Cognitive-aware Short-text Understanding for Inferring Professions

Sayna Esmailzadeh
Iran University of Science and Technology
Saeid Hosseini (sahosseini@su.edu.om)
Sohar University
Sara Kamran
Iran University of Science and Technology
Mohammad Reza Kangavari
Iran University of Science and Technology
Wen Hua
University of Queensland

Research Article

Keywords: Inferring Profession, Cognitive-aware Short-text Understanding, Extracting Linguistic Features, Cognitive-semantic, Occupation Corpus

Posted Date: May 25th, 2022

DOI: https://doi.org/10.21203/rs.3.rs-1682137/v1

Cognitive-aware Short-text Understanding for Inferring Professions

Sayna Esmailzadeh[1], Saeid Hosseini[2], Sara Kamran[1], Mohammad Reza Kangavari[1] and Wen Hua[3]

[1]School of Computer Engineering, Iran University of Science and Technology, Iran.
[2]Faculty of Computing and Information Technology, Sohar University, Oman.
[3]School of IT and Electrical Eng., University of Queensland, Australia.

Contributing authors: sayna.esmailzadeh@gmail.com; saeid.hosseini@uq.net.au; sara.kamran72@gmail.com; kangavari@iust.ac.ir; w.hua@uq.edu.au;

Abstract

Leveraging short-text contents to estimate the occupation of microblog authors has significant gains in many applications. Yet challenges abound. Firstly brief textual contents come with excessive lexical noise that makes the inference problem challenging. Secondly, cognitive-semantics are not evident, and important linguistic features are latent in short-text contents. Thirdly, it is hard to measure the correlation between the cognitive short-text semantics and the features pertaining to various occupations. We argue that the multi-aspect cognitive features are needed to correctly associate short-text contents to a particular job and discover suitable people for the careers. To this end, we devise a novel framework that on the one hand, can infer short-text contents and exploit cognitive features, and on the other hand, fuses various adopted novel algorithms, such as curve fitting, support vector, and boosting modules to better predict the occupation of the authors. The final estimation module manufactures the $\boldsymbol{R^w}$-tree via coherence weight to tune the best outcome in the inferring process. We conduct comprehensive experiments on real-life Twitter data. The experimental results show that compared to other rivals, our cognitive multi-aspect model can achieve a higher performance in the career estimation procedure, where it is inevitable to neglect the contextual semantics of users.

Keywords: Inferring Profession, Cognitive-aware Short-text Understanding, Extracting Linguistic Features, Cognitive-semantic, Occupation Corpus

1 Introduction

Cognitive-semantic approaches [1] combine knowledge from linguistic analytics with intellectual perceptions. Given lexical contents t_j generated by the author a_m, we aim to estimate an occupation o_n through analyzing the cognitive features. This can directly reveal the best person who is mentally fit to handle a particular task. Nowadays, social networks record brief textual contents of the authors that are generated at a high throughput rate. Inferring professions from short-text contents in a cognitive-semantic fashion finds important applications in numerous domains: (i) In crowdsourcing [2], the career module can choose the right crowd to carry out the task [3–5]. (ii) In job recommendation systems, the cognitive module can reflect personality characteristics [6]. (iii)

In workload management, the model can evaluate the suitability of the workforce, which can further improve the efficiency of the team [7]. Inherently, compared to the comprehensive formal documents, individuals tend more frequently to compose informal short-text contents, an easier approach. While the short-text contents can better reveal the linguistic features, we can leverage such informative content to estimate the suitability of professions in real-world scenarios. We address the NP-hard [8] problem of associating the latent cognitive-semantics to the professions in two steps. Firstly, we exploit the linguistic features (e.g. *anger*) and enhance them with cognitive features (e.g. *agreeableness*) to generate the cognitive-semantic vector for each individual. Secondly, we devise a heterogeneous combination of machine learning algorithms to fit the author vectors to the relevant cluster of occupations, where the novel modules are adjusted using the coherence scores. Nevertheless, understanding the professions by cognitive-semantic cues poses certain challenges that we elucidate as follows:

Challenge 1 (Excessive Lexical Noise)

Short-text contents are limited in size and include excessive lexical noise such as various abbreviations and spelling errors, which makes them scientifically tedious to handle [9, 10]. We improve the segmentation algorithm based on the Term Graph (TG) [10], where we consider coherence weight for each term e_t. Also, we adapt Symmetric Conditional Probability (SCP) [11] to evaluate the phrases of up to 5-grams.

Challenge 2 (Hidden Linguistic Features)

Since the words may not carry pertinent instinct meanings, the genuine linguistic features turn latent in short-text contents [12, 13]. To this end, cognitive processes are required to reveal the hidden concepts behind semantics.

Based on the cognitive observations in Table 1, recognizing the meanings can only be comprehensible if the pragmatic human understanding is noted. For instance, while the word *lonely* can reveal the cognitive features of anger and fear, the term *buddy* can highlight the joy in a cognitive perception. Hence, to infer the professions in short-text contents, we should consider the cognitive understanding of the proposed framework.

Challenge 3 (one-to-many correlation between cognitive and job entities)

Furthermore, one aspect of the challenge is that the majority of cognitive features are partially oriented toward multiple professions. Hence, we should devise a model that can infer the pertinence between cognitive features and particular jobs using cognitive insights. To this end, we first extract linguistic features by adopting cognitive textual analytics [14, 15] *(e.g. LIWC)* and subsequently prospect the cognitive features from linguistic features by utilizing the correlation coefficients.

Table 1 Cognitive-semantics

Word	anger	fear	joy
devil, lonely, madness, psychosis	1	1	0
powerful, cash, excite, honest	1	1	1
prune, abstract, account, exercise	0	0	0
purify, absolution, amusing, bride	0	0	1
swim, confidence, elevation, highest	0	1	1

Recent works in inferring professions [16–18] propose the textual relevance between the jobs and the cognitive features. However, they ignore the hidden linguistic features in the microblog noisy context. Furthermore, even though some of the current approaches [19, 20] utilize various machine learning techniques to estimate occupations, they fail to tackle the problem from various aspects, relying only on the Naïve ensemble algorithm [20]).

Challenge 4 (non-linear and multiplex cognitive features)

Inherently, cognitive features involve multiple non-linear characteristics. Therefore, mathematical modeling of heterogeneous approaches like support vector machine, support vector clustering, and Gaussian kernel approaches to specify the occupation boundaries [21] turns non-trivial. Where Lu et al. [22] propose the Gradient Boosting modules to handle non-parametric inputs, but they do not optimize the Mean Squared Error (MSE). Other similar work [23] simultaneously track the feature-specific variations on non-linear input, ignoring the group alignment of the features toward exclusive careers. Hence, we propose a multiplex unified framework that employs three machine learning techniques: SVM-Behavior, Gradboosting, and Isotonic Curve-Fitting module. Such a heterogeneous framework can handle the non-linear cognitive-semantic input from three perspectives. The *SVM-Behavior* module constructs the margins for the occupation clusters

based on the input cognitive features. The Support Vector Cluster (SVC) [21] maps the textual vectors to improve the dimension of the job-specific spaces, which further utilizes the Gaussian behavior kernel. The *Gradboost behavior* module infers the professions using the *Boosting Behavior Tree*. The tree model can adapt the dimension for the occupation cognitive features by dividing and mapping the orients into a binary tree. The *Curve Fitting* module[24] aims to train a mathematical model to estimate the jobs using various cognitive features effectively. Fitted curves can successively visualize the correlation between the cognitive features and each of the inferred professions.

The proposed framework in this paper explores the cognitive cues by extracting the semantic-linguistic features. The multi-component approach employs three aspects of clustering, boosting, and curve fitting to infer the correlation between the cognitive features and professions, which will be collectively adjusted via the coherence weight. We further devise a novel R^w-tree structure that applies quest, update, and insert operations to distinguish occupational boundaries based on the trained weights. In short, we leverage dataset-wide shreds of evidence from noisy short-text contents to exploit the cluster of occupation boundaries. To the best of our knowledge, we propose the first study on inferring professions from microblogs via a cognitive-semantic approach. Our contributions are fourfold:

- We propose an effective segmentation model which firstly utilizes the term-specific cognitive features and computes the correlation scores between each pair of the words and secondly employs an altered stickiness algorithm to extract cognitive-aware multi-words from short-text contents.
- We devise a heterogeneous cognitive-semantic model named LESSN to extract hidden linguistic features and also enhance cognitive-textual analytics to reveal career-related features.
- We propose a multi-component unified framework that can effectively bound the professions to the microblog authors in a cognitive-semantic manner. The experimental results verify the advantage of the recommended solution over other contemporary competitors.
- We develop a novel R^w-tree structure that is consistent with the agglomerative characteristic of the professions where the block indexes in the tree can accommodate the relevant cognitive features for each occupation in the hierarchy.

Table 2 Literature

Category	Approach	Reference
Contextual Semantic	Semantic-Oriented	[10, 25–27]
	Short-text Inference	[27–30]
	Noisy Contents	[9, 28, 29]
	Bag-of-Words	[27, 29, 31]
	Neural Networks	[27, 32, 33]
Inferring Professions	Crowdsourcing	[2, 34]
	Job inferring	[8, 35–38]
Cognitive Perspective	Cognitive features	[39–41]
	Cognitive Analysis	[13–15, 42, 43]

1.1 Paper Organization

We organize the rest of this paper as follows: in Sec. 2, we study the literature; in Sec. 3, we clarify the problem of estimating occupations based on predictive analysis of cognitive cues and elucidate the framework overview; We describe the proposed approaches and experiments in Sec. 4 and Sec. 5 respectively. Finally, we conclude this paper and discuss future work in Sec. 6.

2 Related Work

As briefed in Table 2, the related work comprises contextual semantics, inferring professions, and cognitive perspective.

