
ESTRA

Enhanced Service Ticket Recognition Agent

Sacarea Christian, Matei Cristian, Tolciu Tudor

**We're gonna (try to) make
math interesting!**

Natural Language Processing

- Text and speech processing
- “Understanding” human language
- Computer-Human interaction
- Affective computing



 Cortana

NATURAL LANGUAGE PROCESSING





The problem

- **Receiving incoming Helpdesk E-Mails**
- **Classifying them in specific categories using our trained AI agent (Category, Subcategory)**
- **Updating the database with freshly labeled requests**



Our approach

- **Bag of words**
Basic word counter vectors
- **Lemmatized Bag of words**
BoW but with lemmatized words (verb roots, singulars, no declinations, etc.)
- **Word2Vec**
Word features computed with another agent to form a context map
- **Long short term memory NNs**
Recurrent NN that support sequential input processing,

Tools & Algorithms

Lemmatization



Tensorflow



Bag of Words

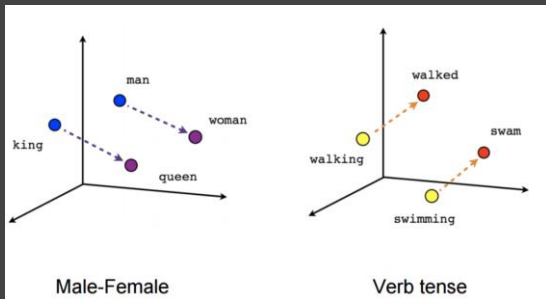
the dog is on the table

0	0	1	1	0	1	1	1
are	cat	dog	is	now	on	table	the

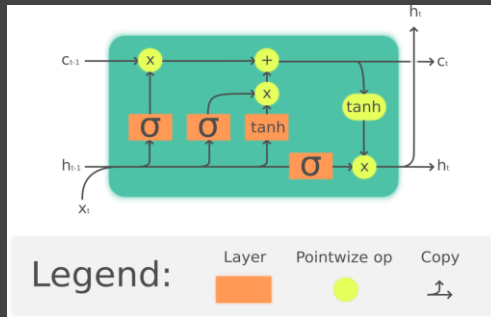
Python (3.7)



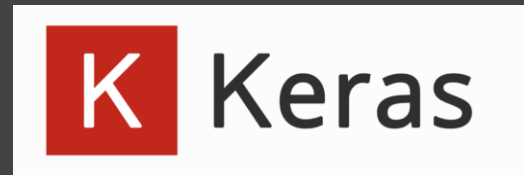
Word2Vec



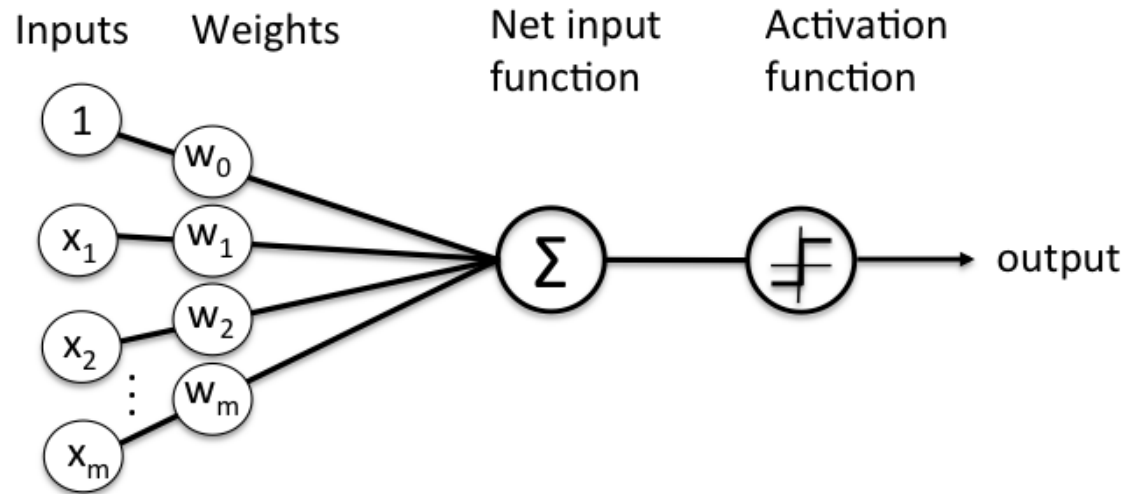
Long short term memory



Keras wrapper

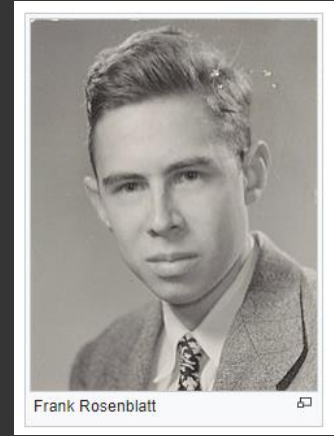


_ AI: The Beginning... (1958)

