A Deep Learning Architecture for Psychometric Natural Language Processing

FAIZAN AHMAD, AHMED ABBASI, JINGJING LI, DAVID G. DOBOLYI, and RICHARD G. NETEMEYER, University of Virginia
GARI D. CLIFFORD, Emory University and Georgia Tech
HSINCHUN CHEN, University of Arizona

Psychometric measures reflecting people's knowledge, ability, attitudes, and personality traits are critical for many real-world applications, such as e-commerce, health care, and cybersecurity. However, traditional methods cannot collect and measure rich psychometric dimensions in a timely and unobtrusive manner. Consequently, despite their importance, psychometric dimensions have received limited attention from the natural language processing and information retrieval communities. In this article, we propose a deep learning architecture, PyNDA, to extract psychometric dimensions from user-generated texts. PyNDA contains a novel representation embedding, a demographic embedding, a structural equation model (SEM) encoder, and a multitask learning mechanism designed to work in unison to address the unique challenges associated with extracting rich, sophisticated, and user-centric psychometric dimensions. Our experiments on three real-world datasets encompassing 11 psychometric dimensions, including trust, anxiety, and literacy, show that PyNDA markedly outperforms traditional feature-based classifiers as well as the state-of-the-art deep learning architectures. Ablation analysis reveals that each component of PyNDA significantly contributes to its overall performance. Collectively, the results demonstrate the efficacy of the proposed architecture for facilitating rich psychometric analysis. Our results have important implications for user-centric information extraction and retrieval systems looking to measure and incorporate psychometric dimensions.

CCS Concepts: • **Computing methodologies** → *Natural language processing*;

Additional Key Words and Phrases: Deep learning, natural language processing, psychometric measures, text classification

ACM Reference format:
Faizan Ahmad, Ahmed Abbasi, Jingjing Li, David G. Dobolyi, Richard G. Netemeyer, Gari D. Clifford, and Hsinchun Chen. 2020. A Deep Learning Architecture for Psychometric Natural Language Processing. *ACM Trans. Inf. Syst.* 38, 1, Article 6 (February 2020), 29 pages.
https://doi.org/10.1145/3365211

6

The authors wish to thank the U.S. National Science Foundation for their support under grants NSF IIS-1816504, IIS-1553109, BDS-1636933, CCF-1629450, and Microsoft Research for its support through CRM:0740129.
Authors' addresses: F. Ahmad, 14 University Circle #4, Charlottesville, VA 22903; email: fa7pdn@virginia.edu; A. Abbasi, J. Li, D. G. Dobolyi, and R. G. Netemeyer, Rouss Hall and Robertson Hall (McIntire School of Commerce), 125 Ruppel Dr, Charlottesville, VA 22903; emails: {abbasi, jl9rf, dd2es}@comm.virginia.edu, rgn3p@virginia.edu; G. D. Clifford, Department of Biomedical Engineering, 313 Ferst Drive, Room 2127 Atlanta, GA 30332; email: gari.clifford@bme.gatech.edu; H. Chen, McClelland Hall 430X 1130 E. Helen St. Tucson, Arizona 85721-0108; email: hchen@eller.arizona.edu.

1046-8188/2020/02-ART6
https://doi.org/10.1145/3365211

1 INTRODUCTION

Psychometrics is concerned with the measurement of knowledge, ability, attitudes, and personality traits. With the increased importance of analytics at the micro-level [10], including understanding and prediction of individuals' behaviors [4], accurate and timely measurement of psychometrics has become of paramount importance. In cybersecurity contexts, self- efficacy and threat perceptions are critical psychometric dimensions known to be strong indicators of end-user susceptibility to phishing attacks [5]. Similarly, financial literacy and psychological traits are important antecedents for downstream financial behaviors [24]. In e-commerce, satisfaction with a website's browsing experience is a crucial lead indicator for purchase propensity and e-loyalty [18]. In health settings, psychometric measures, including health numeracy, subjective literacy, and perceptions of trust and anxiety related to physicians, have been shown to have a profound impact on various health and wellness outcomes such as future doctors' visits and all-around well-being [8, 22, 53]. Hence, accurately and efficiently measuring psychometrics inherent in user-generated content can provide an important information access refinement with positive implications for many real-world tasks, including information retrieval, mobile text analytics, and behavior modeling [4, 20, 25].

Psychometric data collection efforts have traditionally relied on survey-based methods administered on a monthly or quarterly basis. Effectively collecting and measuring relevant constructs in a timely and unobtrusive manner has proven elusive in real-world settings [27]. In recent years, machine learning methods for natural language processing (NLP) have been successfully applied to certain psychometric dimensions such as sentiment and emotion [11, 29]. Such NLP techniques, which analyze user-generated text and automatically score them along the target variable, afford opportunities for real-time, passive monitoring and measurement. However, several gaps and challenges remain:

- *Many rich psychometric dimensions remain underexplored:* Whereas numerous NLP methods have been proposed for sentiment and emotion, other aspects such as attitudes, perceptions, and characteristics have received limited attention [6]. It is unclear how effectively NLP methods can tackle these novel dimensions.
- *User-centric versus task-centric modeling:* Most prior NLP classification objectives and datasets have been arranged around a given task (e.g., sentiment polarity). Psychometric dimensions such as attitudes and perceptions are very individualized, with multiple interrelated target variables of interest associated with each person. There is an opportunity for psychometric NLP methods to incorporate provisions for user-centric modeling.
- *Demographic-sensitive modeling:* Factors such as age, race, gender, and education can have a profound impact on various psychometric measures (e.g., literacy, trust, anxiety) [22]. These differences can be amplified in user-generated text [61]. Several recent studies suggest that machine learning models that fail to properly control for demographics are prone to inaccurate generalizations [21]. Psychometric NLP methods that are accurate across diverse demographic populations are a necessary and understudied research area.
- *Paucity of available text:* Several recent NLP studies have examined "short text" contexts such as Twitter [57] and news articles [37, 59]. User-generated text associated with psychometrics often appears in similarly sparse environments such as comment boxes, text messages, and microblogs, necessitating methods capable of learning patterns from limited linguistic cues.

In order to address these gaps, we propose a novel deep learning architecture for psychometric NLP. Our architecture incorporates provisions to address the aforementioned issues, including novel representation and demographic embeddings and a structural equation modeling (SEM)

encoder, coupled with a robust multitask learning method. The proposed architecture was evaluated on a rich health test bed encompassing three datasets composed of pertinent psychometric dimensions—such as health numeracy, literacy, trust, anxiety, and drug experiences—related to a set of demographically diverse users. The results reveal that the proposed architecture is able to garner markedly better classification accuracy, precision, and recall rates across psychometric dimensions, relative to baseline and benchmark machine learning NLP methods. Ablation analysis shows that each component significantly contributes to overall performance, thereby underscoring the efficacy of the proposed architecture.

2 RELATED WORK

Psychometric dimensions are measures of latent constructs related to knowledge, ability, attitudes, perceptions, and personality traits [52]. These dimensions are known to be important antecedents, mediators, and moderators for important humanistic outcomes and behaviors [38]. For instance, in the health context, health literacy is a subjective reflection of one's "knowledge pertaining to health care issues" [45]. The trust in doctors that patients place and anxiety visiting doctors are additional examples of health-related psychometric dimensions [22, 54]. All three of these, and other related dimensions, have been shown to impact future health outcomes including well-being [45]. However, effectively collecting and measuring such covariates in a timely and unobtrusive manner has proven elusive in real-world settings [27]. Many psychometric dimensions require 10 or more survey responses [14, 46], making them less feasible in persistent measurement environments. Recent studies have suggested that NLP methods applied to user-generated content might offer a complementary or alternative mechanism for measuring psychometric dimensions [27]. However, whereas NLP has a longstanding tradition for certain dimensions such as sentiment polarity (i.e., positive, negative, neutral) and select emotions (e.g., happiness, anger) [11, 33, 65], many important psychometric dimensions have been largely unexplored. Given the potential implications of psychometrics for information retrieval, as noted in the introduction, one of the goals of this study is to demonstrate the efficacy of NLP methods for measuring rich psychometric dimensions from text. Accordingly, in the remainder of this section, we review relevant NLP literature.

2.1 Feature-Based Classifiers for NLP

Feature-based text classifiers—a special type of NLP technique—have demonstrated their effectiveness in extracting certain commonly studied psychometric dimensions, such as sentiment and emotions, from user-generated text [11]. These techniques rely on supervised machine learning methods that leverage rich linguistic features to classify texts into several types (e.g., emotion classes or sentiment polarities) and/or intensity levels (e.g., high vs. low). Past studies have shown that supervised machine learning methods, such as Support Vector Machines [48], Naive Bayes [43], and logistic regression [28], are especially effective for text classification. An important area of focus for feature-based classifiers is to find the ideal feature set for representing the richness of the texts in order to enhance classification performance. This is particularly critical since linguistic feature spaces can be massive, encompassing lexical measures, parts of speech, alternative syntactical patterns, domain-specific and general-purpose semantic lexicons, and pragmatic information. Consequently, several NLP feature ranking techniques have been proposed, such as FSH and FRN [3, 23], to enrich and enhance feature engineering efforts. In contrast, recent studies have found that deep learning affords opportunities for automatic feature engineering [13]. However, deep learning often works better when applied to large-scale texts [57, 59]. Thus, feature engineering and domain adaptation intuitions from feature-based text classification may complement deep learning when facing data paucity in diverse linguistic environments such as those encountered for psychometric NLP.

2.2 RNNs for NLP

Recurrent neural networks (RNNs) selectively pass information across sequence steps while processing sequential data one element at a time [23]. In order to deal with the gradient vanishing problem that commonly appears in long-term sequence learning processes, two major gating mechanisms have been proposed: gated recurrent units (GRUs) [13] and long short-term memory (LSTM) [32]. The basic idea is to use a set of gates to regulate the values (through a weighted sum activation function) flowing into each hidden state so that the gradients are refrained from approximating to zero. These types of RNNs have been shown to be effective for many NLP tasks (e.g., [33, 49]). In the context of psychometric NLP, RNNs could capture long-term linguistic dependencies [12]—which are typically hard to capture through manual feature engineering—in user-generated texts to improve classification performance for previously underexplored psychometric dimensions of interest.

2.3 CNNs for NLP

Convolutional neural networks (CNNs) utilize layers with convolving filters to apply to local features [36]. Originally invented for computer vision, CNNs have recently demonstrated superior performance on several NLP tasks (e.g., [16]). An interesting line of work is to learn character-level CNN embeddings to accommodate possible spelling errors and prefix and suffix information [60]. In addition, Kim et al. [35] used a CNN trained on top of pretrained word vectors for sentence-level classification [19]. Deriu et al. [19] used an ensemble of CNNs with distant supervision and a random forest classifier for message-level sentiment analysis. Coneau et al. [17] proposed a very deep CNN (VD-CNN) to accommodate character-level information for public text classification. In summary, CNNs could be particularly useful for text classification of user-generated psychometric content, which often contains significant misspellings and domain-specific expressions.

2.4 Hybrid Architectures

Several articles (e.g., Cho et al. [13]) have discussed application areas for RNNs and CNNs in NLP. For example, CNNs are well suited for mining local features regardless of position information, whereas RNNs are good at extracting long-term sequential information. Due to their complementary characteristics, researchers often utilize a hybrid architecture comprising RNNs and CNNs to solve complex NLP classification tasks. For example, Zhou et al. [62] used CNNs to learn phrase-level features through a convolutional layer and fed the sequence of such higher-level representations into RNNs to learn long-term dependencies. Inferring psychometric measures may benefit from hybrid architectures capable of accommodating rich and diverse linguistic patterns appearing in user-generated texts.

