Job Skills extraction with LSTM and Word Embeddings

Nikita Sharma
nikita.sharma@student.uts.edu.au
University of Technology Sydney

1 Abstract

In this paper I have compared a few unsupervised and supervised machine learning techniques to address the Job Skills extraction challenge.

The paper proposes the application of Long Short Term Memory [1] (LSTM) deep learning network combined with Word Embeddings [2] to extract the relevant skills from text documents. The approach proposed in the paper also aims at identifying new and emerging skills that haven't been seen already.

I trained my model on free text corpuses from a few online job postings, under Data Science category, and then extended the model to other job postings from other non data science categories, to test how the model is performing cross category. I will also propose some tweaks that can be used to make the model perform better on cross-category text corpuses.

2 Introduction

Job Skills extraction is a challenge for Job search websites and social career networking sites. It is a sub problem of information extraction domain that focussed on identifying certain parts to text in user profiles that could be matched with the requirements in job posts. Job Skills are the common link between Job applications, User resumes and Job postings by companies. Identifying sills in new job postings and new user profiles is an important problem that can fill this gap and provide a pathway for job seekers and hiring organisations.

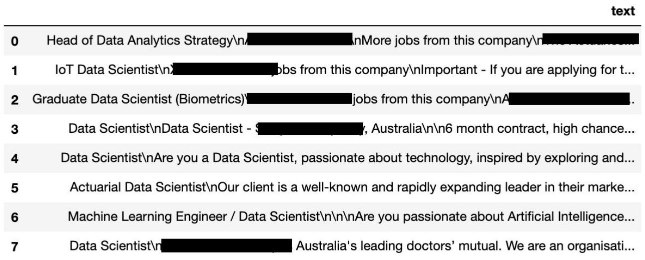
The job skill extraction is often solved by traditional techniques like matching against a static dictionary of skills, but this approach does not extend to new and emerging skills. Updating the dictionary can be manual and tedious, and also needs domain experts a lot of time to identify correct skills that map to a particular domain.

In this paper I have tried out a combination of unsupervised and supervised machine learning techniques to extract relevant skills from our text corpus. I have observed significant improvements in model performance by combining LSTM with Word Embeddings for our problem.

3 Getting data

The most common datasets for this problem are User Profiles / Resumes and Job Postings. For this problem I have limited my training dataset to Job Postings, and under the Data Science category.

I fetched 8 Job Postings from online job search websites [3] that were posted under the Data Science category. The dataset is pretty small. I used this as my bootstrap and training dataset. The jobs have been picked at random with no special attention paid at the Job post content.



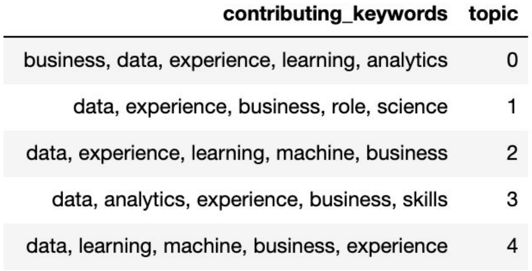
	text
0	Head of Data Analytics Strategy\n[redacted]nMore jobs from this company\n[redacted]
1	IoT Data Scientist\n[redacted]bs from this company\nImportant - If you are applying for t...
2	Graduate Data Scientist (Biometrics)[redacted]jobs from this company\nA[redacted].
3	Data Scientist\nData Scientist - [redacted], Australia\n\n6 month contract, high chance...
4	Data Scientist\nAre you a Data Scientist, passionate about technology, inspired by exploring and...
5	Actuarial Data Scientist\nOur client is a well-known and rapidly expanding leader in their marke...
6	Machine Learning Engineer / Data Scientist\n\n\nAre you passionate about Artificial Intelligence...
7	Data Scientist\n[redacted] Australia's leading doctors' mutual. We are an organisati...