2.1 Contextual semantics

Contextual semantics [25] can be derived using word co-occurrences and sentiment weighting. The principle behind the concept of contextual semantics comes from the dictum-"You shall know words by the job it keeps!" (Firth, 1930- 1955). Saif et al. [25] design a novel lexicon-based approach using a contextual representation of words from Twitter, called SentiCircles, which can capture the hidden semantics of the words from their co-occurrence patterns and subsequently updates pertinent sentiments. However, they utilize supervised learning approaches that require training data for sentiment classifier learning. Wen Hua et al. [10, 26] harvest lexical-semantic relationships between the terms by applying a probabilistic network on the web corpus. To conduct type detection, they introduce Chain and Pairwise models which combine the effects of lexical. For the disambiguation task, the weighted-vote algorithm determines the most appropriate concept for instance. Moreover, [27] leverages the semantic vector space to enrich short-text contents and subsequently infer the contextual (contents+concepts) links between authors. However,

due to the excessive noise in contents [9, 28, 29], the procedure for short-text understanding seems quite challenging. The current approach [31] employs continuous Bag-of-Words (CBOW) to learn the underlying word representations through neural networks. CBOW retrieves the representation of surrounding words with the middle word. CBOW model is determined to be helpful to the understanding of the textual contents. The results for CBOW can be improved by increasing the size of the training dataset and adjusting a better choice for dimensionality. Furthermore, Deep Neural Network (DNN) models such as Convolutional Neural Network (CNN) [32] and Recurrent Neural Network (RNN) [33] are used to learn the low-dimensional semantic vectors for the query authors. [32] presents a Convolutional Deep Structured Semantic Model (C-DSSM) to exploit semantical relevance through similar vectors in the contextual feature space. Graves et al. [33] convert the syllable orders into word sequences and apply RNN to learn the contextual representations of the input sequences. We employ the cognitive aspect of semantics to infer career information from short-texts. To this end, we firstly extract linguistic features. We then train a multi-component heterogeneous model to compute the similarity between cognitive cues and the ground truth, the occupation tags in the microblog contents.

2.2 Inferring professions

Crowdsourcing [2, 34] is a portmanteau of crowd and outsourcing that distributes the tasks between members on the internet. Nowadays, crowdsourcing platforms (e.g. Amazon Mechanical Turk and Freelancer) address microtasks (e.g. image tagging) that are difficult for computers or expensive to be handled by experts. [34] detects trustworthy workers by considering the social positions and the context of the tasks [36]. Job recommendation models [20, 37] match the user preferences to the jobs. Zhang et al. [20] propose suitable jobs for candidates based on their locations, career levels, and roles. They utilize an ensemble approach by combining Collaborative Filtering (CF) and Content-Based Filtering (CBF) modules. While the CF model [38] works based on the ratings of similar users, the CBF method [44] exploits item-to-item correlations using a continuous bag of the words (CBOW) [27] or continuous skip-gram, which produces a distributed representation of words. Schnitzer et al. [37] propose a personalized job recommendation system that applies word embedding to categorize similar job descriptions. This can significantly reduce the target space for the implementation of the personalized classifier and subsequently improve the real-time recommendation. Smart4Job [8] leverages domain knowledge analytics besides a temporal predictor to provide adequate job boards for the dissemination of a new suggestion. The domain knowledge analysis focuses on the experts in the field, and the semantic clarification of job boards relies on textual analytics on a controlled set of vocabularies. A job board is a website that deals specifically with professions or careers (e.g. Linkedin, Indeed). Similarly, we devise a diligent career estimation system that leverages various machine learning modules. Our proposed framework differs from current works that ignore the relevance between occupations and cognitive cues.

2.3 Cognitive Perspective

Exploiting the cognitive features from textual contents can be used to identify the thoughts, behaviors, and relationships between individuals and subsequently can associate each person with a suitable job [39]. The cognitive-behavioral model [41] considers both behavioral and cognitive metrics to promote the job recommendation process. Lents et al. [40] extract an array of adaptive behaviors that people employ to adjust and thrive within educational and work environments during their occupational lifespan. In other words, [40] utilizes the career self-management (CSM) model on the cognitive features to analyze social cognitive variables, including self-efficacy and outcome expectations. Cognitive features [18] comprise five main dimensions (Openness, Conscientiousness, Extroversion, Agreeableness, Neuroticism) [17]. Tadesse et al. [42] utilize Linguistic Inquiry and Word Count (LIWC) dictionary to extract 85 cognitive-linguistic features. LIWC-based models [14, 43] apply the word frequency approach to extract pronouns and identify the psychological and individualized categories. Also, they employ Structured Programming for Linguistic Cue Extraction (SPLICE) to extract 74 linguistic features. SPLICE is a web-based research tool for calculating linguistic cue values based on dictionaries, part-of-speech tagging, and indices. Moreover, they utilize Social Network Analysis (SNA) to investigate the social structure and node

interactions. SNA [13] infers the personality specifications of the users to study five factors in social networks include network-size [45], betweenness [46], density [45], brokerage [46], and transitivity [47]. Aligned with current works, we collectively consider the emotional and sentiment cues in short-text contents to extract cognitive-linguistic features. Furthermore, we analyze Emoji characters that are more appealing for authors to convey their emotions.

3 Problem Statement

We elucidate preliminary concepts, notations, and the framework overview.

3.1 Definition 1 (short-text content)

Each short-text content in the corpus $t_j \in \mathbb{T}$ has an identity (t_j), associated author $a_m \in \mathbb{A}$, and pertinent occupation ($t_j.o_n$). Accordingly, $\mathbb{T}(\mathbb{T} = \{T_1, T_2, ..., T_m\})$ includes all short-text contents, where T_m delineates the set of short-text contents that are owned by the author a_m, where each t_j contains multiple terms $e_t \in t_j$.

3.2 Definition 2 (linguistic feature)

The cognitive-semantics can be interpreted by the conceptual structure. There each conceptual structure can represent a linguistic feature $l_k \in \mathbb{L}$. Each short-text content t_j contains several linguistic features $t_j.L_k$, where l_k represents the linguistic feature that gain several terms $e_t \in t_j$ and w_k denotes the weight of l_k. Each term e_t belongs to one or more linguistic features. The more the number of terms belonging to a linguistic feature, the higher the weight of the linguistic feature will be. Take *"I'm happy you came to visit my gallery."* as a sample short-text. we can analyze textual content to gains linguistic features for this sentence $\{(positive - sentiment, 0.37), (joy, 0.70), (personal - pronoun, 0.69), (negative - emotion, 0.00), ...\}$.

3.3 Definition 3 (cognitive feature)

The associated linguistic features can be interpreted by multiple cognitive features, where features represent an author's cognitive features $p_q \in \mathbb{P}$ that the $\mathbb{P}$ refers to the five-factor traits that can be mapped to $t_j.P_q$. Each p_q represents the cognitive features and $c_q \in [-1, 1]$ the correlation range of p_q. The c_q of -1 indicates a perfect negative linear relationship between linguistic feature l_k and cognitive feature p_q and the value of 1 specifies an absolute positive linear relationship between variables. Take "I'm happy you came to visit my gallery." short-text content as an example. we can analyze linguistic features to exploit the cognitive set as: $\{(openness, 0.83), (extroversion, 0.86), (neuroticism, -0.81), ...\}$.

3.4 Definition 4 (occupation corpus)

Every occupation corpus, denoted by $o_n \in O$ can determine a distinguished occupation that is variously correlated with cognitive features. Given the collection of cognitive features, where each of them is associated with a single short-text content $t_j.P_q$ possessed by the author a_m, the inference model can estimate a relevant career for a_m.

3.5 Problem definition

(extracting linguistic features): *The linguistic features are hierarchically affected by cognitive-semantics. Given the short-text content t_j, our aims is to extract the related linguistic features l_k to t_j.*

(mapping linguistic cues to cognitive features): *The cognitive features are elucidated by linguistic features. Given the linguistic cues L_k, our aim is to map L_k to the pertinent set of cognitive features P_q.*

(inferring professions): *Given the cognitive features P_q, our aims is to estimate an accurate occupation o_n for a_m.*

3.6 Framework Overview

The problem of inferring professions from short-text contents includes two primary steps: (1) to extract linguistic features L_k from short-text t_j. (2) to map the extracted linguistic feature L_k to the most correlated cognitive cues P_q) that are useful to predict the jobs. Figure 1 illustrates our proposed unified framework that can estimate the occupation of short-text authors through leveraging the cognitive features. In the offline part, we utilize the textual information of the query author to retrieve the cognitive orientation of an individual toward a specific occupation. From one side, we take advantage of segmentation in a Term Graph (TG) [10] to understand the coherence weight for each given term, and from the other side, we utilize the Pointwise Mutual Information to better infer the token-wise co-occurrences. Subsequently, we extract linguistic features that conjointly reveal the language habits and the latent semantic knowledge from author's contents. We utilize cognitive-semantic algorithms such as LIWC, Emoji, SPLICE, sentiment Strength, and

NRC lexical dictionaries to exploit cognitive features. We then map the linguistic features of each short-text $t_j.L_k$ to the pertinent cognitive features $t_j.P_q$. In response, aiming to estimate the professions, we fuse three of our behavior machine learning modules including SVM-Behavior, Gradboost behavior, and the curve fitting to calculate the joint coherence weight. In a nutshell, our proposed framework effectively acts as a triplet information pipeline. We collectively transform the cognitive features into trilateral modules including clustering, boosting, and curve fitting to identify professions, where the novel R^w-tree accommodates occupational boundaries. In the online part, we firstly compute contextual vectors for the query author and exploit pertinent cognitive features. To continue, we employ the update and quest procedures in the efficient R^w-tree to attain the top-k most relevant professions for the given query author.

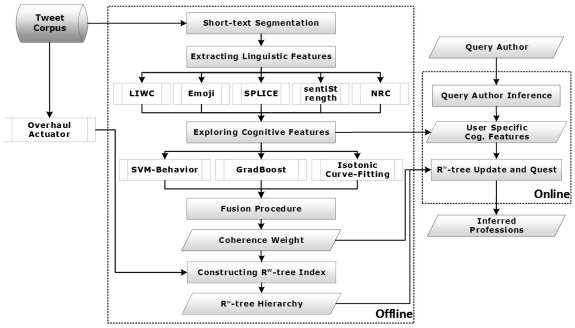

Fig. 1 Framework

4 Methodology

4.1 Offline Processing

4.1.1 Short-text Contents Segmentation

As described in Challenge 1 due to noise and grammatical errors (e.g. slang phrases and ambiguous words), common text mining approaches including tokenization [42] cannot fully obtain either the semantics. To address this issue, we take a three steps procedures: noise reduction, tokenization, and significance computation. For noise reduction, we improve the repeated characters *(e.g. "woooow" to "wow")*, remove emotion irrelevant contents, and apply normalization to enrich the contents. While we use the slang knowledge-base[48] to replace the abbreviations *(e.g. before instead of b4)*, we use the word mover's distance [49] to compute the expansion impact through word vectors.

