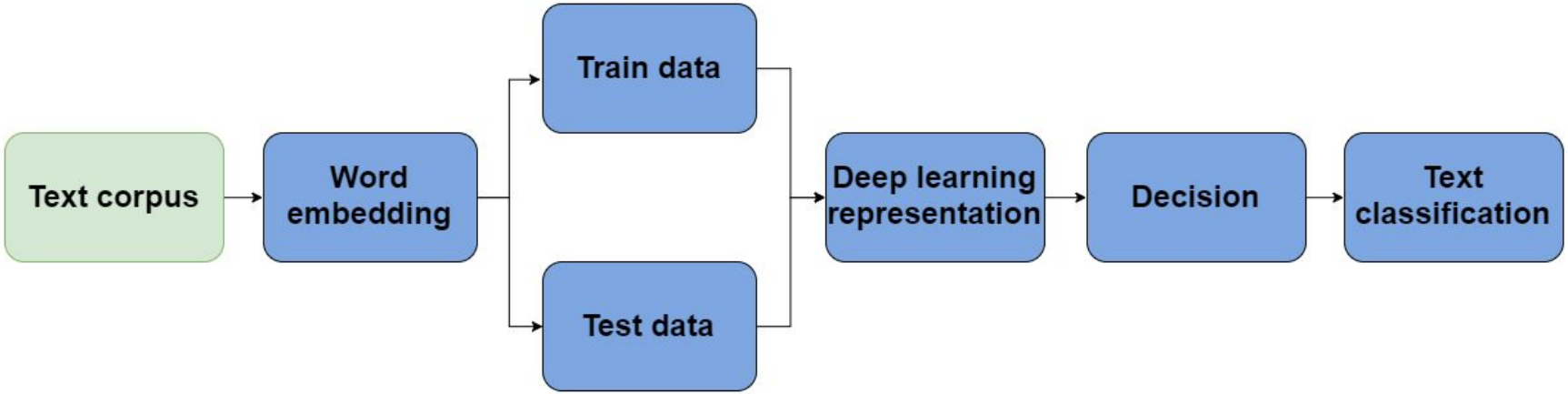


Figure 5. CNN architecture for sentence classification

http://www.wildml.com/2015/11/understanding-convolutional-neural-networks-for-nlp/