4 Unsupervised and Self-supervised approach

Initial approaches were to apply unsupervised and self supervised learning to find patterns in the dataset with an objective to form groups of text, to identify interesting keywords and topics from the documents.

4.1 Topic Modelling

Topic modelling [4] is an unsupervised approach to extract abstract topics from text documents. I identified 5 topics from the combined text corpus created from the 8 job posts that I collected. For each topic I further extracted 5 keywords that contributed the most to the topic. I have added both unigrams and bigrams for this task.



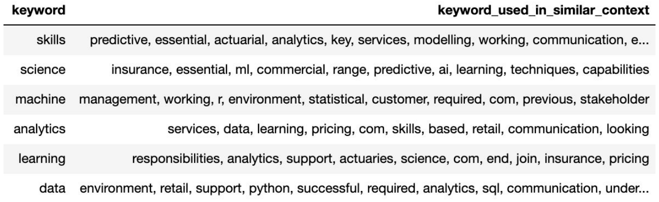
contributing_keywords	topic
business, data, experience, learning, analytics	0
data, experience, business, role, science	1
data, experience, learning, machine, business	2
data, analytics, experience, business, skills	3
data, learning, machine, business, experience	4

Topic modelling identified the context well, along with relevant keywords like *data*, *science*, *machine*, *learning*, *analytics* etc. On the flip side, topic modelling did not provide the complete set of relevant skill sets that we are interested in for our problem statement.

4.2 Word Representations - Word2Vec

Word2Vec [5] is a self supervised neural network that can identify keywords used in similar context and can be used to extract related skills and keywords for any set of provided keywords. The idea is to extend on top of Topic modelling. The base keywords can be extracted from topic modelling and all the skills and keywords used in the same context can be extracted from Word2Vec.

I trained word2vec with the same dataset we created from the 8 job posts. I passed a set of unique keywords extracted from section 4.1 to get new list of skills and keywords.



keyword	keyword_used_in_similar_context
skills	predictive, essential, actuarial, analytics, key, services, modelling, working, communication, e...
science	insurance, essential, ml, commercial, range, predictive, ai, learning, techniques, capabilities
machine	management, working, r, environment, statistical, customer, required, com, previous, stakeholder
analytics	services, data, learning, pricing, com, skills, based, retail, communication, looking
learning	responsibilities, analytics, support, actuaries, science, com, end, join, insurance, pricing
data	environment, retail, support, python, successful, required, analytics, sql, communication, under...

Word2Vec did a good job at identifying few skills like *python*, *predictive*, *ai*, *analytics*, *sql*, *r, ml* etc that can be used for identifying few useful skills. On the flip side, Word2Vec extracted more noise than useful skills, and that makes the results undesirable. While the extracted keywords were useful and were also representative of the Data Science domain, but the output had a lot of noise, and separating useful skills from the noise would be a difficult task.

Tip: Word2Vec accuracy can be improved by providing more text corpuses and more job posts.

5 Supervised Modelling Approach

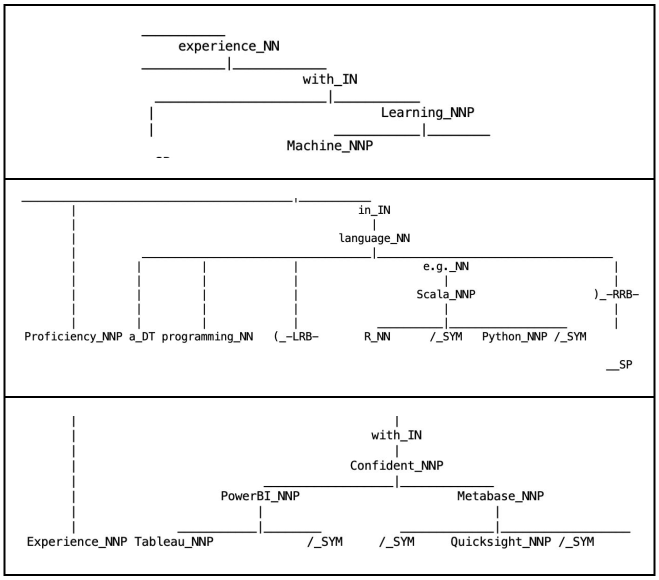
I stopped unsupervised approaches here, and moved towards exploring supervised learning. Supervised learning involved creating training dataset, that we will discuss below in section 5.2.

