Systematic Literature Review on Skills Identification

Hang Song 22/04/2020

Section 0: Introduction

In this document, we investigated established works on the field of skill identification from unstructured data such as job postings and resumes. At first, the general and major challenges in this area of study are discussed. Then with these challenges, we studied how and to what extent these challenges were addressed in the related literature. We will then compare and contrast different approaches carried so far as well as corresponding future works in improving their performances. Finally, the pros and cons are listed in a table for a direct visual comparison.

Section 1: Challenges on skills extraction

In this section, we will list the major challenges in extracting skills entities from text documents like job postings.

The first challenge would be difficulties in dealing with unstructured data. Unlike structured data which fits neatly into databases that are organized by predefined categories like name, address, occupation etc., unstructured data, which in our case is job postings, consists of paragraphs written by companies recruiting for talents with a certain level of freedom. To be more specific, some challenges accompanied would be difficult to locate the paragraph where the company's requirement for skills. Although most job postings would have Jobs Requirement section, it is also possible that additional desirable skills are appearing in other sections of the job posting, e.g. What to expect. Secondly, the job posting may not be as well-formatted as company's internal archives where regularisation is applied. For example, the writer may include tables, bullet points or other formats of texts to make the posting visually pleasant, but this makes the parser difficult to fetch the content.

The second challenge would be the word ambiguity. There are many ambiguities in phrases used in job postings and it varies from field to field. Specifically, some skills can be presented in different ways, e.g. C++ and CPP, C# and C Sharp, which contributes to another challenge of duplication and would be a major obstacle when analyzing the transforming trend of digitisation and how new skill requirements evolve against time. In contrast, there are also the same terms expressing different skills under different contexts (word sense ambiguity). For example, the term organ can be the surgery experience of a doctor but can also mean the instrument that a musician can play, and ZooKeepers can mean Apache Zookeepers as well as actual keeps for a Zoo. To make things more complicated, it can also appear in the resume of a physician that performed an organ surgery but also plays the organ. The third type of ambiguity would be acronyms. For example, BI can be short for Bank Indonesia and also for Business Intelligence,

which leads to the fourth type of ambiguity – skills being overlapped with other known entities. For example, Oracle is both a domain of expertise as well as a company name.

The third major challenge would be duplication. There is one type that is already mentioned in the earlier paragraph – multiple expressions for the same skill. A more common type of duplication would be typos and different ways of writing according to different companies, and also acronyms for skills. For example, companies may use OOP to refer to object-oriented programming. This challenge would need major attention in improving the performance of the project as having duplication affects strongly on the trend analysis.

Section 2: Related Works

There are currently a few pieces of literature on this field and the earliest can be traced back to 1988 (Salton1988), where the author investigated approaches to automatically weight terms to retrieve texts. While more recent works have combined machine learning and Natural Language Processing and word embedding techniques to perform more accurate systems to identify, extract and normalize skill entities from different unstructured input sources like job postings, resumes and paragraphs of texts. In this section, we will focus on how different pieces of literature addressed the common challenges faced in this study. One thing to note that some paper also focuses on the recommendation part of this field, namely matching the candidates' skills with job requirements and building the career path for the candidate through their connections as well as dream jobs. These contents are not included in the following discussions.

In chronological order, we will start from Kivimaki et al study in 2013, where they presented a system that performs skill extraction from text documents by generating the semantic similarity between input text and Wikipedia and then returns a list of skills from a list of Wikipedia links using biased Spreading Activation algorithm. Their study utilizes the skills dictionary from LinkedIn while deleting the skill entities without a hyperlink to avoid ambiguity, as well as API calls for Wikipedia pages for identifying entities as skills. Specific to their study, they encountered the problem of retrieving certain hubs in the Wikipedia link graph no matter what the input is. To address this, they used a biased spreading activation algorithm rather than a regular one. For word sense disambiguation, the spreading activation algorithm is a huge help as it measures conceptual and document similarity when applied to Wikipedia hyperlink graphs. One thing to note is that at the time this paper is published, the word2vec module is not publicly available. The author instead uses a vector space model to embed the phrases. This aspect may be improved now by training the embedding module specifically using job posting data. The authors set the aspects for their future works in terms of improvements and expansions. The LSA and LDA models need further meta-parameter tuning and it is promising to compute the link weights based on contents of linked pages. The author is also interested in extending the system to other languages as well as using alternative data sources. Another thing to note is that the website hosting the GUI of the system is currently down.