2.5 Multitask Adversarial Learning

Multitask learning [63, 64] is an effective approach for improving the performance of a single task by learning multiple tasks jointly. Recent progress in deep learning has offered novel opportunities for implementing multitask learning in a general neural-based framework: learning shared representations across multiple tasks to facilitate feature sharing, and finally mapping to individual tasks via a task-specific predictor. The shared representation could utilize a hybrid architecture, perhaps from lower-level word representations [15] to higher-level contextual representations such as RNNs [49]. This general framework has recently been shown to work well for NLP tasks including sequence tagging [58], text classification, and discourse analysis [40]. A notable improvement of the neural-based multitask learning framework is to adopt adversarial learning [30] to ensure the shared representations only contain common and task-invariant

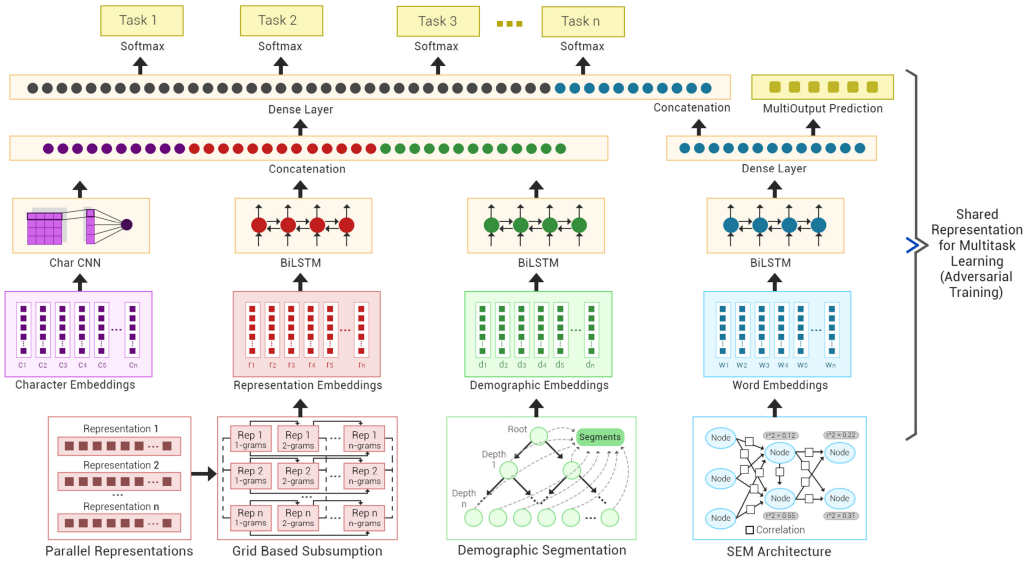

Fig. 1. PyNDA architecture diagram.

information. Such approaches have worked well for several NLP tasks [40] and may also be effective in the context of psychometric NLP.

2.6 Summary of Research Gaps

Based on our review of relevant literature, we have identified three major research gaps. First, although psychometric dimensions such as sentiment and emotion have been studied extensively, there has been limited focus on other rich psychometric dimensions such as trust, anxiety, and literacy. Research examining such psychometric dimensions is of theoretical and practical importance. For instance, effectively capturing such psychometric dimensions necessitates consideration of user-centric modeling techniques capable of considering interrelated dimensions in unison, as well as demographic-sensitive modeling. Second, little work has been done to fuse the rich linguistic resources, methods, and domain knowledge developed in the feature-based NLP classification literature with novel deep learning architectures. Given the complexity of psychometric utterances and paucity of available text, such fusion could facilitate enhanced accuracy by leveraging rich linguistic feature representations in concert with robust deep learning schemes. Third, hybrid deep learning architectures encompassing CNNs, LSTMs, and multitask learning mechanisms have been underexplored. Prior work suggests these approaches offer complementary benefits such as pattern detection from local features, consideration of long-term dependencies, and inclusion of the interplay between closely related user-level psychometric dimensions. In the ensuing section, we propose an architecture expressly designed to address these gaps.

3 PROPOSED ARCHITECTURE: DEEP LEARNING FOR PSYCHOMETRIC NLP

Figure 1 depicts our proposed PyNDA **Ps**ychometric **N**LP **D**eep Le**a**rning Architecture, which encompasses four base neural nets that are fused via a concatenation layer that feeds into dense layers and also leverages a novel multitask learning mechanism. Each component of the architecture is intended to address the aforementioned research gaps, thereby resulting in enhanced text classification capabilities for psychometric dimensions:

- *A character embedding* convolution neural network (CNN) for capturing fundamental spatial syntactic patterns in user-generated texts, at the character and prefix, suffix, and root levels.
- A bidirectional long short-term memory (Bi-LSTM) recurrent neural network that uses a novel underlying *parallel representation embedding* that encompasses an array of topic, sentiment, emotion, and syntactic linguistic representations. This embedding leverages feature subsumption methods capable of ingesting large, diverse feature spaces and refining them into a small set of rich attributes.
- A second Bi-LSTM that incorporates a novel *demographic embedding* scheme intended to better capture nuances and norms inherent across different gender, race, and age segments.
- A structural equation model *(SEM)* Encoder that allows inclusion of related "secondary" attitude and behavior information to allow superior classification of key target psychometric dimensions.
- A novel *multitask learning mechanism* that enables better inclusion of joint information between related target psychometric dimensions.

In the remainder of the section, we describe each component of the proposed architecture.

3.1 Character Embedding

In order to consider the morphological patterns (e.g., prefix, suffix and misspelling) of the input text, we build a character-level embedding using convolutional neural networks. Such neural network-based embeddings have shown great promise on a wide variety of tasks [31, 37, 47]. The input for the character embedding is a sequence of encoded characters. Each character is represented as a one-hot (or one-over-l) vector $g(x) \in [1, l] \rightarrow R$, where l is the size of the alphabet. The alphabet used in our model consists of 70 characters, including 26 English letters, 10 digits, 33 other characters, and the new line character. The convolutional kernel function is defined as $f(x) \in [1, k] \rightarrow R$, where k is the size of the filters. Given the stride of d we can get the convolution $h(y) \in [1, [l - k + 1/d]] \rightarrow R$ between $f(x)$ and $g(x)$ as follows:

$$h(y) = \sum_{x=1}^{k} f(x) \circ g(y \circ d - x + c), \tag{1}$$

where $c = k - d + 1$ is an offset constant. This convolutional layer is later connected to a max-pooling layer, defined as:

$$h(y)_{maxpooling} = max_{x=1}^{k}\, g(y \circ d - x + c). \tag{2}$$

The embedding process uses two convolutional layers, each followed by a max pooling layer. The resulting embedding is fed into two fully connected layers, which are then concatenated with layers from other embeddings for the finally psychometric variable classification.

3.2 Representation Embedding

Examination of rich psychometric dimensions pertaining to diverse user demographics could pose challenges for deep learning methods, particularly in situations involving limited user-generated text. Recent work has shown that rich feature-based methods can often attain text classification performance levels that are comparable to simple deep learning architectures [65], whereas combining the two can often yield enhanced performance [33, 51]. Accordingly, we propose a novel representation embedding that utilizes a rich array of parallel feature representations that capture a bevy of semantic and syntactic information at varying granularities, coupled with grid-based subsumption. The main intuition behind the proposed embedding is similar to

Table 1. An Example Illustrating Major Parallel Representations Employed in the Representation Embedding

	Representation	Example
Semantic	Word	these sawbones are just awful ! escitalopram caused me to gain weight and feel depressed. i 'm better off using google.
	Hypernym	these **DOCTOR** are just awful ! escitalopram caused me to gain **UNIT_OF_MEASUREMENT** and **PROPERTY** depressed. i 'm better off **EXPLOIT** google.
	Sentiment	these **LPOSLNEG** are just **LPOSHNEG** ! escitalopram **LPOSLNEG** me to **LPOSLNEG LPOSLNEG** and **LPOSLNEG** depressed. i 'm **HPOSLNEG** off **LPOSLNEG LPOSLNEG**.
	Affect	these sawbones are just **NEGATIVE-FEAR** ! escitalopram caused me to gain weight and feel **SADNESS**. i 'm better off using google.
	Named Entities	these sawbones are just awful ! escitalopram caused me to gain weight and feel depressed. i 'm better off using **ORGANIZATION**.
	Domain Lexicons	these sawbones are just **REACTION** ! **DRUG REACTION** me to gain **REACTION** and feel **REACTION**. i 'm better off using google.
	Word & Sense	these **sawbones**|_|**01** are just **awful**|_|**01** ! escitalopram **caused**|_|**02** me to **gain**|_|**05 weight**|_|**02** and **feel**|_|**01** depressed. i 'm **better**|_|**03** off **using**|_|**04 google**|_|**01**.
	Word & NE	these sawbones are just awful ! escitalopram caused me to gain weight and feel depressed. i 'm better off using **google**|_|**ORGANIZATION**
	POS	**DT NNS VBP RB JJ. NN VBD PRP TO VB NN CC VB JJ. PRP VBP JJR RP VBG NNP**.
Semantic	Misspellings	these sawbones are just awful ! escitalopram caused me to gain weight and feel **MISSPELLING**. i 'm better off using google.
	Word & POS	**these**|_|**DT sawbones**|_|**NNS are**|_|**VBP just**|_|**RB awful**|_|**JJ !**|_|**. escitalopram**|_|**NN caused**|_|**VBD me**|_|**PRP to**|_|**TO gain**|_|**VB weight**|_|**NN and**|_|**CC feel**|_|**VB depressed**|_|**JJ .**|_|**. i**|_|**PRP 'm**|_|**VBP better**|_|**JJR off**|_|**RP using**|_|**VBG google**|_|**NNP .**|_|**.**
	Legomena	these **HAPAX** are just awful ! escitalopram caused me to gain weight and feel depressed. i 'm better off using google.

standard word embeddings: create a lower-dimensional feature space that captures key patterns. However, as we later demonstrate empirically, the representation embedding is particularly well suited for psychometric NLP, providing strong discriminatory potential. Details regarding the parallel representations and grid-based subsumption are as follows.

3.2.1 Parallel Representations. Table 1 illustrates the major parallel representations incorporated for the input text example: "These sawbones are just awful! Escitalopram caused me to gain weight and feel depresed. I'm better off using Google." The semantic category encompassed topic-, sentiment-, and emotion-related representations. Topic representations included words, named entities, hypernyms, and domain lexicons. Named entities, hypernyms, and lexicons were

employed since they allow tracking of topical information at a less granular level, which can help alleviate the pattern sparsity problem when dealing with limited text. For instance, named entities, which were extracted using Stanford CoreNLP [42], aggregate person, place, and organization-level information (such as "Google" in Table 1). Similarly, hypernyms derived using the hierarchy in WordNet allow aggregation over "type of" relations [44]. As depicted in Table 1, this enables us to associate the very uncommon term "sawbones" with the "DOCTOR" label. Lexicons are used to provide an additional dimension for semantic abstraction. In the health domain example presented in Table 1, lexicons related to common prescription drug names and adverse reactions are incorporated and used to replace words appearing in the respective lexicon term lists (e.g., the drug escitalopram).

A key aspect of these representations is that they all have the same length (i.e., an equal number of tokens). This property allows these "parallel" representations to be merged into feature combinations. Within the semantic category, words are merged with sense tags derived using WordNet in order to allow better word sense disambiguation. Similarly, words can be combined with named entities (e.g., "Google" and "organization") to provide varying levels of granularity.

Additional semantic categories incorporated were sentiment and affect. Sentiment allows us to gauge users' levels of subjectivity and sentiment polarity. Words were mapped to their respective word sense's positive and negative sentiment polarity scores in SentiWordNet [7]. These scores were grouped into high/medium/low bins, resulting in nine potential sentiment polarity tags. For instance, the word "awful" has high negative polarity and a low positive polarity score. Similarly, words were mapped to affect category tags based on WordNet Affect [55]. In the example presented in Table 1, the word "depressed" gets mapped to the SADNESS emotion category.

The syntactic representations incorporated were parts-of-speech (POS) tags, words combined with POS tags, misspellings, and legomena. Misspellings are a major source of sparsity and noise in short, user-generated texts. Accordingly, spellchecking with support for word exclusions was used to correct input text prior to construction of all parallel representations. Additionally, since presence and frequency of misspellings can be important psychometric cues, the misspelling representation was included. In Table 1, the misspelled input "depresed" is corrected for all representations and noted with the MISSPELLING tag. Consistent with prior studies, legomena were included in order to alleviate sparsity attributable to once-appearing words in the training set (e.g., "sawbones") with a HAPAX tag [2]. The combination of words with their respective parts of speech were included as an additional layer of word disambiguation [3].