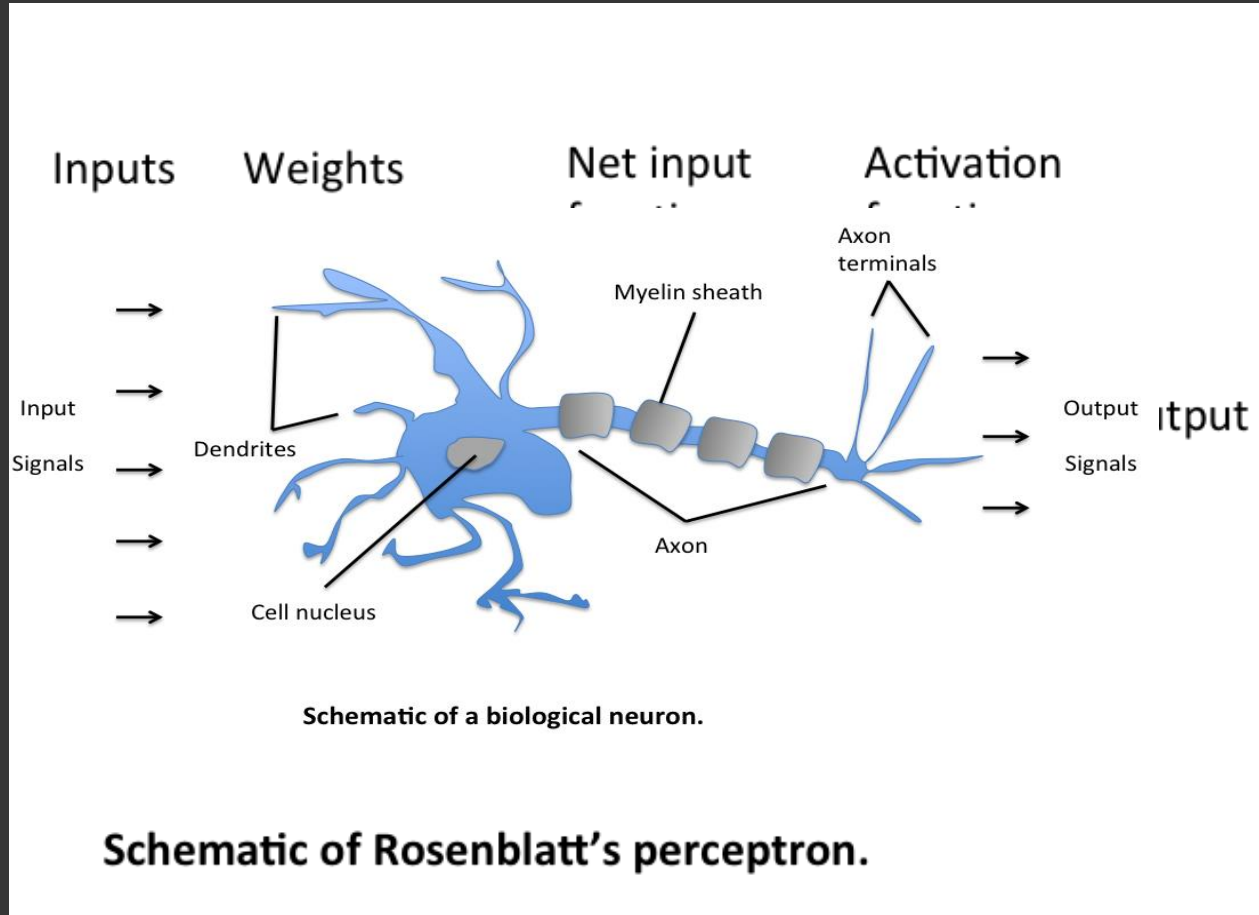


$$f(x_1w_1 + x_2w_2 + \dots + x_mw_m + b_0) = y$$

Schematic of Rosenblatt's perceptron.