Hua et al. [10] associate a weight to each segment solely based on the term graph, treating each term as a node and neglecting the value of semantic measures. As verbalized in Eq. 1, we modified the associate score s(x, y) of each segment:

$$s(x,y) = \max_{i,j}(\epsilon, \frac{w(x_i)+w(y_j)}{2} C(x_i,y_j)) \quad (1)$$

Here $\epsilon > 0$ designates a minimum positive score where x_i and y_j are the given terms and $C(x_i, y_j)$ reflects the semantic coherence between the words. we calculate in the offline phase. Also, $w(x_i)$ and $w(y_j)$ denote the weight of semantic-linguistic features. We calculate the edge score as the maximum score between the corresponding terms. Inherently, we distinguish between terms and phrases. A term is a single vocabulary with semantics and is suitable for cognitive features 4.1.2. Term frequency is the number of times a term appears in the contents of an individual author. Similarly, a phrase comprises two or more terms h_p where the terms can not overlap with each other *(i.e.* $\forall t, e_t \cap e_{t+1} = \emptyset$*)*. "bright morning" is a phrase containing two terms of "bright" and "morning". Accordingly, corpus C includes the short-text contents generated by all the authors. Correspondingly, we compute the term significance relying on the term-document matrix that is capable to alter Naïve word combinations with meaningful phrases. In our previous work [11, 50] we empirically present that the Symmetric Conditional Probability (SCP) can exploit the phrases better than the Point-wise Mutual Information. Similarly, as Eq. 2 shows we employ a modified version of the SCP to integrate the corresponding semantic score through term co-occurrence:

$$SCP(x_i,y_j) = log \frac{s(x_i,y_j) Pr(x_i,y_j)^2}{\frac{1}{n-1}\sum_{i,j=1}^{n-1} Pr(x_i)Pr(y_j)} \quad (2)$$

Here x_i and y_j denote the terms, and $Pr()$ designates the probability of the phrase. As verbalized in Eq. 1, the $s(x_i, y_j)$ computes the edge score for the segment. In an inverse term frequency manner, we observe that the higher the popularity of the phrase, the more authors will tend to utilize it. On the one hand we rank all possible segments by SCP, and on the other hand, we remove the terms and phrases that are excessively used by numerous authors. Through such an arrangement,

the SCP measure can eliminate uninformative phrases. Table 3 presents the notations.

Table 3 Table of important symbols

Definition	Notation
sample short-text	t_j
weight corresponding to linguistic features l_kt	w_k
The set of cognitive features for t_j	$t_j.P_q$
Cluster weight	$\hat{\boldsymbol{w}}_{\boldsymbol{cluster}}$
Boosting score	$\hat{\boldsymbol{w}}_{\boldsymbol{boost}}$
Modified isotonic curve	$\hat{\boldsymbol{w}}_{\boldsymbol{curve}}$
Coherence weight	$\hat{w}_C(P_i, o_j)$

4.1.2 Exploring Cognitive Features

Inherently, we can apply the cognitive textual analysis tools on short-text vectors to retrieve the cognitive orientation of the authors. Hence, we argue that external knowledge (e.g. LIWC) [43] is indispensable to correctly retrieve the semantic of short-text vectors. To this end, we can extract linguistic features by using external knowledge of the short-text contents. Consequently, we can reveal the hidden relevance between cognitive and linguistic features. The genuine linguistic features turn latent in short-text contents. The current methods in the extraction of linguistic features [42, 43] fail to attain the latent features from short-text vectors. To address this issue, as illustrated in Fig. 2, we propose a novel model, named as *LESSN*, to disclose both linguistic and cognitive features. Given $T_m = \{t_1, t_2, ..., t_r\}$ as the set of short-text vectors corresponding to author a_m, the LESSN utilizes five complementary knowledge bases including LIWC, EMOJI, SPLICE, SentiStrength, and NRC to cover various aspects that we further equip them with the weights correlated to each of the linguistic features. Every given weight w_k normalizes the linguistic feature frequency for T_m using the same frequency from corpus C.

Linguistic Inquiry and Word Count (LIWC) knowledge-base [42] can extract the terms like $e_t \in t_j$ that are capable to reveal the feelings, thinking styles, and social concerns that are particularly pertinent to the professions $T_m.L_k^{LIWC}$.

EMOJI [51] can constitute the attitude for emotion analysis and investigates the popular emojis for each cognitive feature based on the overall frequency $T_m.L_k^{EMOJI}$.

SPLICE is another linguistic analysis mean [42] that we utilize to extract various features with sentiment poles to report self-evaluation results for

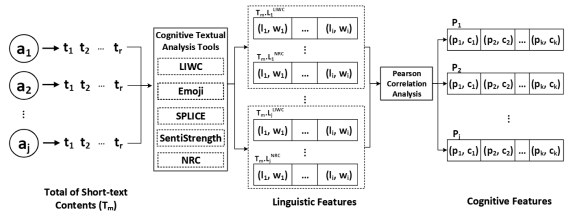

Fig. 2 LESSN: Exploring Cognitive Features

the given author based on complexity and readability scores $T_m.L_k^{SPLICE}$.

SentiStrength [52] is capable to estimate the strength of sentiment positivity in informal short text contents, ranges from -1 (not negative) to -5 (extremely negative) and from 1 (not positive) to 5 (extremely positive). $T_m.L_k^{Senti}$ is the binary output of SentiStrength for the author a_m.

NRC [53] is a lexicon that contains more than 14,000 distinct English words annotated with 8 emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative, positive). To achieve the features $T_m.L_k^{NRC}$ for the author a_m we count the number of terms e_t in each of the emotions and sentiment poles. Successively, we can employ the simple but effective Pearson metric [54] to measure the linear relationship between each weight of linguistic features w_k and the impression scores of the cognitive features. Fig. 2 illustrates how we attain the cognitive features, denoted by $P_q = \{q = 1, ..., k \,|\, (p_q, c_q)\}$. Accordingly, each pair of (p_q, c_q) represents a cognitive feature p_q and its correlation range $c_q \in [-1, 1]$. While c_q renders a linear relationship between l_k and p_q, the values -1 and +1 indicate respective negative and positive absolutes, and the value of zero $c_q = 0$ exhibits irrelevance. Finally, as illustrated in Figure 3, we can invoke three edge weights such as clustering 4.1.3, boosting 4.1.4, and fitting 4.1.5 to collectively compute the association between cognitive features and occupations. Such a portmanteau consisting of three properties can better characterize the correlations between the nodes in two categories of features and professions. While the cluster module automatically groups the features based on the professional boundaries, the supervised boosting model estimates the relevance between cognition and occupation, and the fitting curve can track the variations in input features.

- *Cluster($\hat{\boldsymbol{w}}_{\boldsymbol{cluster}}$)*: explains how the cluster pertaining to each set of cognitive features P_q is

correlated with a specific occupation boundary o_n. To this end, we invoke support vectors of the SVM-behavior to categorize unlabeled cognitive features. The support-vector-clustering can subsequently form the clusters that expose occupational boundaries.

- *Boost($\hat{\boldsymbol{w}}_{\boldsymbol{boost}}$)*: optimizes a loss function based on mean squared error (MSE). In this model, each cognitive feature forms a binary tree, where the goal of the training procedure is to minimize the MSE ratio between the set of trees (a cognitive feature set) and each pertinent occupation.
- *Curve($\hat{\boldsymbol{w}}_{\boldsymbol{curve}}$)*: aligns every non-parametric set of cognitive features to a designated profession. We utilize the common but tractable Isotonic Curve-Fitting (ICF) module to pursue cognitive transformations.

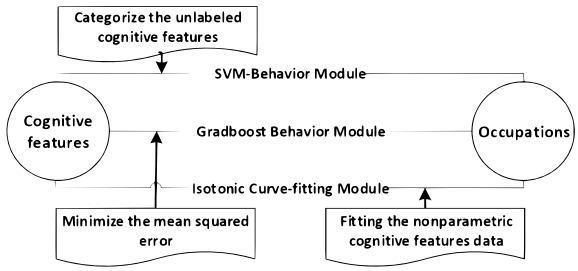

Fig. 3 Weights vector for inferring professions

4.1.3 SVM-Behavior Module

As elucidated in Section 1, the cognitive features can signify a specific profession. However, since the cognitive features are non-linear and multiplex, devising an effective mathematical model to infer pertinent professions is a tedious task. Inherently, it is neither easy to learn the prior knowledge about the occupation clusters nor feasible to estimate the number of such clusters solely based on the noisy short-texts. Hence, we empirically study the clustering procedure from these two perspectives. On the one hand, we utilize TFIDF [55] statistics to compute the significance of the non-stop terms and on the other hand, we combine the results with the set of pertinent cognitive features, denoted by $\hat{\boldsymbol{w}}_{\boldsymbol{cluster}}$. Therefore, we argue that the cognitive-ware SVM-Behavior module can perceive the professions better than the support vector machine [21]. Also, we devise a jointly supervised classifier named as SVM-Behavior, which not only tracks the cognitive features but also consumes the short-text TFIDF vectors. Accordingly, to overcome the curse of dimensionality, we devise a Support Vector Cluster (SVC) module to improve the feature selection and decrease the number of dimensions. From one side, the SVC module projects the textual vectors to the higher dimensional space of the occupations and from the other side, it mutually asserts the Gaussian kernel to further incorporate cognitive orientations. Given two inputs including the set of cognitive features and short-text TFIDF vectors, respectively denoted by $\vec{P}_i$ and $\vec{d}_j$, we can present the corresponding profession-aware labels by o_i and o_j. As clarified in Eq. 3, the training data consumed by the Gaussian kernel $k(\vec{P}_i, \vec{d}_j)$ can comprise both cognitive features and short-text TFIDF vectors. We can address Eq. 3 using the quadratic algorithms.

$$
\begin{aligned}
max \quad & \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i=1}^{l} \sum_{j=1}^{l} \alpha_i \alpha_j o_i o_j k(\vec{P}_i, \vec{d}_j) \\
s.t \quad & \sum_{i=1}^{l} \alpha_i o_i = 0, \quad 0 \leq \alpha_i \leq C, \quad i = 1, 2, ..., l
\end{aligned}
\tag{3}
$$

