Utilizando Python Para Salvar Vidas: O Uso de NLP Para Melhorar o Atendimento de Emergências Médicas

Palestrantes:
Prof. Dr. Wagner de Lara Machado
Dalton Breno Costa

Partnerships

Duke
UNIVERSITY

AD VERVM DVCIT
PUCRS
Pontifícia Universidade Católica do Rio Grande do Sul

GRUPO DE PESQUISA
AVALIAÇÃO EM BEM-ESTAR E SAÚDE MENTAL

João Vissoci

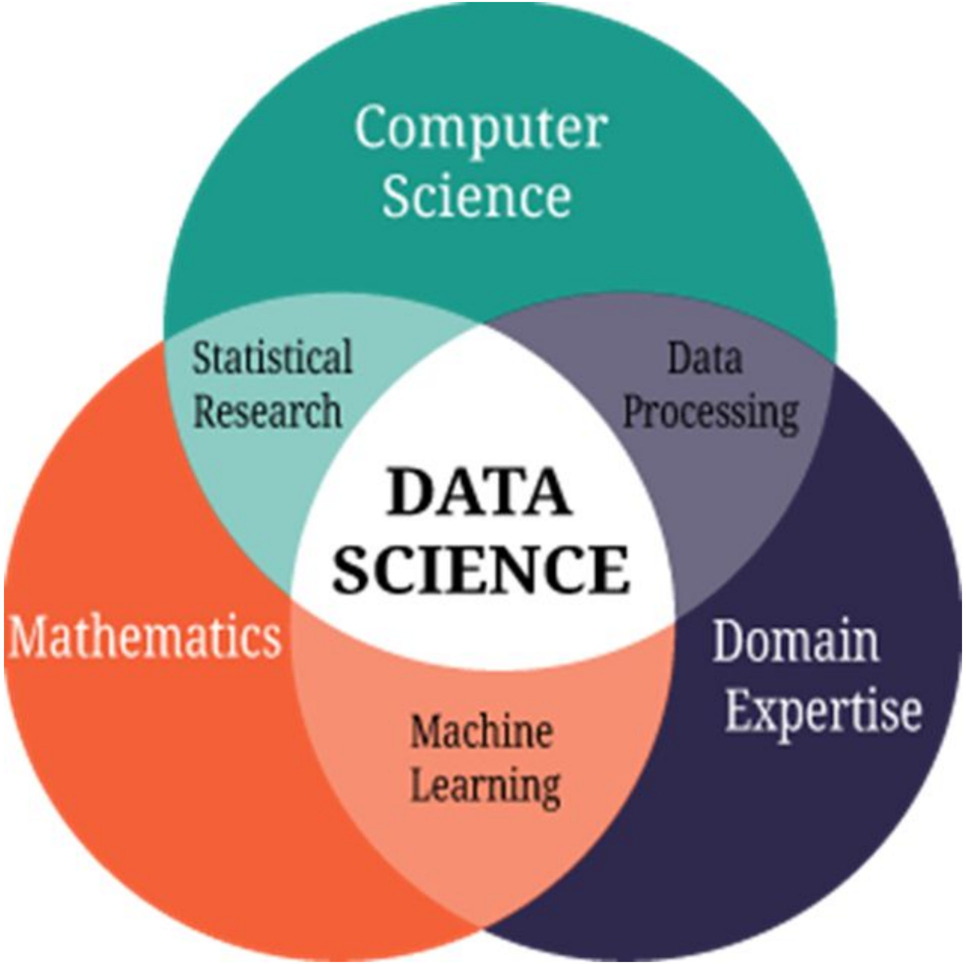
Pesquisador na divisão de Emergency Medicine do departamento de Cirurgia, e na divisão Duke Global Neurosurgery and Neuroscience (DGNN) do departamento de Neurocirurgia, na Duke University.

Wagner de Lara Machado

Psicólogo pela ULBRA , Mestre e Doutor em Psicologia pela UFRGS, realizou estágio de Pós-doutorado na UFRGS, professor do PPG Psicologia da PUCRS e coordenador do grupo de pesquisa Avaliação em Bem-estar e Saúde Mental (ABES - PUCRS).

Dalton Breno Costa

Estudante de Psicologia na UFCSPA, realizou estágio em psicometria na Université de Moncton (Canadá) e atualmente é bolsista de Iniciação Científica do grupo de pesquisa Avaliação, Reabilitação e Interação Humano-animal (ARIHA - PUCRS).


Computer Science
Statistical Research
Data Processing
DATA SCIENCE
Mathematics
Domain Expertise
Machine Learning

Fernandes *et al. BMC Medicine* (2017) 15:80
DOI 10.1186/s12916-017-0849-x

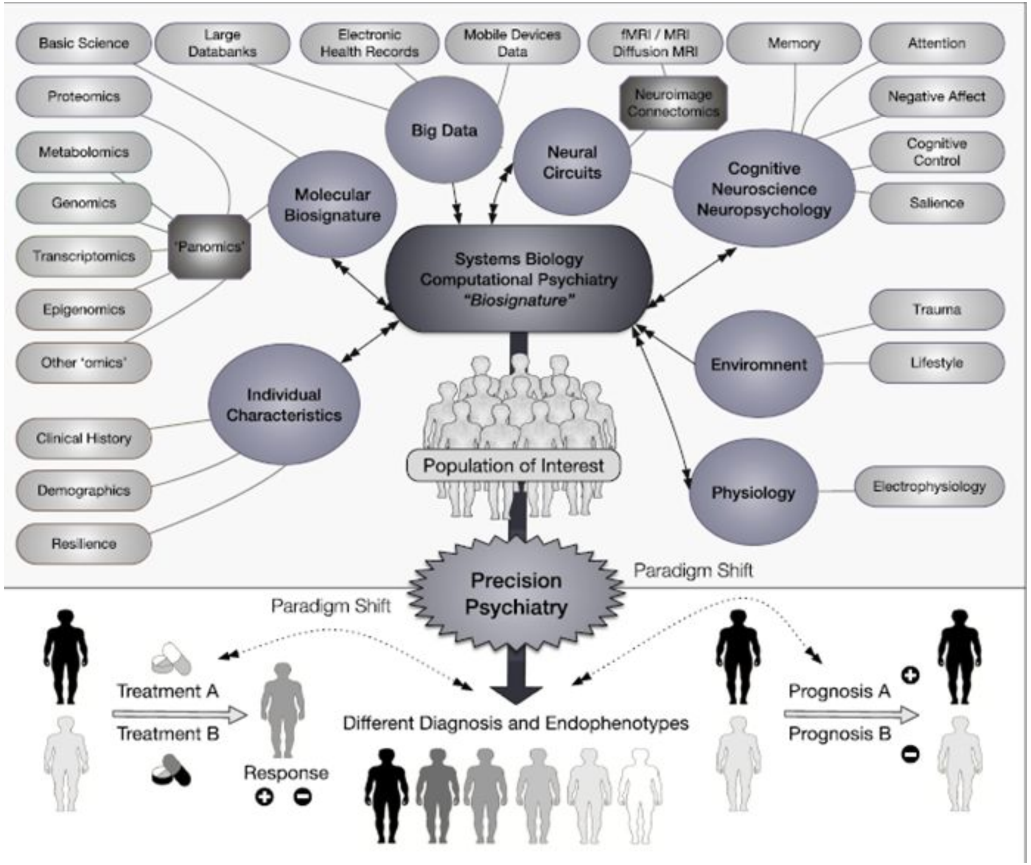
BMC Medicine

OPINION

Open Access

The new field of 'precision psychiatry'

Brisa S. Fernandes[1,2,3*], Leanne M. Williams[4,5], Johann Steiner[6], Marion Leboyer[7], André F. Carvalho[8] and Michael Berk[1,2,9,10]



Basic Science
Large Databanks
Electronic Health Records
Mobile Devices Data
fMRI / MRI Diffusion MRI
Memory
Attention
Proteomics
Negative Affect
Neuroimage Connectomics
Big Data
Metabolomics
Neural Circuits
Cognitive Control
Cognitive Neuroscience Neuropsychology
Molecular Biosignature
Genomics
Salience
'Panomics'
Transcriptomics
Systems Biology
Computational Psychiatry
"Biosignature"
Epigenomics
Trauma
Other 'omics'
Environment
Lifestyle
Individual Characteristics
Clinical History
Population of Interest
Physiology
Electrophysiology
Demographics
Resilience
Precision Psychiatry
Paradigm Shift
Paradigm Shift
Treatment A
Treatment B
Response
Different Diagnosis and Endophenotypes
Prognosis A
Prognosis B

Development of Standardized, Culturally Appropriate Prehospital Chief Complaints in eSwatini: First steps and analytical strategy

