NER Model for ODS Jobs Dataset

Galina Zhakova

May 2022

Abstract

This is a project report on extracting entities from the posts with vacancies. The data source is the ODS jobs slack channel. A link to the project code: `https://github.com/ZhakovaGalina/ods_jobs_ner`.

1 Introduction

In our project, we extract information about jobs titles, salaries, location and work style from messages in ODS job Slack channel. There are solutions to this issue using regular expressions and Yargy-parser. We extract information using the deeppavlov neural network based on marked-up data.

1.1 Team

Zhakova Galina is the author of the project.

2 Related Work

- Data extraction with regular expressions [reg,]
- Data extraction by a Yargy-parser[yar,]

3 Model Description

The information is extracted using DeeppavlovNER based on bert-base-multilingual-cased transformer. Then the extracted information is classified using regular expressions to compare the quality with other approaches.

4 Dataset

The dataset is 500 BIO-marked messages from the slack ODS jobs channel (2020 year).

Tags:

- VAC - information about the vacancy (job title and candidate level)
- LOC - location (city and metro)
- SAL - salary (including information about taxes, not including information about bonuses)
- WST - work style (remote, partially remote, a few days in the office etc.)

	Train	Valid	Test
Articles	402	48	50
Tokens	149,110	17,700	18,004
Vocabulary size		21,924	

Table 1: Statistics of the ODS jobs dataset.

The dataset is a 500 random messages from ODS jobs channel for 2020 year. They were tokenized by sentenses and words with NLTK Python library. And then marked up manually with crowd-sourcing.

5 Experiments

5.1 Metrics

F1 metric is used for results of DeeppavlovNER. F1 metric is used to compare job title and candidate level determination with other approaches. The ratio: (messages with found entities/total number of messages) is used to approximately estimate the quality of location and salary search.

5.2 Experiment Setup

500 messages were randomly divided into train,validation and test with ratio: 80-10-10. We use DeeppavlovNER network with base configuration. Config file can be found in the project: `https://github.com/ZhakovaGalina/ods_jobs_ner`

5.3 Baselines

Data extraction with regular expressions and data extraction with Yargy-parser were used as baselines.

6 Results

In this section, we show the quality of the Deeppavlov NER and compare our results with baselines.

The model is the best at finding a salary information and the worst at detecting the style of work:

	F1
ner f1	69.12
ner token f1	80.35
VAC	70.73
SAL	71.28
LOC	70.73
WST	59.09

Table 2: Result of Deeppavlov NER model on test data.

Our model shows the best results in determining the job title than other approaches:

Job Title F1	NER	Yargy	RegExpr
Total DA,DS,DE	93.20	72.16	87.99
DA	88.89	75	57.14
DS	93.67	72.46	90.9
DE	93.33	70.0	87.5
Any	96.9 0	90.1	

Table 3: Result comparison of Vacancy classification: DeepPavlovNer+Regular Expressions, Yargy Parser, Regular Expressions .

Our model shows the best results in total and in determining senior and lead level than other approaches:

Job Level F1	NER	Yargy	RegExpr
Total	94.33	91.22	92.17
Junior	87.5	88.88	94.73
Middle	95	97.56	95
Senior	95	92.68	93.02
Lead	100	71.42	76.92

Table 4: Result comparison of Level classification: DeepPavlovNer+Regular Expressions, Yargy Parser, Regular Expressions .

The best results above could be achieved due to the quality of regular expressions.

Also below is information about the share of finding information about salary and location in comparison with other approaches. But this information can only be used as an approximate estimate, because it is a quantitative, but not a qualitative indicator.

Detected salaries	Share of messages
DeepPavlovNer+Regular Expressions	92.03
Yargy Parser	67.66
Regular Expressions	94.52

Table 5: Result comparison of Salary detection: DeepPavlovNer+Regular Expressions, Yargy Parser, Regular Expressions .

Detected locations	Share of messages
DeepPavlovNer+Regular Expressions	79.10
Yargy Parser	77.11

Table 6: Result comparison of Location detection: DeepPavlovNer+Regular Expressions, Yargy Parser.

The samples of the information detected by the model could be found in Tab. 7.

'Senior', 'Data', 'Scientist', 'специалиста', 'по', 'ML'
'CV-мидла', 'middle', 'computer', 'vision', 'researcher'
'Разработчик','алгоритмов','машинного','обучения','в','сфере','3D-технологий'

Table 7: Output samples of job information.

7 Conclusion

We have collected a dataset, made a markup for it and apply a model showing good results compared to other models.

References

[reg,] `https://github.com/egorborisov/jobs_article`.

[yar,] `https://github.com/kuk/analyze-ods-jobs`.