Another related paper is Bastion2014 which discussed a large-scale topic extraction pipeline that was applied to LinkedIn in the 'Skills and Expertise' section, the paper then worked on implementing the recommendation system according to the extracted and normalized skills. To

address the issue of ambiguity, where there are multiple meanings of phrases, the authors counted for the phrase frequencies of occurrences in different sections to address this. Furthermore, they generated a set of related phrased of each skill phrase, to catch the content and context to target the exact word sense. For example, when the word 'Organ' refers to the instrument there might be other music-related phrase appearing near the word 'Organ'. The author then used clustering technique for Word Sense Disambiguation. In particular, most phrases would have 1 dominating cluster as they only have 1 meaning, and 2 or 3 clusters when they can represent multiple skills. The clusters are then labelled by industries according to their frequency. The deduplication process is addressed through the rule that when 2 skills are directing to the same Wikipedia page, these 2 skills are considered as the same skill. Finally, this tagging mechanism is scaled through crowdsourcing. For example, the worker is shown the key attributes of a fetched skill and then need to decide among a list of Wikipedia pages which one is the most suitable to disambiguate word sense. The major data source for this system is LinkedIn member profiles, Wikipedia as well as mturk crowdsourcing for scaling. Besides the existing study, the author is also interested in learning how folksonomy evolves and how to best utilize knowledge encoded in it. Like the other paper, the author is interested in extending their study into other languages. For Word Sense Disambiguation, the author is willing to leverage Wikipedia skills in their future works.

A further paper in 2015 by Zhao et al., where they presented a system called SKILL which identifies and normalizes skill entities from job postings and resumes. This is in application inside CareerBuilder. There are 2 components inside the system, namely skills taxonomy generation and skills tagging. Their major data source comes from the postings and applications on CareerBuilder.com, user survey for data-driven evaluation and ESCO as gold-standard for evaluating their system. In their study, their major challenge comes from ambiguities, where they addressed this problem using word2vec embeddings which are trained specifically to skills data. Their workflow is formatted as follows: skill-related contents are first collected and partitioned according to punctuations and sections etc., followed by initial cleaning with an empirical threshold set to remove noise. The seed skill phrases are then assessed by calling Wikipedia API to assemble as surface forms. There is no inference included in this stage as the main purpose is to collect data. These skill surface forms are then parsed into the skills tagging module and embedded as vectors for clustering according to relevancy score. However, there is not much work in this version of the system in terms of addressing Word Sense Disambiguation, excepting for using Wikipedia pages and google search rankings where SVM can be mapped to Support Vector Machine and Zookeeper is always mapped to Apache ZooKeeper. The author of this paper is interested in working on the Hierarchical structure of skills, in terms of having ontologies and graphs of connections in between skills.

A newer version of the research of SKILL system in CareerBuilder is presented in Javed2017 paper. Comparing to the earlier version, the advancements lie in a newly implemented system for Word Sense Disambiguation. In particular, the author implemented Macau to assign the most appropriate skill sense to multi-sense skill entity concerning a given context. One advantage is that compared to other clustering algorithms, Macau does not need the number of clusters