I researched various ways in which text corpus can be represented in the training dataset. I did not want to do a simple skill classifier from the dataset, because this technique will only apply to the known skills/labels and will not extend to new/emerging skills that we are not aware of yet. One great approach was mentioned in the article [6] that proposed the usage of language & grammar as training data.

5.1 Understanding grammar

Every text corpus has a language style and associated grammar. The grammar can be investigated to find patterns in sentences, that can then be used to define phrases that define our skills. I used spacy and nltk to analyse the part-of-speech [7] of our text corpuses.

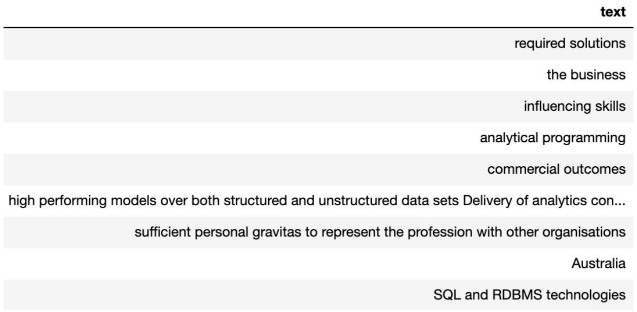
Having a look at the grammar for our text corpuses we can see that most of our skills are represented by Noun entities. This can be a good candidate criteria for preparing our training dataset.


experience_NN
with_IN
Learning_NNP
Machine_NNP
in_IN
language_NN
e.g._NN
Scala_NNP
)_-RRB-
Proficiency_NNP a_DT programming_NN (_-LRB- R_NN /_SYM Python_NNP /_SYM
__SP
with_IN
Confident_NNP
PowerBI_NNP
Metabase_NNP
Experience_NNP Tableau_NNP /_SYM /_SYM Quicksight_NNP /_SYM

I extracted all noun phrases from our text corpuses and used these phrases as our training dataset for the next steps.

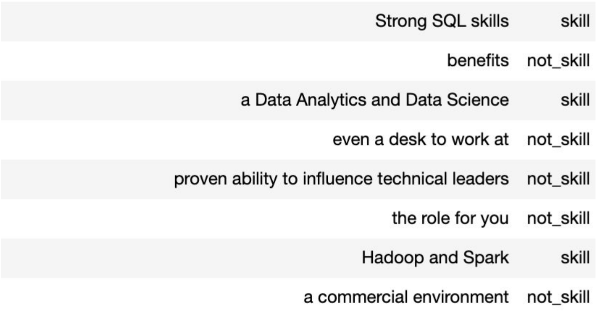
5.2 Preparing training data

I used spacy to extract all the noun phrases from the text corpus. I created a data set from all of these extracted phrases.



text
required solutions
the business
influencing skills
analytical programming
commercial outcomes
high performing models over both structured and unstructured data sets Delivery of analytics con...
sufficient personal gravitas to represent the profession with other organisations
Australia
SQL and RDBMS technologies

I created a training set from this dataset by labelling the dataset as skill or not_skill. This approach is also proposed in the article shared in section 5 introduction. Our challenge is to identify the relevant phrases that represent any form of skill, and separate them out of irrelevant noun phrases. This forms the center of our Skill extractor. The labelled dataset is created as shown in table below. I will use this dataset for skill/not_skill classification.