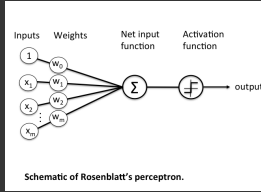


_AI: The neuron

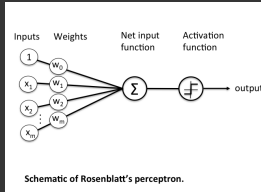


_AI: The neuronal net

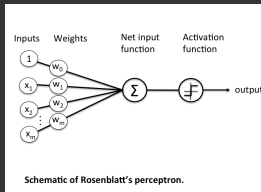
Neuron



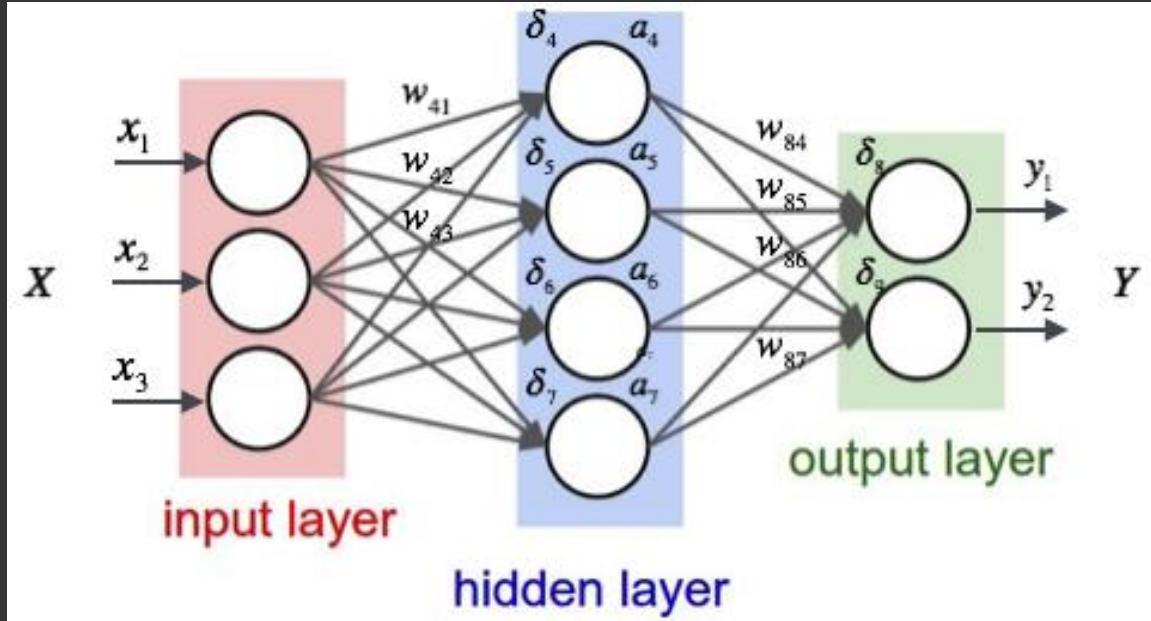
Neuron



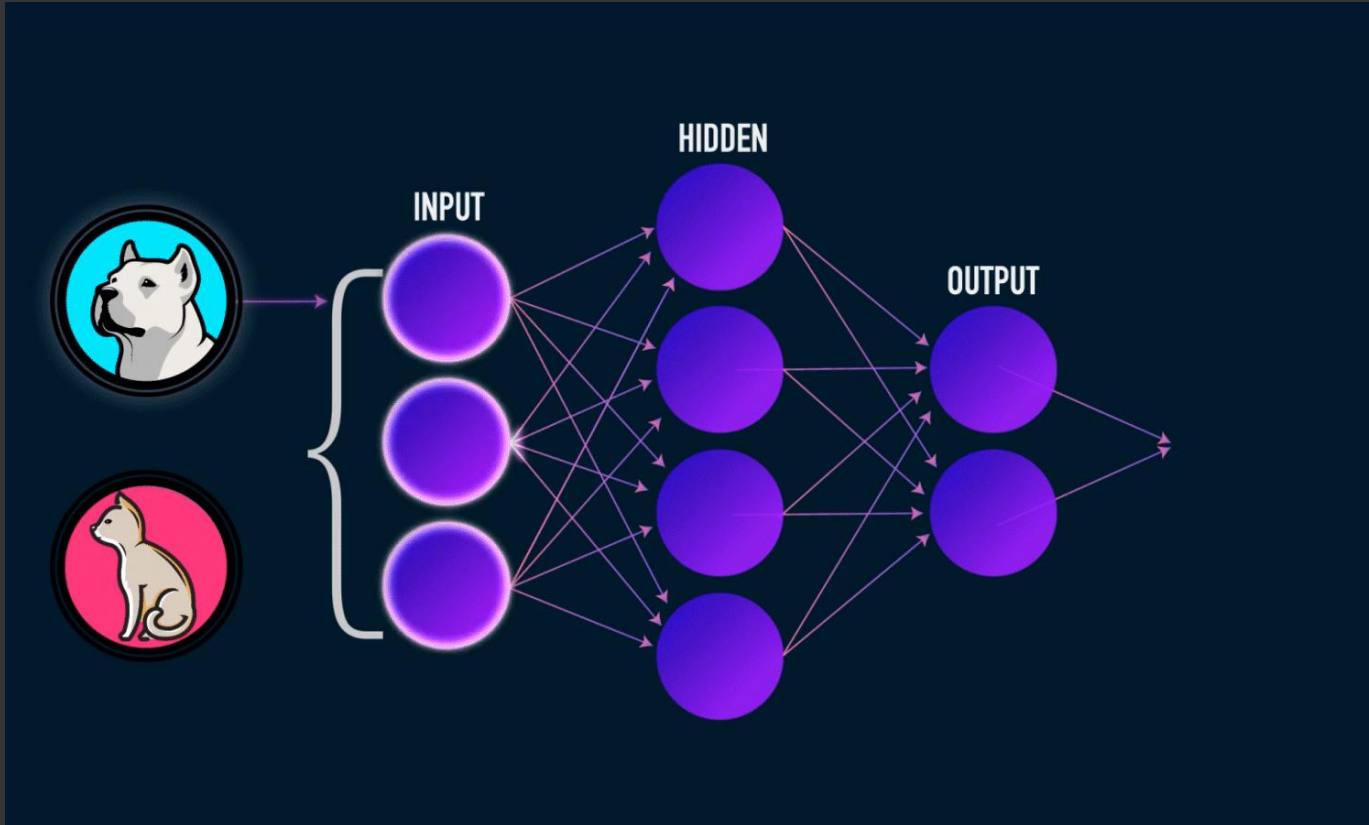
Neuron



Artificial Neural Net



– AI: The neuronal net

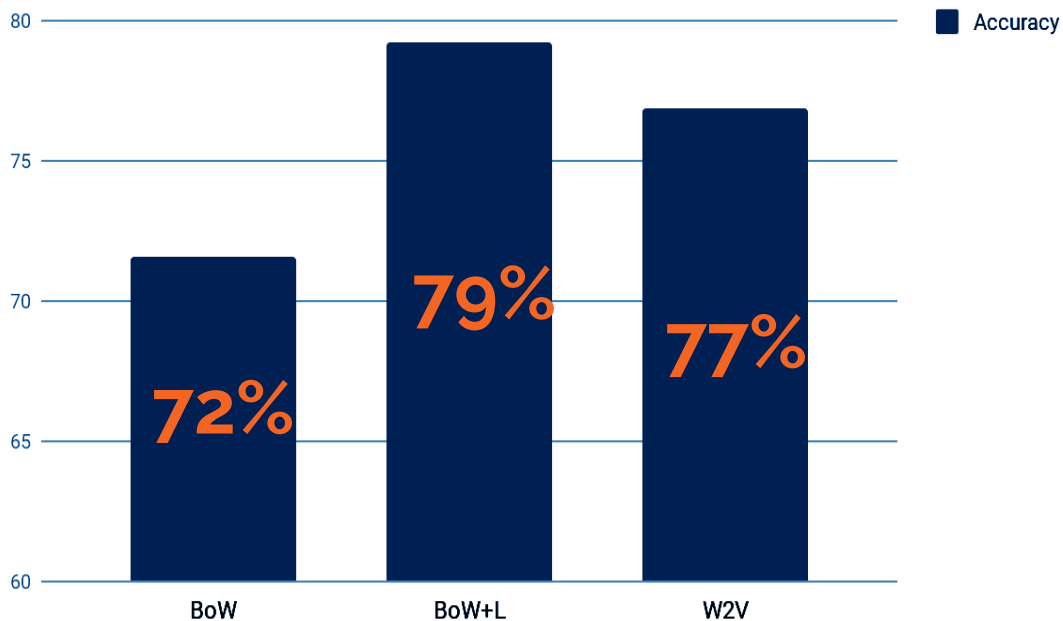


Project steps



Proof of concept results

Results on IMDB review DB



FYI

Models were created and trained using Tensorflow in Python and adjacent libraries such as WordNet and Gensim. Displayed results are all achieved using the same data, and same network hyperparameters (epochs, batch size, etc.). Maximum training cap not reached due to time limits.



