Text Extraction with POS-Tagging and Word Embeddings for support tickets automation

Thesis Project for the MSc. Data Science

Student: Pol
Ribó Supervisor: Simone
Scardapane

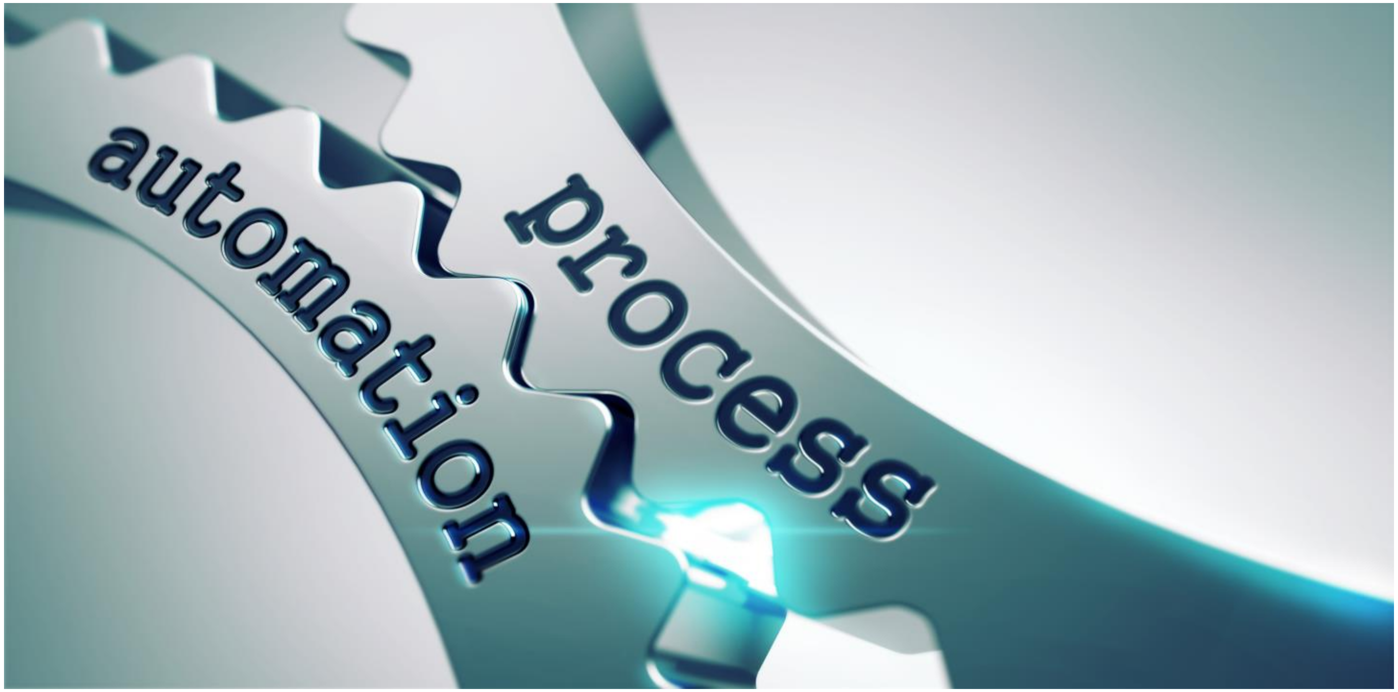
Student Number: 1840853
External Supervisor: Alberto Massidda

SAPIENZA
UNIVERSITÀ DI ROMA

Business Process Automation

Ability of automating processes is usually high-valued in businesses to produce benefits such as productivity increases, saving costs or reducing working hours.

Text extraction as a means to automatise response to customer support tickets.



Thesis objective

Is it possible to extract keywords from any incoming customer ticket?

1. How to approach a keyword extraction task, its caveats and restrictions.
2. Implementation and evaluation of an ensemble NLP model for text extraction.
3. Analysis of each NLP technique employed.
4. Conclusions after the completion of the task, available room for improvement.

Example of support ticket and extraction

"Goodmorning, I need two VMs with 4 cores and 16GB of RAM each. I also need them to have a 100GB SSD disk. Many thanks."

