Domain-Specific Skill Extraction from Job Postings: A Pattern Tree-Based Approach

Zeinab Taqavi Nasab

Department of Computer Engineering, University of Mazandaran, Babolsar, Iran

zbtaqavi@gmail.com

1. Abstract

Extracting and classifying skills in job advertisements has been a challenging task for some time. The importance of skill extraction lies in its application to job matching platforms and the development of extensive job ontologies, which are crucial for automating the recruitment process and placing candidates in the most suitable positions. While large NLP models have dominated this area, in this study, a novel methodology is introduced to score words in a vocabulary based on their probability of being a skill or knowledge term. We utilized syntactic Stanza parser to extract two versions of pattern trees that represent the dependency parse structures surrounding skills and knowledge within a text. These syntactical patterns were then searched within a dependency index of a large corpus of unlabeled, parsed job advertisements. A comparison between this syntactic approach and the NLP models JobBERT and JobSpanBERT from Zhang et al. (2022), whose labeled dataset SkillSpan is employed in this study, demonstrates that comparable results to these models can be achieved without the extensive training required by these models.

Key words: Skill Extraction, Pattern tree

2. Introduction

The problem of extraction skills from job advertisements or CVs has many solutions today with ML models like BERT and its variants working at semantic concept level showing the best results. Despite the rapid growth and success of these ML models, alternative methodologies continue to be worth exploring. For instance, Zhang et al. (2022) employed 3.2 million job advertisement sentences for domain-adaptive pretraining, resulting in the creation of the JobBERT and JobSpanBERT language models. These models achieved F1 scores of 63.88 for predicting knowledge spans and 56.64 for predicting skill spans on the test set. However, considering the substantial time and computational resources required to obtain these results, it is pertinent to question whether there are more cost-effective approaches.

To look at skill extraction as a domain specific IR problem, one traditional solution in an IR system is to use weighted keyword lists as a vocabulary to aid the process of retrieving skill words. For this purpose it's needed to have a scoring system that scores words based on the probability of them being a skill word. In this work the proposed method is to look for grammatical patterns around skill words. It is arguable that words can be identified if they appear with a repeating grammatical pattern in a text like a job advertisement.

To implement this approach, we utilize Stanza, a Python library for the Stanford Parser developed by de Marneffe et al. (2008), which enables the discovery of syntactic dependencies between words in sentences from job advertisements. Initially, the SkillSpan dataset from Zhang et al. (2022), a labeled dataset containing tokens and BIO labels that identify skill and knowledge words and phrases, is used to discover these grammatical patterns. Subsequently, an additional dataset of unlabeled job advertisements is employed to construct a dependency index, in which skill and knowledge terms are searched. The source code for this study has been made publicly available.[1]

3. Related work

The task of skill extraction from job postings has garnered significant attention from researchers, especially with the recent advancements in natural language processing (NLP) models. While these advancements have led to notable improvements, there remain challenges that require further exploration.

In the domain of NLP, Zhang et al. (2022) demonstrated the effectiveness of the BERT (Devlin et al., 2019) base model for skill extraction, achieving enhanced results through adaptive pre-training and refining the model to operate at a span level. This approach has shown promise in capturing more nuanced information from job postings. Similarly, Decorte et al. (2022) extended this line of work by employing RoBERTa (Liu et al.,2019), a variant of BERT, and framing the task as a multi-label classification problem, further advancing the state-of-the-art in skill extraction. Before that Sayfullina et al. (2018) tried to represent soft skills by using methods like soft skill masking and tagging, along with CNN and LSTM (Hochreiter et al.1997) models. Tamburri et al. (2020) reformulate the problem as skill extraction at sentence level. These sentences were from job advertisements and CVs. They predict whether a sentence contains a skill or not.

In the category of skill embedding approaches to build skill taxonomies and ontologies, Javed et al. (2017) utilized word2vector to build a named entity normalization (NEN) system for detecting and normalizing occupational skills from job advertisements and CVs. They used a skills tagger that leverages semantic word vectors to recognize relevant skills. Li et al. (2020) also build their taxonomy including job titles and skills to present a special format of job postings from Linkedin. They used FastText (Joulin et al. 2016) to represent entities of their taxonomy.

Additionally, with the objective of creating a job matching system, Gugnani et al. (2020) utilized Word2Vec to extract skills from job advertisements, subsequently employing Doc2Vec to represent both job advertisements and CVs.