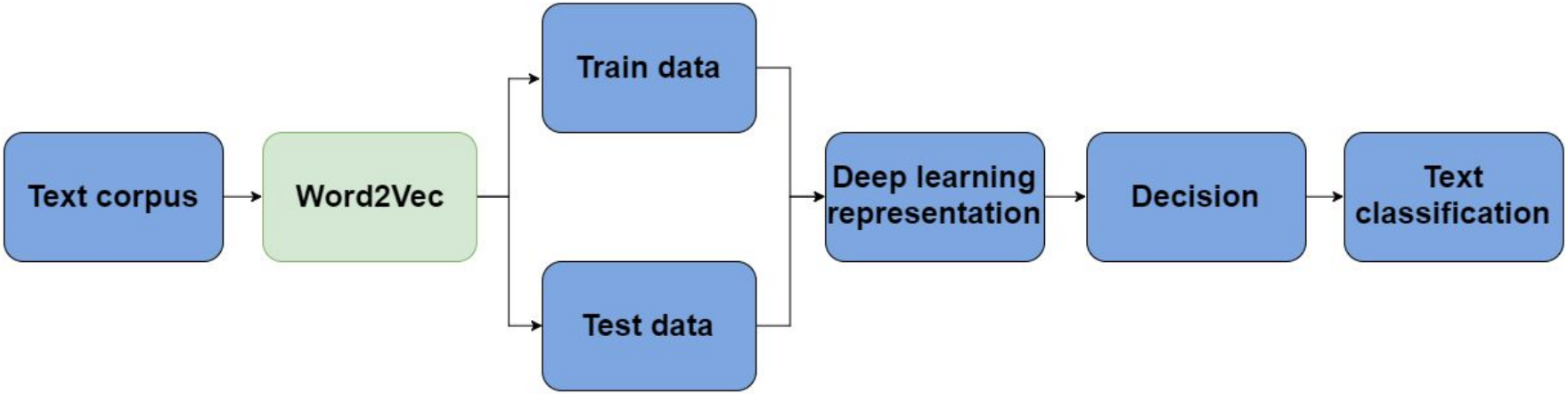
Analysis and Design - Text Cleaning



Analysis and Design - Text Cleaning

- Sentence division
  - (tokenizer from nltk)
- Lemmatization
  - (TreeTagger)
- Stopwords removal
  - (stopwords from nltk)

Analysis and Design - Word2Vec



Analysis and Design - Word2Vec

- Parameters specifications
  - Size
    - 300
  - Window
    - 8
  - Architecture
    - CBOW
  - Minimum count of words
    - 1

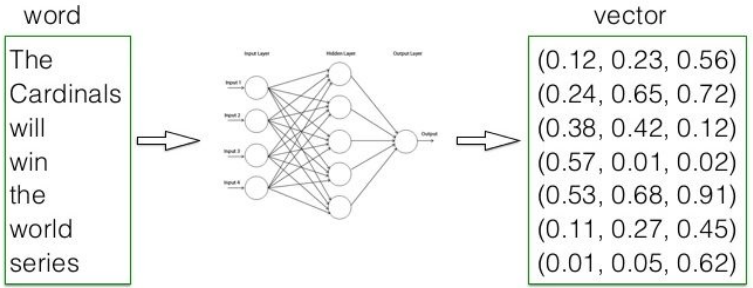
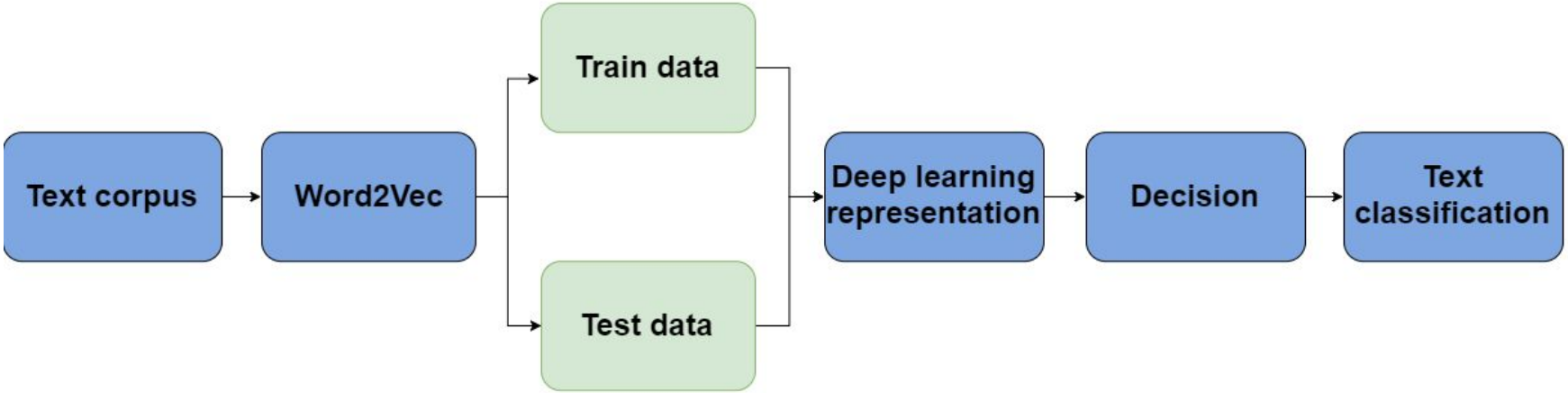