Nevertheless, we need to accommodate the optimized threshold of C inside the occupational boundary to attain a smaller-marginal hyperplane. Moreover, we utilize the parameters α_i and α_j, corresponding to the Lagrange multipliers of $\vec{P}_i$ and $\vec{d}_j$, to retrieve the maximum and minimum local boundaries. As Eq. 4 shows, k_F denotes the cognitive Gaussian kernel function with F as the cognitive linear matrix for the given input space and η as the bias parameter to ensure symmetry.

$$
\begin{aligned}
k_F(\vec{P}_i, \vec{d}_j) &= exp\Big(- \eta \parallel \vec{P}_i^{\,T} F - \vec{d}_j^{\,T} F \parallel^2 \Big) \\
&= exp\Big(- \eta \big((\vec{P}_i - \vec{d}_j)^T F F^T (\vec{P}_i - \vec{d}_j) \big) \Big)
\end{aligned}
\tag{4}
$$

For the next stride, the SVC module takes the Gaussian kernel k_F to project the input data points to the higher dimensional cognitive feature space. To promote the efficiency of the SVC model, we further adjust the boundary information of occupations based on the cognitive features. This proves that our supervised classifier can concurrently support both flexibility and scalability. The classifier module is capable of collecting the appropriate boundaries and, at the same time, can reduce redundant noise. Subsequently, the SVC module can retrieve the optimized sphere, ensuring a minimum radius that can comprise

the majority of the projected data samples. Such sphere when mapped back to the data space, can be partitioned into several components, each exposing an isolated cluster of instances. Eq. 5 reflects the constraint on the spherical radius, denoted by R:

$$\begin{aligned} &\min_{R,\lambda,\phi} \frac{1}{2}R^2 + C\sum_{i}^{l}\xi_i \\ &s.t. \quad \parallel \phi(P_i) - \lambda \parallel^2 \leq R^2 + \xi_i \\ &\lambda = \sum_{j}^{l}\alpha_j\phi(d_j) \end{aligned} \tag{5}$$

Here, $\xi_i \geq 0$ optimizes the empty boundary that relaxes the strict condition of non-linear separability so that each input data can be observed inside the occupation boundary. We adjust the threshold C to control the penalty of the noise in input data. λ is the spherical-center of each occupation boundary. We map $\phi(P_i)$ and $\phi(d_j)$ as a non-linear transformation from non-separable input data to the linear space to classify the features corresponding to professions. As Eq. 6 shows, we compute the clustering weights based on the training data, where the weight $\hat{\boldsymbol{w}}_{\boldsymbol{cluster}}$ computed by the SVM-behavior can assess the selection procedure of the cognitive features.

$$\hat{\boldsymbol{w}}_{\boldsymbol{cluster}} = \sum_{j}^{l} a_j o_j - \sum_{i,j}^{l} a_i a_j o_i o_j k_F(\vec{P_i}, \vec{d_j}) \tag{6}$$

Inherently, it is neither easy to learn prior knowledge about the occupation clusters nor feasible to estimate the number of such clusters solely based on the noisy brief contents. To this end, the SVC model adjusts the boundary information of occupations and employs a supervised classifier to reduce redundancy and noise in short-text vectors.

4.1.4 Gradboost Behavior Module

As explained in Section 1, we devise a supervised boosting approach to iteratively improve the profession-related classifiers, where we increase the cognitive weights of the observations that are difficult to associate with a particular occupation and conversely reduce the weights for those that are easily correlated with a particular job. In other words, we employ the gradient boosting module to minimize the mean squared error between the cognitive features and occupations, denoted by $\hat{\boldsymbol{w}}_{\boldsymbol{boost}}$. This module forms an ensemble of K decision trees where each tree, called Boosting

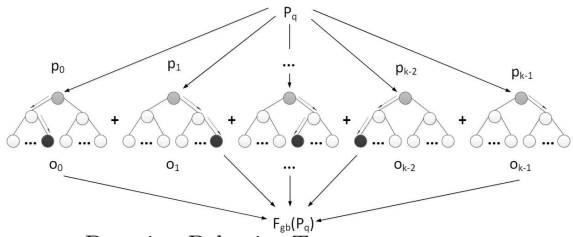

Fig. 4 Boosting Behavior Tree

Behavior Tree (BBT), is tightly correlated with a cognitive vector, subsequently used to estimate a target label. As illustrated in Figure 4, P_q denotes the set of exploited cognitive features, and o_q (e.g. o_1) represents the estimated occupation for the particular cognitive feature (e.g. p_1). Each BBT learns a distinct cognitive attribute and identifies an output gradient space o_q for the input feature. Given the correlation metric between linguistic features and the specific cognitive feature, e.g. p_0, we divide each cognitive node (in gray) into a positive correlation ($c_q \geq 0$) on the left and a negative correlation ($c_q < 0$) on the right. The BBT model successively exploits a suitable occupation (black node) from the set of available jobs and divides the original cognitive feature space R^d into disjoint occupational areas (leaves) where we assign a constant value γ for each region. We compute o_0 by utilizing the loss function $\mathcal{L}(g_i, \gamma)$, which aims to minimize the empirical risk. While g_i delineates the estimated occupation (black node) in Eq. 7, we initialize the supervised learning by a constant value as γ.

$$o_0 = argmin \sum_{i=1}^{n} \mathcal{L}(g_i, \gamma) \tag{7}$$

Eq. 8 and 9 update the approximation of the currently estimated job $o_q^{R_l}$ using the output of the previous tree o_{q-1}. Since each occupation region corresponds to several cognitive features, we select $o_q^{R_l}$ as the most correlated dimension.

$$o_q^{R_l} = argmin \sum_{P_q} \mathcal{L}(g_q, o_{q-1} + h_q) \tag{8}$$

$$o_q = o_{q-1} + h_q.o_q^{R_l} \tag{9}$$

As formulated in Eq. 10, h_q signifies gradient statistic measures for each leaf node. To this end, we employ an iterative procedure where at each round, we use a new cognitive feature tree to associate the input p_q to the L-disjoint output region $\{R_l\}_l^L$. Here, O_m represents the number of leaf nodes.

$$h_q = \sum_{l=1}^{O_m} g_l R_l \tag{10}$$

The BBT method approximates the function F_{gb} as an additive expansion of the cognitive feature learners ($o_i \in \{o_0, o_1, ..., o_{k-1}\}$). We can subsequently determine the best-fitted occupation by applying the simple but effective majority voting on the learning outcomes associated with all of the input feature trees. The boost score $\hat{\boldsymbol{w}}_{\boldsymbol{boost}}$ elucidates the level of minimized MSE between input cognitive features, and the output inferred occupation. For ease of implementation, we first calculate the w_q weight for every input tree p_q (Eq. 11) and in sequence, compute the boosting weight by Eq. 12:

$$w_q = \mathsf{o\_q}(2 - \mathsf{h\_q}) \tag{11}$$

$$\hat{\boldsymbol{w}}_{\boldsymbol{boost}} = \frac{1}{\mathsf{K}} \sum_{q=1}^{k} w_q \tag{12}$$

It is notoriously difficult to learn the weak points to minimize the MSE measure. Hence, the BBT model invokes the cognitive trees to compute behavior functions individually, commencing with equal weights for all cognitive features.

4.1.5 Isotonic Curve-fitting (ICF)

As elucidated in Section 1, many of the cognitive features excessively single out multiple occupations instead of one. The problem can turn up even more challenging when the cognitive features can evolve continuously. Hence, the solution is concerned with a set of non-parametric features with scattered distributions. Therefore, we are required to devise a novel module to direct features toward exclusive professions and simultaneously track the feature-specific variations. Since the cognitive features are inherently nonlinear, the linear regression approaches [23, 42] can not address the challenge. Conversely, we opt for an adopted Isotonic Curve Fitting model, dubbed ICF, that can equip our proposed framework with an unbiased fit to the input cognitive figures. We note that the ICF can successfully form a free-shaping line to follow the feature variations and get aligned toward the real observed data that is not included in the capabilities of the polynomial regressors [56]. However, this approach can encounter computational and statistical overfitting issues in higher dimensions. To address this concern, we build a curve weight between cognitive features data input and occupations $\hat{\boldsymbol{w}}_{\boldsymbol{curve}}$ that can eventually converge to the global isotonic curve. While the proposed modification can often gain better accuracy, it can indirectly resolve the data complexity challenge in the weights of the curve vectors.

As formulated in Eq. 13, given the numerical analysis of the correlation weights for the cognitive features, denoted by P_i, $f_i : [0,1]^d \rightarrow R^d$, our proposed sequential ICF module can retrieve a weighted least-squares fit $P_i \in R^d$ for the corresponding data-oriented vector $\alpha_i \in R^d$ that comes with the weight vector of $w_i \in R^d$.

$$min \sum_{i=1}^{k} w_i (P_i - \alpha_i)^2 \quad \text{where} \quad P_i \leq P_{i+1} \tag{13}$$

Given the observed data $f_i = \{i = 1, 2, ..., k \,|\, (P_i, o_i)\}$, o_i can estimate the pertinent occupation for P_i using Eq. 14:

$$o_i = f_0(P_i) + \mu_{\{i,j\}} \quad \text{where} \quad i \geq 2 \tag{14}$$

Here, $f_0 : [0,1]^d \rightarrow R^d$ is the Borel measure [57], and $\mu_{\{i,j\}}$ denotes the independent noise, the dissimilarity ratio between the occupation, and the cognitive features. The more the occupations are specified by the cognitive features, the less noise will be remarked. Due to excessive noise, we perceive unobserved heterogeneity in occupations that increase the fitting variations and affect generalizability. We bound the noise for the features using the mean and the variance that is respectively denoted by $\mathbb{E}[\mu_{\{i,j\}}]$ and σ^2:

$$Var[\mu_{\{i,j\}}] = \sigma^2, \quad \forall i = P_q \quad \text{and} \quad j = o_n \tag{15}$$

Hence, we utilize the curve weight vectors $\hat{\boldsymbol{w}}_{\boldsymbol{curve}}$ associated with every edge (P_i, o_i) to minimize the uncertainty in the estimation procedure caused by the noise. As Eq. 13 shows, $o_i \in \{o_1, ..., o_k\}$ observes a single instance out of k, corresponding to the weight $w_i \in \{w_1, ..., w_k\}$. We can then estimate the value for each occupation using the relevant cognitive features that are collectively approximated by the combinational variations (Eq. 16):

$$\hat{\boldsymbol{w}}_{\boldsymbol{curve}} \propto \frac{1}{\bar{\sigma}_i^2} = \left[\sum_{i=1}^{k} (o_i - \bar{o}_i)^2 \right]^{-1} \tag{16}$$

While $\bar{o}_i$ is the mean of the occupation responses for the ith cognitive features, o_i denotes the estimated occupation by variable levels.