On a different front, the extraction of grammatical patterns has been explored by Sari et al. (2009), who utilized the Stanford Part-of-Speech (POS) Tagger and the Link Grammar Parser (LG) to perform Named Entity Recognition (NER). Their work highlights the potential of syntactic approaches in identifying relevant patterns within text, which can be crucial for tasks such as skill extraction.

4. Methodology

[1] Link for source code: https://colab.research.google.com/drive/1iPAhiM6dFBW-fZScvswlQ5jdcNi6S8zS?usp=sharing

To improve domain specific IR systems, one way is to use Custom Relevance Scoring algorithms. Our goal is to find a vocabulary for skill and knowledge words and rate each word with the probability that it would be a skill or knowledge word. This algorithm takes two datasets. First the SKillSpan labeled dataset is utilized to find grammatical structures that are presented as pattern trees. Two version of pattern trees are constructed which has different approaches to weight their grammatical branches. A sample pattern tree from v1 is presented in figure 1.

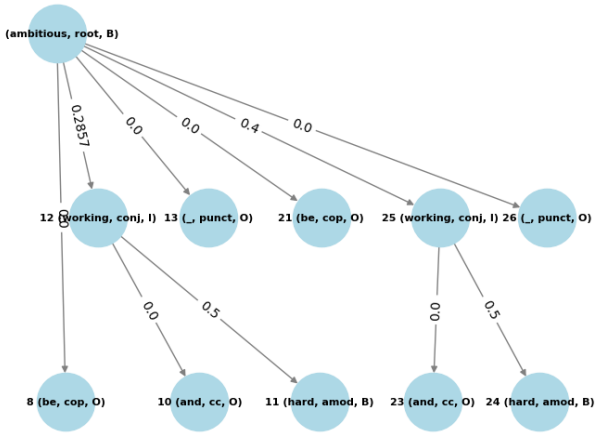

Figure 1. *Pattern Tree for headword 'ambitious' version 1*

To build these trees First job is to find headwords from the dataset. Headwords are the parent of I or B labeled words in a dependency parse tree of a sentence in a job advertisement. The stanza python library is used to build the dependency parse trees. The next step is to join subtrees of all instances of a headword and give weight to their edges. The weights of the edges are equal to the ratio of the number of B or I nodes in the sub-tree of the destination node to the number of B or I nodes in the sub-tree of the source node of that edge. The total number of pattern trees extracted from SkillSpan is 376.

For V2 we define pattern trees in a different way. For each instance of a headword there are children which have different kinds of dependencies to the parent. in dependency parse tree, there are dependency paths which may or may not lead to a B or I node. in v2 every node except root is a dependency and it has a weight that equals to the ratio of number of that path leads to a B or I node to number of all repetition of that path for that headword in all sentences. For Example in Figure 2 it is shown that for the headword ‘administration’ there is a (administration → conj → compound) path with weight 0.333.

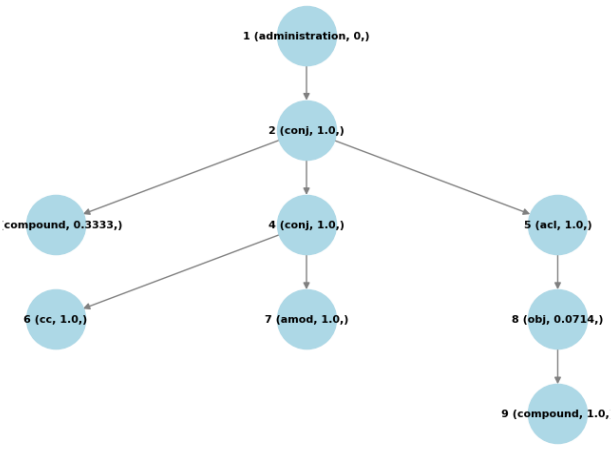

Figure 2. *Pattern Tree for headword 'administration' Version 2*

The unlabeled dataset is used to make an index. for this job we took the dependency parse of all job postings. It contains all extracted terms from job postings. Each term contains the sentence id of which they appeared in and the terms that are their children in the dependency parse tree of that sentence and each term's kind of dependency. With This structure it is able to walk through all paths of a sentence parse tree over both terms and dependencies.