3.2.2 Grid-Based Subsumption (GBS). Although parallel representations can allow inclusion of rich linguistic representations at varying granularities, it also creates potential for inclusion of noise, redundancy, and irrelevant information. For instance, all the nonbolded text in Table 1 is redundant. Even some of the bolded parallel information may not be unique or useful. Accordingly, prior studies have proposed the use of subsumption methods to rectify this concern: feature space reduction techniques specifically crafted for natural language data [3, 50]. However, prior methods use small, predefined subsumption mechanisms that are not scalable or extensible to large, dynamic feature spaces (e.g., [1, 50]). In order to overcome these limitations, we propose a novel GBS method well suited for "winnowing the wheat from the chaff" atop our rich parallel representations. GBS uses a four-stage algorithm, as depicted in Figure 2. In the figure, directional arrows indicate application of subsumption rule-based weighting/removal of n-grams (as done in stages 1, 2, and 4). Solid lines indicate use of correlation-based methods to determine relationships between different parallel representations (stage 2). Lighter-color shading is used to illustrate the down-weighting of redundant or less important features across parallel representations. The idea is that with each subsequent stage, a smaller proportion of the most important features is retained to ensure the ensuing embedding is more effective. The sequence of stages in GBS is intended

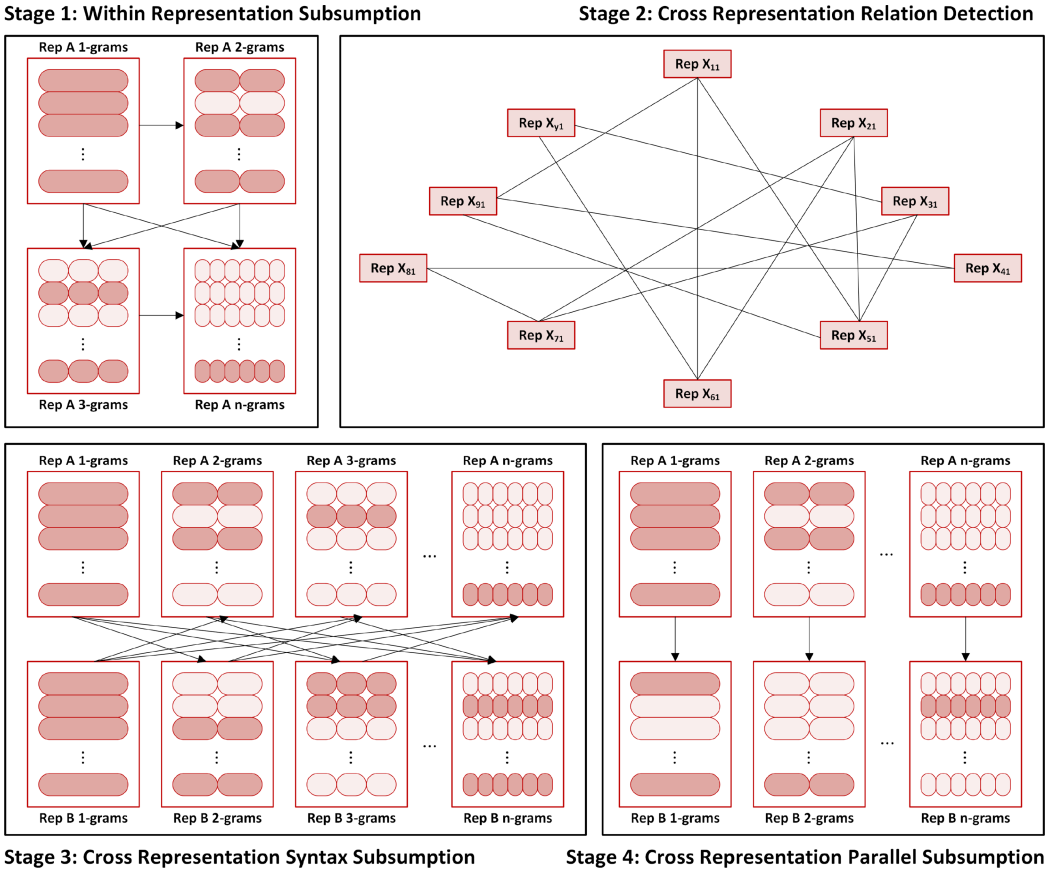

Fig. 2. A four-stage grid-based subsumption (GBS) algorithm.

to balance computational efficiency with subsumption quality. Within-category subsumption (stage 1) is computationally efficient and generally capable of removing 50% of noise and redundancy [3]. Stages 2 and 3 present a novel graph-sensing mechanism for computationally efficient cross-category subsumption. These stages typically alleviate an additional 25% to 35% of noise and redundancy in the parallel representation. Finally, correlation-based parallel subsumption (which is computationally the slowest) is used to alleviate an additional 2% to 5% of noise. Details regarding the four stages of GBS are as follows.

Stage 1 of GBS is mostly consistent with prior subsumption methods [3, 50], where only higher-order n-grams with enhanced discriminatory potential are retained over their lower-order n-gram feature counterparts within the same representation. Given the set of m representations $R = \{r_1, r_2, \ldots, r_m\}$, where each r_x signifies a parallel representation described in Table 1 (e.g., word), we extract all n-gram features such that any f_{ijx} element in feature set F represents the i^{th} feature in n-gram category j for representation r_x, and f_{ijx} is initially weighted as follows:

$$w(f_{ijx}) = \max_{c_a, c_b} \left(p(f_{ijx}|c_a) log \left(\frac{p(f_{ijx}|c_a)}{p(f_{ijx}|c_b)} \right) \right) + s(f_{ijx}), \tag{3}$$

where c_a and c_b are among the set of C class labels, $c_a \neq c_b$, y is one of the d tokens in f_{ijx} with w possible word senses, and function s is the mean semantic orientation score across all token-senses,

computed as the difference between the positive and negative polarity scores for sense q of token f_{ijx} in SentiWordNet:

$$s(f_{ijx}) = \sum_{y=1}^{d} \sum_{q=1}^{w} \frac{pos(f_{ijx}, q) - neg(f_{ijx}, q)}{dw}. \tag{4}$$

The first part of the weighting equation considers the discriminatory potential of the feature based on its log-likelihood ratio, whereas the second part factors in the semantic orientation to ensure that features with opposing orientation (e.g., "like" versus "don't like") are differentiated in terms of overall weights and when making subsumption decisions. Once features are weighted, the within representation r_x subsumption is performed as follows. Each n-gram feature f_{ijx} with $w(f_{ijx}) > 0$ is compared against each lower-order n-gram feature f_{uvx}, where $v < j$, $w(f_{uvx}) > 0$, and f_{uvx} contains some subsequence of tokens from f_{ijx}. If $c(f_{ijx}) = c(f_{uvx})$, where:

$$c(f_{ijx}) = argmax_{c_a, c_b} \left(p(f_{ijx}|c_a) log \left(\frac{p(f_{ijx}|c_a)}{p(f_{ijx}|c_b)} \right) \right) + s(f_{ijx}). \tag{5}$$

Then we determine whether to subsume the higher-order n-gram as follows, where t is a subsumption threshold:

$$w(f_{ijx}) = \begin{cases} 0 & if\ w(f_{ijx}) \leq w(f_{uvx}) + t. \\ w(f_{ijx}) & otherwise \end{cases} \tag{6}$$

Stage 2 entails cross-representation subsumption. Prior studies have relied on manually crafted subsumption graphs encompassing predefined representations and relation links (e.g., [12, 46]). In order to make the subsumption process more dynamic and extensible across an array of novel psychometric dimensions, we propose a graph construction approach. For each unique pair of representations r_x and r_z in R, let A and B signify randomly selected subsets of m features from these representations, where each $f_{ijx} \in A$ and $f_{uvz} \in B$ is such that $j, v = 1$ (i.e., only unigram features). Since representations vary with respect to feature frequency and co-occurrence patterns, it is important to factor in such nuances by considering within-category similarities when making cross-category comparisons. We use k-Means clustering to find the ideal partition over the 2m feature sample encompassing all elements in $A \, U \, B$. With $k = 2$, this results in $G = \{g1, g2\}$ clusters. A link is formed between r_x and r_z if the cross-cluster entropy reduction ratio attributable to representation affiliation information is below a certain threshold:

$$L(r_x, r_z) = \begin{cases} 1 & if\ \dfrac{H(G|r)}{H(G)} \leq l \\ 0 & otherwise \end{cases}, \tag{7}$$

where $H(G)$ is the entropy across clusters and $H(G|r)$ is:

$$H(G|r) = - \sum_{r=\{r_x, r_z\}} P(r) \sum_{g \in G} P(g|r) log_2 P(g|r). \tag{8}$$

In Stage 3, once links are formed between representations as described in Equation (7), cross-representation subsumption between any pair of r_x and r_z where $L(r_x, r_z) = 1$ is performed similarly to the approach described in Equations (5) and (6) of Stage 1. Since the links are bidirectional, $L(r_x, r_z) = L(r_z, r_x)$. Hence, two-way comparisons are made, where each remaining f_{ijx} with $w(f_{ijx}) > 0$ in r_x is compared against each lower-order n-gram feature f_{uvz} where $v < j$, $w(f_{uvz}) > 0$, and f_{uvz} contains some subsequence of tokens from f_{ijx}, and then, each remaining f_{uvz} with $w(f_{uvz}) > 0$ in r_z is compared against its lower-order n-gram counterparts in r_x meeting the same criteria.

Finally, in Stage 4, we account for highly correlated nonsubsuming cross-representation features. For each pair of r_x and r_z where $L(r_x, r_z) = 1$, each remaining f_{ijx} with $w(f_{ijx}) > 0$ in r_x is compared against all remaining f_{uvz} in r_z with weight greater than 0, where $j = v$. If the correlation between f_{ijx} and f_{uvz} is greater than threshold p, $w(f_{ijx}) = 0$.

3.2.3 Embedding and BiLSTM. For each representation, we use word2vec to learn an l-sized embedding vector for each token in that representation's data. However, only tokens with $w(f_{ijx}) = 0$ are included. For all other tokens, the embedding vector is replaced with a vector composed of 0s. This embedding is then fed into a Bi-LSTM layer to learn the sequential dependency among words. Given a sequence of words $x_1, x_2, \ldots, x_t, \ldots, x_T$, where x_t is a vector for word embedding, RNN learns the hidden features of each word based on all previous words in the sequence:

$$h_t = \sigma(W^{hh} h_{t-1} + W^{hx} x_t + b), \tag{9}$$

where W^{hh} is the weights matrix based on the previous hidden features h_{t-1} and W^{hx} is the weights matrix based on the input word vector x_t, b is a bias term, and σ is a nonlinearity function. The equation above learns the hidden features based on previous words. Additionally, we can also create hidden features by learning features based on next words, which is formulated as follows:

$$h_t = \sigma(W^{hh} h_{t+1} + W^{hx} x_t + b). \tag{10}$$

RNNs can theoretically capture long-term dependencies, but it is hard to accomplish this in reality due to the gradient vanishing problem. LSTM uses input, forget, and output gates to maintain more persistent memory to capture the long-term dependencies. It is formulated as follows:

$$i_t = \sigma(W^{(i)} X^{(t)} + U^{(i)} h^{(t-1)}) \tag{11}$$

$$f_t = \sigma(W^{(f)} X^{(t)} + U^{(f)} h^{(t-1)}) \tag{12}$$

$$o_t = \sigma(W^{(o)} X^{(t)} + U^{(o)} h^{(t-1)}) \tag{13}$$

$$\tilde{c_t} = tanh(W^{(c)} X^{(t)} + U^{(c)} h^{(t-1)}) \tag{14}$$

$$c_t = f_t \circ c_{t-1} + i_t \circ \tilde{c_t} \tag{15}$$

$$h_t = f_t \circ tan(c_t), \tag{16}$$

where $W^{(i)}, W^{(f)}, W^{(o)}, W^{(c)}, U^{(i)}, U^{(f)}, U^{(o)}$, and $U^{(c)}$ are weight matrices depending on the input word vector and preceding hidden features. The Bi-LSTM is later concatenated with hidden features of other embeddings as well as a softmax trained on weighted vectors where the binary presence of "1" is replaced with $w(f_{ijx})$ for each token in the text.

3.3 Demographic Embedding

Demographics can have a profound impact on individuals' language usage tendencies and psychometric characteristics [8]. We build a novel demographic word embedding to capture nuances and norms inherent to different demographic segments. More specifically, the demographic embedding identifies segments with the greatest entropy for a target psychometric dimension such that modeling within versus across such demographics may alleviate systematic bias [21] and enhance classification potential by better aligning embeddings with users' underlying semantic intent. Figure 3 illustrates the key intuition behind the demographic embeddings, which are somewhat analogous to the segment-specific modeling idea espoused by techniques such as classification and regression trees (CARTs). Using real data and the demographic embedding described in this section, the figure shows how constructing embeddings within text associated

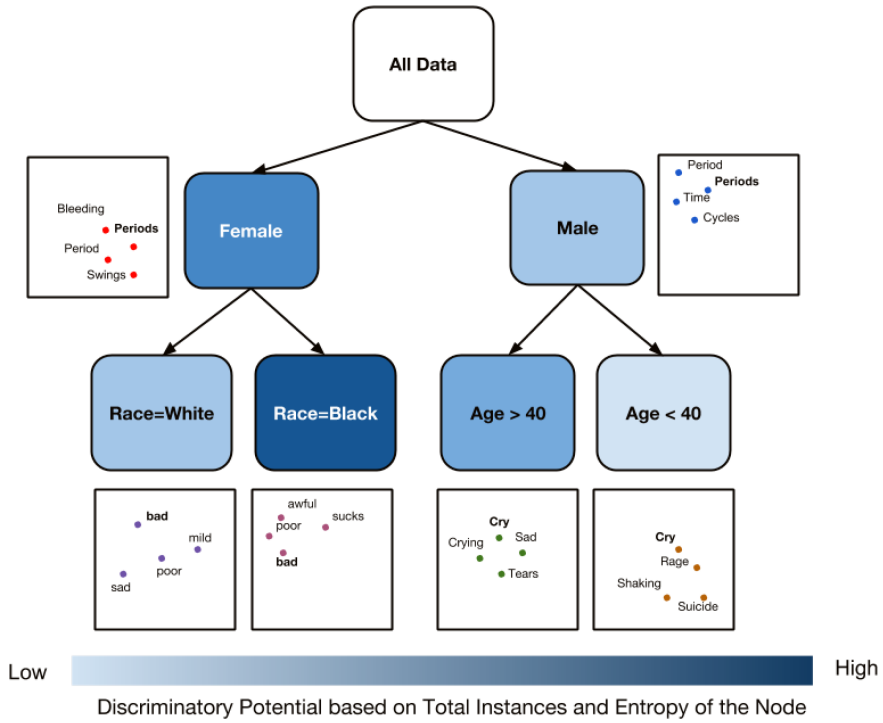

Fig. 3. Illustration of the intuition behind demographic embeddings.

with demographics such as gender, age, and race can alter the word embedding space. For instance, depending on the gender of the writer, terms such as "period" and "periods" can connote biological versus temporal meaning. Words such as "bad" can signify stronger emotional valence when used by certain races. Similarly, the word "cry" may be associated with varying health and wellness implications depending on the user's age. The demographic embedding attempts to calibrate users' embeddings based on various such demographic considerations. Details are as follows.

