Neural Network for Named Entity Recognition for Skills Extraction in Job Postings

List of references – prototype access

- Original Prototype found at: https://www.youtube.com/watch?v=Davcp3-dnX8
- Dataset found at: https://www.kaggle.com/datasets/madhab/jobposts
- SkillNER found at: https://skillner.vercel.app/
- SpaCy found at: https://spacy.io/
- Google Colab found at: https://colab.research.google.com/

All links last accessed: 10th of June, 2022

Pipeline

First the raw data is treated and cleaned by an R-script.

Then the data is annotated using SkillNER and SpaCy.

We treat and clean the annotated data again with an R-script.

Finally we make all job posts the same length, make all data numerical, normalize the data and transform it to tensor-format before feeding it to the Neural Network.

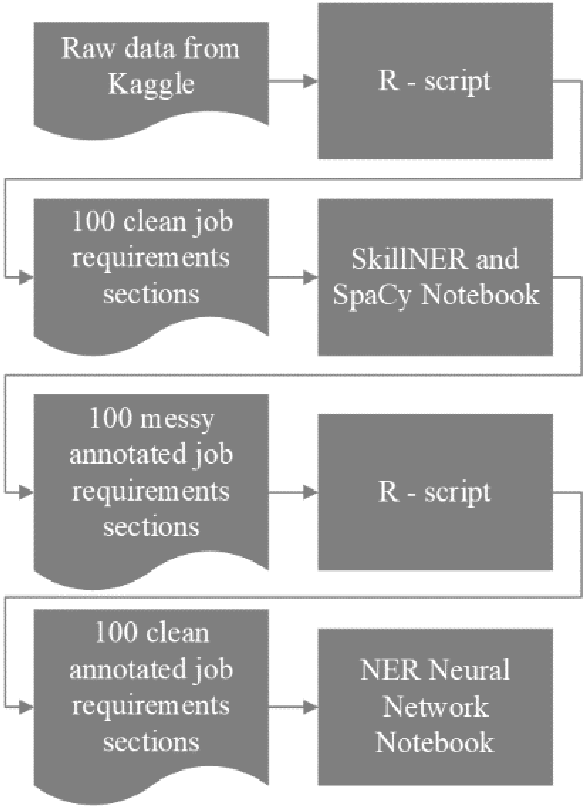
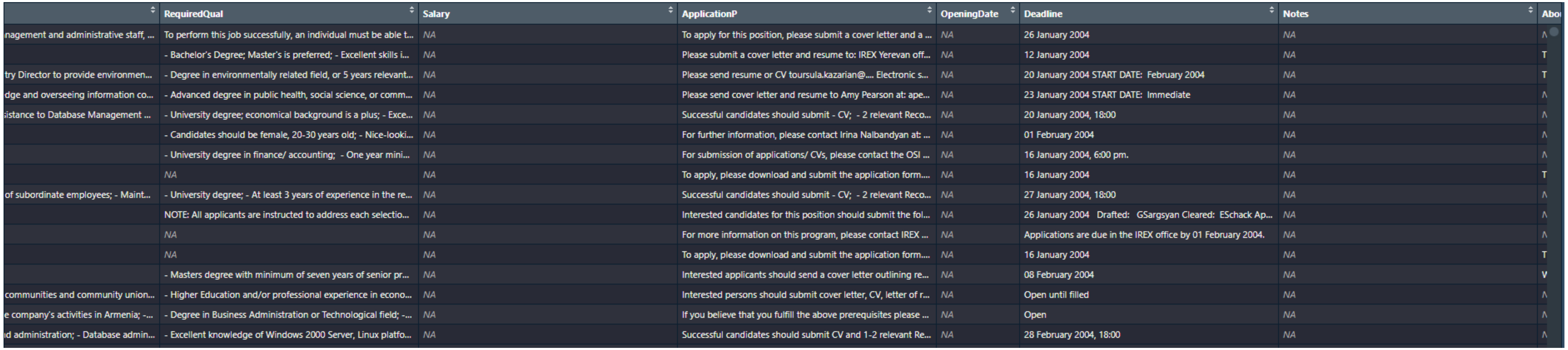

Raw data from Kaggle
R - script
100 clean job requirements sections
SkillNER and SpaCy Notebook
100 messy annotated job requirements sections
R - script
100 clean annotated job requirements sections
NER Neural Network Notebook

Fig.1: Code Pipeline. Source: Own Illustration

The Dataset

The dataset consists of 19’000 job posts that have been posted on the Armenian human resources portal CareerCenter between 2004-2015. It consists of 24 attributes of which only the attribute “Job Requirements” is relevant for the paper.