The construction of these two data structures enables the implementation of the algorithm to ultimately build the vocabulary. vocabulary is the output of algorithm 1 which contains words encountered in the algorithm along with the scores assigned to them. In algorithm 1 every headword is searched in the index and for each occurrence all paths in both the sentence sub-tree and the pattern tree are traversed. We look for similar paths from the headword to its children. For each child with similar dependency to the dependency of the current branch of the pattern tree we add the weight for that branch to the words score in the vocabulary. This process is detailed in algorithm 1. With two versions of pattern trees, Algorithm 1 repeatedly takes one version as input at a time. Consequently, two vocabularies are generated, and their scores are compared across four evaluation tests.

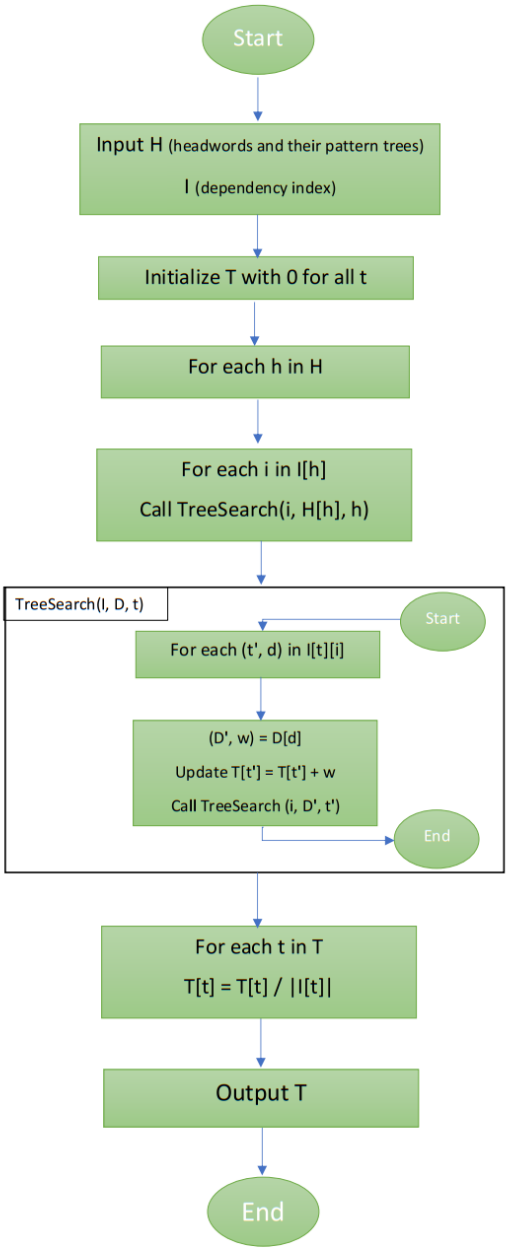

***Figure 3**. Flowchart of Algorithm 1*

5. Dataset

Two datasets were utilized in this study. The first is the SkillSpan[2] dataset, which comprises 200 job postings, containing 5,866 sentences and 122,608 words in the training set. The test set includes 101 job postings, with 4,680 sentences and 57,528 tokens. SkillSpan is a labeled dataset employed in this work to identify the pattern trees. In addition

[2] Linke for SkillSpan dataset: https://github.com/kris927b/SkillSpan.git

to SkillSpan, an unlabeled dataset named "US Job Postings from 2023-05-05[3]" was also used. This dataset contains 33,000 raw job postings, of which the first 1,000 job postings were selected for this study. This unlabeled data was used to construct an index, similar to a traditional inverted index. Subsequently, the vocabulary will be generated from the terms within this index and scored using the pattern trees as outlined in Algorithm 1.

6. Experiments

6.1. Evaluation

In a job matching platform job postings are often tagged with the skills required for that job. We now want to test to extract these skills from a job posting. for validation the labeled dataset is split to 2. one for constructing pattern trees and the rest for tests. To be more precise five-fold cross-validation is applied to measure the f1-score for different thresholds. To assess this, four tests are performed which are repeated for each two versions of the vocabularies. single word test for skills and knowledges. and phrase test. In a single word test we behave every word with a B, I or O tag as a single word and compare their assigned score to their tags. Considering F1-score, precision and recall, multiple values for threshold are tested to find the best for each 4 of tests.