Figure 6. Word2Vec example

*Yoon Kim. Convolutional Neural Networks for Sentence Classification [https://arxiv.org/abs/1408.5882]

https://www.slideshare.net/andrewkoo/word2vec-algorithm

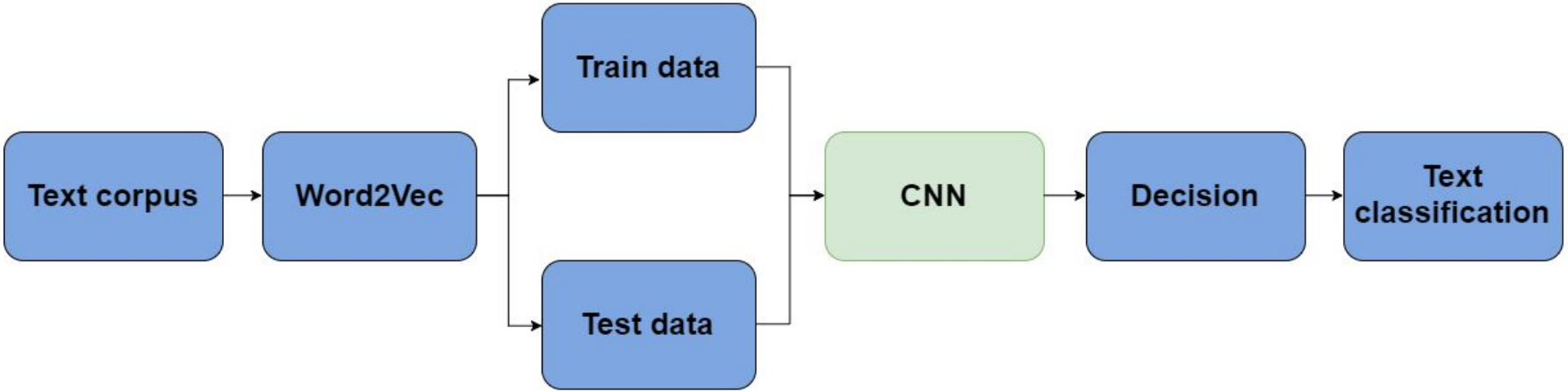
Analysis and Design - Train and Test Data



Analysis and Design - Train and Test Data

- Speeches to sentences
- Sentences to word lists
- Substitution of words for Word2Vec vector representation
- Division of data
  - Train
  - Test

Analysis and Design - CNN Convolutional Layers



Analysis and Design - CNN Convolutional Layers

- Convolution

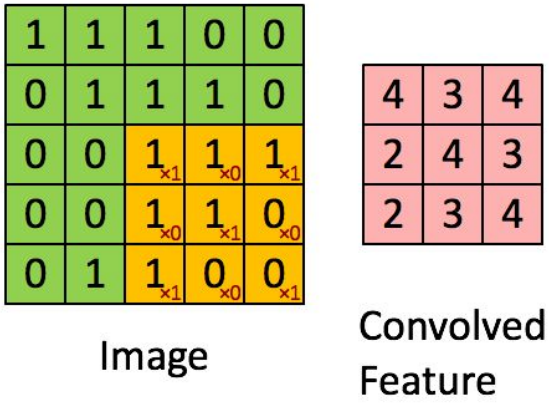


Figure 7. Convolution on images

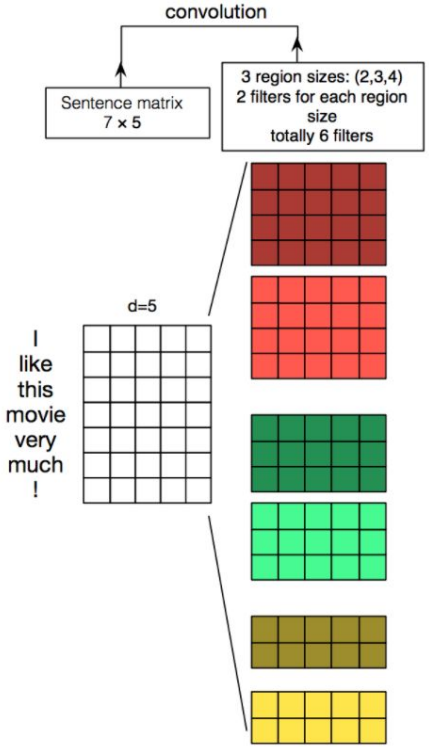


Figure 8. Convolution on text

Analysis and Design - CNN Convolutional Layers

- Embedding
- Dropout
- Convolutional block
  - Convolution1D
  - Max pooling
  - Flatten
- Concatenate
- Dropout
- Activation function