	RequiredQual	Salary	ApplicationP	OpeningDate	Deadline	Notes	Abo
nagement and administrative staff, ...	To perform this job successfully, an individual must be able t...	NA	To apply for this position, please submit a cover letter and a ...	NA	26 January 2004	NA	N
	- Bachelor's Degree; Master's is preferred; - Excellent skills i...	NA	Please submit a cover letter and resume to: IREX Yerevan off...	NA	12 January 2004	NA	T
try Director to provide environmen...	- Degree in environmentally related field, or 5 years relevant...	NA	Please send resume or CV toursula.kazarian@.... Electronic s...	NA	20 January 2004 START DATE: February 2004	NA	T
dge and overseeing information co...	- Advanced degree in public health, social science, or comm...	NA	Please send cover letter and resume to Amy Pearson at: ape...	NA	23 January 2004 START DATE: Immediate	NA	N
istance to Database Management ...	- University degree; economical background is a plus; - Exce...	NA	Successful candidates should submit - CV; - 2 relevant Reco...	NA	20 January 2004, 18:00	NA	N
	- Candidates should be female, 20-30 years old; - Nice-looki...	NA	For further information, please contact Irina Nalbandyan at: ...	NA	01 February 2004	NA	N
	- University degree in finance/ accounting; - One year mini...	NA	For submission of applications/ CVs, please contact the OSI ...	NA	16 January 2004, 6:00 pm.	NA	N
	NA	NA	To apply, please download and submit the application form....	NA	16 January 2004	NA	T
of subordinate employees; - Maint...	- University degree; - At least 3 years of experience in the re...	NA	Successful candidates should submit - CV; - 2 relevant Reco...	NA	27 January 2004, 18:00	NA	N
	NOTE: All applicants are instructed to address each selectio...	NA	Interested candidates for this position should submit the fol...	NA	26 January 2004 Drafted: GSargsyan Cleared: ESchack Ap...	NA	N
	NA	NA	For more information on this program, please contact IREX ...	NA	Applications are due in the IREX office by 01 February 2004.	NA	N
	NA	NA	To apply, please download and submit the application form....	NA	16 January 2004	NA	T
	- Masters degree with minimum of seven years of senior pr...	NA	Interested applicants should send a cover letter outlining re...	NA	08 February 2004	NA	V
communities and community union...	- Higher Education and/or professional experience in econo...	NA	Interested persons should submit cover letter, CV, letter of r...	NA	Open until filled	NA	N
e company's activities in Armenia; -...	- Degree in Business Administration or Technological field; -...	NA	If you believe that you fulfill the above prerequisites please ...	NA	Open	NA	N
d administration; - Database admin...	- Excellent knowledge of Windows 2000 Server, Linux platfo...	NA	Successful candidates should submit CV and 1-2 relevant Re...	NA	28 February 2004, 18:00	NA	N

Fig.2: Job-Posts Dataframe. Source: https://www.kaggle.com/datasets/madhab/jobposts

Pre-processing 1

The raw data consists of 19’000 job posts, posted on the Armenian Human Resources portal CareerCenter between 2004 and 2015. It consists of 24 attributes, but since only the requirements section is important for skills extraction all other 23 attributes are neglected. Then 100 random, not empty, samples are taken and cleaned. This is done in R and corresponding libraries:

```
#Format the text
for (h in 1:nrow(job_requirements_export)) {

  job_requirements_export[h,1] <- gsub('-', '', as.character(job_requirements_export[h,1]), fixed = TRUE)
  job_requirements_export[h,1] <- gsub(';', '.', as.character(job_requirements_export[h,1]), fixed = TRUE)
  job_requirements_export[h,1] <- gsub(':', '.', as.character(job_requirements_export[h,1]), fixed = TRUE)
  job_requirements_export[h,1] <- gsub('\n', ' ', as.character(job_requirements_export[h,1]), fixed = TRUE)
  job_requirements_export[h,1] <- gsub('/', '', as.character(job_requirements_export[h,1]), fixed = TRUE)
  job_requirements_export[h,1] <- gsub('  ', ' ', as.character(job_requirements_export[h,1]), fixed = TRUE)
  job_requirements_export[h,1] <- gsub(',', '', as.character(job_requirements_export[h,1]), fixed = TRUE)
  job_requirements_export[h,1] <- trimws(job_requirements_export[h,1], which = c("both", "left", "right"))

}
```

All observed special characters are not needed are cleaned.