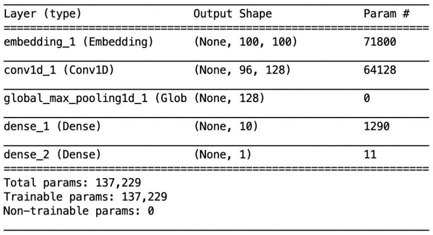


Strong SQL skills	skill
benefits	not_skill
a Data Analytics and Data Science	skill
even a desk to work at	not_skill
proven ability to influence technical leaders	not_skill
the role for you	not_skill
Hadoop and Spark	skill
a commercial environment	not_skill

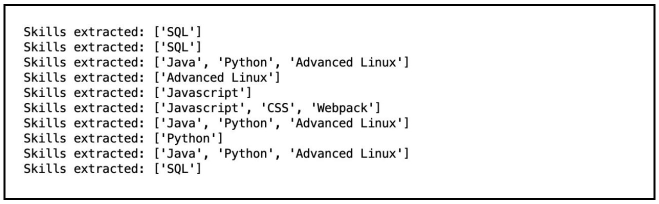
The training data was labelled manually. I created a training dataset of ~1k such noun phrases. The phrases were labelled by plain intuition and not by domain expertise. There are few phrases like '*proven ability to influence technical leaders'* which represent a skill, but were labelled as non-skill since there were no data science related tools/tokens mentioned in the phrase.

5.3 Word Embeddings + Convolution

A simple word embeddings based classifier (diagram below) was trained on our newly prepared dataset from 5.2. New noun phrases are created from a new job post that we want to extract skills from. The new noun phrases are checked against our model. All the phrases classified as *skill* are then selected for noun keyword extraction. These are our extracted skills.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_1 (Embedding)      (None, 100, 100)          71800     
_________________________________________________________________
conv1d_1 (Conv1D)            (None, 96, 128)           64128     
_________________________________________________________________
global_max_pooling1d_1 (Glob (None, 128)               0         
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290      
_________________________________________________________________
dense_2 (Dense)              (None, 1)                 11        
=================================================================
Total params: 137,229
Trainable params: 137,229
Non-trainable params: 0
_________________________________________________________________
```

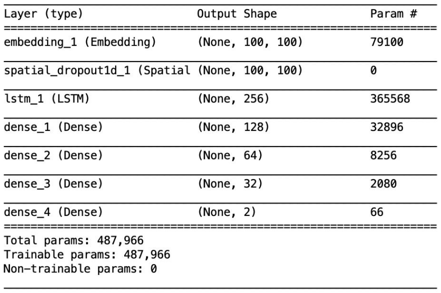


```
Skills extracted: ['SQL']
Skills extracted: ['SQL']
Skills extracted: ['Java', 'Python', 'Advanced Linux']
Skills extracted: ['Advanced Linux']
Skills extracted: ['Javascript']
Skills extracted: ['Javascript', 'CSS', 'Webpack']
Skills extracted: ['Java', 'Python', 'Advanced Linux']
Skills extracted: ['Python']
Skills extracted: ['Java', 'Python', 'Advanced Linux']
Skills extracted: ['SQL']
```

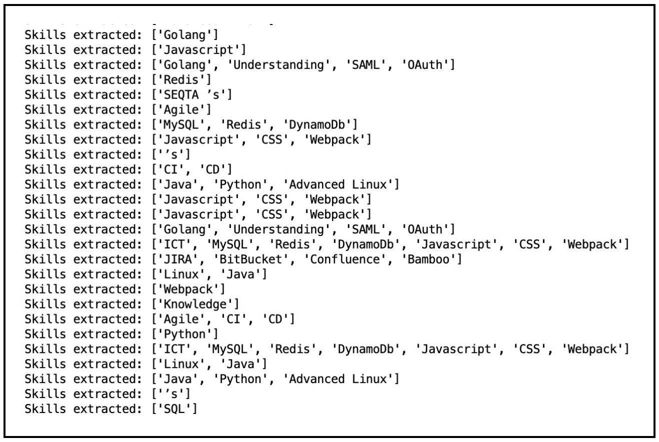
The word embedding were able to extract a lot of useful skills from the job post.

5.4 LSTM + Word embeddings

LSTM is a deep learning technique which is very popular with text data. Using the word embeddings along with LSTM improves the accuracy of the skill classification and also extracts a lot more keywords from the same job post used in section 5.3.