Prof. Dr. Wagner de Lara Machado, Prof. Dr. João Vissoci e Dalton Costa

Project Context

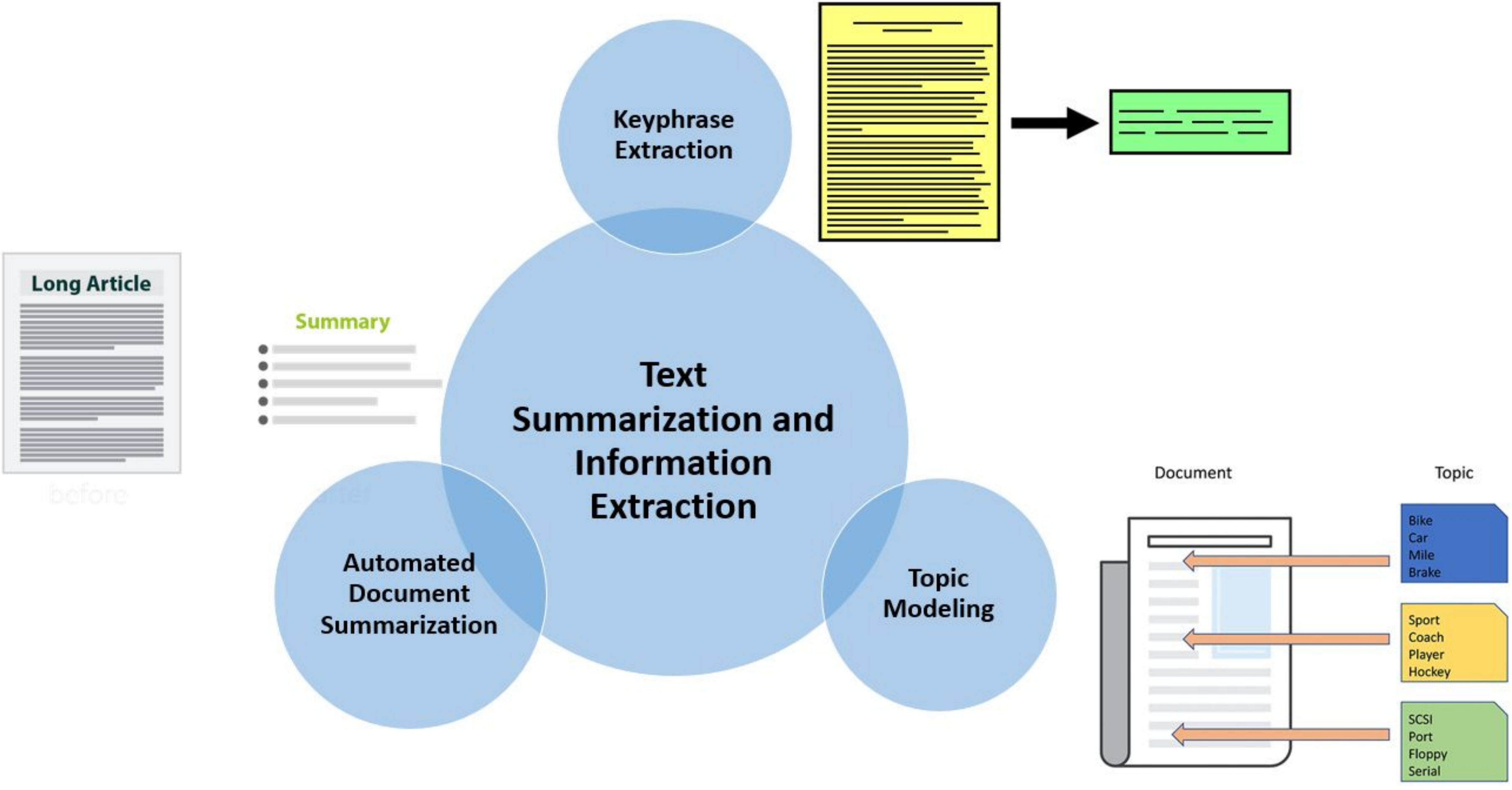
- Emergency conditions: death rates in LMICs.
- Emergency care systems: underdeveloped or non-existence.
- Lack and poor quality standardized patient data.
- Records: free-text form.
- Problems to obtain reliable and informative data.

Project aim

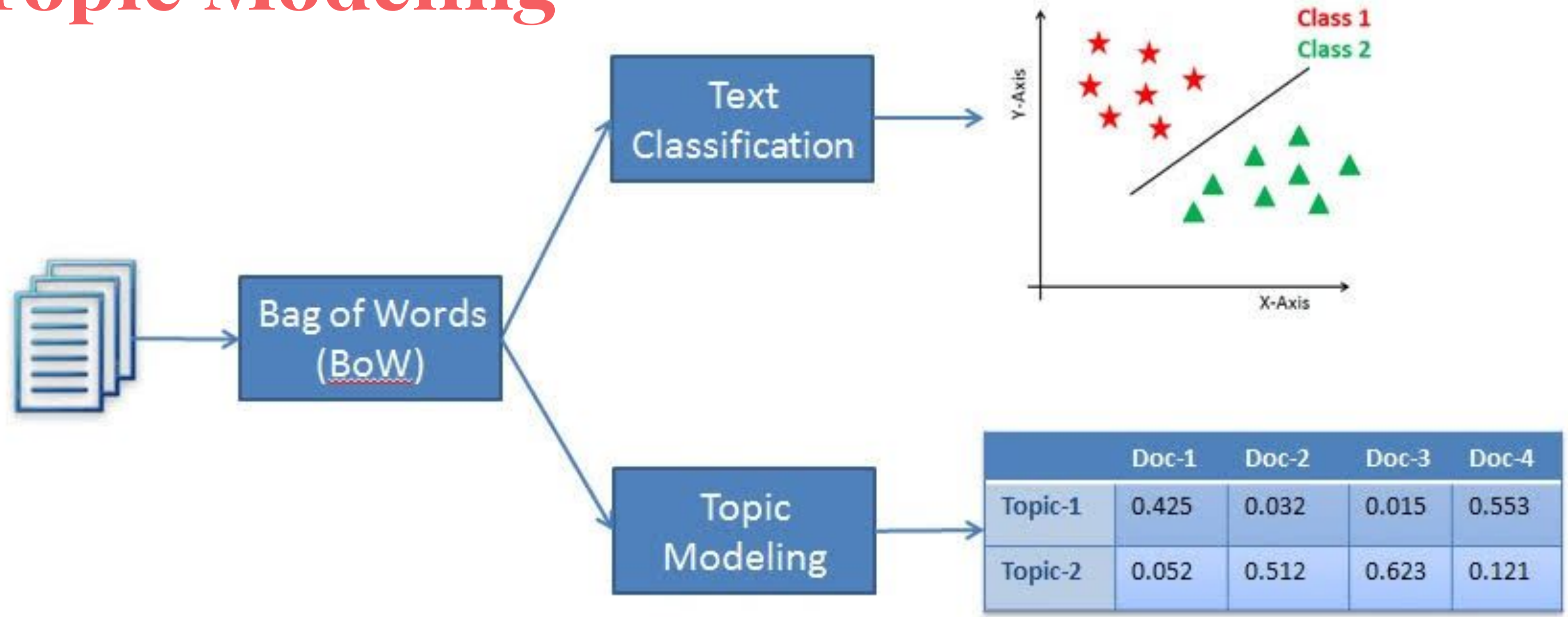
- Utilize natural language processing to develop and prospectively validate culturally-relevant chief complaint categories for use in prehospital services in the country of eSwatini (formerly Swaziland).
- Based on pre-existing emergency call center data.

Project Objectives

- Improve: Public Health and Emergency Health Care.
- Data driven policies and interventions.


Keyphrase Extraction
Long Article
Summary
Text Summarization and Information Extraction
Automated Document Summarization
Topic Modeling
Document
Topic
Bike
Car
Mile
Brake
Sport
Coach
Player
Hockey
SCSI
Port
Floppy
Serial

Topic Modeling



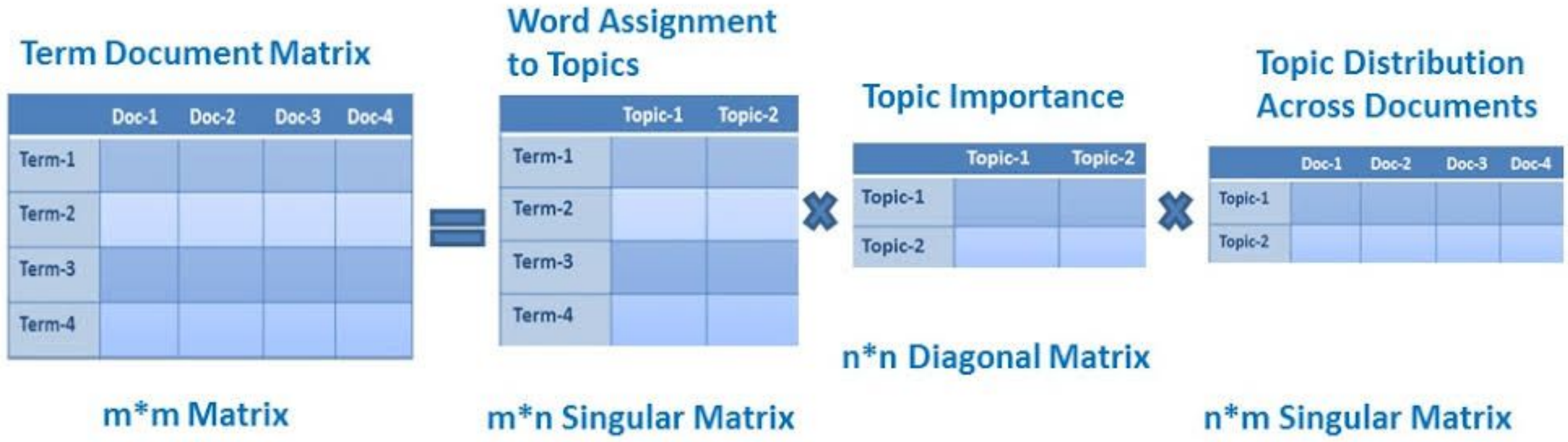
	Doc-1	Doc-2	Doc-3	Doc-4
Topic-1	0.425	0.032	0.015	0.553
Topic-2	0.052	0.512	0.623	0.121