4.2 Online Processing

In the online phase, we estimate the occupation of the input query author using short-text contents

through two tasks: (i) *Query Author Inference* that collectively processes the contents of the input user and extracts the cognitive features. (ii) *R^w-tree Update and Quest* that firstly refreshes the index of the R^w-tree by the coherence weight that we calculate using the user-specific features in the offline phase. Secondly, given the cognitive features of the input author, we unveil the most relevant occupations.

4.2.1 Query Author Inference

The procedure of Query Author Inference comprises two main tasks: *generating query author segments* and *extracting cognitive features*. To generate the textual segments from query author contents, we firstly combine all short-texts into a single document. The reason is that the model is already built-in the offline section, and the cold-start users that come with limited content can specifically benefit from such merged contentment. We note that posting a few short-text contents can not sufficiently reveal an adequate number of cognitive features to estimate the occupation of a cold-start user. To continue, we carry out noise reduction and tokenization. Similar to Sec. 4.1.1, we utilize the SCP model to extract final segments in the form of output N-Grams, resulting in sufficient semantics. Aiming to explore the cognitive features, we adopt the proposed approach in Sec. 4.1.2, which can be done fairly quickly for the single query author. We initially exploit the latent linguistic features by textual analysis tools and the external knowledge-base. Subsequently, the LESSN module can attain cognitive cues. On the one hand, we integrate all exploited linguistic features as an output set, and on the other hand, the Pearson correlation can retrieve the relationship between linguistic and cognitive features.

What is the usage of the Overhaul Actuator? Inherently, the more varied the exploited linguistic features are, the higher the diversity of the cognitive features will be. The overhaul actuator is triggered by the sense that the level of variations in linguistics passes a threshold, and the framework gets alarmed to rearrange the features within R^w-tree. Such an automatic procedure not only improves the correlation between linguistic and cognitive features but also improves the generalizability of our proposed model by assigning more accurate professions to the unseen forthcoming authors.

4.2.2 Computing Coherence Weight

In this section, we aim to adjust the parameters that can collectively explain the correlation between a set of cognitive features and a single occupation. Inherently, the professions follow a hierarchical pattern. For instance, a pair of occupations, including C# and C++ programmers, firstly inherit from computer engineering. Accordingly, a computer engineer can comprise the cognitive features that belong to both profession leaves: C# and C++ programmers. Adversely, if an individual possesses the cognitive features of a C# programmer, he can be categorized as a computer engineer too. To model the hierarchy, we utilized R^w-tree as a modified version of the R-tree [58]. While the R^w-tree structure is consistent with the agglomerative attribute of occupations, the tree blocks can accommodate the set of cognitive features for each specific occupation.

As verbalized in Eq. 17, the coherence weight, denoted by $\hat{w}_C(P_i, o_j)$ can collectively associate a set of cognitive features P_i to an occupation boundary o_j. Based on Sec. 4.1.2, we can compute the correlations through $\hat{w}_{cluster}$, ($\hat{w}_{boost}$), and $\hat{w}_{curve}$. Instead of naïve tuning, we adopt the simple but effective PageRank algorithm to collectively learn the best values for the bias parameters, α, and β.

$$\hat{w}_C(P_i, o_j) = (1 - \alpha - \beta) \times \hat{w}_{cluster} + \alpha \times \hat{w}_{boost} + \beta \times \hat{w}_{curve} \tag{17}$$

Lemma. *Given an arbitrary number of classes, one can exploit the occupation boundaries via cognitive orientations in an iterative adjustment procedure on R^w-tree.*

Proof: Let the number of arbitrary classes be κ. Hence, there will be a unified final result for constructing κ classes. Based on the coherence weight adjustment $\hat{w}^{\kappa}_C(P_i, o_j)$ on the trilateral scores each result can represent a particular cognitive feature for each of the given κ classes. Thus, there will be $|\kappa|$ decisions associated with each of modules $\hat{w}^{\kappa}_{cluster}$, $\hat{w}^{\kappa}_{boost}$, and $\hat{w}^{\kappa}_{curve}$, where we can collectively maximize the accuracy of the final estimated class in the R^w-tree structure. Consequently, using a repetitive adjustment procedure, we can take advantage of the decision on the R^w-tree structure, commencing with the first score denoted by $\hat{w}^1_C$ on $\kappa = 1$. With this logic, the coherence weight can construct the R^w-tree hierarchy on an arbitrary set of occupations.

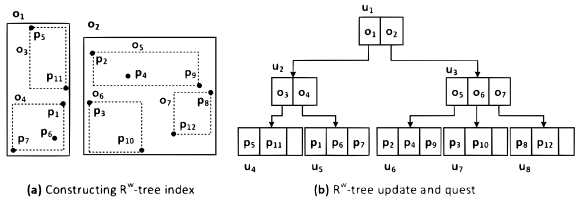

Fig. 5 R^w-tree data structure to gain the occupational rectangle

4.2.3 Constructing R^w-tree Index Hierarchy

We employ a modified version of the R-tree [58] named R^w-tree and utilize collection weight $\hat{w}_C(P_i, o_j)$ to distinguish the relevant boundaries. The R^w-tree aims to find a profession based on a small set of cognitive feature nodes. As Figure 5 illustrates, each p_q denotes a cognitive orientation point, where δ highlights the minimum number of acceptable points in each profession-related rectangle. For instance in Figure 5 (a), where the δ threshold is set to 3, we can nominate the o_4 profession boundary, comprising p_1, p_6, and p_7. In other words, we aim to specify δ number of cognitive points to create an occupation rectangle. We consider a parameter u_i for nodes. The internal u_i keeps the outcome of the Minimum Bounding Rectangle (MBR)[59] for the number of cognitive features.

As algorithm 1 shows the quest and update methods to obtain the list of occupations O_{result} based on the input cognitive features r in our R^w-tree, denoted by $\hat{R}^w$. Where r represents the input cognitive orientations, O_{result} highlights the output list of occupations. Moreover, we use u as the bounding box for the R^w-tree to appoint a cognitive orientation node for each leaf and the occupation node for the parent nodes.

To start, we achieve the list of cognitive features in the bounding box O_{rec} by initializing the rectangles in the R^w-tree(line 2). We use the update and quest procedures to find the suitable occupational boundaries for the input cognitive feature set of r. If the input is not discovered in O_{rec}, we trig the update procedure through the MBR function (line 12) and update the occupational rectangles based on the r and δ threshold. On the contrary, where r exists in the O_{rec}, the quest method (line 8) returns the occupational boundaries that are pertinent to the input cognitive feature set. Where we assert that the node u comes with a child node, we recursively call the quest procedure to reveal all the children in the hierarchy (line 23). Otherwise, where the node u does not have a child (as an occupation rectangle), the search procedure can compare the input r with the properties in the current object $u.P$ (line 27). Eventually, if the resultant output is not empty, the algorithm will return the parent of the node u that represents the profession rectangle (lines 8-9), finalizing in a list of occupations O_{result}.

Overhaul Actuator. Linguistic features may evolve during streaming. Accordingly, the distribution of the cognitive features associated with each of the career boundaries may significantly change in the R^w-tree. Inherently, such skews can affect the dependencies between cognitive features and negatively affect the performance of the

Algorithm 1 R^w-tree update & quest procedures

Input: $\hat{R}^w, r, \delta$
Output: O_{result}

1: $O_{result} \leftarrow []$
2: $O_{rec} \leftarrow initializeRectangle(\hat{R}^w)$
3: **if** r ! $\in O_{rec}$ **then**
4: $\quad \hat{R}^w \leftarrow Update(r)$
5: $\quad O_{rec} \leftarrow initializeRectangle(\hat{R}^w)$
6: **end if**
7: **for** o in $\hat{R}^w.nodes()$ **do**
8: $\quad Quest(o, r)$
9: **end for**
10: **Procedure Update**(r_u):
11: **if** $r_u.count() \geq \delta$ **then**
12: $\quad u' \leftarrow MBR(\hat{R}^w, r_u)$
13: $\quad$ **return** u'
14: **else**
15: $\quad$ print('Minimum δ orientations are required.')
16: $\quad$ **break**
17: **end if**
18: **Procedure Quest**((u_q, r_q)):
19: **if** $u_q.child()$ **then**
20: $\quad$ **for** v in $u_q.children()$ **do**
21: $\quad\quad Quest(v, r_q)$
22: $\quad$ **end for**
23: **else**
24: $\quad$ **if** $r_q == u_q.P$ **then**
25: $\quad\quad o' \leftarrow u_q.parent()$
26: $\quad\quad O_{result}.append(o')$
27: $\quad$ **end if**
28: **end if**
29: **return** O_{result}

profession inference module. Hence the Overhaul Actuator should estimate under what intervals one must re-initiate the R^w-tree. To this end, we employ an adapted divide and expansion method to approximately estimate the actuation interims. As algorithm 2 shows, we first take the whole or a portion of the data $\mathcal{D}$ if the size is massive. We then divide the dataset into two equal parts denoted by $d = \{d[0], d[1]\}$. To commence the evaluation part, we process the first subset $d[0]$ to explore the cognitive features and acquire the occupational boundaries, based on which we can employ any model $Sub_{tree}()$ to exploit the set of sub-trees from $\tilde{R}$. The evaluation function $E_w(\tilde{R})$ can measure the effectiveness ω^0 of the novel job inference module using the proposed benchmark in Section 5.4. Subsequently, we incrementally add up the initial portion $d[0]$ by ζ percent from the second half $d[1]$. We then recompute the effectiveness of ω^i for iteration i. We can use the insufficiency ratio $(\omega^0 - \omega^i)$ to compute the error level in R^w-tree that is owing to the augmented data. Where the MAE size is greater than the threshold ϵ, the sensing procedure can trig the actuation unit to enforce the reconstruction of the modified R^w-tree. Conversely, if the error rate does not exceed ϵ, one can justify that either an incorrect expansion rate has been selected, that resulting in over-fitting, or the error is so small that the next iteration must be followed.