```
_________________________________________________________________
Layer (type)                 Output Shape              Param #   
=================================================================
embedding_1 (Embedding)      (None, 100, 100)          79100     
_________________________________________________________________
spatial_dropout1d_1 (Spatial (None, 100, 100)          0         
_________________________________________________________________
lstm_1 (LSTM)                (None, 256)               365568    
_________________________________________________________________
dense_1 (Dense)              (None, 128)               32896     
_________________________________________________________________
dense_2 (Dense)              (None, 64)                8256      
_________________________________________________________________
dense_3 (Dense)              (None, 32)                2080      
_________________________________________________________________
dense_4 (Dense)              (None, 2)                 66        
=================================================================
Total params: 487,966
Trainable params: 487,966
Non-trainable params: 0
_________________________________________________________________
```



```
Skills extracted: ['Golang']
Skills extracted: ['Javascript']
Skills extracted: ['Golang', 'Understanding', 'SAML', 'OAuth']
Skills extracted: ['Redis']
Skills extracted: ['SEQTA ’s']
Skills extracted: ['Agile']
Skills extracted: ['MySQL', 'Redis', 'DynamoDb']
Skills extracted: ['Javascript', 'CSS', 'Webpack']
Skills extracted: ['’s']
Skills extracted: ['CI', 'CD']
Skills extracted: ['Java', 'Python', 'Advanced Linux']
Skills extracted: ['Javascript', 'CSS', 'Webpack']
Skills extracted: ['Javascript', 'CSS', 'Webpack']
Skills extracted: ['Golang', 'Understanding', 'SAML', 'OAuth']
Skills extracted: ['ICT', 'MySQL', 'Redis', 'DynamoDb', 'Javascript', 'CSS', 'Webpack']
Skills extracted: ['JIRA', 'BitBucket', 'Confluence', 'Bamboo']
Skills extracted: ['Linux', 'Java']
Skills extracted: ['Webpack']
Skills extracted: ['Knowledge']
Skills extracted: ['Agile', 'CI', 'CD']
Skills extracted: ['Python']
Skills extracted: ['ICT', 'MySQL', 'Redis', 'DynamoDb', 'Javascript', 'CSS', 'Webpack']
Skills extracted: ['Linux', 'Java']
Skills extracted: ['Java', 'Python', 'Advanced Linux']
Skills extracted: ['’s']
Skills extracted: ['SQL']
```

6 Comparing Results

LSTM combined with Word embeddings provided us the best results on the same test job posts. The training data was also a very small dataset and still provided very decent results in Skill extraction. More data would improve the accuracy of the model.

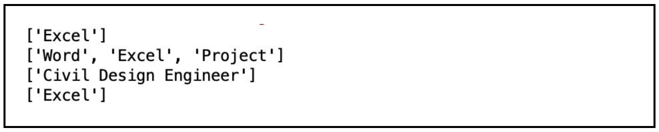
Approach	Accuracy	Pros	Cons
Topic modelling	n/a	Few good keywords	Very limited Skills extracted
Word2Vec	n/a	More Skills extracted compared to Topic Modelling	Lot of noise introduced
Word embeddings	Test accuracy - 0.6803	Lot of relevant skills extracted. No extra feature engineering for curating training data.	Still misses some skills
LSTM + Embeddings	Test accuracy - 0.7658	Best skill extraction. No extra feature engineering for curating training data.	Best so far

7 Extending to different job category

The model was trained on Data Science category and I wanted to test the same model on categories other than Data Science.

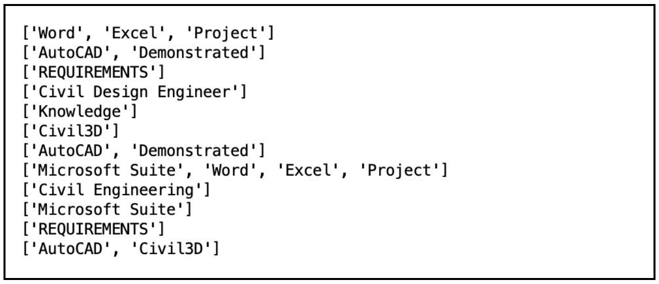
The same model was applied to a Civil Engineer job post and the results were very satisfying. The reason for the consistent accuracy of the model is because of the same grammar structure of the job post. All these job posts were written in similar language structure where Skills were represented by Noun phrases.

Skills extracted via word embeddings network:



```
['Excel']
['Word', 'Excel', 'Project']
['Civil Design Engineer']
['Excel']
```

Skills extracted via LSTM + Word embeddings network:



```
['Word', 'Excel', 'Project']
['AutoCAD', 'Demonstrated']
['REQUIREMENTS']
['Civil Design Engineer']
['Knowledge']
['Civil3D']
['AutoCAD', 'Demonstrated']
['Microsoft Suite', 'Word', 'Excel', 'Project']
['Civil Engineering']
['Microsoft Suite']
['REQUIREMENTS']
['AutoCAD', 'Civil3D']
```

8 Limitations of model

Since the core training dataset for the model is Noun phrases, if our job post doesn't follow the same language structure the model performance would be worse. Lot of job posts are represented by Verb phrases, and in that case we would have to create a new training dataset with verb phrases and create a new model.

The LSTM based model also has a very small amount of noise extracted, eg. *requirements*, *empty quotes* etc.These can be avoided by adding some text cleanup or keywords cleanup.

9 Future work

Future scope would be to try out the techniques on Job profiles defined by verb phrases. Another future task is to try Sentence embeddings [8] instead of Word embeddings to see if it can generalise for multiple cross job categories.

10 Conclusion

LSTM and Word embeddings are a powerful and simple way to extract useful information from our text corpuses. The network was able to provide decent results by training on a very small dataset and can be extended to other job categories.

The model also doesn't use a static list of Job Skills, or a static Skill classifier. This enables the model to pick up new skills rather than being limited to a set of known skills.

11 Acknowledgement

The work was possible by taking pointers from lot of blogs and existing whitepapers. All the blogs and papers are references in section 12 - References.

I acknowledge the Job portals from where I fetched the sample job postings for training the model. The data has only been used for my personal research and has been used within the privacy agreement provided on the website.

I also acknowledge the awesome python libraries Keras, Pandas, Nltk, Spacy, Sklearn etc.These are great libraries that made the research possible.

12 References

[1] ong Short Term Memory [https://en.wikipedia.org/wiki/Long_short-term_memory Wikipedia]

[2] Word embeddings [https://en.wikipedia.org/wiki/Word_embedding Wikipedia]

[3] Job search portal [https://seek.com.au]

[4] opic modelling [https://en.wikipedia.org/wiki/Topic_model Wikipedia]

[5] Word2Vec [https://en.wikipedia.org/wiki/Word2vec Wikipedia]

[6] Intuition Engineering, Deep learning for specific information extraction from unstructured texts [https://towardsdatascience.com/deep-learning-for-specific-information-extraction-from-unstructured-texts-12c5b9dceada]

[7] Part of Speech [https://en.wikipedia.org/wiki/Part_of_speech Wikipedia]

[8] Sentence embeddings [https://en.wikipedia.org/wiki/Sentence_embedding Wikipedia]

Author Information

Nikita Sharma, Student, Master of Data Science and Innovation, University of Technology Sydney - UTS, Sydney, Australia. Sept 2019.