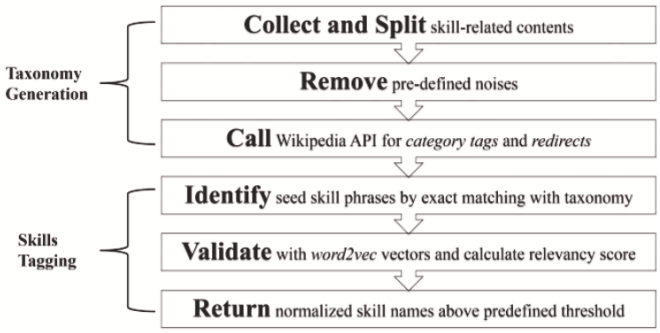

Figure 1: Pipeline schematic of Zhao2015.

to be predefined. Furthermore, the tagging system is redesigned where the relevancy score is assigned a more even distribution for better rankings and WSD is integrated into the tagging module. The data source is consistent with the earlier version and so is the evaluation system. The author is interested in investigating case-sensitive tagging to improve false positives, as well as building a more comprehensive skill hierarchy as the skills are interconnected with each other and also inclusive. Furthermore, when applying WSD to a large-scale dataset, the author plans to use MCMC clustering algorithm to parallelize the process.

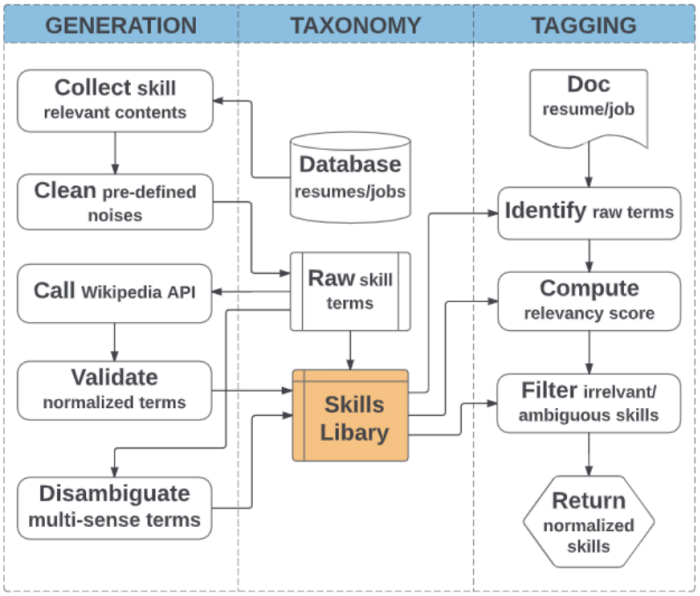

Figure 2: Pipeline schematic of Javed2017.

An open-source system for extracting skills and classifying occupations using pre-defined ontologies is introduced by a team of researchers from the University of Chicago (Crockett et al.). The system integrated different stages in the study, namely NLP for parsing text inputs, word embedding to map skill phrases into vector spaces, as well as training ML model for classifications into a packed library for miscellaneous used to investigate labour market-related studies. The training of the ML model involves crowdsourcing of manual labelling of a text corpus. Other data sources include O*NET, ESCO ontologies and other open-source job posting data. The system uses multiple word embedding models including word2vec, doc2vec and FastText. In this sys-

tem, the workflow is as following: the job posting data is pre-processed to generate text corpus for training etc. followed by embedding into vector space. Then the matching between the result of the earlier stage with pre-defined ontology is carried out and the result can be iteratively fed into the system to enrich the ontology. Together with the skill phrase, the surrounding text is also fetched for disambiguation purposes, while the detailed approaches are not addressed in the article.

The newest literature is written by Gugnani et al. in February 2020, where the author presented a job recommender system with the first stage being extracting skill-related phrases from job posting data. The research is carried with data coming from various open-source websites which contain 1.1M job descriptions in more than 50 domains. The author used data set from Maheshwary and Misra (2018) for evaluating the performance of their system, as well as using the skills dictionary from O*Net and ComputerHope which includes 53293 skills. The workflow is as follows: The raw text is parsed through 4 different modules: Named Entity Recognition, Part of Speech Tagger, Word2Vec embedder and Skill dictionary from O*Net etc. The outcome from the 4 modules is weighted individually and combined as an overall metric to define whether or not the given phrase is a skill. An innovative approach this paper includes is that it considers the possibilities of implicit skills, i.e. the skills might be required by this position but now shown in the description. This attribute is fetched through cross-reference between similar job postings and identify the missing skills. The author is interested in expanding the dataset given time.