First application

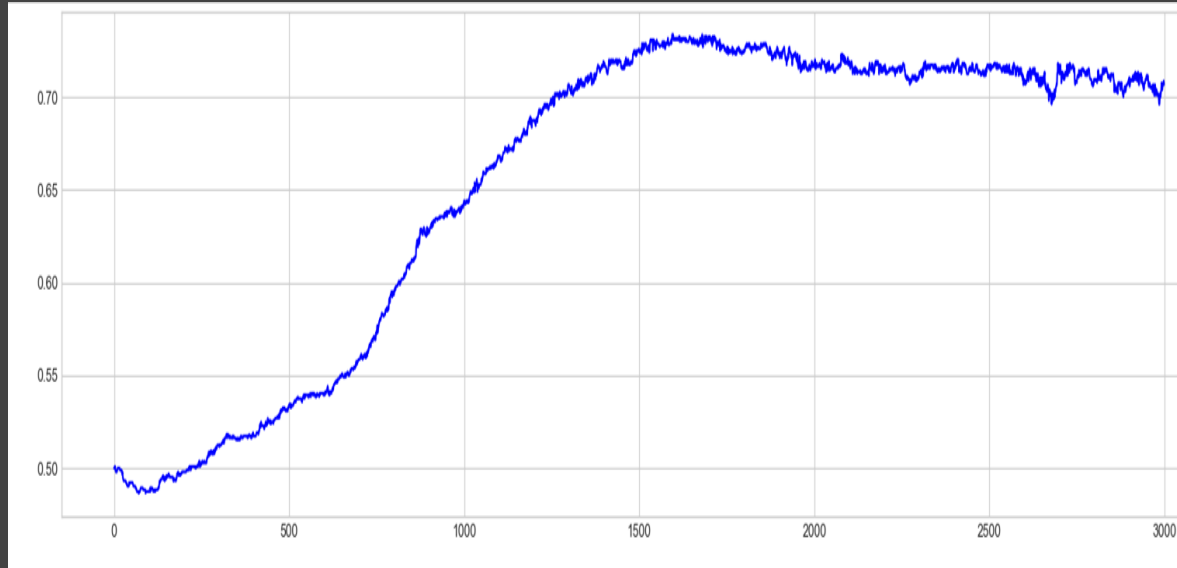
After the proof of concept experiment we conducted, we received our first batch of data: mail that had to be classified into two categories: **Service Request** and **Incident**

Service Request: the user requests something. It needs access for something, or some specific device or software

Incident: there has been a problem with the equipment. Something does not function as designed



First application: standard method results (Accuracy)



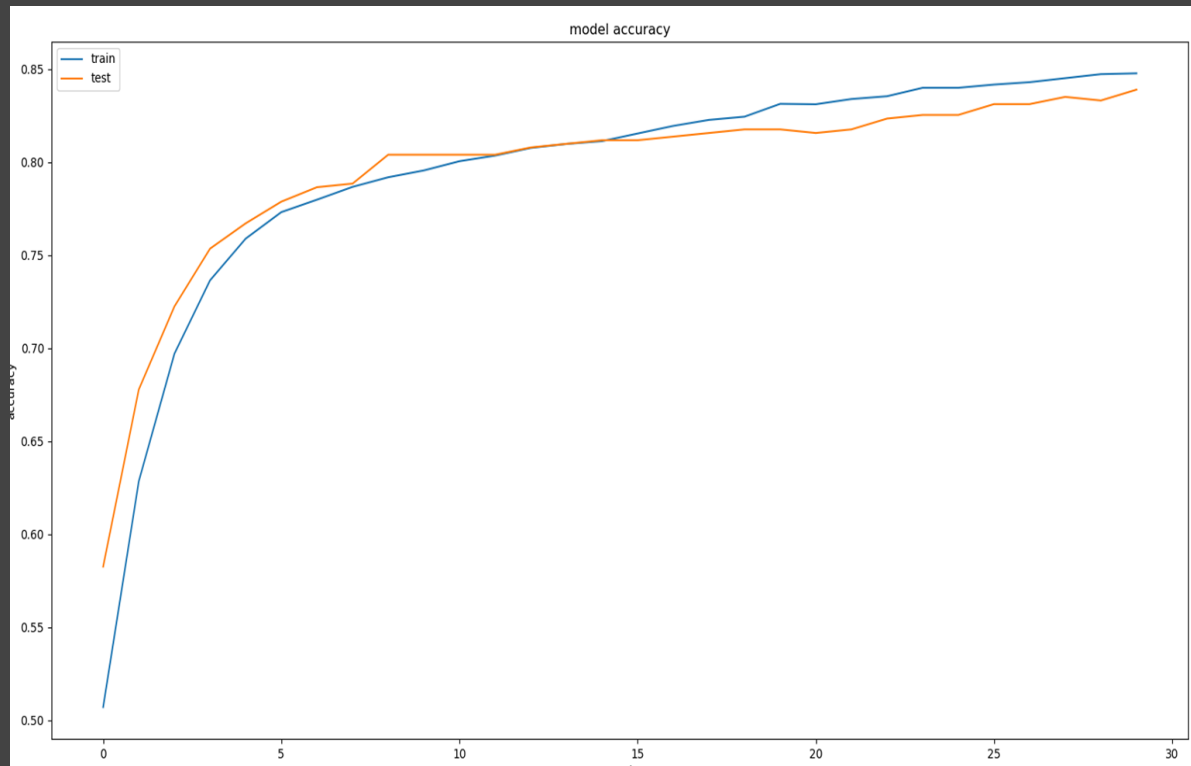
FYI

Accuracy: cca. 74%

Blue: validation data

! At epoch 1600:
Overfitting

Upgraded Algorithms results (Keras, LSTM) (Accuracy)