Topic Modeling

- There are several algorithms for creating Topic Modeling:
  - **Scikit-Learn Library:**
    - Latent Semantic Indexing (LSI)
    - Latent Dirichlet Allocation (LDA)
    - Non-negative Matrix Factorization (NMF)
  - **Gensim Library**:
    - Latent Semantic Indexing (LSI)
    - Latent Dirichlet Allocation (LDA)
    - Hierarchical Dirichlet Process (HDP)

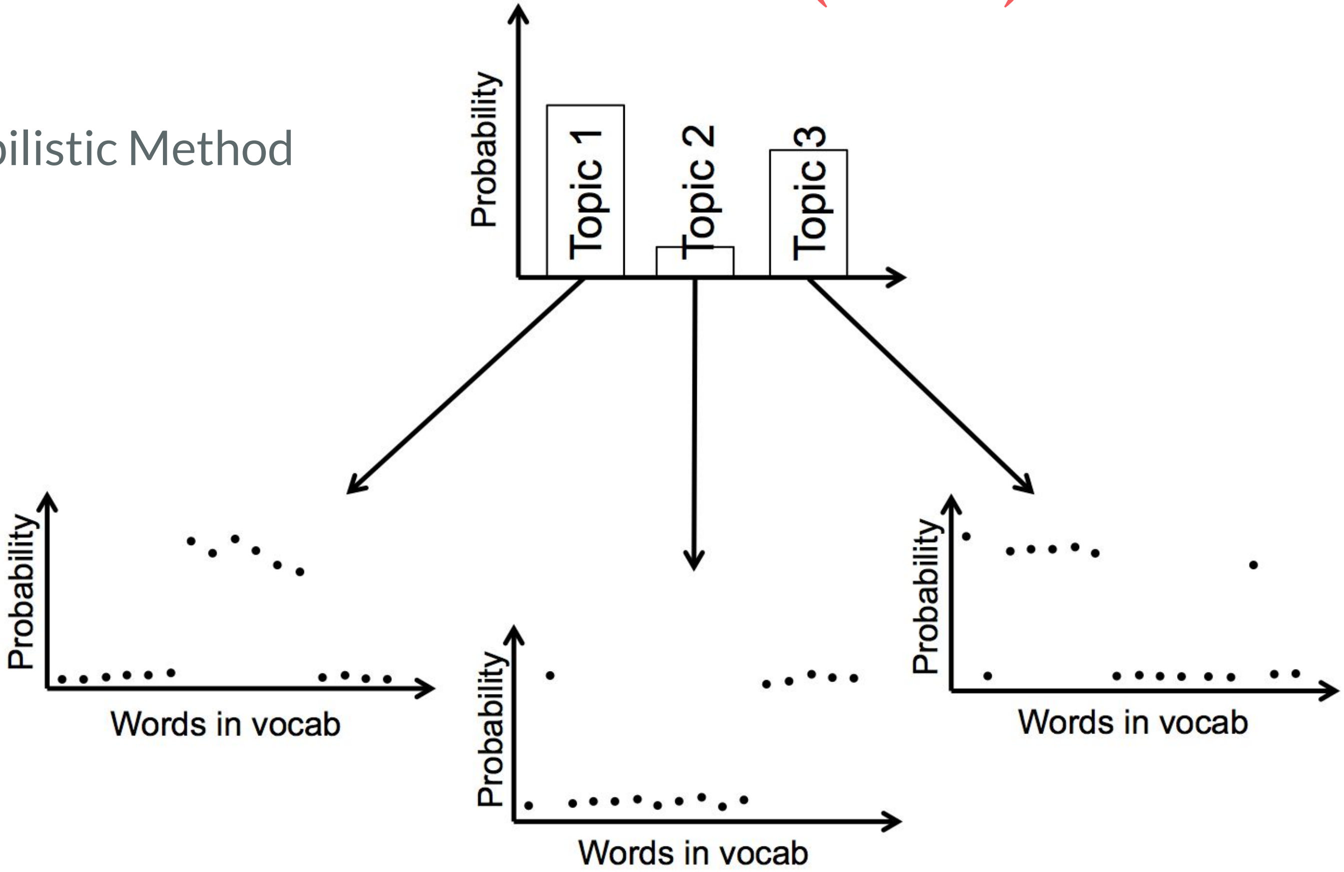
Latent Semantic Indexing (LSI)

- LSI is also known as **Latent Semantic Analysis (LSA)**.
- LSI is based on the principle that words that are used in the same contexts tend to have similar meanings.



Latent Dirichlet Allocation (LDA)

- Probabilistic Method

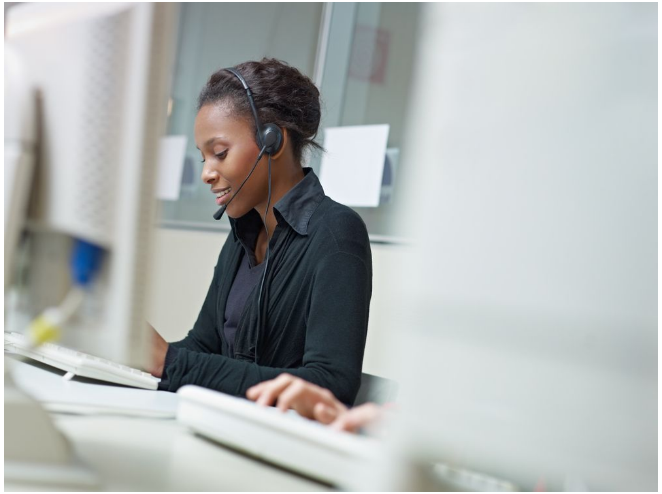


The aim of the experiment...

- Create and compare two Topic Modeling models (LSI and LDA) from the Gensim library.
- Interpret the categories created.
- Create and compare two neural network models using the Keras library.

Data

- The data comes from a medical emergency call center in Eswatini, Africa.
- The tests were conducted using data from 2017.