For phrase tests, skill or non-skill and knowledge or non-knowledge phrases are made by joining words from the validation set. a skill phrase is a phrase made of B or I tagged words where non-skill phrases are made of O tagged words. A phrase score is defined as the ratio of the sum of the scores of the words within the phrase to the length of the phrase. Phrase tests are also repeated to find the best threshold.

The results of evaluation are given in figure 3. the average performance of the algorithm in F1 for 8 validation tests are shown. For each test the result is calculated with the best threshold. As mentioned before, all four primary tests are duplicated to assess the 2 versions of scoring with two kinds of pattern trees.

	V1	V2
Single skill word	0.5173738167176085	0.523807358695685
Single knowledge word	0.4689511864017243	0.4786140261516357
Skill phrase	0.46148719613938294	0.4788161019179381
Knowledge phrase	0.37666073453971616	0.42642937524970803

Table 1. *performance of the Syntactic Method with 5-fold cross-validation with 2 versions of scoring the vocabulary*

6.2. Analysis of F1 Score Across Varying Thresholds

In the evaluation, five-fold cross-validation was employed to determine the optimal threshold at which the method performs best across eight validation tests. A detailed

[3] Link for US Job Postings from 2023-05-05: https://www.kaggle.com/datasets/techmap/us-job-postings-from-2023-05-05

report of the averaged F1-scores from five runs, with the applied thresholds, is presented in Table 2.

		Version 1		Version 2	
		threshold	F1-score	threshold	F1-score
Detect skills	Single Term	0.12	0.5170	0.07	0.5233
		0.13	0.5173	0.08	0.5237
		0.14❊	0.5173	0.09❊	0.5238
		0.15	0.5172	0.1	0.5232
		0.17	0.5172	0.11	0.5221
	Phrase	0.22❊	0.4614	0.06	0.4751
		0.21	0.4509	0.08❊	0.4788
		0.14	0.4492	0.09	0.4753
Detect knowledge	Single Term	0.47	0.4549	0.17	0.4702
		1.41*	0.4689	0.28	0.4788
		1.51	0.4638	0.3*	0.5041
		1.8	0.4654	0.38	0.4702
	Phrase	0.77	0.3460	0.17	0.4021
		0.88	0.3521	0.19	0.4120
		1.85	0.3746	0.23	0.4250
		1.96*	0.3766	0.24*	0.4264
		2.45	0.3502	0.36	0.3553

Table 2. *performance of the Syntactic Method with different Tresholds in 8 Evaluation Tests. For each Test the Best Threshold Is Marked with *.*

6.3. Syntactic Method vs. JobBERT & JobSpanBERT

The syntactic method utilized the SkillSpan dataset to construct vocabularies for skills and knowledge. Zhang et al. (2022) introduced the JobBERT and JobSpanBERT models, which are variants of the BERT base model fine-tuned using the SkillSpan dataset. A comparative analysis of these models in single-task learning (STL) revealed minimal differences in performance when compared to the syntactic method. The results of both approaches are presented in Table 3.

Given the extensive corpus of data and the significant amount of time required to develop these models, it is noteworthy that the syntactic method achieved comparable performance with considerably less effort and time. This suggests that the syntactic method holds promise as an efficient alternative.

For comparison, F1 scores for phrase detection were measured against the single-task learning (STL) phase of the JobBERT and JobSpanBERT models. In Version 1 of the syntactic method, a threshold of 0.08 was applied for skill detection, and a threshold

of 0.04 was used for knowledge detection. In Version 2, the thresholds were set at 0.06 for skills and 0.02 for knowledge. The pattern trees were extracted from the training set of the SkillSpan dataset, and the resulting vocabulary was evaluated on both the development and test sets of SkillSpan.

		Skill detection	Knowledge detection
validation	JobBERT(STL)	60.05	60.66
	jobSpanBERT(STL)	60.07	59.47
	Syntactic Method V1	58.869	59.202
	Syntactic Method V2	60.747	59.424
test	JobBERT(STL)	56.11	63.88
	jobSpanBERT(STL)	56.64	61.06
	Syntactic Method V1	51.902	51.770
	Syntactic Method V2	54.488	51.649

Table 2. *Comparison of Syntactic Method vs. JobBERT and JobSpanBERT models Based on F1-Score*

7. Conclusion

In this study we tried to find skills in job advertisements which can be later applied to build larger job matching platforms. This work states that extracting syntactical patterns around skill and knowledge terms and phrases can be a good guide to identify them in unlabeled job advertisements. Our method works because job advertisements, even with varied vocabulary, tend to use similar grammar. For feature works it is promising that along with identifying skills and splitting them in general classes of skill and knowledge, comprehensive classification of skills become feasible based on this methodology or with combinations to other approaches.