The first task is to identify demographic variables that significantly affect the psychometric dimensions of interest. We use a decision tree model to accomplish this task. Given a dataset $\{a_1, a_2, \ldots, a_M, C\}$, where $A = a_1, \ldots, a_m, \ldots, a_M$ is the set of input demographic attributes and $C = \{c_1, c_2, \ldots, c_N\}$ is the target psychometric classes, the decision tree partitions this dataset S into subsets using "nodes" according to input attribute a_m at certain splitting values $v \in V(a_m)$. $V(a_m)$ is the set of all possible values for attribute a_m. The goal is to create tree subdivisions that provide discriminatory potential for a given target class c_n. In this study, we use the entropy-based information gain metric as the node selection metric.

Given a target class C with possible values $\{c_1, c_2, \ldots, c_m\}$ and probability mass function $P(C)$, the entropy H for the target class is defined as:

$$H(C) = -\sum_{i=1}^{m} P(c_i) log_2 P(c_i). \tag{17}$$

The information gain measures the reduction of entropy for target classes when further splitting the dataset by a new input attribute a_m. Discretization is applied to continuous attributes before

calculating the information gain. Specifically, the information gain of introducing an attribute a_m is defined as:

$$G(C, a_m) = H(C) - H(C|a_m = v_m), \tag{18}$$

where $H(C)$ is the entropy of the class label C and the second term is the expected entropy after the dataset is partitioned using attribute a_m at value v_m.

For the demographic embedding, we build two types of decision trees. The first type utilizes all demographic variables, termed as "global tree" T_g. The second type consists of a collection of "local trees" T_{lj}, each of which excludes one among the demographic variables. In the same spirit as the random forest algorithm [9], these local trees build on a random subset of input attributes to alleviate the possible dependency on a few dominant attributes. In order to be computationally feasible, we employ a binary tree structure and use depth parameter $d = 1, 2, \ldots, D$ to control the tree size. The demographic trees are formulated as follows:

$$T_g = \{a_m = v_m | a \in A, ht(T_g) = d\} \tag{19}$$

$$T_{lj} = \{a_m = v_m | a \in A, \{a_i\}, ht(T_{lj} = d)\}, \tag{20}$$

where $ht()$ is the height function of the tree. The most prominent demographic conditions affecting the psychometric classes are selected based on node score I:

$$I(a_m = v_m) = \frac{NA(a_m = v_m)}{H(C|a_m = v_m)} + \frac{N(a_m = v_m)}{N(S)}, \tag{21}$$

where $a_m = v_m$ is the node representing a condition defined by an attribute a_m and its splitting value vm (e.g., Age = 35); $NA(a_m = v_m)$ is the average of the accuracies of all the leaves underneath this node; $H(C|a_m = v_m)$ is the entropy with regard to class label for this node; $N(a_m = v_m)$ is the number of data points belong to this node; and $N(S)$ is the total number of data points in the dataset.

The final set of demographic conditions M incorporated include the root node of the global tree and the top $K - 1$ nodes (ranked by node score I) from the local trees:

$$M = \{a_0 = v_0 | T_g\} U \{a_m = v_m | T_{lk}, l(a_m = v_m) \in r_{tl}(I_1, I_2, \ldots, I_{K-1})\}, \tag{22}$$

where $a_0 = v_0 | T_g$ is the root node condition for the global tree and $r_{tl}(I_1, I_2, \ldots, I_{K-1}$ is the top $K - 1$ node scores for the local trees. The demographic embedding leverages this information as follows:

(1) Let m_k represent one of the K elements in M. For each m_k, we identify a subset of individuals satisfying that condition in the training set and construct a subcorpus comprising text only belonging to those individuals. We use word2vec to learn an l-sized word embedding vector for each word j in the subcorpus m_k such that $w_{kj} = (w_{kj1}, w_{kj2}, \ldots, w_{kjl})$. We also train a general word embedding across each word in the entire training set $w_j = (w_{j1}, w_{j2}, \ldots, w_{jl})$.

(2) For each individual u^i, we can identify the subset $M_s = \{m_1, m_2, \ldots, m_s | m \in M\}$ of demographic conditions applicable to that user. Following the average embedding idea [61], the demographic embedding weight wd_{ij} for word j appearing in a text instance associated with individual u^i is defined as the weighted average of node score I_w and the node-specific word embedding w_{sj}:

$$wd_{ij} = \begin{cases} \frac{(\sum_s^{|M_s} I_w * w_{sj})}{|M_s|} & if\ |M_s| > 0 \\ wd_{ij} & if\ |M_s| = 0. \end{cases} \tag{23}$$

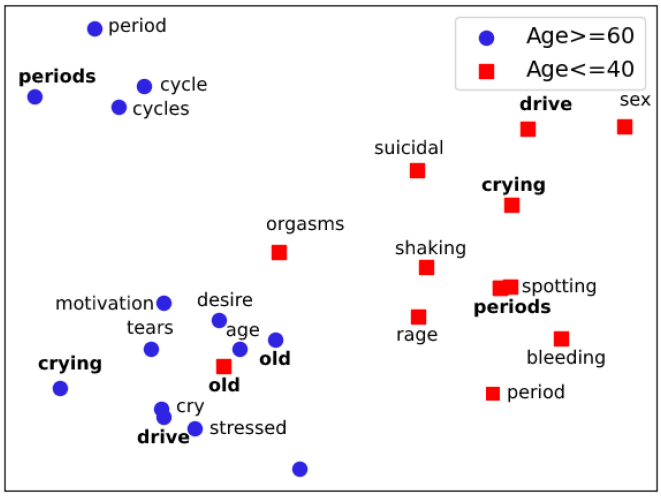

Fig. 4. Example embedding calibration based on demographics.

As discussed earlier in the beginning of the section using a small real-data-based illustration, demographic embeddings allow the word embeddings for a user to be calibrated based on the user's demographics. The visuals appearing in Figure 3 are naturally illustrative, not exhaustive. Figure 4 delves deeper by showing another example of how that calibration looks, in this case for users above or below the age of 40. From the figure we can see that words such as "drive," "crying," and "old" have different semantic connotations for different age groups. As we later demonstrate in the evaluation section, the inclusion of such demographic embeddings can allow more precise representation of words that might otherwise be susceptible to mean-centering across diverse subsets of the user population when modeled using a single global word embedding.

3.4 Structural Equation Model (SEM) Encoder

Psychometric dimensions are inherently correlated. For example, a patient with high anxiety associated with seeing a physician may also have low self-esteem [22]. In order to incorporate such secondary psychometric dimension information in PyNDA, we propose a novel Structural Equation Model (SEM) encoder. The underlying intuition behind our encoder is similar to the feature augmentation idea commonly used in multitask learning, which has been shown to offer significant performance lifts. Similarly, as illustrated in the ablation analysis in Section 4, our SEM encoder significantly enhances performance for classification of psychometric dimensions. Details are as follows. In order to incorporate such secondary psychometric dimension information in PyNDA, we propose a novel SEM encoder. The underlying intuition behind our encoder is similar to the feature augmentation idea commonly used in multitask learning, which has been shown to offer significant performance lifts. Similarly, as illustrated in the ablation analysis in Section 4, our SEM encoder significantly enhances performance for classification of psychometric dimensions. Details are as follows.

SEM is a general multivariate statistical modeling technique to depict and test relationships among variables related to psychometric measures [36]. It models the psychometric dimensions as latent variables and discovers their most suitable relationships based on data. Figure 5 illustrates the important components of an SEM. The circles represent latent variables (or psychometric dimensions), whereas the straight arrows signify the relationships between them. P denotes the path coefficients for relationships. R^2 is the variance explained in a consequent variable by a set

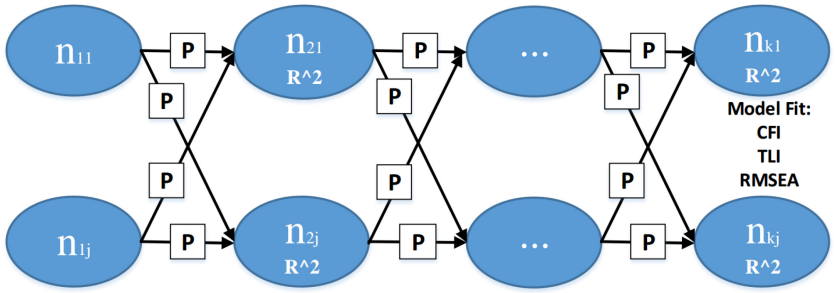

Fig. 5. Structural equation models for psychometric dimensions.

of antecedent variables. Model fit refers to the ability of an SEM model to fit the data. Common model fit indices include Comparative Fit Index (CFI), Tucker Lewis Index (TLI), and Root Mean Squared Error of Approximation (RMSEA). CFI and TLI are comparative fit indices that compare the model of interest with an alternative, such as a null model. They range between zero and one, with a higher value indicating a better fit. RMSEA is an absolute measure of fit based on the noncentrality parameter. Higher RMSEA indicates poor model fit. The SEM encoder aims to incorporate these multivariate, structured correlations between psychometric dimensions into PyNDA. Specifically, we build a series of SEM models for a given target psychometric dimension of interest along with other dimensions potentially affecting it. Let S represent a set of SEM models for a target psychometric dimension. Each model G_i in S can be considered a directed graph containing latent variables (or nodes) and directed links, arranged in a linear sequence with K levels and J nodes for each level such that node n_{kj} at level $k \geq 1$ connects to each n_{k+1j} in the next level. P is the path coefficient from an antecedent n_{kj} leading to a consequent variable n_{k+1j} with variance R^2 across all of its inbound antecedent links from level k. The target psychometric dimension only appears in level $k \geq 2$ to ensure it has antecedent variables and valid P and R^2. For each G_i, we can obtain the model fit indices CFI, TLI, and RMSEA. In order to include a balanced model fit measure, we use $MF = (CFI + TLI + (1 - RMSEA))/3$ to depict the average model fit indices. For each nontarget variable $v \in V$, we find a subset S^* of all the SEM models containing them. We use a scoring function that weights path coefficients and model fit indices equally to summarize the relevance of any v to the target variable:

$$w(v) = \frac{1}{S^*}\left|\sum_{S^*} P\right| + \frac{1}{2|S^*|}\left(\sum_{S^*} R^2 + \sum_{S^*} MF\right). \tag{24}$$

Finally, for each target variable we can derive the top K from V based on $w(v)$ values. In order to avoid future leaks, we assume that V is unknown for test instances and must be predicted. Hence, a model is built on the training data to jointly score each selected v. This is done using a standard word embedding, followed by a Bi-LSTM layer and a fully connected dense layer to classify the selected independent variables. The learned dense layer is then directly concatenated with the ones yielded by other embeddings to classify the target psychometric dimensions of interest.