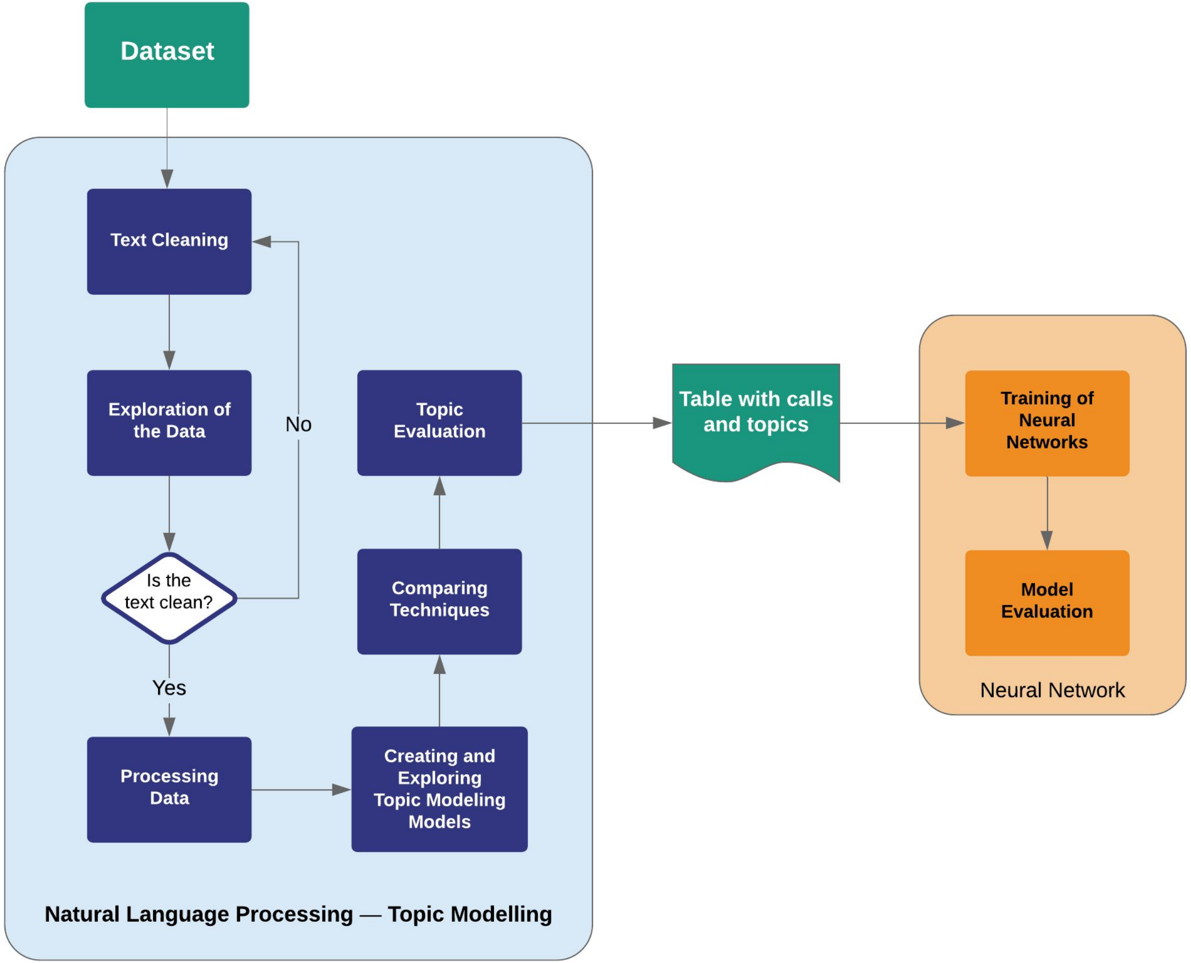


Year	2014/2015	2015/2016	2017
n	40,091	72,859	29,416

Record Example

J	K	L	
allType	location	cond	callStatus
Primary Call	Moyeni	Speech disturbance, right hemi paralysis	Dispatched
Primary Call	Manzini	abdomina pain and vomiting since yesterday	Dispatched
Primary Call	Mahlanya	coughing hematemisis weak HIV positive but not on art CD$ count is 648	Dispatched
Primary Call	Mbabane	collapsed conscious sweating fever been to Maputo	Not Dispatched
Primary Call	Mayiwane	diabetic patient unconscious	Dispatched
Primary Call	Zulwini	maternal case Primi Gravida Labour Pains Discgarging Fluids full term	Not Dispatched
Primary Call	Moneni	no injuries but transported to hospital	Dispatched
Primary Call	Mavalela	maternal case primi gravida full term EDD 26 january labour pains contractions 5minut	Dispatched
Primary Call	Lomshiyo	abdominal pains diarrhoea with blood weak not ambulant anorexia since yesterday sl	Dispatched
Primary Call	Rockland	loss of appetite anorexia pedal eadema and swollen knees not ambulant and weak	Dispatched
Primary Call	Nkoyoyo	HIV positive since 2012 loss of strength anorexia not yet on art	Dispatched
Primary Call	Nkoyoyo	maternal case labour pains since 0100hrs para 2 gravida 2 contractions of interval of 5	Dispatched
Primary Call	Sibane Hotel	epleptic pt diabetic patient collapsed and conscious	Dispatched

Method


Dataset
Text Cleaning
Exploration of the Data
No
Is the text clean?
Yes
Processing Data
Creating and Exploring Topic Modeling Models
Comparing Techniques
Topic Evaluation
Natural Language Processing — Topic Modelling
Table with calls and topics
Training of Neural Networks
Model Evaluation
Neural Network

Step 1 - Text Clearing

Spell Correction

```
>>>from textblob import TextBlob

>>>b = TextBlob("I havv goood speling!")

>>>print(b.correct())
```

I have good spelling!

+600 substituições manuais de palavras

```
>>> text = text.str.replace('abdominalapains', 'abdominal pain')
```

Step 1 - Text Clearing

- All words were replaced to **lowercase**.
- All **numbers, punctuation, line breaks, and whitespace** were removed.
- **Stopwords** were removed using the NLTK library.

Step 1 - Text Clearing

- **Stemming** : process that consists in normalizing the words to their root.

  Example: "likes", "liked", "likely", "liking" → like

```
from nltk.stem import PorterStemmer
def stemmer(text):
 st = PorterStemmer()
 text = text.apply(lambda x: " ".join([st.stem(word) for word in x.split()]))
 return(text)
```

- Calls composed of **three or fewer** words were excluded.

```
def remove_short_sentences(text):
 return(pd.Series(map(lambda x: x[1], filter(lambda x: (len(x[1].split(" ")) > 3), text.iteritems()))))
```

Step 2 - Exploration of the Data


```python
from wordcloud import WordCloud
def show_wordcloud(data, title = None):
    wordcloud = WordCloud(background_color='white', max_words=200,
    max_font_size=40, scale=3, random_state=1).generate(str(data))
```

Cleaning Results

Original:
29.403 calls
383.522 words

-22,40 %

-56,12 %

After cleaning:
22.815 calls
168.271 words

Step 3 - Preprocessing Data

- **Word Tokenization**: procedure of dividing a sentence into pieces, each piece is called a Token. Example:
    - input: "ower abdomin pain sport blood weak dizzi pas urin onset"
    - output: ['lower', 'abdomin', 'pain', 'sport', 'blood', 'weak', 'dizzi', 'pas', 'urin', 'onset']

```
from nltk.tokenize import RegexpTokenizer
def preprocess_data(doc_set):
    tokenizer = RegexpTokenizer(r'\w+')
        # list for tokenized documents in loop
    texts = []
    # loop through document list
    for i in doc_set:
        tokens = tokenizer.tokenize(i)
        # add tokens to list
        texts.append(tokens)
    return texts
```

Step 3 - Preprocessing Data with the Gensim Library

- **Bigrams and Trigrams**: process that joins words that are composed or has a better meaning together. Bigrams are compositions of two words and trigrams of three words. Example:
    - “difficult” and “breath” → difficult_breath

```
from gensim.models import Phrases
bigram = Phrases(Phrase_Token, min_count=30, threshold=15)
for idx in range(len(Phrase_Token)):
    for token in bigram[Phrase_Token[idx]]:
        if '_' in token:
            Phrase_Token[idx].append(token)
trigram = Phrases(bigram[Phrase_Token], min_count=30, threshold=15)
for idx in range(len(bigram[Phrase_Token])):
    for token in trigram[bigram[Phrase_Token][idx]]:
        if '_' in token:
            bigram[Phrase_Token][idx].append(token)
```

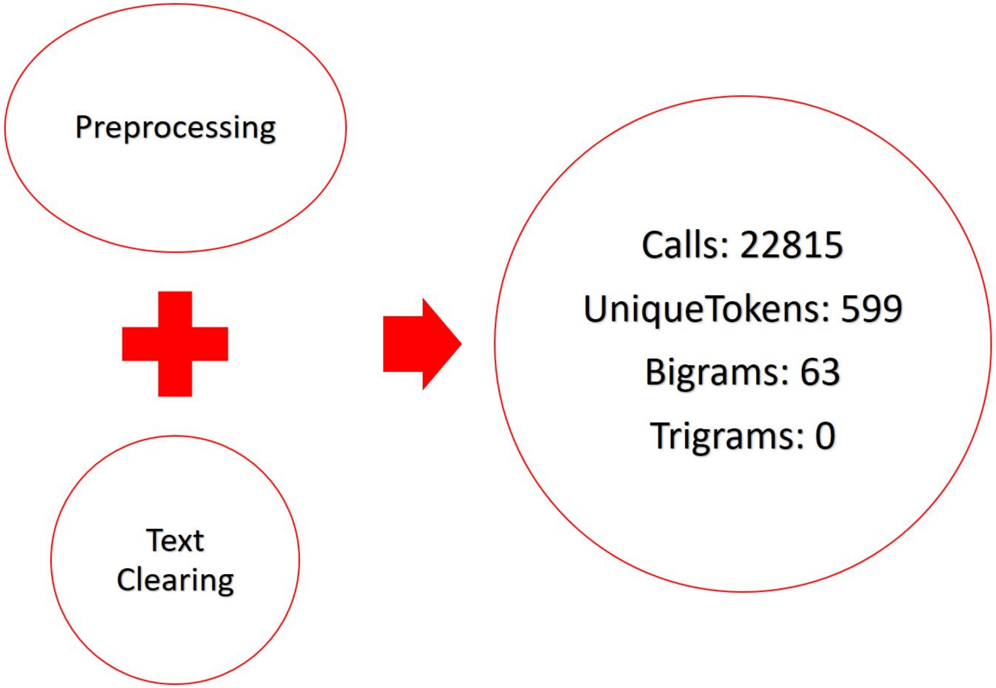
Step 3 - Preprocessing Data with the Gensim Library