Refrences

Mike Zhang, Kristian Nørgaard Jensen, Sif Dam Sonniks, Barbara Plank. 2022. SKILLSPAN: Hard and Soft Skill Extraction from English Job Postings. arXiv:2204.12811 [cs.CL], https://doi.org/10.48550/arXiv.2204.12811

Smith, E., Weiler, A., Braschler, M. (2021). Skill Extraction for Domain-Specific Text Retrieval in a Job-Matching Platform. In: Candan, K.S., et al. Experimental IR Meets Multilinguality, Multimodality, and Interaction. CLEF 2021. Lecture Notes in Computer Science(), vol 12880. Springer, Cham. https://doi.org/10.1007/978-3-030-85251-1_10

Jens-Joris Decorte, Jeroen Van Hautte, Johannes Deleu, Chris Develder, Thomas Demeester. 2022. Design of Negative Sampling Strategies for Distantly Supervised Skill Extraction. arXiv:2209.05987 [cs.CL], https://doi.org/10.48550/arXiv.2209.05987

Y. Sari, M. F. Hassan and N. Zamin, "Creating Extraction Pattern by Combining Part of Speech Tagger and Grammatical Parser," 2009 International Conference on Computer Technology and Development, Kota Kinabalu, Malaysia, 2009, pp. 515-519, doi: 10.1109/ICCTD.2009.227.

Elena Senger, Mike Zhang, Rob van der Goot, and Barbara Plank. 2024. Deep Learning-based Computational Job Market Analysis: A Survey on Skill Extraction and Classification from Job Postings. In *Proceedings of the First Workshop on Natural Language Processing for Human Resources (NLP4HR 2024)*, pages 1–15, St. Julian's, Malta. Association for Computational Linguistics.

Luiza Sayfullina, Eric Malmi, and Juho Kannala. 2018. Learning representations for soft skill matching. In *International Conference on Analysis of Images, Social Networks and Texts*, pages 141–152.

Sepp Hochreiter, Jürgen Schmidhuber, and Corso Elvezia. 1997. Long short-term memory. *Neural Computation*, 9(8):1735–1780.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Liu Zhuang, Lin Wayne, Shi Ya, and Zhao Jun. 2021. A Robustly Optimized BERT Pre-training Approach with Post-training. In *Proceedings of the 20th Chinese National Conference on Computational Linguistics*, pages 1218–1227, Huhhot, China. Chinese Information Processing Society of China.

Marie-Catherine de Marneffe and Christopher D. Manning. 2008. The Stanford Typed Dependencies Representation. In *Coling 2008: Proceedings of the workshop on Cross-Framework and Cross-Domain Parser Evaluation*, pages 1–8, Manchester, UK. Coling 2008 Organizing Committee.

Damian A Tamburri, Willem-Jan Van Den Heuvel, and Martin Garriga. 2020. Dataops for societal intelligence: a data pipeline for labor market skills extraction and matching. In *2020 IEEE 21st International Conference on Information Reuse and Integration for Data Science (IRI)*, pages 391–394. IEEE.

Hoang, P., Mahoney, T., Javed, F. and McNair, M. 2018. Large-Scale Occupational Skills Normalization for Online Recruitment. *AI Magazine*. 39, 1 (Mar. 2018), 5-14. DOI:https://doi.org/10.1609/aimag.v39i1.2775.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. arXiv preprint arXiv:1607.01759 (2016).

Shan Li, Baoxu Shi, Jaewon Yang, Ji Yan, Shuai Wang, Fei Chen, and Qi He. 2020. Deep Job Understanding at LinkedIn. In Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '20). Association for Computing Machinery, New York, NY, USA, 2145–2148. https://doi.org/10.1145/3397271.3401403

Gugnani, A. and Misra, H. 2020. Implicit Skills Extraction Using Document Embedding and Its Use in Job Recommendation. *Proceedings of the AAAI Conference on Artificial Intelligence*. 34, 08 (Apr. 2020), 13286-13293. DOI:https://doi.org/10.1609/aaai.v34i08.7038.