3.5 Structural Multitask Learning

Given the user-centric nature of psychometric analysis, structural relationships among target psychometric measures provide a unique opportunity for multitask learning. For example, if "trust in doctors" and "anxiety of seeing physicians" are correlated, we can share their input text features

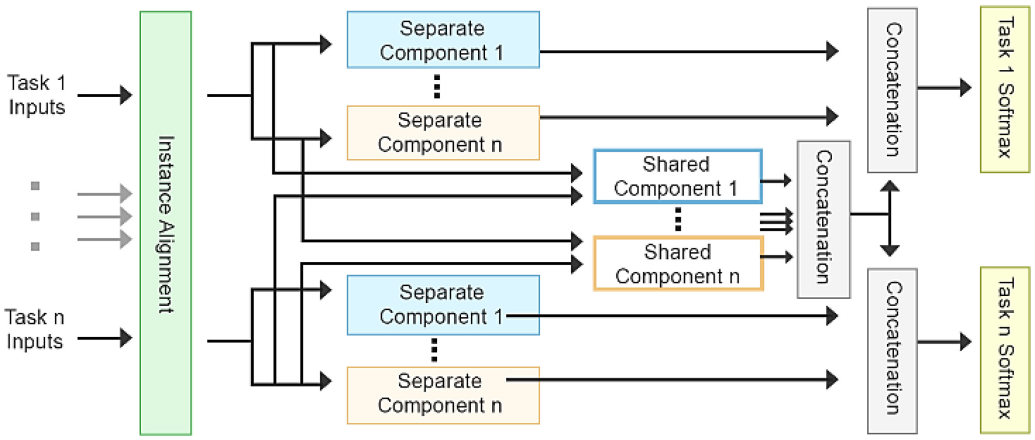

Fig. 6. Structural multitask learning for psychometric dimensions.

and jointly train the two classifiers together to augment the feature set for the current task. Notably, the relationship among psychometric measures follows a hierarchical structure, as discussed in the SEM encoder section. Therefore, we can build a structural feature sharing representation to reflect this unique property.

Figure 6 presents our proposed MTL approach. Suppose we have four target variables of interest and wish to share features among them. Following [41], we create "separate LSTMs" consisting of task-specific features and a single "shared LSTM" as a cross-task representation that reflects common patterns and cues across the different classification tasks. We jointly train these classifiers to allow feature sharing. In order to maintain orthogonality between shared and separate representations [41], adversarial training is used to optimize the purity of the shared representations. The idea is to build two neural networks, generator and discriminator, to combat one another. The generator tries to generate the purest shared feature set, while the discriminator attempts to distinguish the shared features into specific tasks. In our case, the single "shared LSTM" is used as the generator that works adversarially against a multilayer perceptron as the discriminator. For the generator to learn to purify the shared feature set, it needs the gradients from the discriminator, which are propagated back via a "Gradient Reversal Layer (GRL)" [26]. GRL applies an identity function on the inputs in the forward pass and sends the negative gradients back in the backward pass, which enables learning in the generator network. To train the discriminator, we simply map the shared representations into a probability distribution predicting what kinds of tasks can be inferred. Cross-entropy loss is used to train the discriminator.

The tension between two networks results in convergence when the discriminator is no longer able to perform such differentiation, thereby resulting in better feature sharing. Collectively, this is accomplished via the loss function $L = L_{Task} + \lambda L_{Adv} + \gamma L_{Diff}$. As depicted in Figure 4, we extend this idea for our context in two ways. First, we perform MTL across various components of our architecture, including the final concatenation layer, demographic embedding LSTM, representation embedding LSTM, and character CNN:

$$L_{Task} = \sum_{k=1}^{K} \alpha_k \, L(x(k), y(k)), \tag{25}$$

where $L(x(k), y(k))$ is the cross-entropy of the true/prediction distributions on the training set, K is the product of the tasks and architecture components, and k is a given task architecture component with a learned weight α_k. This allows flexibility in accounting for interaction effects between and across tasks and architecture components in our MTL setup.

Second, we perform cross-task training instance alignment by minimizing the distance between concurrent inputs (e.g., anxiety and trust text instances). Since back-propagation occurs across all tasks via the unified loss function, the idea is to input texts from different tasks that are similar on psychometric dimensions in order to challenge the discriminator (i.e., L_{Adv}). For instance, given the target task variable is anxiety, for an input instance with high anxiety, we may wish to pair it with a low or high trust text instance from a different user that also has high anxiety. Since minimizing such distance between concurrent training instances is an NP-hard problem, we use a simple greedy heuristic where the tasks concurrent to the target are ranked based on their $w(v)$ as defined in Equation (25). The target tasks' training instances are then paired with the nontarget task with the highest $w(v)$ based on Euclidean distance between users' psychometrics. These tuples are then compared with instances from the next task based on minimal distance, and so on, resulting in concurrent instance sets that are somewhat homogenous across psychometric dimensions, allowing MTL to learn richer separate and shared features. As later demonstrated in the ablation analysis, structural multitask learning with adversarial training provides an additional performance lift for our psychometric extraction tasks.

4 TESTBED

In order to evaluate the proposed PyNDA architecture, an extensive research testbed was constructed, comprising three datasets and 11 total classification tasks. While psychometrics are known to be important in various application domains including security and e-commerce, in this study we focused on the health domain. The first two datasets encompassed four important psychometric dimensions known to be predictive of health outcomes.

1) *Health Literacy (HL)* - In essence, HL is a subjective construct reflecting how much one thinks one knows about health [46]. Low HL has been associated with increased mortality, increased hospitalization, and poor adherence and self-maintenance to a host of chronic diseases such as diabetes, heart disease, and risk of stroke [46].

2) *Health Numeracy (HN)* - Conversely, health numeracy (HN) is an objective construct reflecting the ability to calculate, use, and understand numeric and quantitative concepts in the context of health issues. HN has been associated with outcomes such as the ability to understand dosage in medication and adherence to treatment [14].

3) *Trust in Doctors (TD)* - Perceptions of trust in physicians/doctors (TD) can have an important mediating role on health outcomes [22].

4) *Anxiety Visiting Doctors (AV)* - Anxiety when visiting the doctor's office is another strong mediator for health outcomes such as future doctors' visits and wellness [54].

For each of the four aforementioned psychometric dimensions (HL, HN, TD, and AV), well-established survey items have been developed in the literature. These items can be used to compute individuals' scores on a fixed continuous scale (e.g., 1–10). In order to construct our user-generated text datasets, we developed equivalent free response questions with accompanying text boxes that immediately followed the survey items. These questions were validated through pretesting and were found to nicely represent the target variable for each user's collected text.

Table 2 summarizes the three datasets and related classification tasks incorporated in our testbed. Consistent with several prior psychometric and NLP studies, for our first dataset we used Amazon Mechanical Turk (AMT) since it is considered somewhat representative of the broader Internet population. Each respondent provided quantitative and text responses for all four target

Table 2. Summary of Test Bed: Three Datasets and 11 Tasks

Characteristics	AMT	Qualtrics	HealthForum
Text Instances	4,262	4,240	138,998
Classification Tasks	Subjective Literacy (HL), Health Numeracy (HN), Trust in Doctors (TD), Anxiety in Visiting (AV)	Subjective Literacy (HL), Health Numeracy (HN), Trust in Doctors (TD), Anxiety in Visiting (AV)	Drug Experience, Age, Gender
Demographics			
Race	81.2% white, 7.4% black	50% white, 50% black	Unavailable
Age (Mean)	37.4	45.6	39.9
Gender (Male)	48.3%	24.2%	29.4%
Income (USD)	62% < $55K	67% < $55K	Unavailable
Education (College Grads)	44.6%	32.1%	Unavailable

dimensions of interest, some additional secondary dimensions, plus demographics such as age, gender, race, income, and so forth. A total of 4,262 usable user responses were collected.

The third dataset was composed of 138,998 user drug experience assessments collected from an online health forum. The major psychometric dimension of interest in this dataset was users' self-reported prescription drug experience ratings, which appear on a 1-to-5 scale. In addition to the ratings, users in this forum provide some basic demographics such as age and gender, and text comments describing their experience, reasons for taking the drug, and potential side effects. Although demographics such as age and gender are not psychometric dimensions, due to their close relation to psychometrics, we included user age and gender as additional classification tasks for evaluating PyNDA and comparison methods. The health forum dataset was included due to its complementary nature to the AMT and Qualtrics datasets with respect to number of instances, dimensions, and response collection mechanism. Collectively, the testbed was composed of a diverse array of datasets, tasks, and user content channels.

5 EVALUATION

5.1 Experiment Results - Benchmark Methods

In order to assess the performance of our PyNDA architecture, we conducted an extensive benchmark evaluation in comparison with 16 text classification techniques, presented in Table 3. The comparison methods belong to five categories: feature-based classifiers, CNNs, LSTMs, hybrid deep learning architectures, and multitask deep learning methods. While the selected techniques are not an exhaustive list, they are representative of state-of-the-art approaches in each of the five categories. Included were well-established feature-based classifiers, such as the Multinomial Naive Bayes [43], Logistic Regression [28], FRN [3], FastText, and Linear SVM [48]. Also selected were widely used CNN architectures, such as CNNSent [35], CNNChar [60], VD-CNN [17], SWISS-CHEESE [19], and SENSEI-LIF [42]. In order to further enrich the evaluation, we also built several custom CNN architectures, such as CNNWordRep, which uses CNN to build word and representation embeddings before inserting these into the dense layers for final classification, and CNNCombine, which uses CNN to build word and representation embeddings simultaneously. Prominent LSTM architectures were also selected, including the basic LSTM [39], LSTMWordRep [56], and LSTMCombine. The LSTMsThenCNNs [15] method was included as a hybrid deep learning architecture.

For feature-based classifiers, consistent with prior work, we adopted unigram features with tfidf weighting since these yielded the best performance. Additionally, for multinomial Naive Bayes,

Table 3. Summary of Benchmark Results

Category	Method	Accuracy	Precision +High	Precision -Low	Recall +High	Recall -Low	F-score +High	F-score -Low	ROC AUC
Feature	FastText [34]	73.9	73.4	74.4	74.9	72.9	74.2	73.6	73.9
	FRN [3]	74.9	73.0	75.0	74.3	74.8	73.7	74.9	74.9
	Linear SVM [48]	73.5	73.3	73.7	73.8	73.1	73.6	73.4	73.5
	LogisticRegression [28]	74.6	73.9	75.6	76.1	73.2	75.0	74.3	74.7
	Multinomial NB [43]	73.4	72.9	74.8	75.7	71.2	74.3	72.9	73.5
CNN	CNNSent [35]	74.4	73.4	76.4	77.9	70.8	75.6	73.5	74.4
	CNNChar [60]	66.6	66.2	68.0	69.4	63.7	67.8	65.8	66.6
	CNNWordRep	73.6	73.4	74.4	74.7	72.6	74.0	73.5	73.6
	CNNCombine	73.2	72.2	74.7	75.2	71.1	73.7	72.8	73.2
	SENSEI-LIF [42]	73.4	73.2	74.1	74.4	72.3	73.8	73.2	73.4
	SWISSCHEESE [19]	74.1	73.6	73.9	76.3	71.8	74.9	72.9	74.1
	VeryDeepCNN [17]	58.9	58.1	60.4	63.2	54.2	60.5	57.1	58.7
LSTM	LSTM [39]	72.6	72.5	73.5	73.5	71.8	72.9	72.6	72.6
	LSTMCombine	74.8	74.8	75.1	75.5	74.1	75.2	74.6	74.8
	LSTMWordRep [56]	74.5	74.6	74.7	74.6	74.3	74.6	74.5	74.4
Hybrid	LSTMsThenCNNs [15]	75.1	74.1	76.8	77.6	72.5	75.8	74.5	75.1
PyNDA		**81.1**	**80.2**	**82.1**	**82.7**	**79.4**	**81.4**	**80.7**	**81.0**

we used Laplace smoothing, and we ran logistic regression using the L2 penalty with LibLinear solver with 100 maximum iterations. For SVM, we adopted the L2 penalty with squared hinge loss function, with 1,000 maximum iterations. For all the deep learning benchmarking models as well as PyNDA, we tuned the settings and parameter values on a validation set. The text representations for all benchmarking deep learning models were those used in the original studies.

All methods were evaluated using fivefold cross-validation (with 80% training and 20% testing per fold). For PyNDA, hyperparameters were tuned lightly on a validation subset within the training data. For comparison methods, a more in-depth combination of grid and random search was used in order to ensure a fair comparison and that PyNDA's performance lift was not simply attributable to parameter settings.

Using this process, for Fasttext [43], a learning rate of 0.001 was used. For CNNSent [28], the batch size was set to 6 and the number of dimensions for the word embedding was 128. For SENSEI-LIF [23], 128 dimensional embeddings were utilized along with 64 filters in the CNN. The LSTMsThenCNNs [57] was run with one layer for both CNN and LSTM, with 64 filters in the CNN and 128 nodes per layer in the LSTM. VeryDeepCNN [65] performed best using a three-layer CNN with 64 filters per layer. CNNChar [37] also performed best with one layer and 64 filters. SWISSCHEESE [42] was run with two layers and 128 dimensional embeddings. For all comparison methods with CNNs, we tried different kernel sizes (3, 5, 7, 12) and found size = 7 to typically work the best for benchmark techniques.

