Deep learning-based methods for technologist skill named entity recognition

Tirthankar Mittra, Abulitibu Tuguluke

University of Colorado Boulder

Definitions

- 'Technologist': An expert in a particular field of technology.
  e.g.: Software Engineer, Data Scientist.
- 'Skill set': The skill that associate with job title of a candidate.
  e.g.: C#, SQL, Python.
- 'Named-entity recognition': Information extraction that seeks to locate and classify name entity.
  e.g.: C#, Data Scientist as Skill, Job title.

Introduction

The US, and the world in general, is facing a tech-talent shortage, which can be translated into revenue decrease. One way to hire talent is through job posting, yet spotting the skills in a job still heavily depends on text parsing. In this project, we propose a new concept: Technologist Skill Name Entity Recognition (TSNER), where we apply name entity recognition to identify which organization requires what type of technology skill sets for a particular job title from a lengthy job description text.

Related Work

1. Deep learning-based methods for natural hazard named entity recognition. This model is tailored for natural hazard named entity, where there are only 10 categories, ours is focused on thousands of technologist skills and company names along with titles.
2. Portuguese named entity recognition using bert-crf. Unlike this paper we did not pre-train our BERT model from scratch to learn word embeddings and then do transfer learning, we directly used $BERT_{BASE}$ cased based model's embeddings trained on English language.
3. Bidirectional lstm-crf models for sequence tagging. We used this model to understand how training a deep neural network architecture from scratch would look like for a new NER task and we also wanted to compare it's performance with our BERT based model.

Methods

The first step is pre-processing the JSON file of 1.4 million data point to extract the relevant job data, We used bag of words for labelling to just set up our model. For our project we use three main categories of named entities, i.e. skill (S), job title (J) and company or organization (C), while the output for rest of the inputs will be labelled as (O) except for the padding tokens which are labelled as (UNK). Next we used 3 algorithms for our name entity recognition task.

- Base model: Conditional random field (CRF)
- BERT-CRF model
- BI-LSTM-CRF

Motivation

A rule free, regex free name entity recognition deep learning parser for job postings, where we can 'recognize' company name (C), job title (J), tech skill (S) and more.

Experimental Design

We chose 2000 randomly selected well labeled data (with over 128 words in each description) from 1 million data set to train test and validate our models, the labeled data was split into training and validation set, we used recall, precision and f1 score metrics specifically because it's a standard used for many other NER tasks and also because the distributions of the different named entities in a corpus are generally skewed. Loss on the validation data set(early stopping) was used to stop over-fitting. We didn't do much hyper parameter tuning so the need for a separate test dataset as the results was not so imperative.

Experimental Results

Training/ Validation size	entity	precision	recall	f1-score
1.5k/0.5k	C	1.000	0.953	0.976
	J	0.999	0.999	0.999
	O	0.983	0.986	0.985
	S	0.991	0.988	0.990
15k/5k	C	1.000	0.993	0.996
	J	1.000	1.000	1.000
	O	0.999	1.000	0.999
	S	1.000	0.999	0.999

Table 1:Conditional random field (CRF) metric

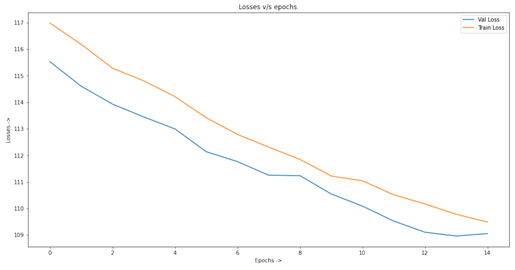

Figure 1:BERT-CRF

Experimental Results (Cont.)

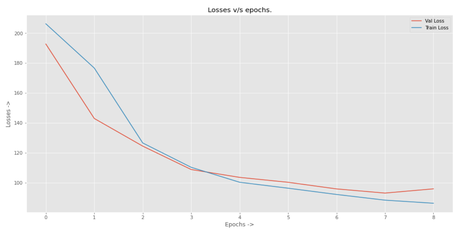

Figure 2:BiLSTM-CRF

Method	entity	precision	recall	f1-score
BiLSTM-CRF	C	0.000	0.000	0.000
	J	0.545	0.992	0.704
	S	0.000	0.000	0.000
	O	0.000	0.000	0.000
	UNK	0.980	0.998	0.989
BERT-CRF	C	0.000	0.000	0.000
	J	0.581	0.870	0.697
	S	0.268	0.026	0.048
	O	0.341	0.105	0.161
	UNK	0.727	0.936	0.818

Table 2:BiLSTM-CRF vs BERT-CRF metric

Conclusions

- Data collection and labeling is the bottleneck for supervised learning.
- Traditional CRF with POS tagging performs the best, high accuracy and has no memory issue.
- Even though BERT layer is defined as not trainable, it required extra GPU otherwise there was RAM overflow, making it perform less than CRF, and less feasible.
- Myth buster: Not every algorithm should be about Deep Learning. CRF is lightweight, practical easy to train model and should be used on new NER tasks.

Ethical Implications

This study only focused on the STEM job parsing in the job market, and paid no attention to the social science and skill related to that field, the ethical dilemma is all these studies will tilt toward engineering majors, while ignoring the 'engineering of the mind' majors which play a key role in our society. There should not be any special treatment for any job just because of the market value and popularity, every harmonious society should accept the message that both social and technical knowledge are keys to success and harmony. The message we should not be sending is that skills not related to tech are not important. In the future, we hope to replicate the same work for non-STEM job postings.

Acknowledgements

The data in this study will be used only for research purposes and in ways that will not be shared due to DHI Group Inc.'s data retention policy.

Contact Information

- Email:Tirthankar.Mittra@colorado.edu
- Email: Abulitibu.Tuguluke@colorado.edu