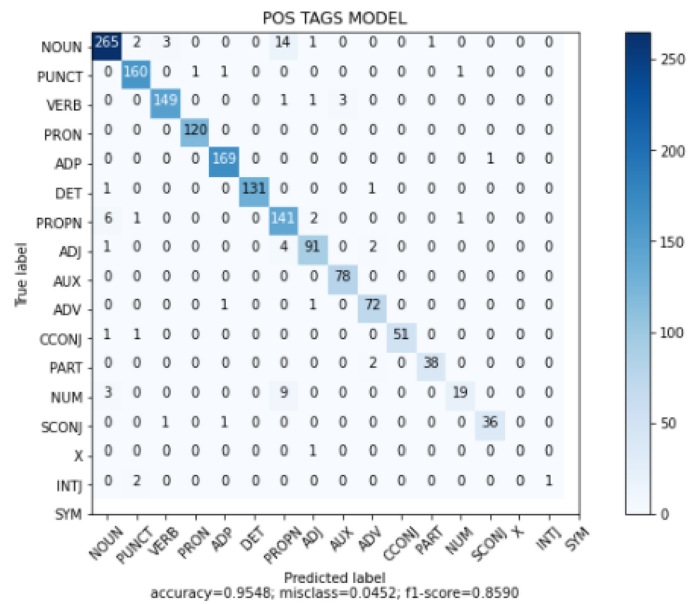
- cpu: <number of cores> (mandatory)
- ram: <number of gigabytes> (mandatory)
- disk: list of disks (mandatory); a disk has the following structure:
  1. size: <number of gigabytes> (mandatory)
  2. type: <label whether is ssd, magnetic> (optional, defaults to magnetic)
- name: <string> (optional)

’Server’: [2, VM], ’Cores’: [’4’, ’core’], ’Ram’: [’16’, ’ram’], ’Disk’: [’100’, ’disk’]

NLP: Part-of-Speech Tagging

Task	Part-of-Speech tagging.
Goal	Determine the grammatical category of a word.
Data	UDPOS Dataset.
Model	Pre-trained BERT model with added 1 Linear Layer and Softmax Layer.
Results	Trained for 5 epochs; evaluation score = 0.9235.

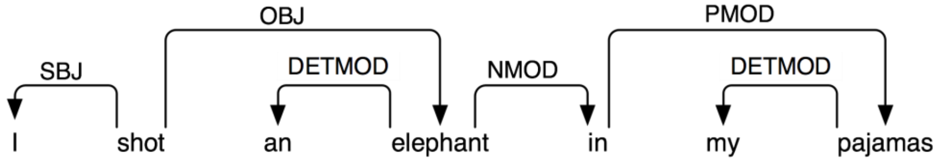
	Precision	Recall	F1-score	Support
NOUN	0.9567	0.9266	0.9414	286
PUNCT	0.9639	0.9816	0.9726	163
VERB	0.9739	0.9675	0.9707	154
PRON	0.9917	1	0.9959	120
ADP	0.9826	0.9941	0.9883	170
DET	1	0.9850	0.9924	133
PROPN	0.8343	0.9338	0.8812	151
ADJ	0.9381	0.9286	0.9333	98
AUX	0.9630	1	0.9811	78
ADV	0.9351	0.9730	0.9536	74
CCONJ	1	0.9623	0.9808	53
PART	0.9744	0.9500	0.9620	40
NUM	0.9048	0.6129	0.7308	31
SCONJ	0.9730	0.9474	0.9600	38
X	0	0	0	0
INTJ	0	0	0	1
SYM	1	0.33	0.5	3
				1593



Dependency Parsing

Task	Dependency Parsing
Goal	Capture the relation between words in a sentence.
Model	Spacy built-in Transition Based non-monotonic Parser.
Usage	Used to retrieve the quantity associated to keyword.