Data labeling

Those 100 random cleaned job posts are annotated using SkillNER. SkillNER uses the large language model from SpaCy, a self-created skill database and a self created phrase matcher to annotate data. This is done using python and the corresponding libraries on Google Colab.

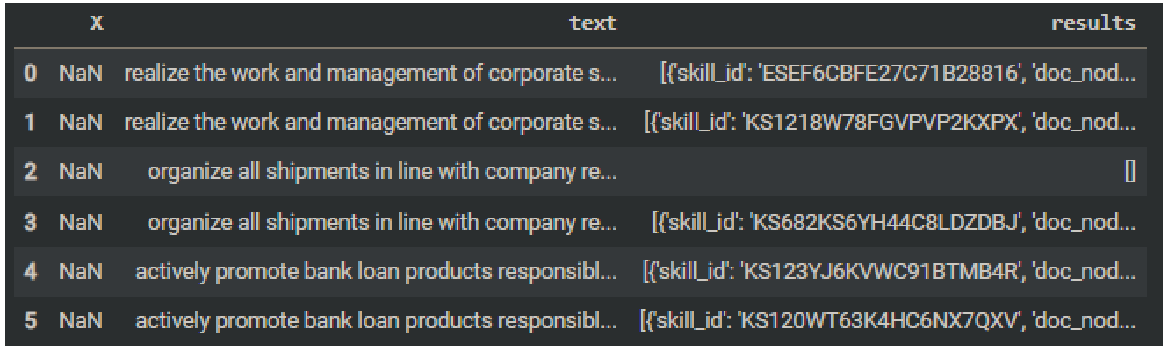
```
#create an empty dataframe with the format of the annotations created by the skill_extractor.
annotations_df = pd.DataFrame(columns = ['X', 'text', 'results'])
#extract skills from requirements_df

for ind, row in requirements_df.iterrows():

  job_description = requirements_df.iloc[ind,1]
  job_description
  annotations = skill_extractor.annotate(job_description)
  #temporary data frame to hold the current annotations
  df = pd.DataFrame(annotations)
  #add temporary data frame to final dataframe
  annotations_df = annotations_df.append(df, ignore_index = True)

annotations_df.head(6)
```

The skill extractor is the main workhorse in this part.

Pre-processing 2

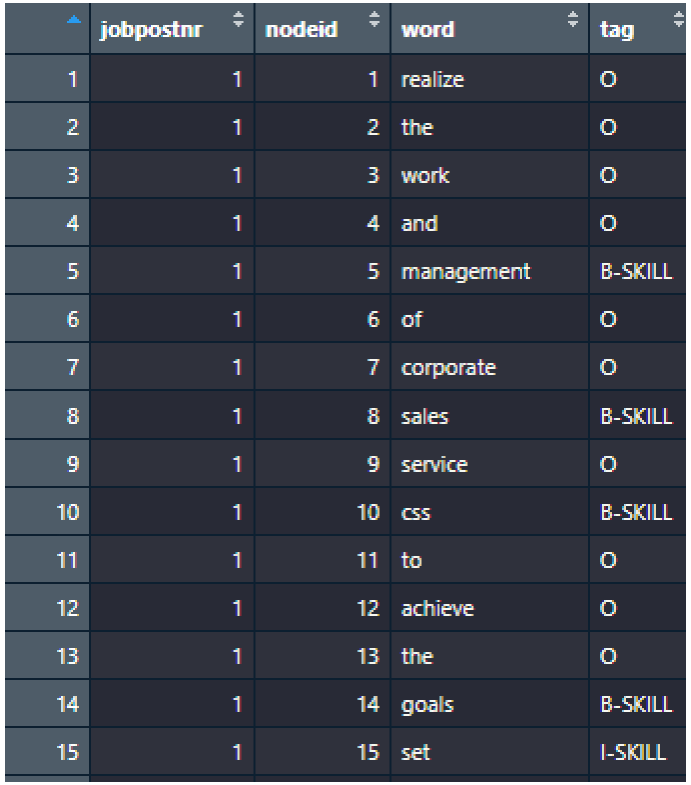
The 100 annotated job posts have a very messy format when leaving SkillNER's skill extractor.

Therefore a lot of cleaning in R is again needed. The code is very long, therefore no code samples are shown here. Instead the transformation of the data is shown.