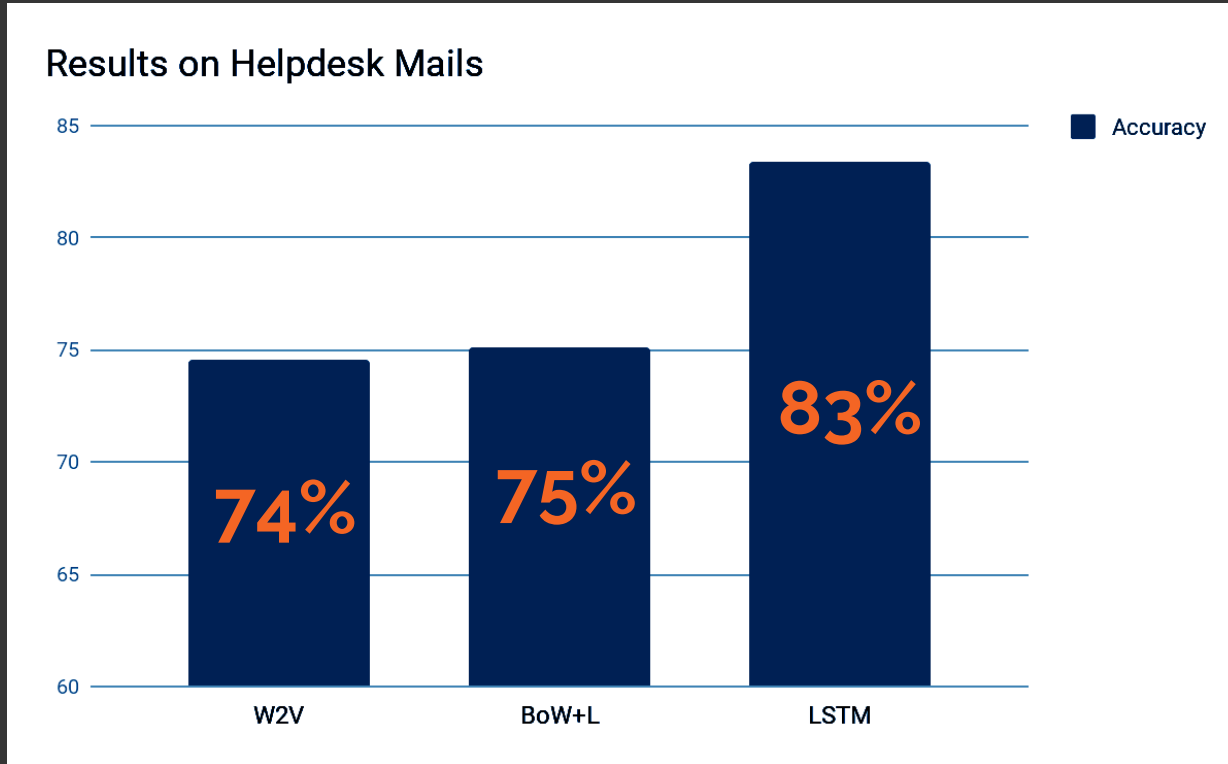
FYI

Accuracy: cca. 83%

Blue: training data

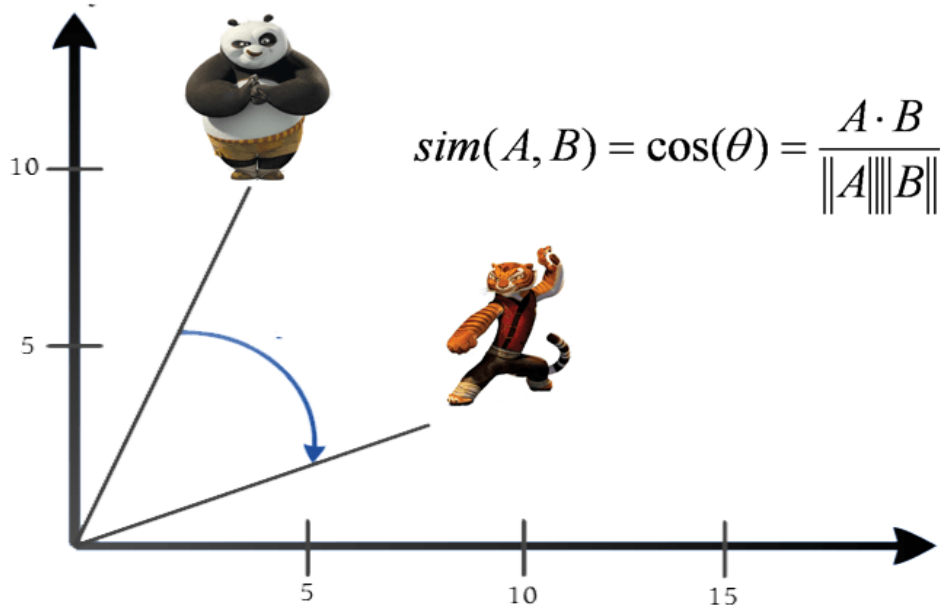
Orange: validation data

Results: ServiceRequest vs. Incident



Model Analysis: Similarity of tickets

Cosine Similarity



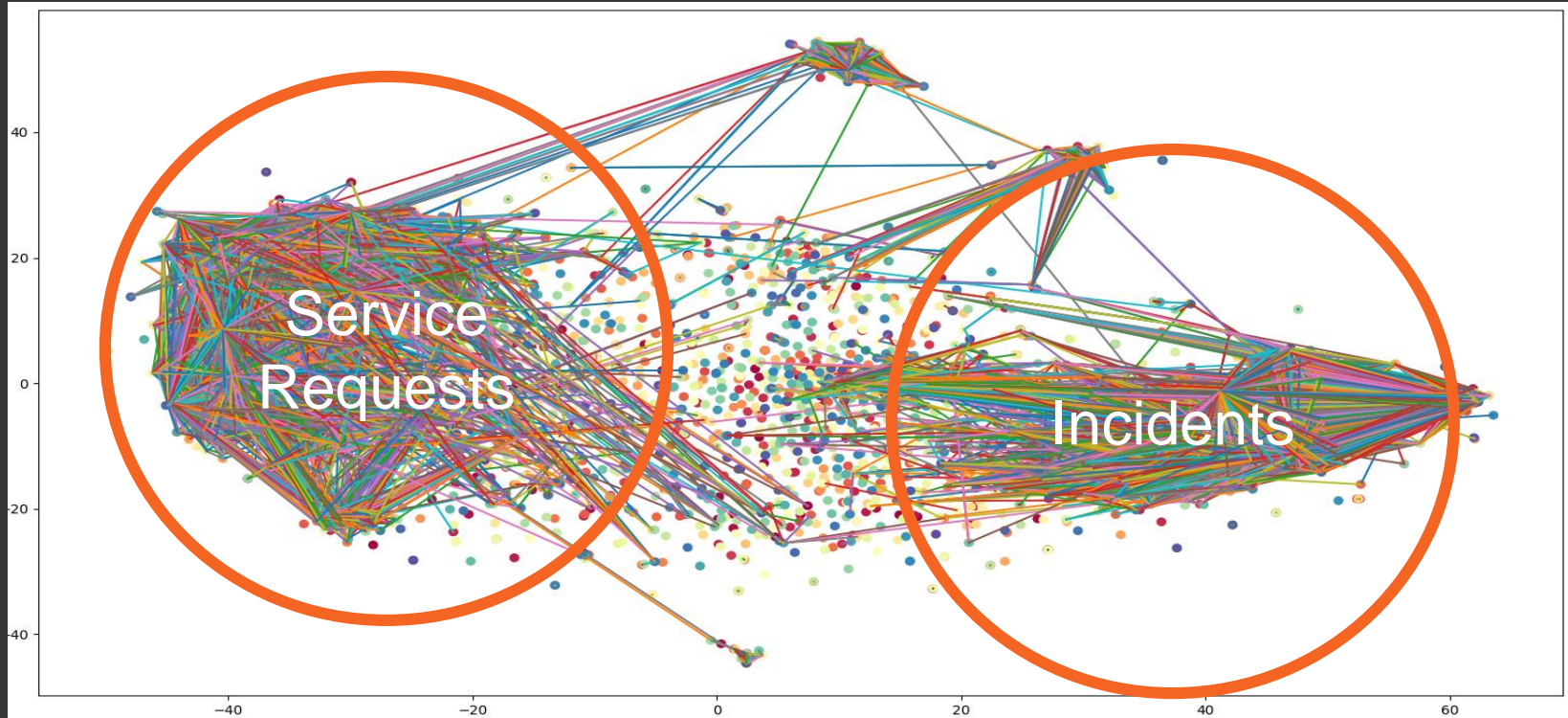
The Idea

After the training of the model, we use the output of one of the networks layers to receive an embedding (Doc2Vec), similar to a semantic vector.

Afterwards, we can calculate the similarity between 2 inputs using the cosine distance of 2 vectors in an N-dimensional vector space.

Doc2Vec Representation (TSNE)