4.2.4 Complexity Comparison: *R*-tree versus R^w-tree

This section compares the ordinary R-tree against our proposed R^w-tree from a theoretical computer science perspective. Intuitively, every basic R-tree can manipulate the indexing rectangles, denoted by o_j, where each of them can specify an optimum number of pertinent cognitive features P_i. Note that our proposed R^w-tree structure dedicates a weight $(\hat{w}_C(P_i, o_j))$ to each o_j, initialized when the occupation node is inserted. Here is where the R^w-tree behaves differently compared to the trivial R-tree. Let γ be the set of the cognitive features $(\mathbb{P})$ and ι as the number of boundary classes (O), in this case, the original R-tree will run in $O(log_{\iota}^{\gamma+\iota})$. As elucidated in Section 4.2.3, our proposed method constructs the R^w-tree that further divides the data into three levels based on the coherence weights. As algorithm 3 shows, we can assign the highest weight of occupations in the middle-level of R^w-tree and reserve the lowest weight for the top-level. Therefore, the expected time for R^w-tree to run each procedure will result in $O(log_{\iota}^{\gamma+\frac{\iota}{2}})$, with an apparent lower complexity. Based on algorithm 3, each $\mathcal{O}_c$ node represents

Algorithm 2 R^w-tree Overhaul Actuator Interval Estimation

Input: $\mathcal{D}, \zeta, \epsilon$
Output: d[1]

1: $d \leftarrow Split(\mathcal{D}, 2)$
2: $\tilde{R} \leftarrow Sub_{tree}(d[0])$
3: $\omega^0 \leftarrow E_w(\tilde{R})$
4: $q_s \leftarrow d[0]$
5: **for** i in Range$(1, \mathsf{d[1]}/\zeta)$ **do**
6: $\quad q_s \leftarrow q_s +$ Subset$(d[1], i)$
7: $\quad \tilde{q}_s \leftarrow Sub(q_s)$
8: $\quad \omega^i \leftarrow \mathbf{E}_w(\tilde{q}_s)$
9: $\quad$ **if** $\omega^0 - \omega^i \geq \epsilon$ **then**
10: $\quad\quad$ **return** $i \times \zeta$
11: $\quad$ **end if**
12: **end for**
13: **return** $\mathsf{d[1]} = 0$

an occupation and comes with the weight, cognitive orientation (the value of γ), and a name. The l parameter specifies the number of levels in the tree. In the insertion procedure, we create a new sub-tree of occupation nodes for each boundary denoted by r, and examine the child nodes (line 4). Additionally, we store the cognitive features of the child node in the last level r^l, the child node of the boundary in the middle-level $r^{(l-1)}$, and the parent node of the boundary in the first level of the object $r^{(l-2)}$. Finally, we return the R^w-tree. By applying the R^w-tree algorithms 1, and 3 we can increase the performance of the procedures over the R-tree algorithm. To get the top-k nodes, we consider the points that the middle-level boundary will gain a higher rank than the top-level. Hence the quest procedure will be divided $(\frac{\iota}{2})$ based on the values in the middle-level.

Scalability. Where the size of the dataset is small, i.e. $|\gamma + \iota| = 10k$, the performance of both trees will be reasonable. However, in big data scenarios, where the number of boundaries augments, the performance of the R-tree gets flattened. But since R^w-tree accommodates the nodes based on the value of the weights in various levels, higher levels with lower weights, and vice versa, the performance of the R^w-tree decreases less. We also

observe that compared to the total number of boundaries in the R-tree, the proposed R^w-tree can execute the construct, updates, and quest procedures based on the levels that are determined by the range of the weights.

Algorithm 3 Insertion in R^w-tree

Input: $\mathcal{O}_c, l, \hat{R}^w$
Output: $\hat{R}^w$

$l \leftarrow l_i$
Procedure Insert($\mathcal{O}_c$):
$r \leftarrow$ new $\hat{R}^w.Subtree()$
if $\mathcal{O}_c.child()$ **then**
  for i in $\mathcal{O}_c.children()$ **do**
    $r^l \leftarrow \{i[orients]\}$
    $r^{l-1} \leftarrow \{i[name], i[weight]\}$
  end for
end if
$r^{l-2} \leftarrow \{\mathcal{O}_c[name], \mathcal{O}_c[weight]\}$
return $\hat{R}^w$

5 Experiments

We conducted extensive experiments on a real-world Twitter dataset [11] to examine the performance of our cognitive-semantic approach in inferring the professions from short-text contents. We used a machine with 64GB of RAM with Intel Core i7-7700K CPU of 4.20GHz. (Code and data)[1].

5.1 Data

We collected the Twitter dataset [26] with more than 2 million tweets through *Twitter API* and selected $5K$ users, and subsequently retrieved up to 300 records from each tweet history. The final dataset contains $76K$ words of $16M$ collocations.

5.2 Effectiveness of Exploring Cognitive Features

As a prerequisite to the estimation of the professions, we need to examine how successfully one can exploit the cognitive cues from short-text vectors (section 5.2). Accordingly, we first compare the performance of the cognitive-semantic approaches. Our proposed model, named LESSN, leverages five cognitive analysis modules, including LIWC, Emoji, SPLICE, sentiStrength, and NRC, (Section 4.1.2). To investigate the performance of competitors, we use a cognitive dataset

[1] https://sites.google.com/view/infer-jobs-cognitive-approach

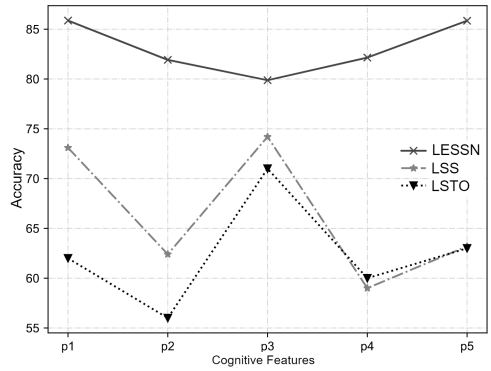

Fig. 6 Performance of the exploring cognitive features models

[60] with a 96-dimension feature space for both individual and combined attributes. We aim to discover the model that, on the one hand, suits best the short-text noisy contents, and on the other hand, is the most competent candidate to infer cognitive features. We compare the LESSN model against LSS [42] and LSTO [13]. The former is equipped with LIWC, SNA, and SPLICE, and the latter takes advantage of the LIWC, SNA, and other Time-related and auxiliary features. Figure 6 compares the accuracy of the competitors where the dimension varies. The LESSN and LSTO respectively gain the highest and the lowest performance. The LESSN, as the most noise-resilient model, overcomes the LSS model because it can better extract the linguistic features in the training stage. The excessive noise in microblog contents can lead to hidden linguistic features that can significantly reduce the performance of LSS and LSTO. This is where the sentiStrength and NRC modules assure better accuracy for the LESSN. The highest and the average accuracies for LSS in the estimation of the cognitive feature are respectively 74.85% and 66.59%, where the same numbers for LSTO are 71.38% and 63.02%. Our proposed framework achieves the accuracy of 86.10% that is approx. 11.25% more than [42].

5.3 Baselines

The baselines to infer professions are as follows:

- $Inprs_{cluster}$: this method [21] uses SVM-Behavior to compute the weights and categorize job boundaries.
- $Inprs_{boost}$: [22] divides each cognitive feature by a single tree to obtain the boosting score.
- $Inprs_{curve}$: is motivated by [24] and generates the curve weight via utilizing the ICF module.

- $Inprs_{conjunct}$: is our proposed novel framework that adjusts parameters (Sec. 4.2.2) to construct the R^w-tree that identifies cognitive profession boundaries.
- *Smart4Job*: this time-aware approach [8] combines the semantic classifier with the time series module to single-out the popularity of the occupations.
- *JobSeeker*: this embedding model [61] automatically extracts the skills relating to each user profile.

Table 4 Baselines Compares

Methods	F-Measure	Precision	Recall
$Inprs_{cluster}$	0.711	0.770	0.657
$Inprs_{boost}$	0.849	0.809	0.890
$Inprs_{curve}$	0.855	0.813	0.898
$Inprs_{conjunct}$	**0.890**	**0.858**	**0.922**
Smart4Job	0.799	0.814	0.783

5.4 Benchmark

Intuitively, we define statistical hypothesis parameters to evaluate the effectiveness of the competitors in inferring the professions. We investigate how the models succeed in assigning the right occupation to each of the microblog users. We dedicate 80% of each dataset for parameter setting and perform the test on the remaining portion. For the ground truth, we rely on two sources: The career tags in the Bio and the job labels that the experts add for the test users. The *Recall* is the number of authors for whom we have successfully assigned a valid occupation. To compute the *Precision*, we divide the number of authors with correct jobs by the total number of users with the assigned profession tags. Finally, we judge the leading method using the *F1-Score*, measured by Precision and Recall [62, 63].

5.5 Effectiveness of the Profession Estimation

As Table 4 reports, this section compares the effectiveness of various competitors (Section 5.3) in inferring the professions using cognitive-semantics cues. For our tripartite framework, $Inprs_{curve}$ overcomes both $Inprs_{cluster}$ and $Inprs_{boost}$ because it tracks job-specific dynamic alterations through the isotonic fitting. However, obtaining the best-adjusted parameters like the number of clusters remains a dilemma for $Inprs_{boost}$ and $Inprs_{cluster}$. Employing the fixed margin to distinguish the career clusters, suddenly changed by an unobserved instance, makes $Inprs_{cluster}$ gain the least performance. The contribution of various aspects makes our framework fit to overcome other rivals.