Layer (type)	Output Shape	Param #	Connected to
input_1 (InputLayer)	(None, 30)	0	
embedding (Embedding)	(None, 30, 300)	3653400	input_1[0][0]
dropout_1 (Dropout)	(None, 30, 300)	0	embedding[0][0]
conv1d_1 (Conv1D)	(None, 28, 100)	90100	dropout_1[0][0]
conv1d_2 (Conv1D)	(None, 27, 100)	120100	dropout_1[0][0]
conv1d_3 (Conv1D)	(None, 26, 100)	150100	dropout_1[0][0]
max_pooling1d_1 (MaxPooling1D)	(None, 14, 100)	0	conv1d_1[0][0]
max_pooling1d_2 (MaxPooling1D)	(None, 13, 100)	0	conv1d_2[0][0]
max_pooling1d_3 (MaxPooling1D)	(None, 13, 100)	0	conv1d_3[0][0]
flatten_1 (Flatten)	(None, 1400)	0	max_pooling1d_1[0][0]
flatten_2 (Flatten)	(None, 1300)	0	max_pooling1d_2[0][0]
flatten_3 (Flatten)	(None, 1300)	0	max_pooling1d_3[0][0]
concatenate_1 (Concatenate)	(None, 4000)	0	flatten_1[0][0] flatten_2[0][0] flatten_3[0][0]
dropout_2 (Dropout)	(None, 4000)	0	concatenate_1[0][0]
dense_1 (Dense)	(None, 50)	200050	dropout_2[0][0]
dense_2 (Dense)	(None, 6)	306	dense_1[0][0]

Total params: 4,214,056
Trainable params: 4,214,056
Non-trainable params: 0

*Yoon Kim. Convolutional Neural Networks for Sentence Classification [https://arxiv.org/abs/1408.5882]

Analysis and Design - CNN Convolutional Layers

Convolution1D

	Kernel sizes	**Number of filters**	**Activation function**
Paper*	3, 4, 5	100	ReLU
Our adaptation	3, 3	300	ReLU

*Yoon Kim. Convolutional Neural Networks for Sentence Classification [https://arxiv.org/abs/1408.5882]