- Removal of tokens that are **very frequent or very rare.**

- Creation of the **Corpus or Bag of Words (BoW).**

	it	is	puppy	cat	pen	a	this
it is a puppy	1	1	1	0	0	1	0
it is a kitten	1	1	0	0	0	1	0
it is a cat	1	1	0	1	0	1	0
that is a dog and this is a pen	0	2	0	0	1	2	1
it is a matrix	1	1	0	0	0	1	0

Final Corpus



Step 4 - Creating and Exploring Topic Modeling Models

- To generate models with Gensim it is necessary to provide the **Numbers of Topics and Corpus**.

```
from gensim.models import LdaModel, LsiModel
```

```
lsimodel = LsiModel(corpus=corpus, num_topics=2, id2word=dictionary)
ldamodel = LdaModel(corpus=corpus, num_topics=9, id2word=dictionary)
```

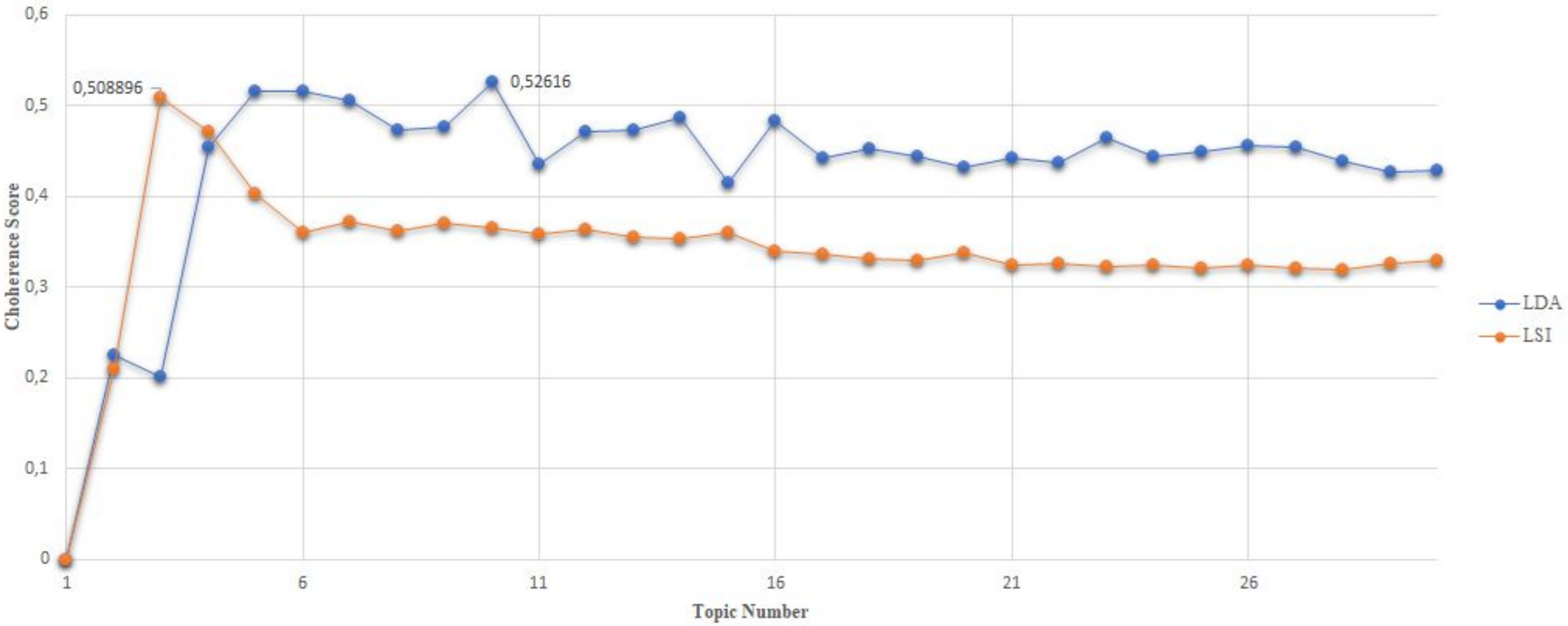
- What is the optimal number of topics?
    - **Topic Coherence**: measures that provide the degree of semantic similarity between words and topic.

Step 4 - Creating and Exploring Topic Modeling Models

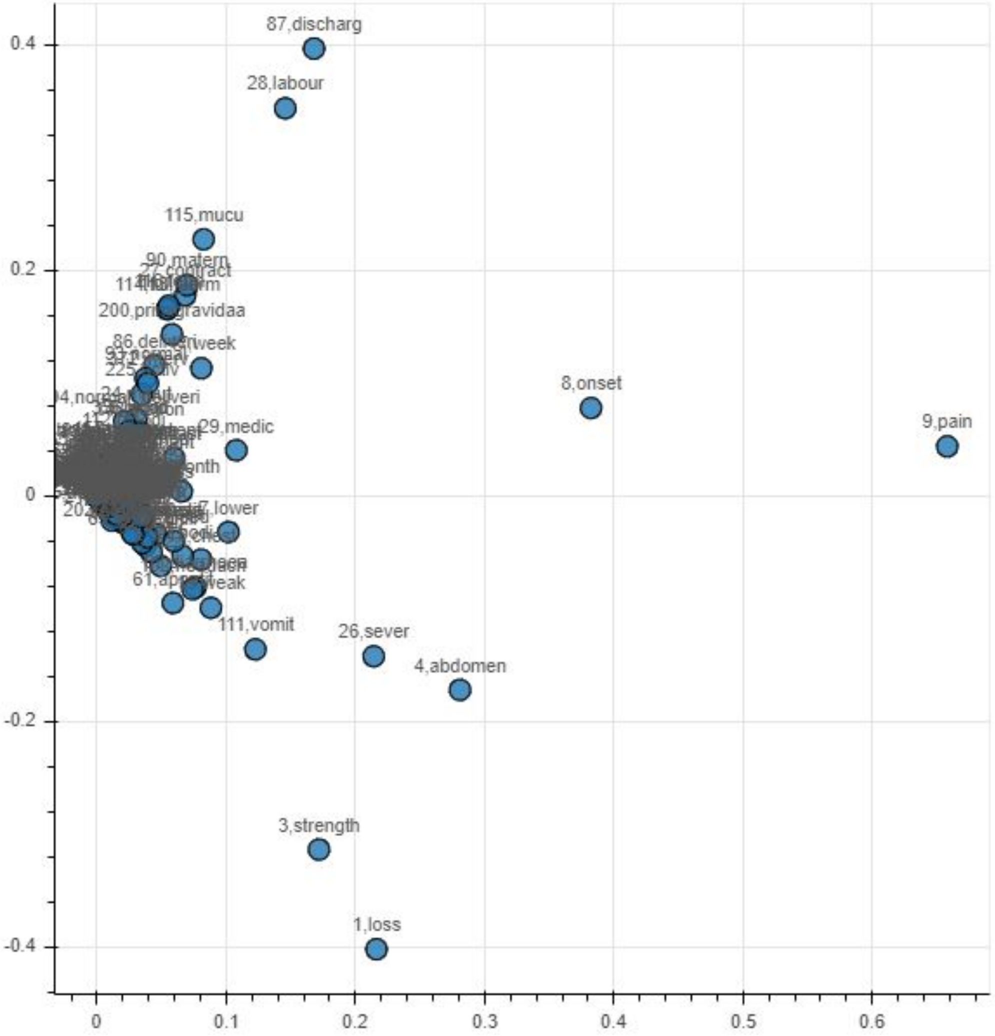
- Topic Coherence Exploration

```
from gensim.models import LsiModel
from gensim.models.coherencemodel import CoherenceModel
def Find_N_Topic_LSA(dictionary, corpus, texts, limit):
   c_v = []
   lm_list = []
   for num_topics in range(1, limit):
       lm = LsiModel(corpus=corpus, num_topics=num_topics, id2word=dictionary)
       lm_list.append(lm)
       cm = CoherenceModel(model=lm, texts=texts, dictionary=dictionary, coherence='c_v')
       c_v.append(cm.get_coherence())
   return lm_list, c_v
```