The parameter settings for our proposed architecture were as follows. For the character embedding, we used one hot encoding—the CNN embedding consisted of two layers of 1D convolution filters followed by maxpooling layers. We used 128 filters for each layer with a kernel size of 7 and maxpool size of 3. The BiLSTM layers for representation and demographic embedding used two layers of bidirectional LSTMs with 128 units in each layer. Finally, the dense layers for all the embeddings consisted of two fully connected layers with 256 units each. Regularization was done using a dropout value of 0.5. We used a batch size of 16, and 10 epochs.

For parallel representations, n-grams up to n = 4 were extracted for representations depicted in Table 1. For GBS, the subsumption threshold t was set to 0.05, and the cross-category thresholds l and p were each set to 0.95. For the SEM encoder, K was learned dynamically from the training data for each fold of a given target psychometric dimension, and ranged between 2 and 8. For our MTL method, we set λ and γ to 0.05 and 0.01 as done in [41].

Bifurcation was performed on each dataset to convert the continuous psychometric target class variables into binary high/low classification variables. Consistent with prior studies, this was done by only using instances from the end quartiles as the low and high class labels, respectively.

Table 3 presents the PyNDA results along with the 16 benchmarking methods, averaged across the 11 datasets. PyNDA significantly outperformed the benchmarking methods across all evaluation metrics, including accuracy, precisions, recalls, and receiver operating characteristic curve area-under-the-curve (ROC). The overall accuracy, F-measures, and ROC for PyNDA were at least 5% higher than the second best method. Among benchmarking methods, LSTM architectures were better than CNN, underscoring the importance of capturing long-term dependencies among texts for more effective psychometric classification. CNNChar yielded the worst results, suggesting that morphological patterns may not be critical indicators for psychometric-related texts. Instead, word- or sentence-level features may have more predictive power, as illustrated by the relatively higher performance for CNNSent and CNNWord. Given the recent effectiveness of CNN approaches in sentiment classification tasks (e.g., [19, 42]), the relative superiority of LSTM in our context reinforces previously stated notions of the complexity of the nuanced psychometric dimensions examined in our study. Feature-based classifiers demonstrated reasonable performance, relative to alternative benchmarking methods.

Figure 7 depicts the accuracy and F-measures of the best method for each category, broken down by the 11 psychometric classification tasks across the three datasets. In general, PyNDA outperformed the second-best methods on 10 (out of 11) measures regarding accuracy, all measures in terms of F-measure for the positive class, and 10 measures for F-measure for negative class. Collectively, the results showcase the efficacy and utility of our proposed architecture. The results also suggest that the amalgamation of CNN, LSTM, and multitask learning coupled with rich underlying embeddings and encoders offers robust classification performance across myriad datasets and psychometric classification tasks.

5.2 Experiment Results - Ablation Analysis

PyNDA encompasses novel embeddings, encoders, and a multitask learning scheme. In order to examine the additive impact of each component of the architecture, ablation analysis was performed. We compared the full PyNDA against a base version encompassing only the CNN with character embedding. We then incrementally added the parallel representations with an LSTM, the full representation embedding using GBS, the demographic embedding and SEM encoder, and, finally, multitask learning.

The top half of Table 4 shows the summary (averaged) results across all 11 tasks associated with our three test beds. From the table, it is apparent that each additional component included in the ablation analysis bolstered performance. For instance, including the parallel representations over the base character embedding enhanced accuracy by 8 percentage points. Inclusion of the full representation embedding added an additional 4-point lift. Similarly, the demographic embedding, SEM encoder, and MTL enhanced accuracy by about 2 percentage points, on average. Paired t-test results across the five folds of the 11 datasets (i.e., n = 55) revealed that each additional component significantly enhanced performance over the prior ablation setting (all p-values < 0.05; df = 54).

We also evaluated two alternative MTL setups. The first was a separate word embedding and separate and shared LSTMs, exactly as proposed in [41], concatenated with the rest of our

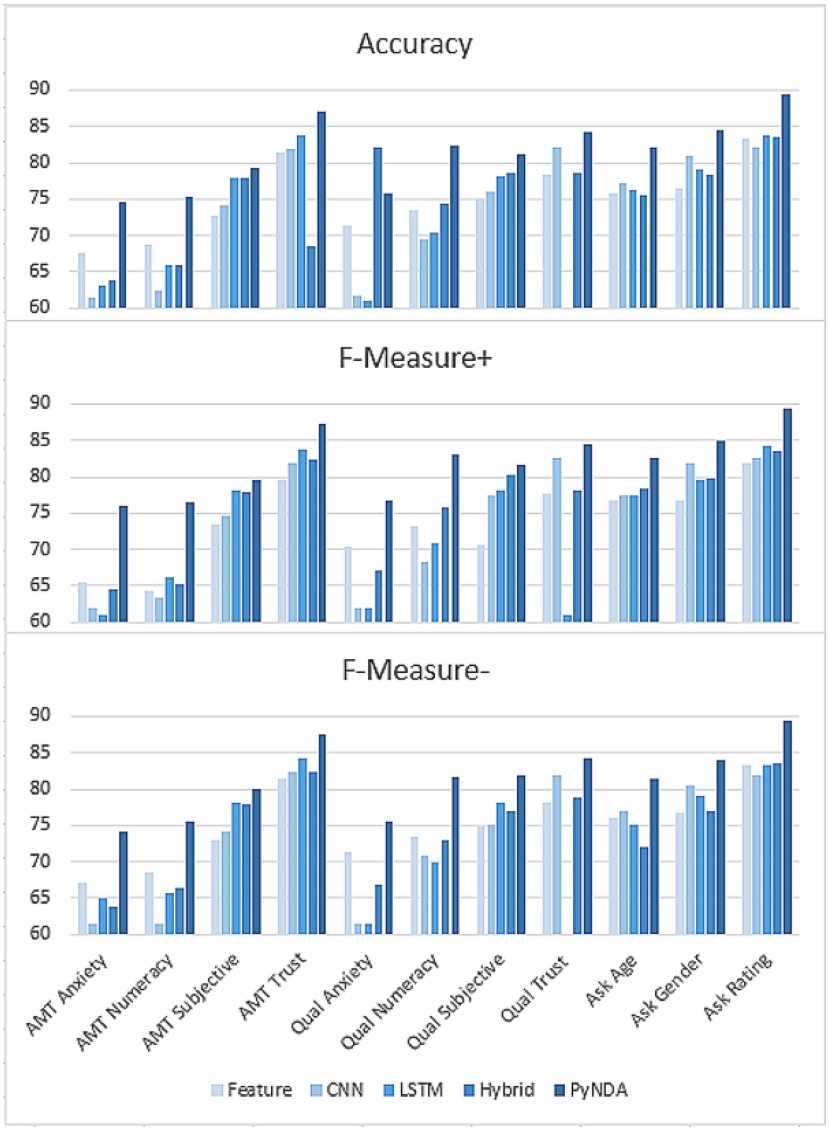

Fig. 7. Accuracy and F-measures for each task across three datasets.

architecture. The second was our MTL with only the final concatenation layer (i.e., no embedding LSTM-level weights). The results, depicted in the middle of Table 4, show that the more holistic application of MTL in PyNDA, with inclusion of finer-grained component-level weights in L_{Task} coupled with training instance alignment, boosts accuracy by 1.5% to 4% over alternative setups.

Finally, we examined the effectiveness of the demographic embeddings and parallel representation-weight-based LSTMs relative to the use of pretrained embeddings. The results from these comparisons appear at the bottom part of Table 4. Replacing the demographic embeddings used in our proposed PyNDA architecture with a pretrained word embedding decreased accuracy and class-level precision and recall by 1% to 2% across the board. We also explored the lift of the demographic embedding versus simply adding the demographic variables

Table 4. Summary of Ablation Analysis Results

Ablation Setting	Acc	Prec+	Prec-	Rec+	Rec-
CharEmbeddingCNN	66.6	66.2	68.0	69.4	63.7
+ParallelRepsLSTM	74.8	74.8	75.1	75.5	74.1
+RepEmbedding	79.4	78.9	80.1	80.6	78.2
+DemEmbd&SEMEnc	80.3	79.5	81.3	82.0	78.6
+MultiTaskLearning	81.1	80.2	82.1	82.7	79.4
Alternative Multitask Learning (MTL) Setups					
MTLSeparateWord	77.0	77.1	77.2	76.9	77.0
MTLNoComponents	79.6	79.3	80.0	80.5	78.7
Demographic vs. Pretrained Embedding					
DemEmbd&SEMEnc	80.3	79.5	81.3	82.0	78.6
DirectDem&SEMEnc	79.5	78.9	79.8	80.6	78.3
PretrainedDemEmbd&SEMEnc	79.2	78.6	79.9	80.1	78.1
Parallel Rep LSTM vs. Pretrained Embedding					
ParallelRepsLSTM	74.8	74.8	75.1	75.5	74.1
PretrainedEmbdsLSTM	73.1	72.8	73.6	73.8	72.2

directly to PyNDA (called DirectDem in Table 4)—the embedding enhanced overall results by 1%. As later illustrated in Figure 11, the deltas are even more pronounced for certain key statistical (and otherwise) minority groups in the data. The results further underscore the utility of the explicit split representation employed by the proposed demographic embedding for enhancing predictive power while debiasing. Similarly, replacing the "ParallelRepsLSTM" in PyNDA with the pretrained embedding weights dropped accuracy, precision, and recall by 1.5% to 2%, highlighting the value of the representation embeddings with LSTMs relative to the pretrained embeddings.

Figure 8 shows ablation analysis accuracies for each of the 11 tasks in our three datasets. The x-axis shows the impact of the five ablation settings. For almost every task, we observe a general upward trajectory as additional components of PyNDA are incrementally introduced. Though not depicted here due to space constraints, similar plots were observed for positive and negative class F-measures. The figure demonstrates the robustness of each PyNDA component's additive contribution across the 11 tasks.

In order to dig deeper into the impact of the five components of PyNDA, we examined the effectiveness of various combinations of component subsets relative to use of all five components. Table 5 presents the experiment results. Due to space constraints, we do not report all possible combinations. Rather, select good-performing pairs, triples, and four-component combinations are presented along with the results for the full PyNDA configuration. Once again, results were averaged across the 11 datasets. In general, PyNDA outperforms various subset combinations by 1 to 2 percentage points. Interestingly, looking at the occurrence frequency for components in top-performing combinations, we can see that the parallel representations are utilized the most (i.e., the representation embedding and the representation-weight-based LSTM), followed by the demographic embedding + SEM encoder and multitask learning. With the exception of the baseline character embedding CNN, all components seem to complement one another. However, it is worth noting that the results presented here are averaged across the 11 datasets. As depicted earlier in Figure 8, performance for components can vary a bit depending on the psychometric dimension of interest. Nevertheless, the results further underscore the design efficacy and robustness of the various components of PyNDA.

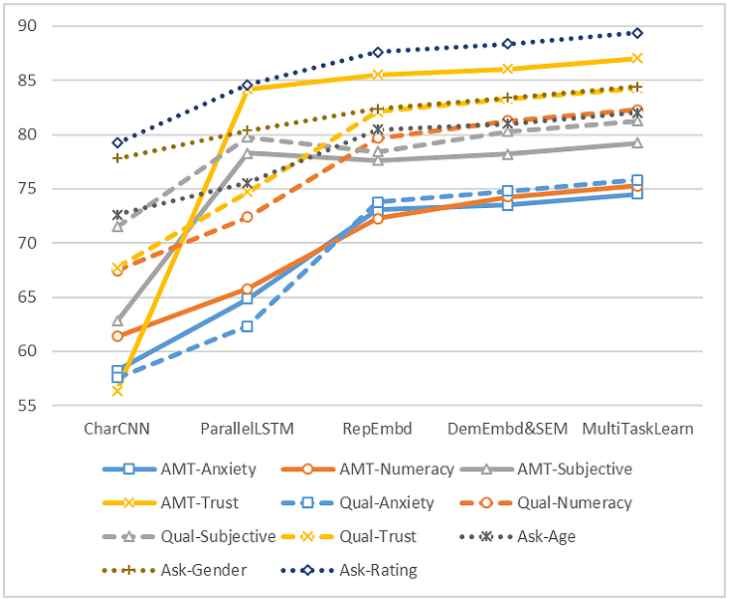

Fig. 8. Ablation analysis accuracy for each task across three datasets.