Analysis and Design - CNN Pooling Layer

- MaxPooling1D
  - Pool size
    - 2

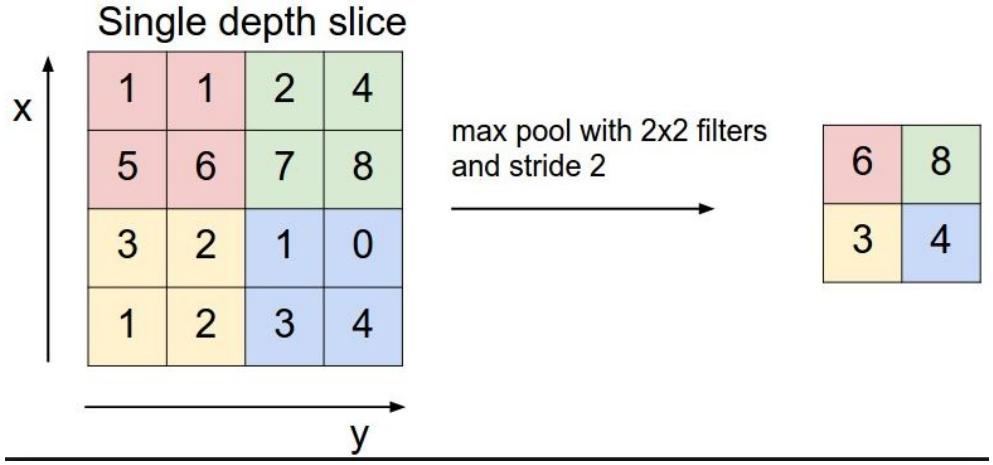
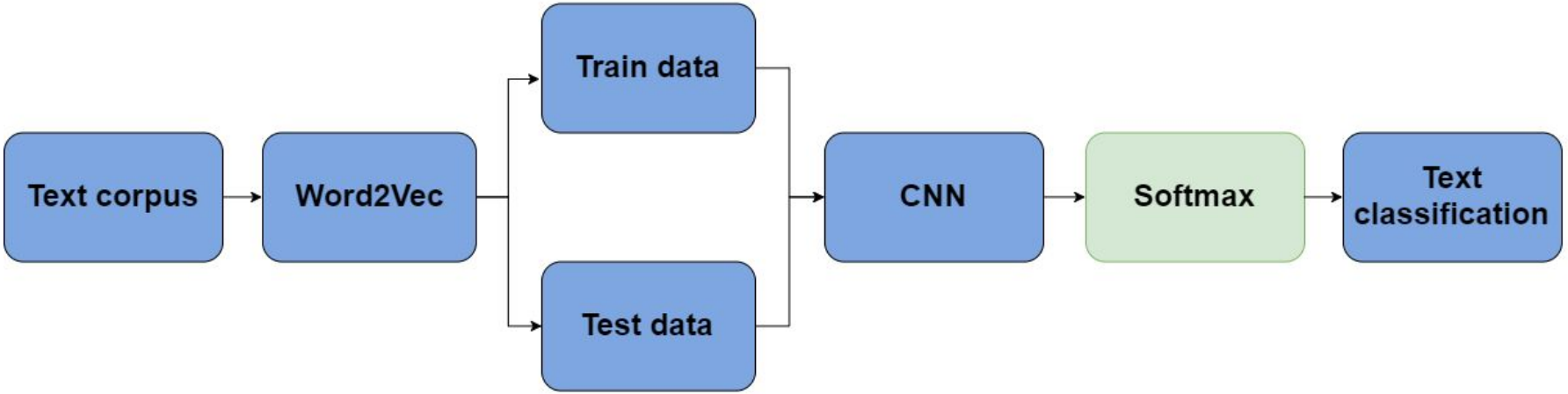


Figure 9. Max pooling example with 2x2 filter

*Yoon Kim. Convolutional Neural Networks for Sentence Classification [https://arxiv.org/abs/1408.5882]