Topic Coherence Exploration



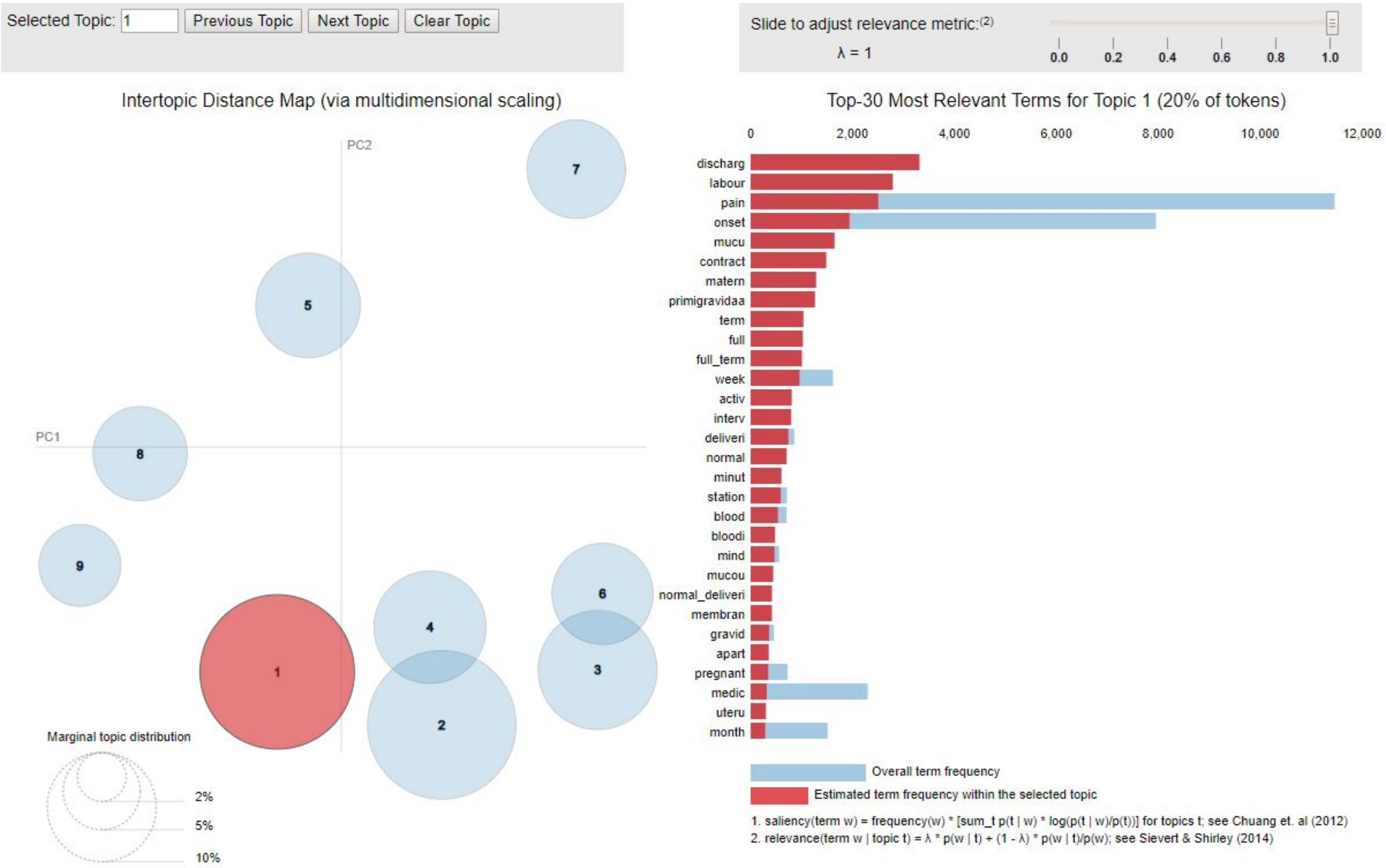
LSI


87,discharg
28,labour
115,mucu
200,primigravidaa
29,medic
8,onset
9,pain
7,lower
111,vomit
26,sever
4,abdomen
3,strength
1,loss
0.4
0.2
0
-0.2
-0.4
0
0.1
0.2
0.3
0.4
0.5
0.6