Table 5. Summary of Combinatorial Analysis Results

PyNDA Component Combination Setting	Acc	Prec+	Prec-	Rec+	Rec-
Individual Components					
CharEmbeddingCNN	66.6	66.2	68.0	69.4	63.7
ParallelRepsLSTM	74.4	74.3	75.0	75.2	73.7
RepEmbedding	76.2	75.1	76.8	76.9	74.8
DemEmbd&SEMEnc	75.8	75.2	76.6	76.8	75.0
MultiTaskLearning (MTL)	75.8	75.0	76.8	76.8	75.1
Select Two-Component Combinations					
RepEmbd+DemEmbd&SEMEnc	77.4	77.1	77.8	77.9	77.0
MTL+ParallelRepsLSTM	75.9	75.6	76.4	76.8	75.1
MTL+RepEmbd	77.8	77.3	78.1	78.6	76.9
CharEmbdCNN+MTL	76.0	75.2	76.9	77.4	74.6
ParallelRepsLSTM+DemEmbd&SEMEnc	76.3	75.3	77.0	77.9	74.8
Select Three-Component Combinations					
CharEmbdCNN+ParallelRepsLSTM+DemEmbd&SEMEnc	76.3	75.9	76.7	77.1	75.6
MTL+DemEmbd&SEMEnc+ParallelRepsLSTM	77.9	77.7	78.7	78.9	77.3
RepEmbd+MTL+DemEmbd&SEMEnc	80.2	80.0	80.3	80.9	79.4
MTL+DemEmbd&SEMEnc+ParallelRepsLSTM	80.0	79.6	80.3	81.2	79.1
Select Four-Component Combinations					
RepEmbd+MTL+ParallelRepsLSTM+DemEmbd&SEMEnc	80.6	79.5	81.4	81.8	79.2
CharEmbdCNN+MTL+DemEmbd&SEMEnc+RepEmbd	80.3	79.8	80.8	81.3	79.1
PyNDA	**81.1**	**80.2**	**82.1**	**82.7**	**79.4**

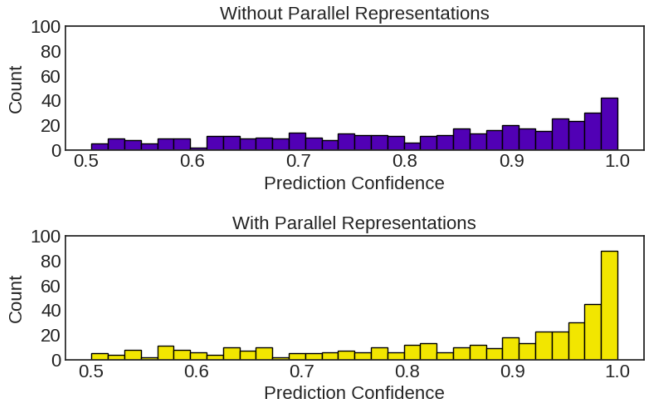

Fig. 9. Impact of parallel representations on bi-LSTM prediction confidences for true positives/negatives.

Table 6. Illustration of How Representation Embeddings Enhance Semantic Richness

Representation	Word	Neighbors in Representation Embedding	Neighbors in Default Embedding (word2Vec)
Sentiment	good	happy, strong, healthy	**bad**, strong, **dangerous**
	bad	serious, hard, heavy	**good**, low, **high**
	mild	minimal, constant, minor	**weird**, slight, minor
Domain Lexicons	back	sleep, pregnant, muscle	**down**, **off**, **through**
	stomach	chest, heart, abdomen	**mood**, breast, **allergies**
	pill	tablet, antibiotic, injection	**shot**, **round**, **method**

One critical component of PyNDA is the parallel representations and the related representation embedding that utilizes GBS to allow greater semantic, syntactic, and stylistic richness in the input space. Figure 9 shows how the Bi-LSTM weights for true positive and negative cases are enhanced by inclusion of the representation embedding. In general, the Bi-LSTM classifier becomes more confident for its correct predictions, suggesting that the parallel representations are indeed serving as an effective mechanism for enhancing regularization.

In order to dig deeper into the value proposition of the representation embeddings, Table 6 depicts the nearest neighbors for select sentiment and (health) domain-specific words in the representation embedding versus a standard word embedding trained using word2Vec. Here, nearest neighbors were computed using the standard cosine similarity measure between the words' n-dimensional embedding vectors. From the table, we can see that the default word embedding groups words with opposing sentiment polarity next to one another (e.g., neighbors for "good" and "bad"), whereas in the parallel representation, through inclusion of sentiment lexicons such as SentiWordNet, neighbors for these words are ones with similar sentiment polarity. Similarly, for domain lexicon terms such as "stomach," the parallel representation identifies other anatomy terms, and "pill" is in closer proximity to other drug administration terms such as "tablet" and "injection." Likewise, the term "back" is considered more anatomical (noun sense) and associated with sleep, pregnancy, and muscular considerations. In order to allow readers to more easily visually examine the differences in the embeddings, we visualize the embeddings in a two-dimensional plot (Figure 10), which shows the sentiment and lexicon embeddings' effectiveness in focusing on alternative semantic representations relative to the standard word embeddings. The results in

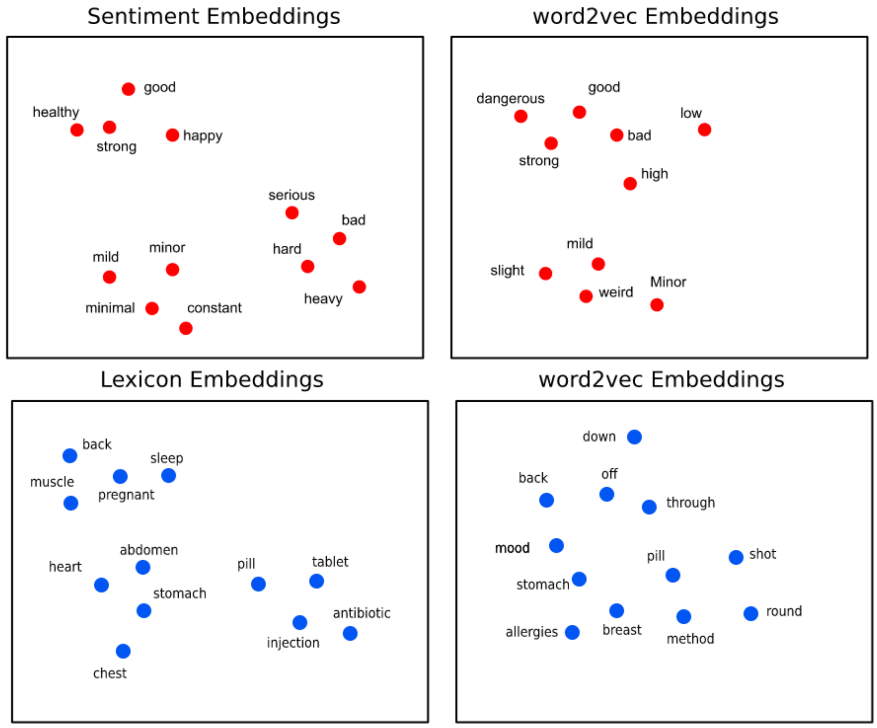

Fig. 10. Word neighborhoods in representation vs. word2vec embeddings.

Figures 9 and 10 and Table 6 shed light on how the parallel representations bolster the richness of the input space, allowing enhanced psychometric NLP capabilities.

In order to further examine the value proposition of the demographic embedding and its ability to alleviate bias, we compared its performance against a "No Demographic" variation of the architecture comprising CharCNNEmbedding, ParallelRepLSTM, and the RepEmbedding only. We also compared against the same architecture with the demographic variables concatenated to the architecture via an artificial neural network, called "Direct Demographic." The demographic embeddings outperformed each comparison method by 1% to 2% or more in average accuracy across the 11 tasks. However, the performance deltas were especially pronounced on the demographic segments identified by our demographic embedding method. As one illustration, Figure 11 depicts accuracies for three related segments on the numeracy classification task, on the Qualtrics dataset. For those without college education, income less than $55,000, and black racial affiliation, the demographic embedding outperformed the no-demographic and direct variable approaches by 3 to 5 percentage points.

It is worth noting that the segment-specific embeddings such as these ones each accounted for between 10% and 20% of all users in the dataset. Words weighted differently by the demographic embeddings for the numeracy task included "capable," "able," "interest," "understand," "capacity," and "complex." Similarly, words such as "anxious" and "worry" were weighted differently when uttered by such segments in the context of the anxiety classification task. The results reinforce the notion that calibrating the psychometric discriminatory potential of utterances based on demographic considerations can alleviate bias and enhance classification accuracy. Collectively, the ablation analysis results further underscore the robustness of the embeddings, encoder, and multitask learning environment proposed in PyNDA.

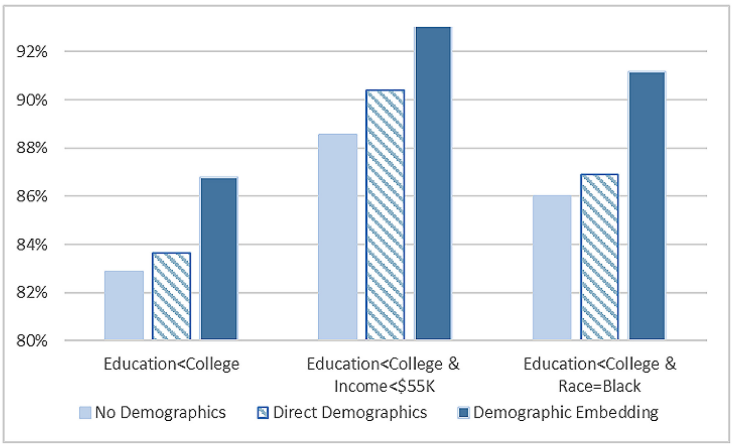

Fig. 11. Demographic embedding accuracy for select segments on qualtrics numeracy task.

6 CONCLUSION

Psychometric measures reflecting people's knowledge, ability, attitudes, and personality traits have profound implications for many important, real-world challenges, such as e-commerce, health care, and cybersecurity. However, effectively measuring and extracting rich psychometric dimensions from user-generated content in a timely and unobtrusive manner has proven elusive. In this article, we propose a novel deep learning architecture, PyNDA, to extract critical psychometric dimensions such as literacy, numeracy, trust, anxiety, and experience ratings from natural language texts. In order to address the paucity of the user-generated texts as well as to reflect the demographic sensitivity and user-centric characteristics of psychometric dimension extraction, PyNDA is composed of several proposed components, including a representation embedding, a demographic embedding, an SEM encoder, and a multitask learning mechanism. Our experiments on 11 tasks pertaining to three datasets show that PyNDA markedly outperforms traditional feature-based classifiers as well as state-of-the-art deep learning architectures.

We believe the biggest novelty of our work lies in the representation engineering phase, which encompasses the representation embedding, demographic embedding, and SEM encoder in order to effectively represent rich and diverse psychometric information. To the best of our knowledge, the ideas of utilizing an array of semantically parallel nonredundant embeddings, demographically calibrated embeddings, and structural equation modeling information in a deep learning NLP architecture are all relatively new. Further, given the limited prior work on such psychometric dimensions, designing and developing a deep learning architecture that effectively fuses these models is certainly a nontrivial undertaking. Case in point, our ablation analysis and combinatorial expeirments show that arbitrarily fusing these representational components and applying multitask learning in an unprincipled manner doesn't work well. Hence, we believe the model fusion and multitask learning arrangements constitute a key secondary technical contribution that future work can expound upon.

Given the lack of prior work focused on natural language processing methods for deriving psychometrics from secondary data, the results have important implications for information retrieval and behavior modeling. For instance, by adding attitudes and beliefs as an additional information refinement, psychometric dimensions could be used to enrich contextual search efforts that have focused on using task, sentiment, omni-channel cross-device journey, and spatial-temporal information to enhance search-related outcomes. Our work also contributes to the growing body of

literature on user modeling by introducing demographically calibrated embeddings and structural equation modeling concepts in text extraction and categorization contexts. Furthermore, PyNDA has profound practical implications: for example, it could be used to infer users' psychometric attitudes and beliefs, which drive key behaviors in various critical contexts such as health, cybersecurity, and e-commerce. Within the health domain, such models could be deployed via mobile apps to help infer patients' mental statuses related to chronic diseases in a timelier manner using mobile-generated text, thereby helping physicians conduct informed decision making and also allowing patients to better self-regulate their health statuses. In the future, we hope to extend our model to some of the other aforementioned application domains and also to deploy them in real-time synchronous chat contexts. We believe that this present study constitutes an important initial step toward these future real-world applications.

REFERENCES

[1] Ahmed Abbasi. 2010. Intelligent feature selection for opinion classification. *IEEE Intelligent Systems* 25, 4 (2010), 75–79.

[2] Ahmed Abbasi and Hsinchun Chen. 2008. Writeprints: A stylometric approach to identity-level identification and similarity detection in cyberspace. *ACM Transactions on Information Systems (TOIS)* 26, 2 (2008), 7.