Clausal Argument Relations	Description
NSUBJ	Nominal Subject
DOBJ	Direct Object
IOBJ	Indirect Object
CCOMP	Clausal Complement
XCOMP	Open clausal complement
Nominal Modifier Relations	**Description**
NMOD	Nominal modifier
AMOD	Adjectival modifier
NUMMOD	Numerical modifier
APPOS	Appositional modifier
DET	Determiner
CASE	Prepositions and other markers
Other Relations	**Description**
CONJ	Conjunct
CC	Coordinating conjunction



Word Embeddings

Approach

Data collection

- Gather data from Wikipedia distributed categories database of articles. Total of 6.311 articles.

Data pre-processing

- Tokenization
- Lemmatization
- Removal of stopwords
- Removal of special characters

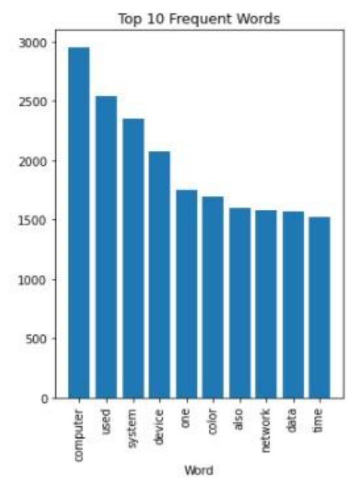
Data modelling

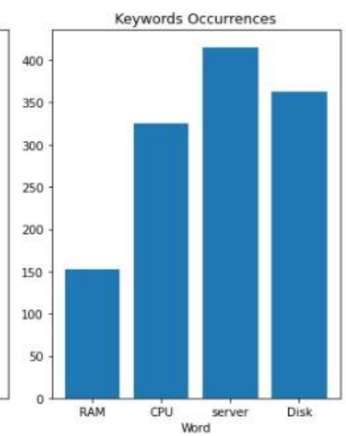
- Fasttext sub-word model
- Skip-gram

Hyperparameters

- Embedding size: 60
- Window size: 40
- Words minnimum: 30
- Down-sampling rate: 1e-2
- Iterations: 100

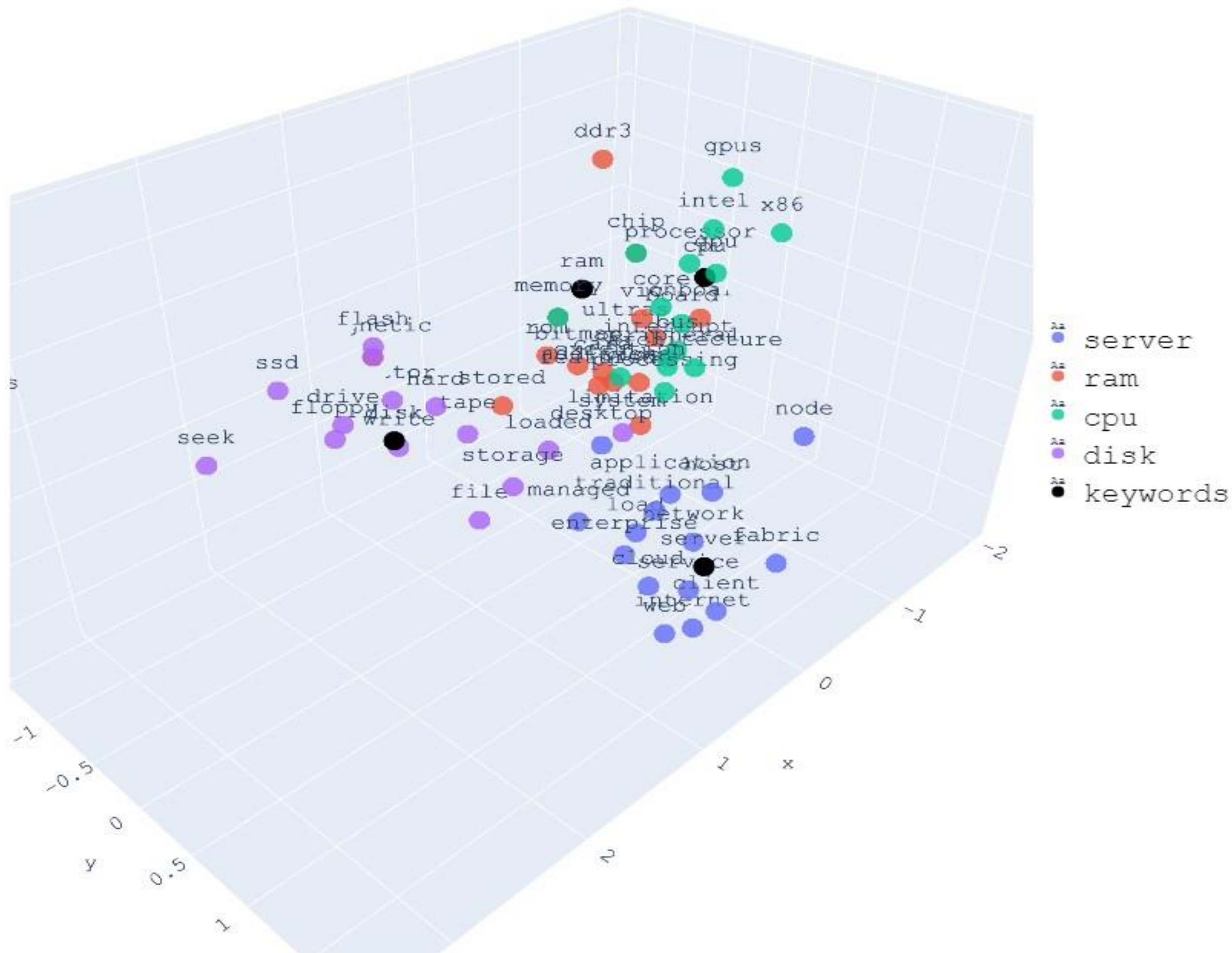
Dataset Description:





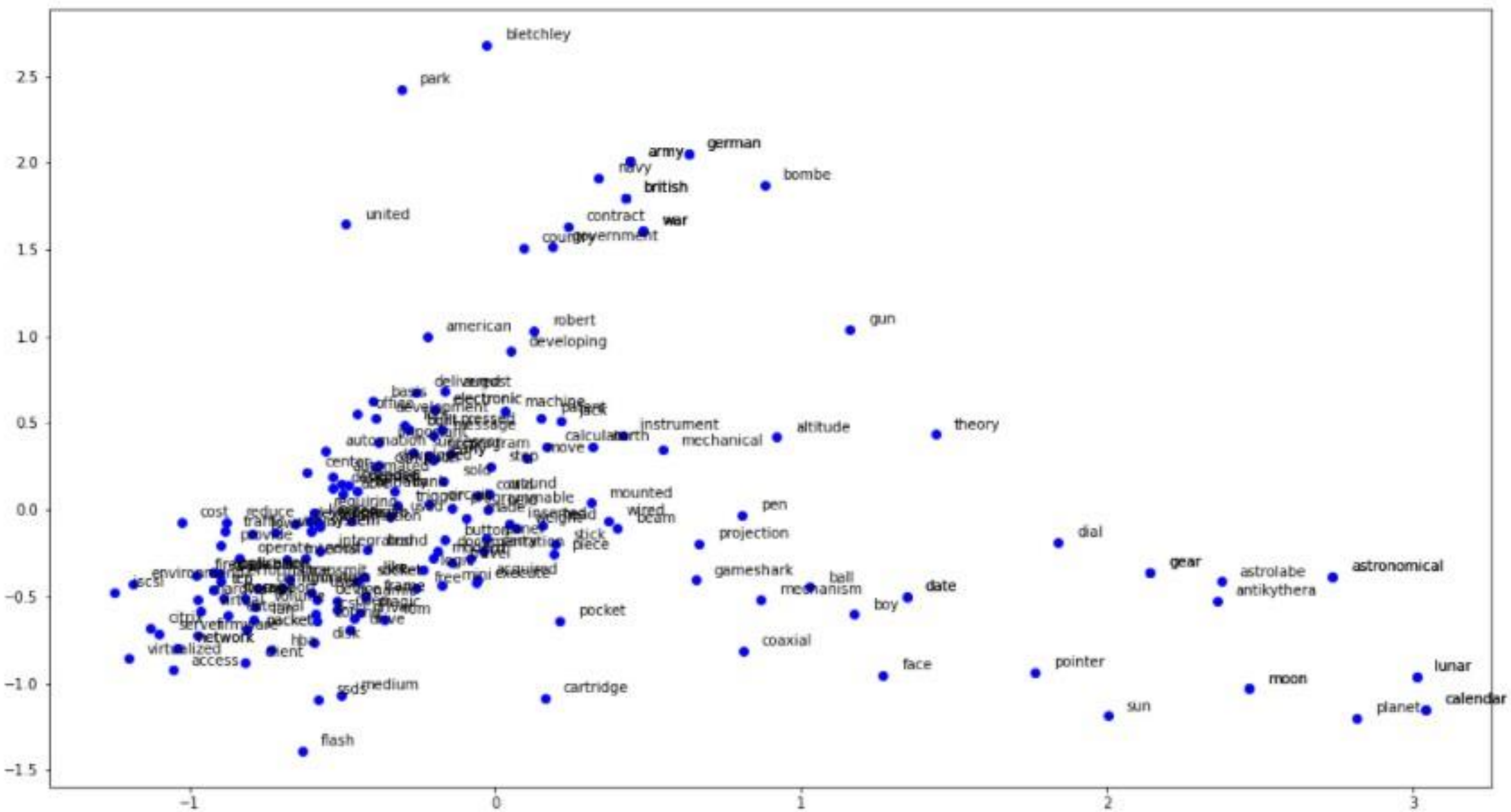
Visualization of Word Embeddings

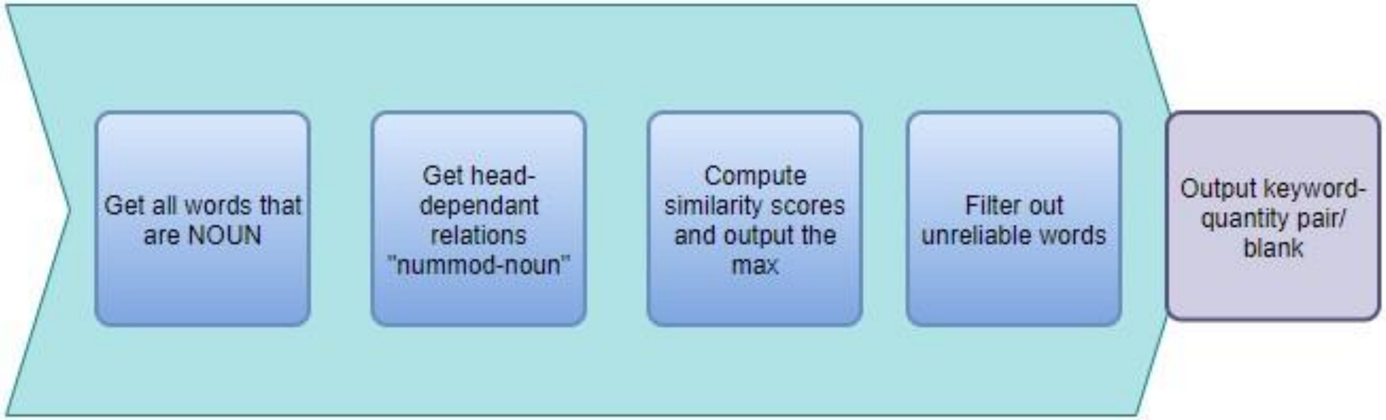
PCA 3D Plot of keyword embeddings



Visualization of Word Embeddings

PCA 2D Plot of embeddings





Pipeline & Results

- -A pipeline for every keyword.
- -Grid-search for threshold setting.
- -Test tickets generated by template.
- -Own tests score: 77.1% accuracy.

Conclusions

- Approach the task as process where every step discards words.
- Scalable, could work for any type of word.
- Create own embeddings is better.
- Next steps involve send keywords automatically to VM deployment program.