Visualização do Modelo de LDA

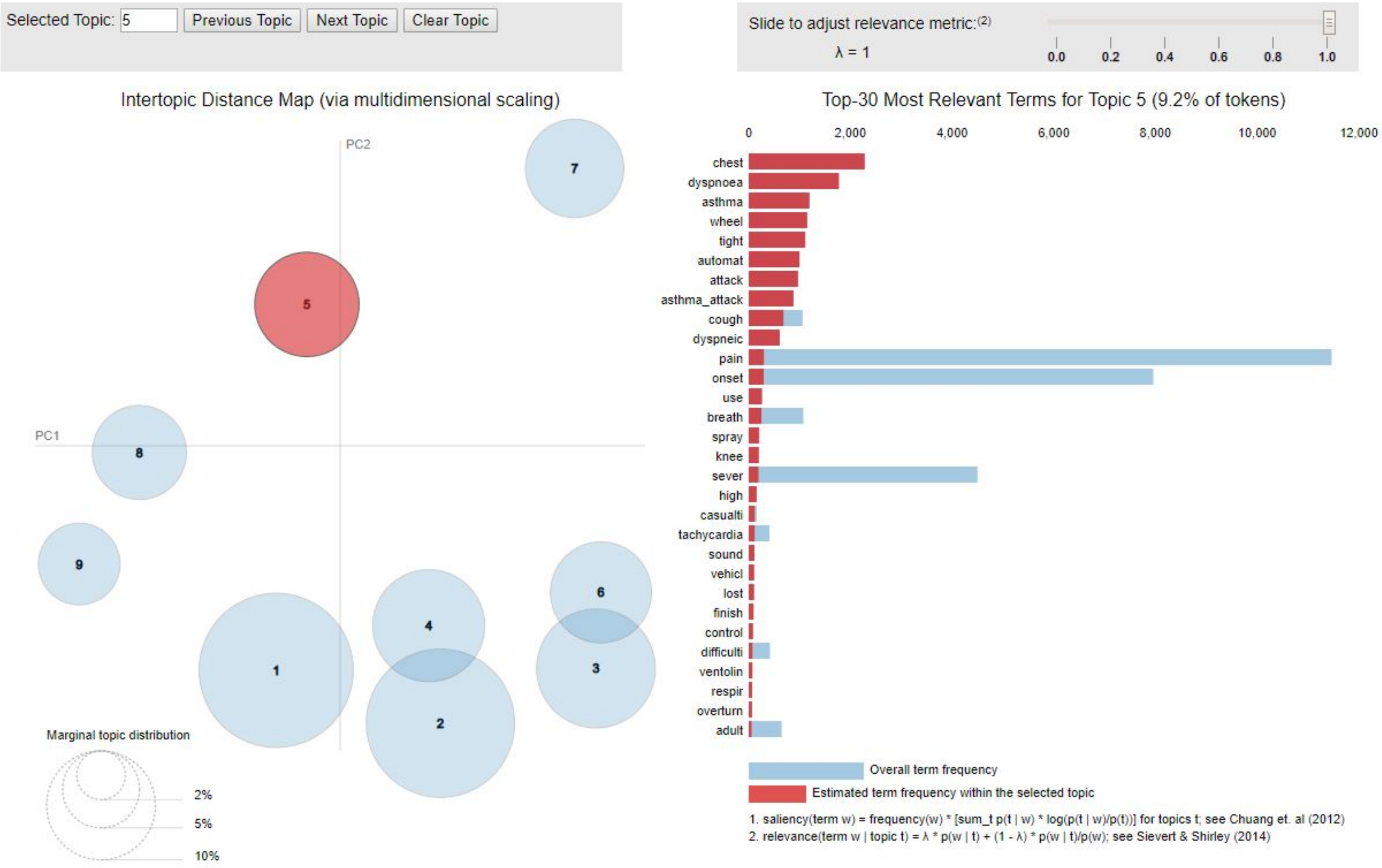
https://daltonbc96.github.io/vis.html

LDA

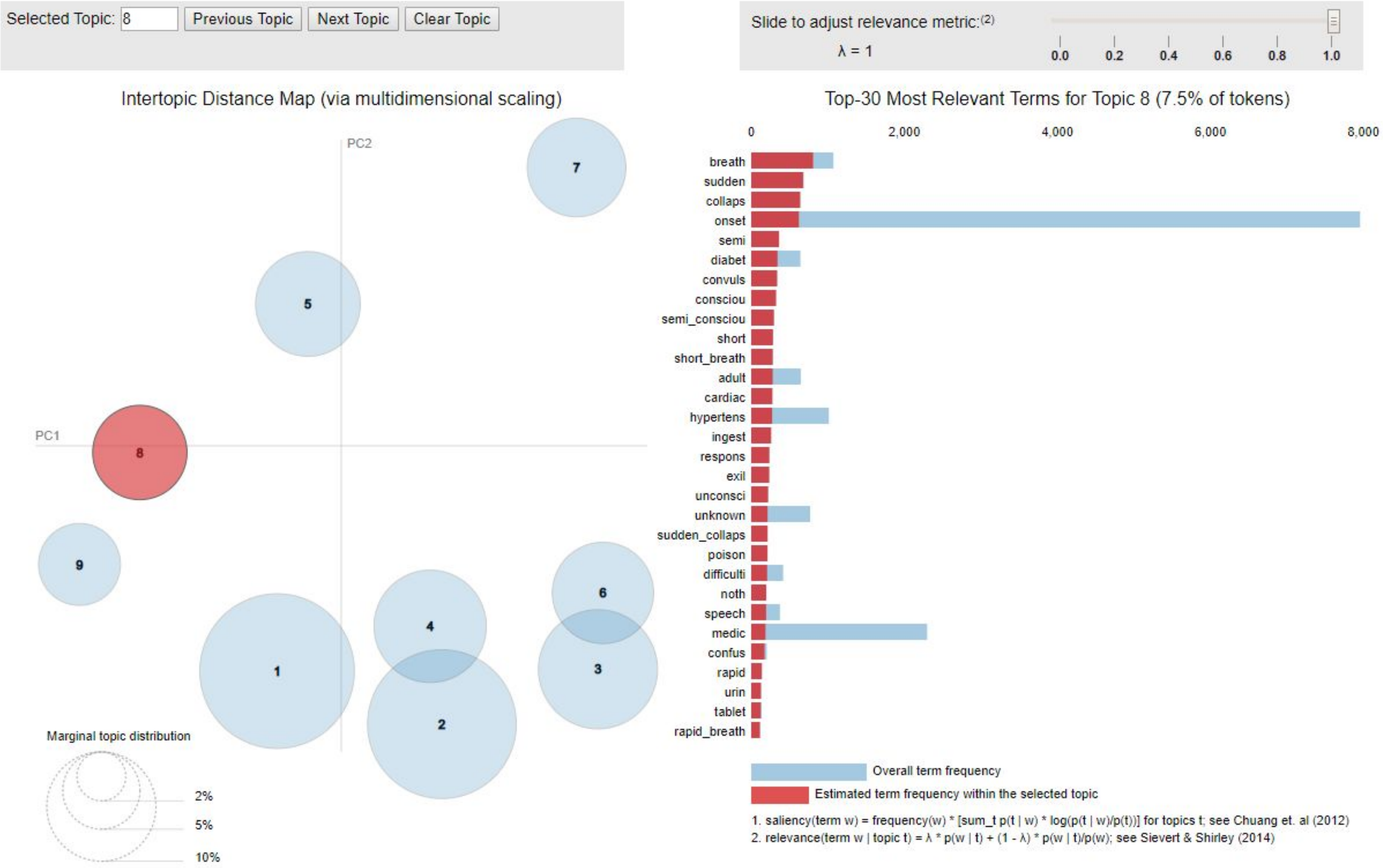


Selected Topic: 1
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
1
2
3
4
5
6
7
8
9
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 1 (20% of tokens)
0
2,000
4,000
6,000
8,000
10,000
12,000
discharg
labour
pain
onset
mucu
contract
matern
primigravidaa
term
full
full_term
week
activ
interv
deliveri
normal
minut
station
blood
bloodi
mind
mucou
normal_deliveri
membran
gravid
apart
pregnant
medic
uteru
month
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

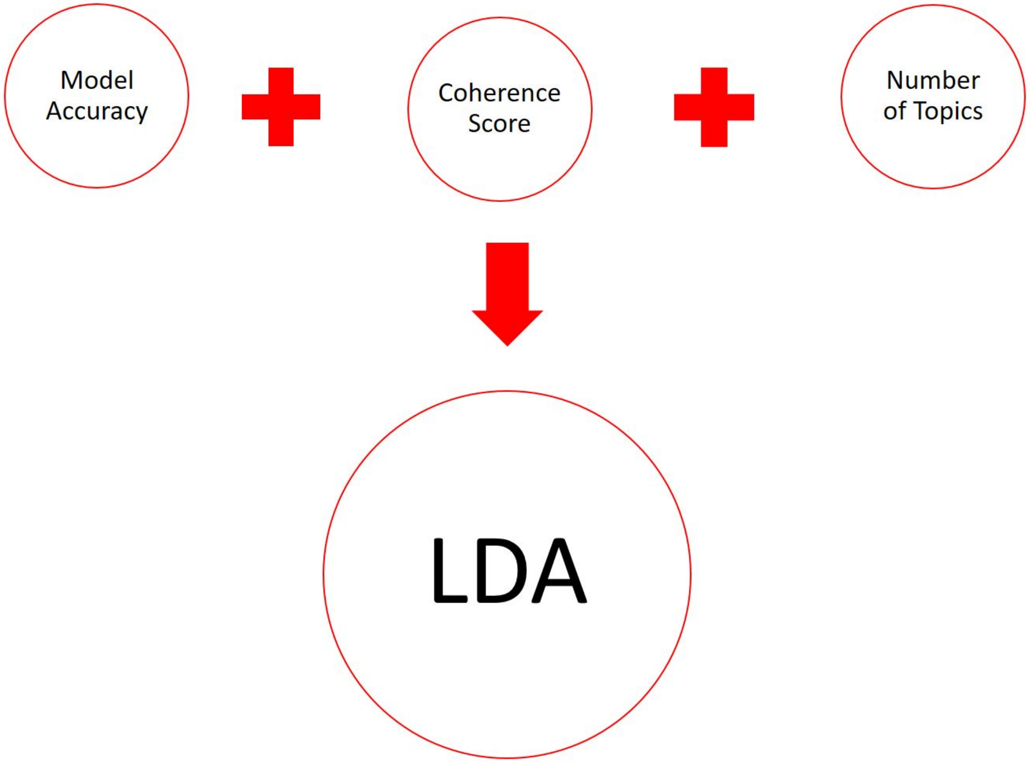
LDA


Selected Topic: 5
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
7
5
8
9
1
4
2
6
3
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 5 (9.2% of tokens)
0
2,000
4,000
6,000
8,000
10,000
12,000
chest
dyspnoea
asthma
wheel
tight
automat
attack
asthma_attack
cough
dyspneic
pain
onset
use
breath
spray
knee
sever
high
casualti
tachycardia
sound
vehicl
lost
finish
control
difficulti
ventolin
respir
overturn
adult
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

LDA


Selected Topic: 8
Previous Topic
Next Topic
Clear Topic
Slide to adjust relevance metric:(2)
λ = 1
0.0
0.2
0.4
0.6
0.8
1.0
Intertopic Distance Map (via multidimensional scaling)
PC2
PC1
7
5
8
9
1
4
2
6
3
Marginal topic distribution
2%
5%
10%
Top-30 Most Relevant Terms for Topic 8 (7.5% of tokens)
0
2,000
4,000
6,000
8,000
breath
sudden
collaps
onset
semi
diabet
convuls
consciou
semi_consciou
short
short_breath
adult
cardiac
hypertens
ingest
respons
exil
unconsci
unknown
sudden_collaps
poison
difficulti
noth
speech
medic
confus
rapid
urin
tablet
rapid_breath
Overall term frequency
Estimated term frequency within the selected topic
1. saliency(term w) = frequency(w) * [sum_t p(t | w) * log(p(t | w)/p(t))] for topics t; see Chuang et. al (2012)
2. relevance(term w | topic t) = λ * p(w | t) + (1 - λ) * p(w | t)/p(w); see Sievert & Shirley (2014)