Analysis and Design - CNN Activation function



Analysis and Design - CNN Activation function

- Softmax

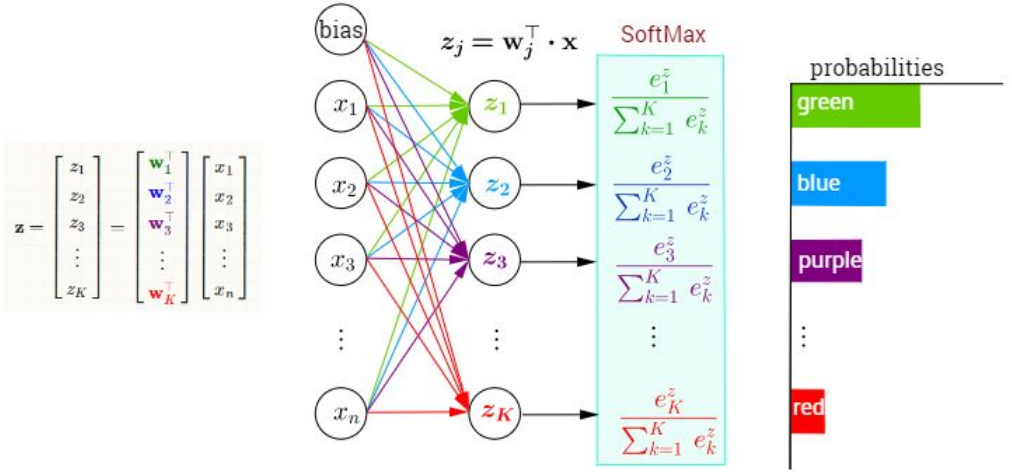
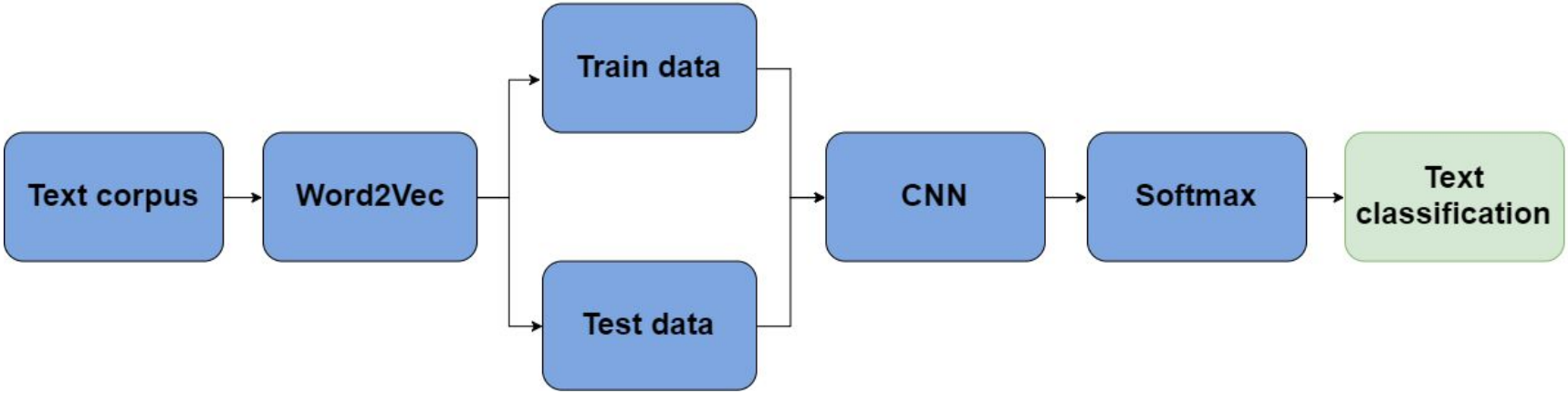


Figure 10. Multi-Class Classification with NN and SoftMAx Function

https://stats.stackexchange.com/questions/265905/derivative-of-softmax-with-respect-to-weights/271944