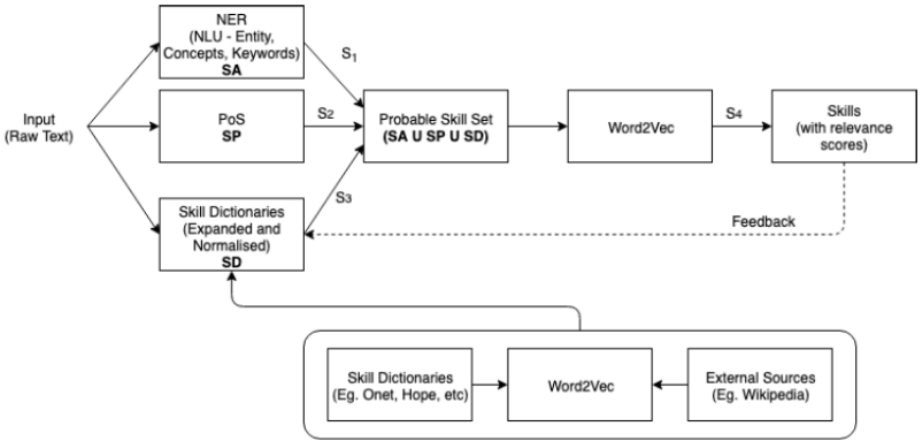

Figure 3: Pipeline schematic of Gugnani2020.

Section 3: General comparison

From the literature review carried out earlier, it can be seen that in addressing the study of extracting skills from the unstructured text input document, the approaches evolve from matching keywords to utilising advanced machine learning models to process linguistics, implement word sense disambiguation and de-duplication. Although not all the processes are fully automated and there are still manual tuning and empirical thresholds are set, document are crowdsourced for labelling, the field has seen much advancements and the scale of the database is constantly

expanding. For targeting phrases as skills, researchers mapped the phrases into a vector space of semantics to perform techniques like clustering, with the help of surrounding words. There is literature about cross-referencing and disambiguation that could improve the performance of the skill identification systems significantly. Now with the related works investigated, we could have a better look at the data with purposes. After understanding the data, we have, we can start carefully designing the pipeline from collecting and generating text corpus and taxonomy to designing, pruning and refining the ontology to be used to target down the skill phrase. We can then have an initial assessment of the first model and target on specifically which part after improvement could give the best performance etc.

Section 4: Pros and Cons

In this section, we will assess the pros and cons of the literature we studied. Note that not all literature listed above are presented due to the lack of source code and information in the paper for assessment. Furthermore, the pros and cons listed here is a superficial initial assessment and is not based on the actual performance of the different systems and models introduced earlier. Besides, the metrics of different models in the evaluation should not be compared as they are based on different evaluation rules and dataset. Unfortunately, at this stage, we do not have other solutions for comparison.

Author	Year	Pros	Cons
Kivimaki	2013	• Logical flow of normalization • Real-time extraction • Fast web service (down currently)	• LinkedIn skill without hyperlinks are removed, leading to reduced size of skill dictionary • biased testing set due to mismatch of skill phrases • not fully automatic optimization scheme • limited number of models tested • modules not trained specifically for job postings (~)
Bastian	2014	• through deduplication process • disambiguation • large-scale dataset • regularized tag input	• clustering metric is not properly justified • does not consider typo (~)
Gugnani	2020	• disambiguation • multiple NLP modules implemented to normalize and filter	• manual tuning and weight assignment