Step 5 - Comparing Techniques

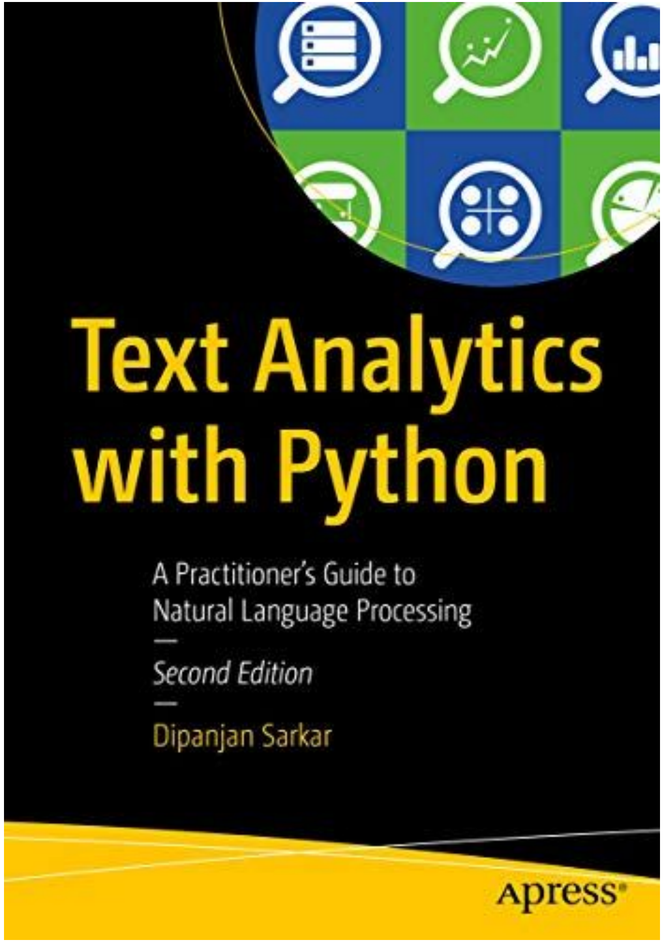


Step 6 - Topic Evaluation

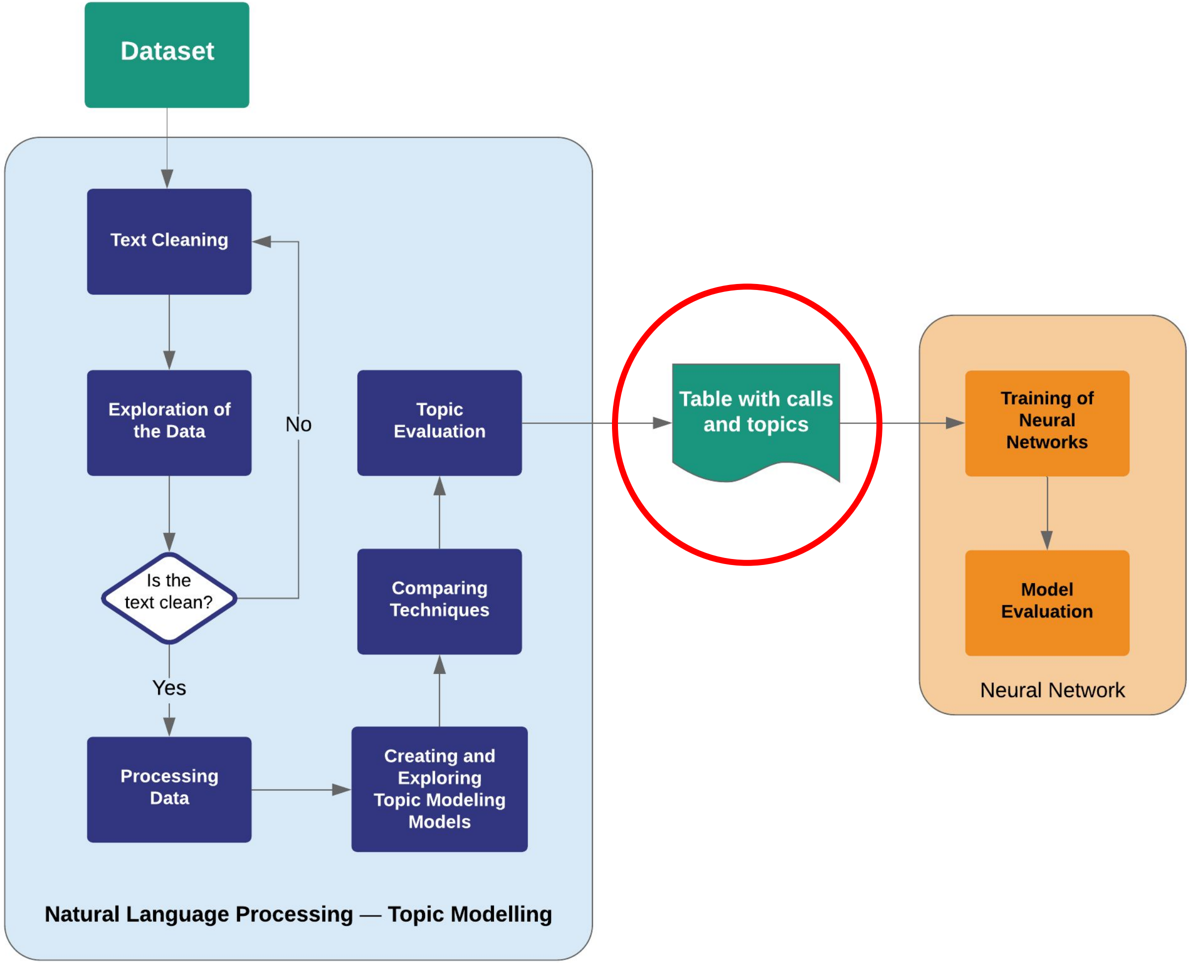
Topic	Description
1	Bleeding, emergencies with pregnant women, abortion, bleeding, injury or pain in the abdominal region
2	Severe pain, fractures, falls, limb displacement, swelling, edema
3	Traumas and head injuries, assault injuries, deep wounds, stab yeast, deep laceration
4	Dyspnea, asthma attack, tachycardia, chest pain, use of sprays, coughing
5	Start of labor, mucu, contractions, pain, bleeding, full term pregnancy
6	Severe headache, fever, weak body, dizziness, body pain
7	Loss of strength, vomiting, diarrhea, abdominal pain, poor appetite, ulcers
8	Complications of medication ingestion, postpartum complications and other medical emergencies
9	Sudden collapse, convulsions, semiconscious or unconscious, slow or rapid breathing, heart problems

Reading Indication

https://medium.com/ensina-ai/inteligencia-artificial-saude-psicologia-psiquiatria-23dbdbcb2e17



2019


Dataset
Text Cleaning
Exploration of the Data
Is the text clean?
No
Yes
Processing Data
Creating and Exploring Topic Modeling Models
Comparing Techniques
Topic Evaluation
Natural Language Processing — Topic Modelling
Table with calls and topics
Training of Neural Networks
Model Evaluation
Neural Network

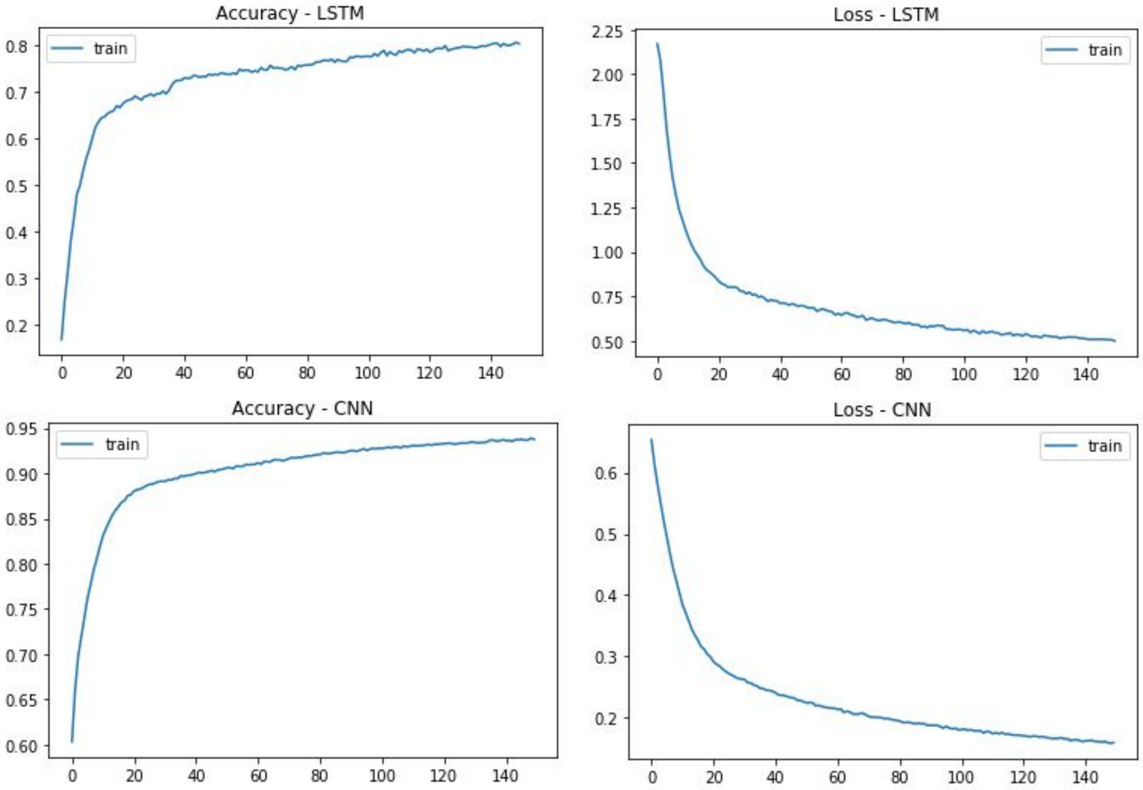
Step 7 - Training of Neural Networks with the Keras library

- **Word Tokenization** with the Keras library.

- Training and Test Data
  - Training: 15970 calls (70%)
  - Test: 6845 calls (30%)

- Neural Networks:
  - Long Short Term Memory (LSTM)
  - Convolutional Neural Network (CNN)

Step 7 - Training of Neural Networks



LSTM

Accuracy : 0.887

Loss: 0.650

CNN

Accuracy : 0.957

Loss: 0.140

Conclusion

- LDA was coherent and effective in recovering the structure of medical emergency data. CNN had the best performance in classifying emergency data.
- This categorization system may enable the specialization of emergency services.
- More insights and information about emergencies.

Next Steps

- Improve the unsupervised model and then invest in supervised learning.
- Correlate the learned model to the clinical outcome.
- Improve records of emergency calls (quality of writing).

Obrigado!

Utilizando Python Para Salvar Vidas:

O Uso de NLP Para Melhorar o Atendimento de Emergências Médicas

Prof. Dr. Wagner de Lara Machado
wag.lm.psico@gmail.com

Dalton Breno Costa
dalton.bc96@gmail.com