Inherently, the quantity measure can alter various pertinent boundaries. For instance, the domain knowledge of *computer engineering* can conversely differ from the *secretary*. Therefore, the more specific the profession, the higher the probability will be for the user to get assigned a job, resulting in a higher recall. On the contrary, *Smart4Job* can achieve better F-measure than *JobSeeker* because it utilizes semantic classification, directly increasing the domain knowledge, while *JobSeeker* cannot compete merely via textual analysis tools. Hence, the variation between occupation boundaries is more authoritative than the textual representation of the job-specific tasks. Accordingly, the embedding approaches gain a lower performance compared to the isotonic regressors. Finally, due to the dataset-specific attributes, even the most straightforward boosting models surpasses the *Smart4Job* that comprises both the classification and forecasting modules.

5.6 Effectiveness of Short-text Contents History

In this part, we examine how the data incompleteness can affect the performance of the career inference process. Firstly, we chronologically sort the tweet history of the users. We then eliminate a portion of older content and repeat the evaluation in each iteration. Figure 7 depicts the performance metrics through the removal of three ratios 20%, 30%, and 50%. We observe that where we remove up to 30% of the contents, disregarding the recall, which behaves as overfitted, the precision increases continuously and, we reach the best F1-Score in 70%, i.e. 0.96. We may further improve the performance by removing the contents that are irrelevant to the cognitive cues, that is why the effectiveness of our proposed model fosters compared to the entire corpus. This can be due to the trends within data, where the older textual contents can mislead the pertaining cues of the jobs. Moreover, the noisy event-oriented sentences (e.g. about a new year) in the range of 100%-80% can negatively influence the recall metric. In a nutshell, more data does not assure a better result, especially when the amount of career-independent or event-oriented contents are abundant.

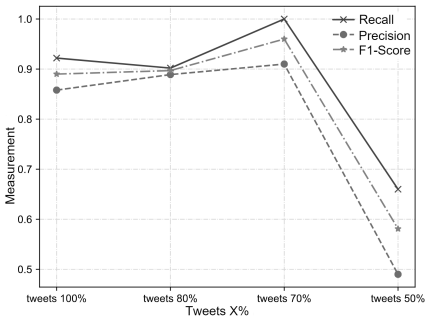

Fig. 7 Impact of short-text history

5.7 Impact of Parameter Adjustment

It is inherently appealing to estimate professions by utilizing various data perspectives through different algorithms, where we incorporate three modules in our framework: (i) Cluster attribute ($\hat{w}_{cluster}$) as an SVM-Behavior module that considers the distribution impact of the cognitive feature clusters. (ii) Boost attribute($\hat{w}_{boost}$) that appoints a distinct decision tree for each cognitive cue. (iii) Curve attribute $\hat{w}_{curve}$ that avoids any negligence of the odd cognitive features through Isotonic Curve-Fitting. Nevertheless, the parameter adjustments can play a key role in promoting our proposed framework $Inprs_{conjunct}$. Hence, we should include impact of every module separately. In other words, the more optimized parameters are, the better the performance will be.

Adjusting α and β: Figure 8 illustrates a schematic of the tuning procedure to select the best values for α and β. We adopted performance metrics to ensure that our method could surpass other baselines. Concerning Eq. 17, we adjusted α and β (ranging $\in$ [0,1]) to compute the best coherence score ($\hat{w}_C(P_i, o_j)$) for each profession o_j based on the given input set of cognitive features P_i. We used F1-Score to choose the best performance as it implicitly comprises both precision and recall measures. We further conducted a separate experiment for each pair of α and β and used a random set of cognitive features. The best value for α and β are respectively 0.35 and 0.5, which is reported for the best F1-score 0.890.

5.8 Impact of R^w-tree on Efficiency

In this part, we compare the efficiency of the R^w-tree versus the original R-tree in the construction of job boundaries. Fig. 9 depicts that as the value for boundaries increases, the latency of all two methods slightly grows. The reason is that

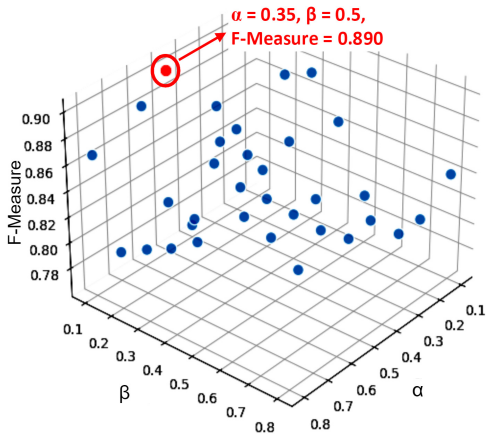

Fig. 8 Impact of Parameters Adjustment

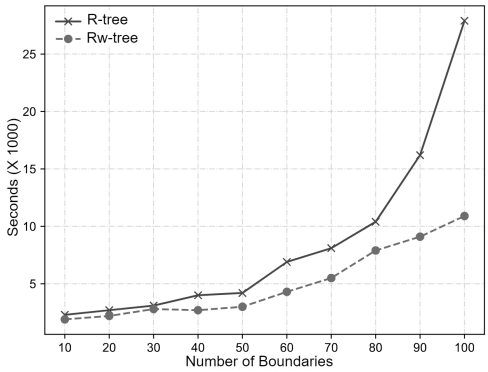

Fig. 9 Compare Efficiency Between R^w-tree and R-tree

choosing a bigger value for boundaries increases the size of the cognitive features, which results in higher processing times. The results show that our model can efficiently exploit an execution time equal to 11.2 ms for 100 boundaries, Where R-tree takes 27.5 millisecond. The reason is that R^w-tree separates the nodes with higher-weights from lower-weights in a separate layer and processes the operations for higher weight nodes and delay the rest.

6 Conclusion

In this work, we devise a unified cognitive-semantic framework that consumes short-text contents to predict career boundaries. To this end, we firstly extract linguistic features from temporal-textual data using the proposed textual analysis tool (LESSN), a better understanding of contextual semantics. More specifically, the LESSN model considers linguistic features to categorize the similarity between short-text vectors in semantic dimensions, for example, destruct and hate in the anger dimension, and then explores

the cognitive features, such as openness, extraversion, and etcetera. Correspondingly, we attain the career boundaries using tripartite weighting modules such as clustering, boosting, and isotonic fitting that collectively learn how every cognitive orientation is prevalent to the given careers. Consequently, where the coherence weight maximizes the performance, we construct a novel R^w-tree index hierarchy to adapt each cognitive-semantic category to the given agglomerative profession boundaries. The extensive experiments on a real-world microblog dataset demonstrate the superiority of our proposed model versus trending competitors. While we can leverage the deep learning algorithms to improve the effectiveness, we can also scale-out the components to promote efficiency, parallel processing. We leave these tasks as future work.

References

[1] Rajdeep Singh. A cognitive approach to the semantics in the sacred context: Semantic and symbolic function of sacred words. *English Linguistics*, 7(3), 2018.

[2] Kim Bartel Sheehan. Crowdsourcing research: data collection with amazon's mechanical turk. *Communication Monographs*, 85(1):140–156, 2018.

[3] Joaquin Navajas, Tamara Niella, Gerry Garbulsky, Bahador Bahrami, and Mariano Sigman. Aggregated knowledge from a small number of debates outperforms the wisdom of large crowds. *Nature Human Behaviour*, 2(2):126–132, 2018.

[4] Alasdair Brown and J James Reade. The wisdom of amateur crowds: Evidence from an online community of sports tipsters. *European Journal of Operational Research*, 272(3):1073–1081, 2019.

[5] Florian Daniel, Pavel Kucherbaev, Cinzia Cappiello, Boualem Benatallah, and Mohammad Allahbakhsh. Quality control in crowdsourcing: A survey of quality attributes, assessment techniques, and assurance actions. *ACM Computing Surveys (CSUR)*, 51(1):1–40, 2018.

[6] Md Rajibul Hasan, Ashish Kumar Jha, and Yi Liu. Excessive use of online video streaming services: Impact of recommender system use, psychological factors, and motives. *Computers in Human Behavior*, 80:220–228, 2018.

[7] Roger D Dias, Heather M Conboy, Jennifer M Gabany, Lori A Clarke, Leon J Osterwei, George S Avrunin, David Arney, Julian M Goldman, Giuseppe Riccardi, Steven J Yule, et al. Development of an interactive dashboard to analyze cognitive workload of surgical teams during complex procedural care. In *2018 IEEE Conf. on CogSIMA*, pages 77–82. IEEE, 2018.

[8] Sidahmed Benabderrahmane, Nedra Mellouli, Myriam Lamolle, and Patrick Paroubek. Smart4job: A big data framework for intelligent job offers broadcasting using time series forecasting and semantic classification. *Big Data Research*, 7:16–30, 2017.

[9] Saeid Hosseini. Location inference and recommendation in social networks. 2017.

[10] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. Short text understanding through lexical-semantic analysis. In *2015 IEEE 31st Int. Conf. on Data Eng.*, pages 495–506. IEEE, 2015.

[11] Saeid Hosseini, Sayan Unankard, Xiaofang Zhou, and Shazia Sadiq. Location oriented phrase detection in microblogs. In *Int. Conf. on Database Sys. for Advanced Apps*, pages 495–509. Springer, 2014.

[12] Chia-Huei Wu, Ying Wang, Sharon K Parker, and Mark A Griffin. Effects of chronic job insecurity on big five personality change. *Journal of Applied Psychology*, 2020.

[13] Golnoosh Farnadi, Susana Zoghbi, Marie-Francine Moens, and Martine De Cock. Recognising personality traits using facebook status updates. In *7th Int. AAAI Conf. on Social Media*, 2013.

[14] Iuliana F Iatan. Predicting human personality from social media using a fuzzy neural

network. In *Issues in the Use of Neural Networks in Info. Retrieval*, pages 81–105. Springer, 2017.

[15] Ahmed Al Marouf, Md Kamrul Hasan, and Hasan Mahmud. Comparative analysis of feature selection algorithms for computational personality prediction from social media. *IEEE Tran. on Computational Social Systems*, 2020.

[16] Tianran Hu, Haoyuan Xiao, Jiebo Luo, and Thuy-vy Thi Nguyen. What the language you tweet says about your occupation. In *Tenth Int. AAAI Conf. on Web and Social Media*, 2016.

[17] Maria Törnroos, Markus Jokela, and Christian Hakulinen. The relationship between personality and job satisfaction. *Personality and Individual Differences*, 145:82–88, 2019.

[18] Meghna Basu Thakur, Priya Kewalramani, and Rida Sheikh. A correlational study between personality, job stress and job performance across individualistic and collectivistic cultures. *Our Heritage*, 68(1):1042–1056, 2020.