Alpha-Clustering Algorithmus based upon Cosine-Similarity of vectors





Second application

After the plausible results of our first application we received further data to be analysed. We developed further models for the prediction of different types of classification.

Procedure Type:

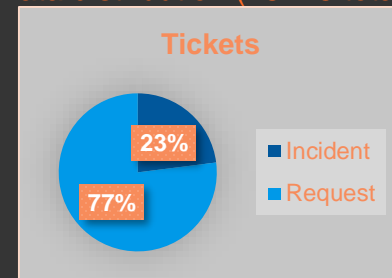
ServiceRequest vs Incident, type described and researched in the first application.

Request type: More detailed classification type, that contains a total of 12 classes such as *VpnConnection, Mail, Exchange, Accessories, etc.*

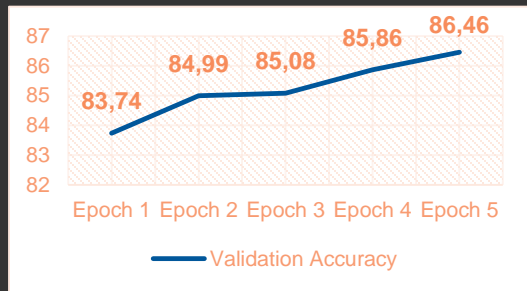


— Procedure type: Incident vs Request

Data distribution (48775 total)