[3] Ahmed Abbasi, Stephen France, Zhu Zhang, and Hsinchun Chen. 2011. Selecting attributes for sentiment classification using feature relation networks. *IEEE Transactions on Knowledge and Data Engineering* 23, 3 (2011), 447–462.

[4] Ahmed Abbasi, Raymond Y. K. Lau, and Donald E. Brown. 2015. Predicting behavior. *IEEE Intelligent Systems* 30, 3 (2015), 35–43.

[5] Ahmed Abbasi, F. Mariam Zahedi, and Yan Chen. 2016. Phishing susceptibility: The good, the bad, and the ugly. In *2016 IEEE Conference on Intelligence and Security Informatics (ISI'16)*. IEEE, 169–174.

[6] Ahmed Abbasi, Yilu Zhou, Shasha Deng, and Pengzhu Zhang. 2018. Text analytics to support sense-making in social media: A language-action perspective. *MIS Quarterly* 42, 2 (2018), 427–464.

[7] Stefano Baccianella, Andrea Esuli, and Fabrizio Sebastiani. 2010. Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Vol. 10. 2200–2204.

[8] Nancy D. Berkman, Stacey L. Sheridan, Katrina E. Donahue, David J. Halpern, and Karen Crotty. 2011. Low health literacy and health outcomes: An updated systematic review. *Annals of Internal Medicine* 155, 2 (2011), 97–107.

[9] Leo Breiman. 2001. Random forests. *Machine Learning* 45, 1 (2001), 5–32.

[10] Donald E. Brown, Ahmed Abbasi, and Raymond Y. K. Lau. 2015. Predictive analytics: Predictive modeling at the micro level. *IEEE Intelligent Systems* 30, 3 (2015), 6–8.

[11] Erik Cambria, Bjorn Schuller, Yunqing Xia, and Catherine Havasi. 2013. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems* 28, 2 (2013), 15–21.

[12] Chao Che, Cao Xiao, Jian Liang, Bo Jin, Jiayu Zho, and Fei Wang. 2017. An RNN architecture with dynamic temporal matching for personalized predictions of Parkinson's disease. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 198–206.

[13] Kyunghyun Cho, Bart Van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. 2014. On the properties of neural machine translation: Encoder-decoder approaches. *Arxiv Preprint Arxiv:1409.1259* (2014).

[14] Philip J. Ciampa, Chandra Y. Osborn, Neeraja B. Peterson, and Russell L. Rothman. 2010. Patient numeracy, perceptions of provider communication, and colorectal cancer screening utilization. *Journal of Health Communication* 15, sup3 (2010), 157–168.

[15] Ronan Collobert and Jason Weston. 2008. A unified architecture for natural language processing: Deep neural networks with multitask learning. In *Proceedings of the 25th International Conference on Machine Learning*. ACM, 160–167.

[16] Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research* 12, (Aug. 2011), 2493–2537.

[17] Alexis Conneau, Holger Schwenk, Loïc Barrault, and Yann Lecun. 2016. Very deep convolutional networks for text classification. *Arxiv Preprint Arxiv:1606.01781* (2016).

[18] Dianne Cyr. 2008. Modeling web site design across cultures: Relationships to trust, satisfaction, and e-loyalty. *Journal of Management Information Systems* 24, 4 (2008), 47–72.

[19] Jan Deriu, Maurice Gonzenbach, Fatih Uzdilli, Aurelien Lucchi, Valeria De Luca, and Martin Jaggi. 2016. Swisscheese at semeval-2016 task 4: Sentiment classification using an ensemble of convolutional neural networks with distant supervision. In *Proceedings of the 10th International Workshop on Semantic Evaluation*. 1124–1128.

[20] Yuxiao Dong, Nitesh V. Chawla, Jie Tang, Yang Yang, and Yang Yang. 2017. User modeling on demographic attributes in big mobile social networks. *ACM Transactions on Information Systems* 34, 4 (2017), 35.
[21] Julia Dressel and Hany Farid. 2018. The accuracy, fairness, and limits of predicting recidivism. *Science Advances* 4, 1 (2018), eaao5580.
[22] Elizabeth Dugan, Felicia Trachtenberg, and Mark A. Hall. 2005. Development of abbreviated measures to assess patient trust in a physician, a health insurer, and the medical profession. *BMC Health Services Research* 5, 1 (2005), 64.
[23] Jeffrey L. Elman. 1990. Finding structure in time. *Cognitive Science* 14, 2 (1990), 179–211.
[24] Daniel Fernandes, John G. Lynch Jr., and Richard G. Netemeyer. 2014. Financial literacy, financial education, and downstream financial behaviors. *Management Science* 60, 8 (2014), 1861–1883.
[25] Tianjun Fu, Ahmed Abbasi, Daniel Zeng, and Hsinchun Chen. 2012. Sentimental spidering: Leveraging opinion information in focused crawlers. *ACM Transactions on Information Systems* 30, 4 (2012), 24.
[26] Yaroslav Ganin and Victor Lempitsky. 2014. Unsupervised domain adaptation by backpropagation. *Arxiv Preprint Arxiv:1409.7495* (2014).
[27] David Gefen and Kai Larsen. 2017. Controlling for lexical closeness in survey research: A demonstration on the technology acceptance model. *Journal of the Association for Information Systems* 18, 10 (2017), 727–757.
[28] Alexander Genkin, David D. Lewis, and David Madigan. 2007. Large-scale Bayesian logistic regression for text categorization. *Technometrics* 49, 3 (2007), 291–304.
[29] Lin Gong and Hongning Wang. 2018. When sentiment analysis meets social network: A holistic user behavior modeling in opinionated data. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*. 1455–1464.
[30] Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in Neural Information Processing Systems*. 2672–2680.
[31] Christophe Van Gysel, Maarten de Rijke, and Evangelos Kanoulas. 2018. Neural vector spaces for unsupervised information retrieval. *ACM Transactions on Information Systems* 36, 4, Article 38 (June 2018), 25 pages. DOI: https://doi.org/10.1145/3196826
[32] Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* 9, 8 (1997), 1735–1780.
[33] Minlie Huang, Qiao Qian, and Xiaoyan Zhu. 2017. Encoding syntactic knowledge in neural networks for sentiment classification. *ACM Transactions on Information Systems* 35, 3, Article 26 (June 2017), 27 pages. DOI: https://doi.org/10.1145/3052770
[34] Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016. Bag of tricks for efficient text classification. *Arxiv Preprint Arxiv:1607.01759* (2016).
[35] Yoon Kim. 2014. Convolutional neural networks for sentence classification. *Arxiv Preprint Arxiv:1408.5882* (2014).
[36] Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86, 11 (1998), 2278–2324.
[37] Chenliang Li, Yu Duan, Haoran Wang, Zhiqian Zhang, Aixin Sun, and Zongyang Ma. 2017. Enhancing topic modeling for short texts with auxiliary word embeddings. *ACM Transactions on Information Systems* 36, 2, Article 11 (Aug. 2017), 30 pages. DOI: https://doi.org/10.1145/3091108
[38] Jingjing Li, Kai Larsen, and Ahmed Abbasi. 2020. TheoryOn: A design framework and system for unlocking behavioral knowledge through ontology learning. *MIS Quarterly* (2020), 1–48.
[39] Jiwei Li, Minh-Thang Luong, Dan Jurafsky, and Eudard Hovy. 2015. When are tree structures necessary for deep learning of representations? *Arxiv Preprint Arxiv:1503.00185* (2015).
[40] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2016. Recurrent neural network for text classification with multi-task learning. *Arxiv Preprint Arxiv:1605.05101* (2016).
[41] Pengfei Liu, Xipeng Qiu, and Xuanjing Huang. 2017. Adversarial multi-task learning for text classification. *Arxiv Preprint Arxiv:1704.05742* (2017).
[42] Christopher Manning, Mihai Surdeanu, John Bauer, Jenny Finkel, Steven Bethard, and David McClosky. 2014. The Stanford corenlp natural language processing toolkit. In *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics: System Demonstrations*. 55–60.
[43] Andrew McCallum and Kamal Nigam. 1998. A comparison of event models for naive Bayes text classification. In *AAAI-98 Workshop on Learning for Text Categorization*, Vol. 752. Citeseer, 41–48.
[44] George A. Miller. 1995. WordNet: A lexical database for English. *Communications of the ACM* 38, 11 (1995), 39–41.
[45] Richard Netemeyer, David Dobolyi, Ahmed Abbasi, Gari Clifford, and Herman Taylor. 2019. Health literacy, health numeracy, and trust in doctor: Effects on key patient health outcomes. *Journal of Consumer Affairs* 53 (June 2019), 1–40.
[46] Richard H. Osborne, Roy W. Batterham, Gerald R. Elsworth, Melanie Hawkins, and Rachelle Buchbinder. 2013. The grounded psychometric development and initial validation of the Health Literacy Questionnaire (HLQ). *BMC Public Health* 13, 1 (2013), 658.

[47] Sinno Jialin Pan, Zhiqiang Toh, and Jian Su. 2013. Transfer joint embedding for cross-domain named entity recognition. *ACM Transactions on Information Systems* 31, 2, Article 7 (May 2013), 27 pages. DOI: https://doi.org/10.1145/2457465.2457467

[48] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up?: Sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing-Volume 10*. Association for Computational Linguistics, 79–86.

[49] Nanyun Peng and Mark Dredze. 2016. Multi-task multi-domain representation learning for sequence tagging. *CoRR, abs/1608.02689* (2016).

[50] Ellen Riloff, Siddharth Patwardhan, and Janyce Wiebe. 2006. Feature subsumption for opinion analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 440–448.

[51] Mickael Rouvier and Benoit Favre. 2016. SENSEI-LIF at SemEval-2016 Task 4: Polarity embedding fusion for robust sentiment analysis. In *Proceedings of the 10th International Workshop on Semantic Evaluation (SemEval'16)*. 202–208.

[52] John Rust and Susan Golombok. 2014. *Modern Psychometrics: The Science of Psychological Assessment*. Routledge.

[53] Marilyn M. Schapira, Cindy M. Walker, Tamara Miller, Kathlyn E. Fletcher, Pamela S. Ganschow, Elizabeth A. Jacobs, Diana Imbert, Maria O'Connell, and Joan M. Neuner. 2014. Development and validation of the numeracy understanding in medicine instrument short form. *Journal of Health Communication* 19, sup2 (2014), 240–253.

[54] Charles Donald Spielberger. 1989. *State-trait Anxiety Inventory: A Comprehensive Bibliography*. Consulting Psychologists Press.

[55] Carlo Strapparava and Alessandro Valitutti. 2004. Wordnet affect: An affective extension of Wordnet. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'04)*, Vol. 4. Citeseer, 1083–1086.

[56] Peilu Wang, Yao Qian, Frank K. Soong, Lei He, and Hai Zhao. 2015. A unified tagging solution: Bidirectional LSTM recurrent neural network with word embedding. *Arxiv Preprint Arxiv:1511.00215* (2015).

[57] Lei Xu, Chunxiao Jiang, Yong Ren, and Hsiao-Hwa Chen. 2016. Microblog dimensionality reduction—A deep learning approach. *IEEE Transactions on Knowledge and Data Engineering* 28, 7 (2016), 1779–1789.

[58] Zhilin Yang, Ruslan Salakhutdinov, and William Cohen. 2016. Multi-task cross-lingual sequence tagging from scratch. *Arxiv Preprint Arxiv:1603.06270* (2016).

[59] Zheng Yu, Haixun Wang, Xuemin Lin, and Min Wang. 2016. Understanding short texts through semantic enrichment and hashing. *IEEE Transactions on Knowledge and Data Engineering* 28, 2 (2016), 566–579.

[60] Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems*. 649–657.

[61] Yulei Zhang, Yan Dang, and Hsinchun Chen. 2011. Gender classification for web forums. *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans* 41, 4 (2011), 668–677.

[62] Chunting Zhou, Chonglin Sun, Zhiyuan Liu, and Francis Lau. 2015. A C-LSTM neural network for text classification. *Arxiv Preprint Arxiv:1511.08630* (2015).

[63] Jiayu Zhou, Jianhui Chen, and Jieping Ye. 2011. Clustered multi-task learning via alternating structure optimization. In *Advances in Neural Information Processing Systems*. 702–710.

[64] Jiayu Zhou, Jianhui Chen, and Jieping Ye. 2011. Malsar: Multi-task learning via structural regularization. Arizona State University, 21.

[65] David Zimbra, Ahmed Abbasi, Daniel Zeng, and Hsinchun Chen. 2018. The state-of-the-art in twitter sentiment analysis: A review and benchmark evaluation. *ACM Transactions on Management Information Systems (TMIS)* 9, 2 (2018), 5.

Received April 2019; revised August 2019; accepted September 2019