	X	text	results
0	NaN	realize the work and management of corporate s...	[{'skill_id': 'ESEF6CBFE27C71B28816', 'doc_nod...
1	NaN	realize the work and management of corporate s...	[{'skill_id': 'KS1218W78FGVPVP2KXPX', 'doc_nod...
2	NaN	organize all shipments in line with company re...	[]
3	NaN	organize all shipments in line with company re...	[{'skill_id': 'KS682KS6YH44C8LDZDBJ', 'doc_nod...
4	NaN	actively promote bank loan products responsibl...	[{'skill_id': 'KS123YJ6KVWC91BTMB4R', 'doc_nod...
5	NaN	actively promote bank loan products responsibl...	[{'skill_id': 'KS120WT63K4HC6NX7QXV', 'doc_nod...

Fig.3: Annotated data, directly outputted by SkillNER



	jobpostnr	nodeid	word	tag
1	1	1	realize	O
2	1	2	the	O
3	1	3	work	O
4	1	4	and	O
5	1	5	management	B-SKILL
6	1	6	of	O
7	1	7	corporate	O
8	1	8	sales	B-SKILL
9	1	9	service	O
10	1	10	css	B-SKILL
11	1	11	to	O
12	1	12	achieve	O
13	1	13	the	O
14	1	14	goals	B-SKILL
15	1	15	set	I-SKILL

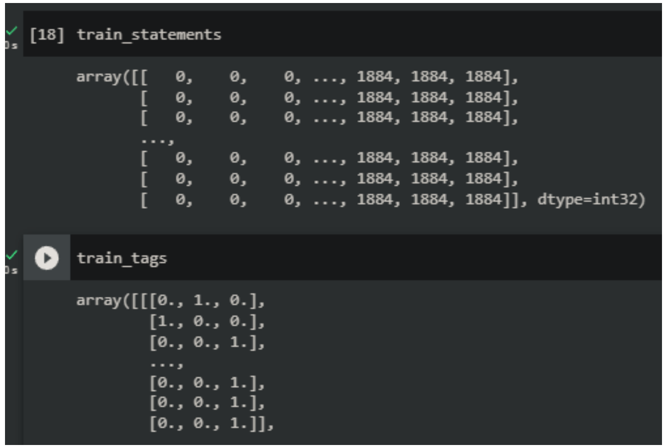
Fig.4: Dataframe usable by the Neural Network Notebook

NN Notebook

Tensor Transformation

The first part of the NN Notebook is again pre-processing of data. From the dataframe of Fig. 4 it is the aim to arrive at having a training- and testset in numerical, normalized, tensor format. In our case we arrive at a shape of (100, 530, 3).

100 job postings, each job posting is 530 words long and each word is tagged to be one of three possible one-hot-encoded tags.



```
[18] train_statements

array([[   0,    0,    0, ..., 1884, 1884, 1884],
       [   0,    0,    0, ..., 1884, 1884, 1884],
       [   0,    0,    0, ..., 1884, 1884, 1884],
       ...,
       [   0,    0,    0, ..., 1884, 1884, 1884],
       [   0,    0,    0, ..., 1884, 1884, 1884],
       [   0,    0,    0, ..., 1884, 1884, 1884]], dtype=int32)
```

```
train_tags

array([[[0., 1., 0.],
        [1., 0., 0.],
        [0., 0., 1.],
        ...,
        [0., 0., 1.],
        [0., 0., 1.],
        [0., 0., 1.]],
```

Fig.5: Final format of data before feeding it to the Neural Network

NN Notebook
The Neural Network

The most important layers, parameters and metrics from our prototype are covered here:

Input layer: Takes the inputs from the tensors, # neurons has to be tensor-length.

LSTM-layer: Consists of neurons that can learn order in sequence prediction and are therefore very useful for our case. Past sequence information is taken into account.

Time Distributed layer: apply the same function for every time-step and help when sequential data is present, but the data should also be looked at unsequential..

Output layer: Predicts the tag for the current entity, # neurons has to be length of one-hot encoded tag vector.

We can also experiment with the loss function, the optimizer type, and the performance metrics.

In this protoype we focus on the prediction accuracy and the loss function. Currently we achieve 97% precision and recall on the test set. Precision indicates how many predicted positives were true positives. Recall indicates how many of the true positives were found. Results have to be taken with a pinch of salt, because no data was manually annotated. May be useful for transfer learning.