[19] Erik Brynjolfsson, Tom Mitchell, and Daniel Rock. What can machines learn, and what does it mean for occupations and the economy? In *AEA Papers and Proceedings*, volume 108, pages 43–47, 2018.

[20] Chenrui Zhang and Xueqi Cheng. An ensemble method for job recommender systems. In *Proceedings of the Recommender Systems Challenge*, pages 1–4. 2016.

[21] Niusha Shafiabady, Lam Hong Lee, Rajprasad Rajkumar, VP Kallimani, Nik Ahmad Akram, and Dino Isa. Using unsupervised clustering approach to train the support vector machine for text classification. *Neurocomputing*, 211:4–10, 2016.

[22] Hsin-Min Lu, Jih-Shin Chen, and Wei-Chun Liao. Nonparametric regression via variance-adjusted gradient boosting gaussian process regression. *IEEE Tran. on Knowledge and Data Eng.*, 2019.

[23] Zaixu Cui and Gaolang Gong. The effect of machine learning regression algorithms and sample size on individualized behavioral prediction with functional connectivity features. *Neuroimage*, 178:622–637, 2018.

[24] Meng Liu, Xuyun Zhang, Chi Yang, Qiang He, and Jianbing Zhang. Curve fitting based efficient parameter selection for robust provable data possession. *Journal of Parallel and Distributed Computing*, 120:62–76, 2018.

[25] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Contextual semantics for sentiment analysis of twitter. *Information Processing & Management*, 52(1):5–19, 2016.

[26] Wen Hua, Zhongyuan Wang, Haixun Wang, Kai Zheng, and Xiaofang Zhou. Understand short texts by harvesting and analyzing semantic knowledge. *IEEE Tran. on Knowledge and data Eng.*, 29(3):499–512, 2016.

[27] Saeed Najafipour, Saeid Hosseini, Wen Hua, Mohammad Reza Kangavari, and Xiaofang Zhou. Soulmate: Short-text author linking through multi-aspect temporal-textual embedding. *IEEE Tran. on Knowledge and Data Eng.*, 2020.

[28] Wen Hua, Kai Zheng, and Xiaofang Zhou. Quality-aware entity-level semantic representations for short texts. *IEEE Data Eng. Bull.*, 39(2):93–105, 2016.

[29] Masumi Shirakawa, Kotaro Nakayama, Takahiro Hara, and Shojiro Nishio. Wikipedia-based semantic similarity measurements for noisy short texts using extended naive bayes. *IEEE Tran. on Emerging Topics in Computing*, 3(2):205–219, 2015.

[30] Peipei Li, Lu He, Haiyan Wang, Xuegang Hu, Yuhong Zhang, Lei Li, and Xindong Wu. Learning from short text streams with topic drifts. *IEEE Tran. on cybernetics*, 48(9):2697–2711, 2017.

[31] Tomas Mikolov, Kai Chen, Greg Corrado, and Jeffrey Dean. Efficient estimation of word representations in vector space. *arXiv*

preprint arXiv:1301.3781, 2013.

[32] Yelong Shen, Xiaodong He, Jianfeng Gao, Li Deng, and Grégoire Mesnil. Learning semantic representations using convolutional neural networks for web search. In *Proceedings of the 23rd Int. Conf. on World Wide Web*, pages 373–374, 2014.

[33] Alex Graves. Supervised sequence labelling. In *Supervised sequence labelling with recurrent neural networks*, pages 5–13. Springer, 2012.

[34] Nuno Luz, Nuno Silva, and Paulo Novais. A survey of task-oriented crowdsourcing. *Artificial Intelligence Review*, 44(2):187–213, 2015.

[35] Yanbo Huang. Exploiting embedding in content-based recommender systems. 2016.

[36] Cataldo Musto, Giovanni Semeraro, Marco de Gemmis, and Pasquale Lops. Learning word embeddings from wikipedia for content-based recommender systems. In *European Conf. on Information Retrieval*, pages 729–734. Springer, 2016.

[37] Steffen Schnitzer, Dominik Reis, Wael Alkhatib, Christoph Rensing, and Ralf Steinmetz. Preselection of documents for personalized recommendations of job postings based on word embeddings. In *Proceedings of the 34th ACM/SIGAPP*, pages 1683–1686, 2019.

[38] Santiago Alonso, Jesús Bobadilla, Fernando Ortega, and Ricardo Moya. Robust model-based reliability approach to tackle shilling attacks in collaborative filtering recommender systems. *IEEE Access*, 7:41782–41798, 2019.

[39] Robert W Lent, Taylor R Morris, Lee T Penn, and Glenn W Ireland. Social–cognitive predictors of career exploration and decision-making: Longitudinal test of the career self-management model. *Journal of counseling psychology*, 66(2):184, 2019.

[40] Robert W Lent, Ijeoma Ezeofor, M Ashley Morrison, Lee T Penn, and Glenn W Ireland. Applying the social cognitive model of career self-management to career exploration and decision-making. *Journal of Vocational Behavior*, 93:47–57, 2016.

[41] Ji Geun Kim, Haram J Kim, and Ki-Hak Lee. Understanding behavioral job search self-efficacy through the social cognitive lens. *Journal of Vocational Behavior*, 112:17–34, 2019.

[42] Michael M Tadesse, Hongfei Lin, Bo Xu, and Liang Yang. Personality predictions based on user behavior on the facebook social media platform. *IEEE Access*, 6:61959–61969, 2018.

[43] Veronica Ong, Anneke DS Rahmanto, Derwin Suhartono, Aryo E Nugroho, Esther W Andangsari, Muhamad N Suprayogi, et al. Personality prediction based on twitter information in bahasa indonesia. In *2017 Federated Conf. on FedCSIS*, pages 367–372. IEEE, 2017.

[44] Joeran Beel, Bela Gipp, Stefan Langer, and Corinna Breitinger. paper recommender systems: a literature survey. *Intnl. Journal on Digital Libraries*, 17(4):305–338, 2016.

[45] Han Lin and Lin Qiu. Sharing emotion on facebook: network size, density, and individual motivation. In *CHI'12 Extended Abstracts on Human Factors in Computing Systems*, pages 2573–2578. 2012.

[46] Scott E Seibert, Maria L Kraimer, and Robert C Liden. A social capital theory of career success. *Academy of management journal*, 44(2):219–237, 2001.

[47] Mohammad Aghagolzadeh, Iman Barjasteh, and Hayder Radha. Transitivity matrix of social network graphs. In *2012 IEEE Statistical Signal Processing Workshop (SSP)*, pages 145–148. IEEE, 2012.

[48] Liang Wu, Fred Morstatter, and Huan Liu. Slangsd: building, expanding and using a sentiment dictionary of slang words for

short-text sentiment classification. *Language Resources and Evaluation*, 52(3):839–852, 2018.

[49] Ian Beaver. Machine based expansion of contractions in text in digital media, January 23 2020. US Patent App. 16/513,073.

[50] Saeid Hosseini, Hongzhi Yin, Xiaofang Zhou, Shazia Sadiq, Mohammad Reza Kangavari, and Ngai-Man Cheung. Leveraging multi-aspect time-related influence in location recommendation. *World Wide Web*, 22(3):1001–1028, 2019.

[51] Felipe Taliar Giuntini, Larissa Pires Ruiz, Luziane De Fatima Kirchner, Denise Aparecida Passarelli, Maria De Jesus Dutra Dos Reis, Andrew Thomas Campbell, and Jó Ueyama. How do i feel? identifying emotional expressions on facebook reactions using clustering mechanism. *IEEE Access*, 7:53909–53921, 2019.

[52] Geetha Sitaraman. *Inferring big 5 personality from online social net.* PhD thesis, 2014.

[53] Tushar Maheshwari, Aishwarya N Reganti, Upendra Kumar, Tanmoy Chakraborty, and Amitava Das. Revealing psycholinguistic dimensions of communities in social networks. *IEEE Intelligent Systems*, 33(4):36–48, 2018.

[54] Jacob Benesty, Jingdong Chen, and Yiteng Huang. On the importance of the pearson correlation coefficient in noise reduction. *IEEE Tran. on Audio, Speech, and Language Processing*, 16(4):757–765, 2008.

[55] Seyyed Mohammad Hossein Dadgar, Mohammad Shirzad Araghi, and Morteza Mastery Farahani. A novel text mining approach based on tf-idf and support vector machine for news classification. In *2016 IEEE Int. Conf. on Eng. and Tech. (ICETECH)*, pages 112–116. IEEE, 2016.

[56] Rebekka Weidmann, Felix D Schönbrodt, Thomas Ledermann, and Alexander Grob. Concurrent and longitudinal dyadic polynomial regression analyses of big five traits. *Journal of Research in Personality*, 70:6–15, 2017.

[57] Qiyang Han, Tengyao Wang, Sabyasachi Chatterjee, Richard J Samworth, et al. Isotonic regression in general dimensions. *The Annals of Statistics*, 47(5):2440–2471, 2019.

[58] Lakshmi Balasubramanian and M Sugumaran. A state-of-art in r-tree variants for spatial indexing. *Int. Journal of Computer Apps*, 42(20):35–41, 2012.

[59] XIANG Yuan-ping, HE Yan-Ping, WEI Yulin, LIANG Huan, and GUO Ben-chu. Algorithm for minimum bounding rectangle. *Computer and Modernization*, (2):58, 2016.

[60] DJ Stillwell and M Kosinski. mypersonality project website, 2015.

[61] Jorge Carlos V. Rebaza, Ricardo Puma, Paul Bustios, and Nathalia C Silva. Job recommendation based on job seeker skills: An empirical study. In *Text2Story@ ECIR*, pages 47–51, 2018.

[62] Saeid Hosseini, Saeed Najafipour, Ngai-Man Cheung, Hongzhi Yin, Mohammad Reza Kangavari, and Xiaofang Zhou. Teags: Time-aware text embedding approach to generate subgraphs. *arXiv preprint arXiv:1907.03191*, 2019.

[63] Leila Khalatbari, M.R. Kangavari, Saeid Hosseini, Hongzhi Yin, and Ngai-Man Cheung. Mcp: A multi-component learning machine to predict protein secondary structure. *Computers in biology and medicine*, 110:144–155, 2019.