June 03, 2016

System for recommending job titles based on user provided titles and categories

Xuejun Tao

Hao Jiang

Roger Hernandez Jr

Mark Rivera

Christian Posse

See next page for additional authors

Inventor(s)
Xuejun Tao, Hao Jiang, Roger Hernandez Jr, Mark Rivera, Christian Posse, Maria Helena Buhr, and Sarah Lin

SYSTEM FOR RECOMMENDING JOB TITLES BASED ON USER-PROVIDED TITLES AND CATEGORIES

ABSTRACT

The present disclosure relates to a system and method of recommending job titles based on user-provided titles and categories. The system leverages an occupational ontology with multiple levels such as a job title, minor occupational category and major occupational category. The system receives job title or category as inputs from a user. It then classifies the job title or job category into one or more minor job categories. The system then uses a number of recommendation algorithms to generate several job titles. It then presents recommended job titles to the user in one final recommendation list. The user interface could either display all recommended job titles or a subset of job titles. In a variation of the method, the system could sense the user's reactions to the recommended titles to improve recommendation algorithms over time.

BACKGROUND

Generally, there are several reasons why job title recommendations are important for job search and job-matching applications. Many job seekers are not aware of specific job titles. This is especially true for millennials and entry level job candidates. Imperfect knowledge about job titles creates a problem for online job searches because users do not know which titles to search for in the application. One possible solution is to recommend job titles to users based on their interests in general job categories. Furthermore, job title recommendations could help more experienced job seekers because job titles could be characteristic to specific organizations or even divisions of the same organization. Job seekers often have imperfect knowledge about what a relevant job may be called in different organizations, and therefore, could miss out on job

opportunities in online job searches. For example, “Software Developer”, “Software Engineer”, “SWE”, “SW Developer”, “SW Engineer”, “Application Developer”, “App Developer”, “Java Developer”, and “Mobile Developer” are all titles that could be used to describe one and the same role. A job seeker using any one of these titles to search online might fail to see relevant opportunities with the other titles. Furthermore, job title recommendations could help large organizations screen many possible job requisitions. Recruiters run the risk of searching for experienced candidates by means of one title but may fail to identify a good candidate who chose to use a different title on the resume.

In synonym expansions, search engines often use synonyms as a method to expand search queries to gather a larger set of relevant data. This method works well in cases where multiple search terms are near perfect synonyms. However, in cases where there are subtle differences based on search terms, users may benefit from being able to use their own discretion to select suggested search terms instead of auto populating synonyms. This is particularly the case in job searches, where small differences in terminology may spell the difference between an ideal result and one that is suboptimal. For example, consider a case where two software developers – one an expert in Java programming and the other in Python programming – enter the same search term, “Software Developer”. A potential synonym for this search term, “Java Developer”, would be highly relevant for one user but not for the other. It is impossible for the system to determine whether this synonym should be included in the search without additional information either provided by user direction or inferred from the user’s profile, online behavior, history, or context. Indeed, in this case, its inclusion would result in the addition of many irrelevant results for one user, but its exclusion would result in poor recall for the other job seeker.

In collaborative filtering, some online search and recommendation systems make use of collaborative filtering to address this issue, by suggesting content that other users with similar interests have chosen. This solution, too, imperfectly addresses the problem, when collaborative data is sparse and few users have co-viewed items or when additional data is required to disambiguate search results as the above example of Java vs. Python developers. Thus, the job seeker would benefit by an improved method of recommending job titles based on user provided titles and categories.

DESCRIPTION

The present disclosure relates to a system of recommending job titles based on user-provided titles and categories. The system leverages an occupational ontology as shown in FIG. 1.

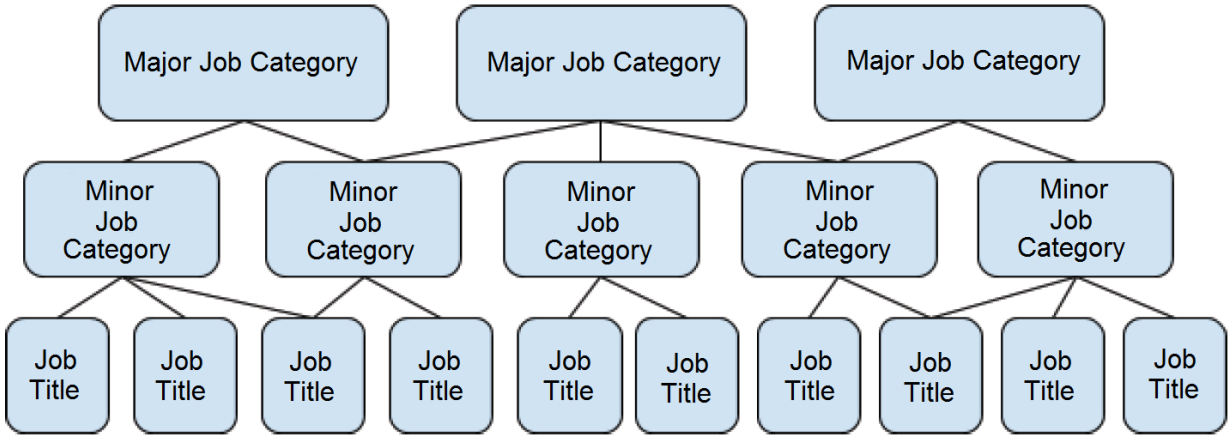

Figure 1: Occupational ontology

The ontology has multiple levels. In the given example, there are three levels, specifically, job title, minor occupational category and major occupational category. Job title is the most detailed level in the ontology. Examples of job titles are “emergency room registered nurse” and “java developer”. Minor occupational category consists of one or more job titles. For