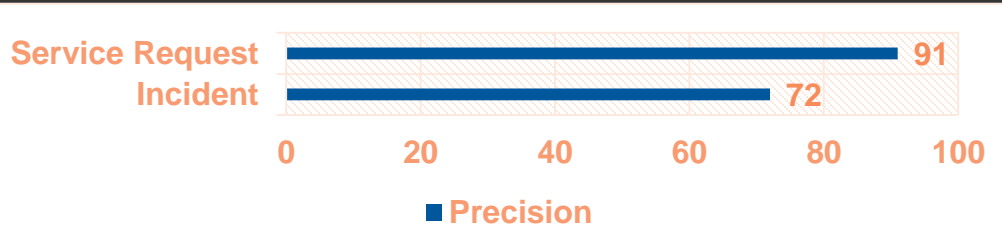


▶ Validation Accuracy: 86.66%



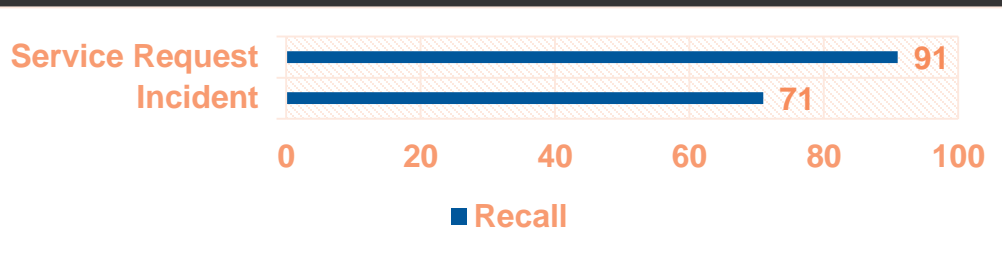
▶ Precision

- "How many selected items where relevant?"

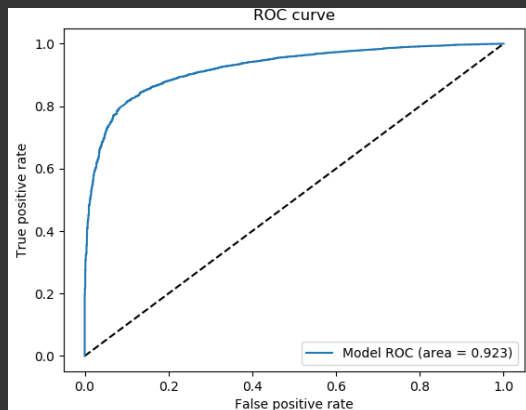


▶ Recall

- "How many relevant items where selected?"