Analysis and Design - Classification



Analysis and Design - Classification

- Batch size
  - 64
- Epochs
  - 100

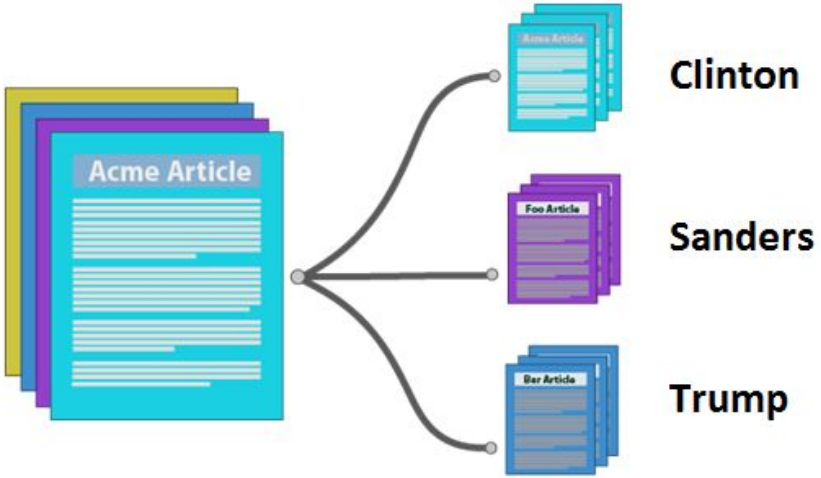
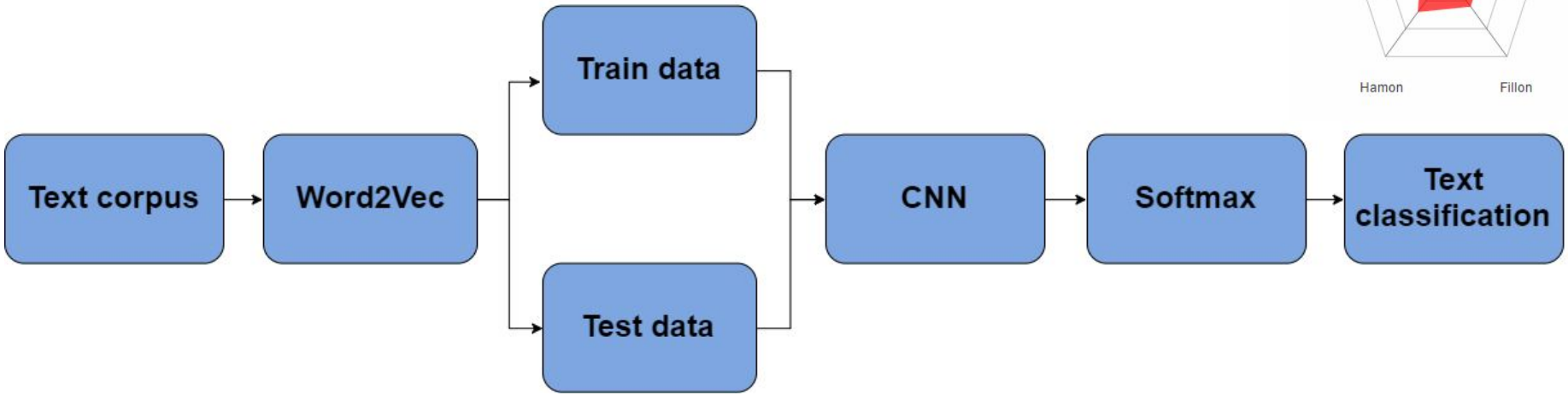


Figure 11. Classification example

Implementation



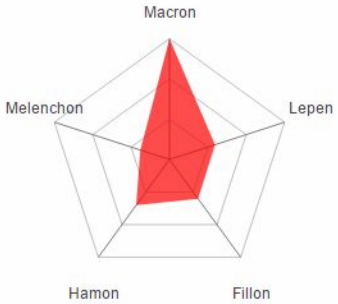


Figure 12. Block diagram of implementation

Testing Approaches

1. Word2Vec model per candidate
    a. Without expert knowledge
2. Word classification
    a. Too inaccurate
3. Sentence classification
    a. Dynamic size of sentences
        i. Not possible to apply deep learning
    b. Fixed size*
        i. CNN-static
            1. Word2Vec is not used
        ii. CNN-non-static

*Yoon Kim. Convolutional Neural Networks for Sentence Classification [https://arxiv.org/abs/1408.5882]

Testing

- USA National 2016 Elections
  - 3 candidates
  - Same number of speeches per candidate
    - 10
  - Similar number of sentences per candidate
- France National 2017 Elections
  - 6 candidates
  - Different number of speeches per candidate
    - 1-11
  - Different number of sentencesper candidate

Results

	Candidates	Training speeches	Testing speeches	Results
USA	3	24	6	57.92 %
France	6	27	6	72.45 %
France 2.0	4	*	*	55.72 %

* Speeches were divided into sentences and then the model was trained.

Conclusion and Evaluation

- Knowledge acquired
  - Text analysis (NLP)
  - Python
  - Word2vec
  - Deep Learning (CNN)
- Interesting project
  - Applied into recent data
- Better results were expected
  - Not enough training data

Further Work

- Improve accuracy
  - Neural Networks increase performance with more data for training
  - Training group of sentences (paragraphs) instead of single sentences
    - If enough data, training the whole speech
  - Different parameters in word2vec
  - Different parameters in Neural Network
  - More epochs
- Deconvolution

Thank you!

GALDO SEARA Luis
GONZALEZ HUESCA Juan Manuel