example, the job title “emergency room registered nurse” could be part of a minor occupational category called “registered nurse”. Similarly, “java developer” could be part of the minor occupational category “software developer”. Major occupational category consists of one or more minor occupational categories. For example, the minor occupational category “registered nurse” could be part of the major occupational category “healthcare”. The minor occupational category “software developer” could be part of the major category “computer or IT”.

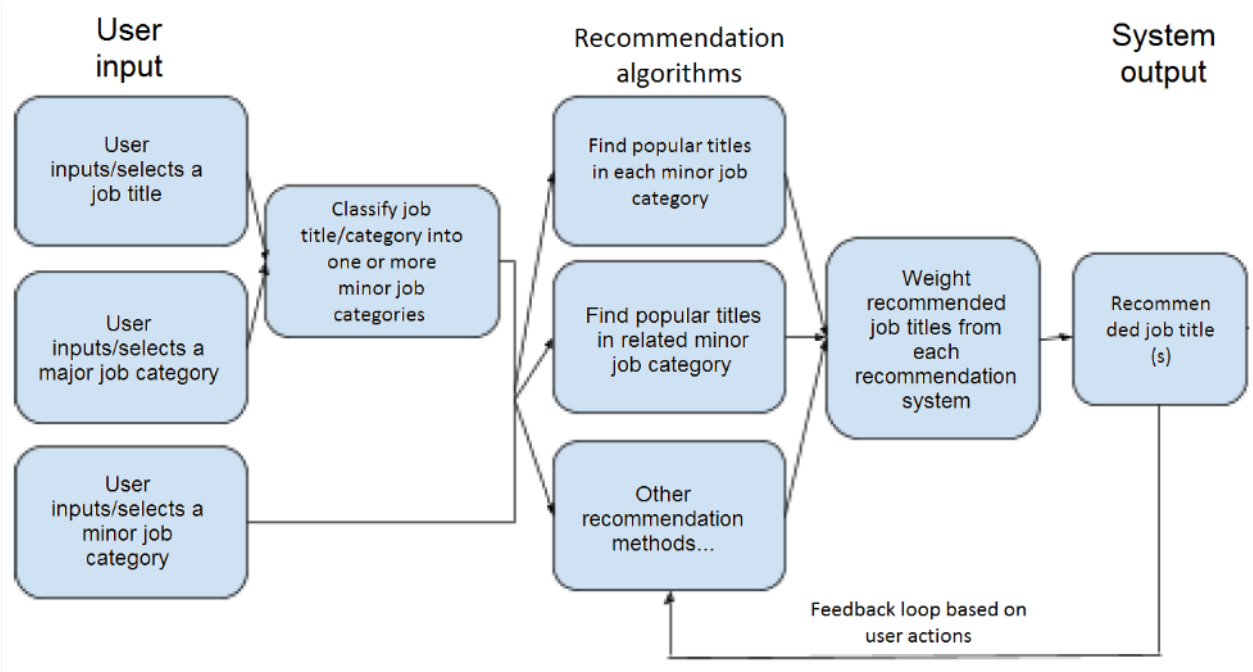

Figure 2: Flowchart describing how the system recommends jobs

As shown in FIG. 2, the flowchart describes how the system recommends job titles based on the given input. The method follows the steps detailed here:

Step 1: Receive user inputs. The system accepts three forms of input such as a job title, a minor occupational category or a major occupational category as earlier discussed with reference to FIG. 1.

Step 2: Classify job titles and major occupational categories into the one or more minor occupational categories. Minor categories are powerful inputs for our recommendation algorithms because these categories have high within-category homogeneity with respect to skills and job requirements.

Step 3: The system uses an assembly of several different recommendation algorithms. Each recommendation algorithm could use information about minor occupational categories to identify job title recommendations. Examples of algorithms include (a) popular titles in the same occupational category or (b) popular titles in related occupational categories. The framework can be extended to any number of recommendation algorithms.

Step 4: Weight recommendations from different recommendation algorithms and combine the job titles into one final recommendation list.

Step 5: Present recommended job titles to the user with options for user input. The options may call for marking a particular job title as positive, neutral or negative.

Step 6: In this step, the user's inputs are fed back to the system to improve the recommendation algorithms.

The recommendation system thus generates one or more recommended job titles to the user that are generated in a structured way. The user interface could either display all possible recommendations obtained from the system, or only present a subset of job titles at a time. The user's inputs in terms of positive, neutral or negative reactions when seeing recommended titles could be used to improve the recommendation algorithms over time.