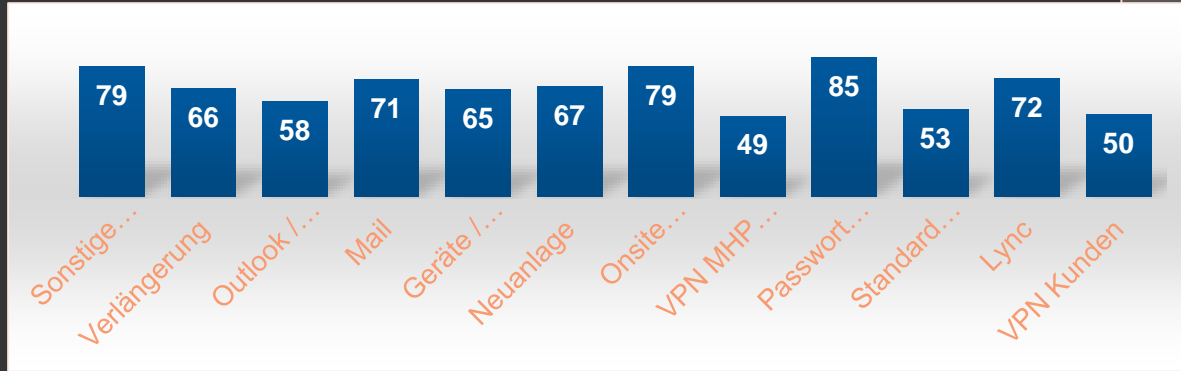


▶ Area under the curve: 0.923

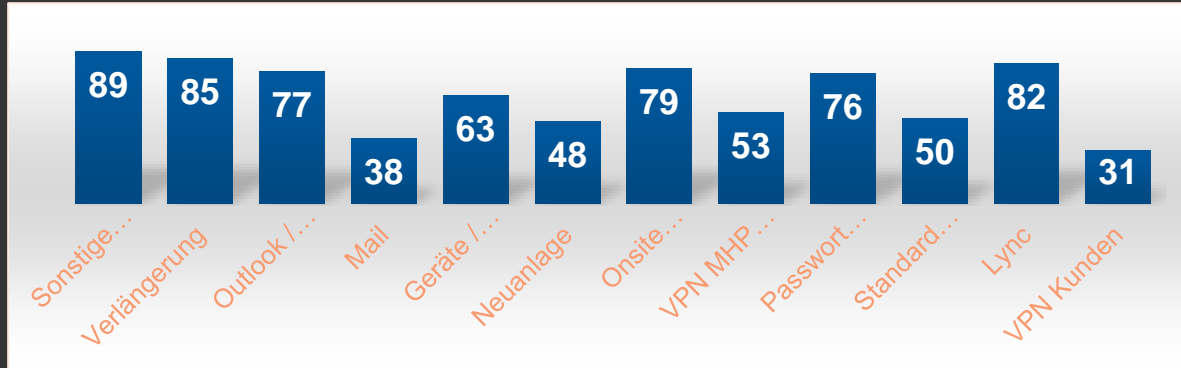


Request Type

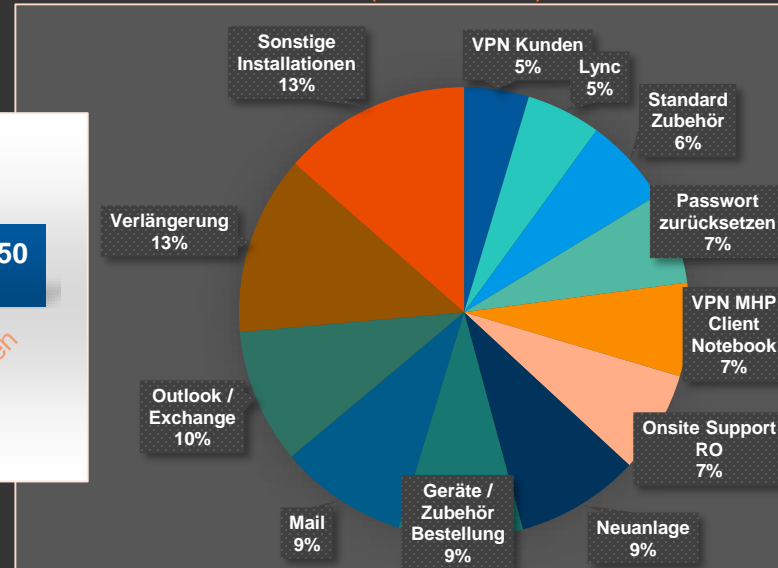
► Precision: "How many selected items were relevant?"



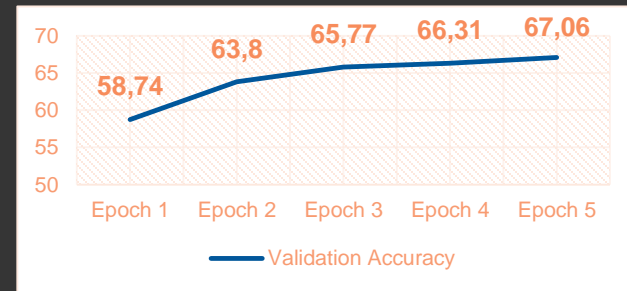
► Recall "How many relevant items were selected?"



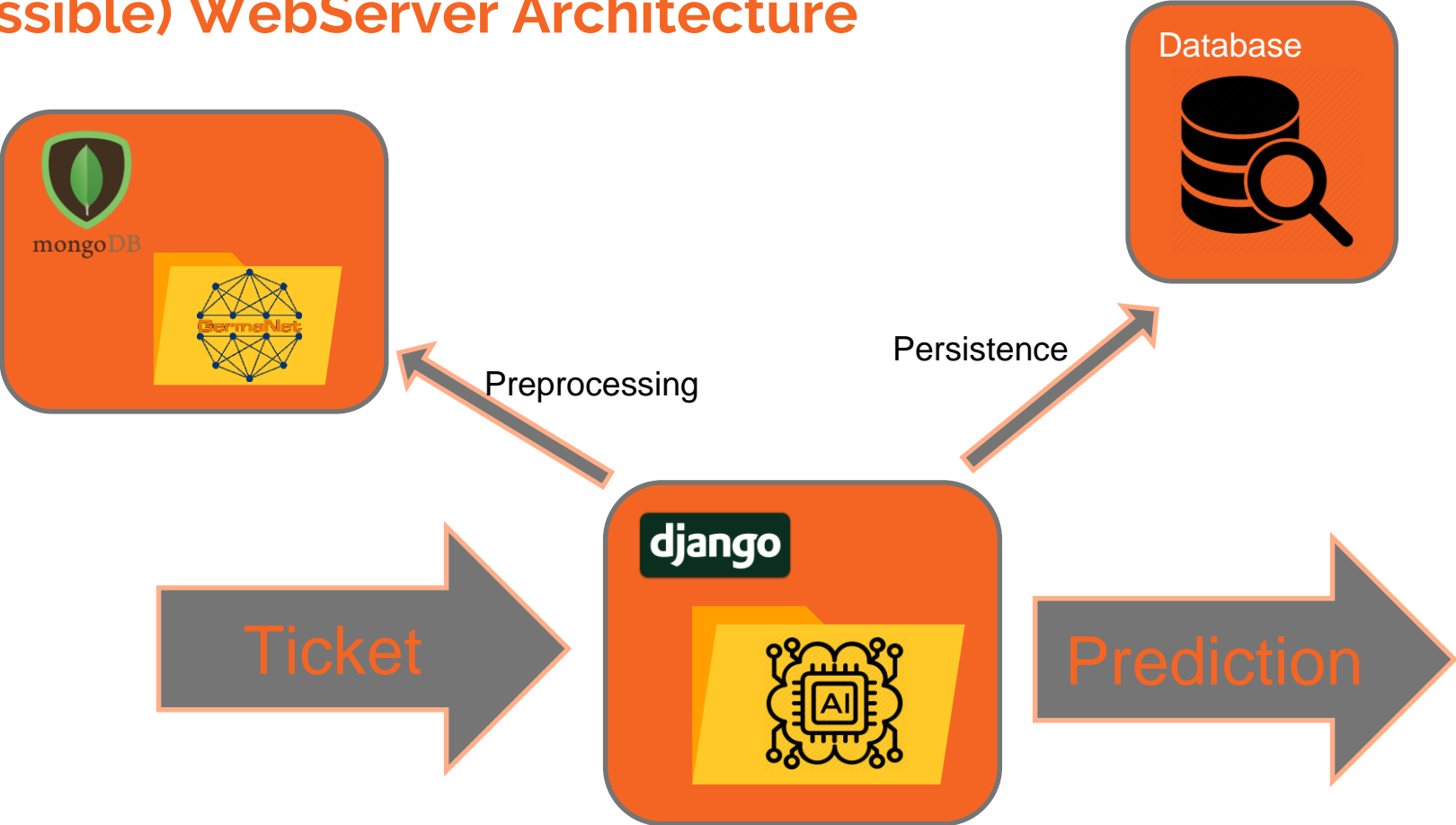
► Data distribution (22134 total)



► Validation Accuracy: 67.06%



(Possible) WebServer Architecture



Next steps



**Thank you for your
attention!**
