Dynamic, Multi-dimensional, and Skillset-specific Reputation Systems for Online Work (Online Appendix)

Marios Kokkodis
Carroll School of Management, Boston College, Chestnut Hill, MA 02467,
kokkodis@bc.edu

Reputation systems in digital workplaces increase transaction efficiency by building trust and reducing information asymmetry. These systems, however, do not yet capture the dynamic multidimensional nature of online work. By uniformly aggregating reputation scores across worker skills, they ignore skillset-specific heterogeneity (reputation attribution), and they implicitly assume that a worker's quality does not change over time (reputation staticity). Even further, reputation scores tend to be overly positive (reputation inflation), and as a result, they often fail to differentiate workers efficiently.

This work presents a new augmented intelligence reputation framework that combines human input with machine learning to provide dynamic, multi-dimensional, and skillset-specific worker reputation. The framework includes three components: The first component maps skillsets into a latent space of finite competency dimensions (word embedding), and as a result, it directly addresses reputation attribution. The second builds dynamic competency-specific quality assessment models (hidden Markov models) that solve reputation staticity. The final component aggregates these competency-specific assessments to generate skillset-specific reputation scores. Application of this framework on a dataset of 58,459 completed tasks from a major online labor market shows that, compared with alternative reputation systems, the proposed approach (1) yields more appropriate rankings of workers that form a closer-to-normal reputation distribution, (2) better identifies "non-perfect" workers who are more likely to underperform and are harder to predict, and (3) improves the ranking of within-opening choices and yields significantly better outcomes. Additional analysis of 77,044 restaurant reviews shows that the proposed framework successfully generalizes to alternative contexts, where assigned feedback scores are overly positive and service quality is multidimensional and dynamic.

Key words: Reputation frameworks, Reputation inflation, Reputation attribution, Reputation staticity, Online labor markets, Hidden Markov models, Word embedding

A. Transition functions

To model transitions the framework considers the following three functions:

$$g^d \in \{\text{ Multinomial Logit, Ordered logit, Constrained ordered logit }\}\ \forall d \in \mathcal{D}\,. \tag{12}$$

Application of the multinomial logit is straightforward. The transition probability from state s_k to s_l is given by the following:

$$Pr(s_l|s_k,\mathbf{\Gamma},\mathbf{Z}_{t-1}) = \frac{\exp(\boldsymbol{\gamma}_{kl}\mathbf{Z}_{t-1})}{\sum_{m\in K}\exp(\boldsymbol{\gamma}_{km}\mathbf{Z}_{t-1})} = \text{softmax}(\boldsymbol{\gamma}_{kl}\mathbf{Z}_{t-1})\,, \tag{13}$$

where I dropped the superscript d for simplicity. This model does not assume any order, so for $k>l$, state s_k might model higher or lower quality from state s_l.

To the contrary, the ordered logit formulation assumes that states are ordered in terms of increasing quality, such that state s_k has a larger constant term than s_l when $k>l$ (Ghose and Todri 2016, Ghose et al. 2017, Todri et al. 2020, Zucchini et al. 2017). The transition probability from state s_k to a state s_l is as follows:

$$Pr(s_l|s_k,\mathbf{\Gamma},\mathbf{Z}_{t-1}) = \begin{cases} \Pr(\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}+\varepsilon \le \alpha^1) = \Lambda(\alpha^1-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}), & \text{if } l=1 \\ \Pr(\alpha^{l-1} < \boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}+\varepsilon \le \alpha^l) = \Lambda(\alpha^l-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}) - \Lambda(\alpha^{l-1}-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}), & \text{if } 1<l<K \\ \Pr(\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}+\varepsilon > \alpha^{K-1}) = 1-\Lambda(\alpha^{K-1}-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}) & \text{if } l=K \end{cases}. \tag{14}$$

In the previous equation, ε is the unobserved error term that I assume to follow a logistic distribution (i.e., $\varepsilon_i|\boldsymbol{Z}_{t-1} \sim Logistic(0,1)$), Λ is the logit function, and $\boldsymbol{\alpha} = [\alpha^1,\alpha^2,...,\alpha^{K-1}]'$ are state-specific thresholds (Wooldridge 2010).

The constrained ordered logit adjusts Equation 14 to not allow transitions to lower-quality states. Specifically:

$$Pr(s_l|s_k,\mathbf{\Gamma},\mathbf{Z}_{t-1}) = \begin{cases} 0, & \text{if } l<k \\ \Pr(\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}+\varepsilon \le \alpha^1) = \Lambda(\alpha^1-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}), & \text{if } l=k=1 \\ \Pr(\alpha^{l-1} < \boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}+\varepsilon \le \alpha^l) = \Lambda(\alpha^l-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}) - \Lambda(\alpha^{l-1}-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}), & \text{if } 1<k<l<K \\ \Pr(\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}+\varepsilon > \alpha^{K-1}) = 1-\Lambda(\alpha^{K-1}-\boldsymbol{\gamma}_k\boldsymbol{Z}_{t-1}) & \text{if } l=K \end{cases}. \tag{15}$$

B. HMM identification

Given the structure of the HMM I focus on estimating the parameter vectors $\boldsymbol{\Theta}^d, \boldsymbol{\Gamma}^d$. To do so, I maximize the conditional probability of the set of observations given the HMM. (For simplicity, in the following analysis I drop the superscript d. However, keep in mind that this estimation process happens independently for each dimension $d \in \mathcal{D}$.)

Let us assume that I have the following sequence of M observations for a given worker i:

$$\boldsymbol{Y}_i = Y_{i1}, Y_{i2}, \ldots, Y_{iM} \,. \tag{16}$$

These observations correspond to a sequence of input vectors:

$$\boldsymbol{X}_{1:M} = \boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_M \,. \tag{17}$$

Furthermore, let us assume that $\boldsymbol{Y}_i$ is the result of a sequence of latent states, $\boldsymbol{S}_i$:

$$\boldsymbol{S}_i = S_{i1}, S_{i2}, \ldots, S_{iM} \,, \tag{18}$$

where $S_{im} \in \mathcal{S}$. This sequence of states is affected by the sequence of historic vectors $\boldsymbol{Z}_{1:M-1}$:

$$\boldsymbol{Z}_{1:M-1} = \boldsymbol{Z}_1, \boldsymbol{Z}_2, \ldots, \boldsymbol{Z}_{M-1} \,. \tag{19}$$

Figure 10 shows these sequences along with their interactions. Based on the structure of the graph, I get the conditional likelihood of observing sequence $\boldsymbol{Y}_i$:

$$\Pr(\boldsymbol{Y}_i | \boldsymbol{S}_i; \boldsymbol{\Theta}, \boldsymbol{X}_{1:M}) = \prod_{t=1}^{M} \Pr(Y_{it} | S_{it}; \boldsymbol{\Theta}, \boldsymbol{X}_t) \,, \tag{20}$$

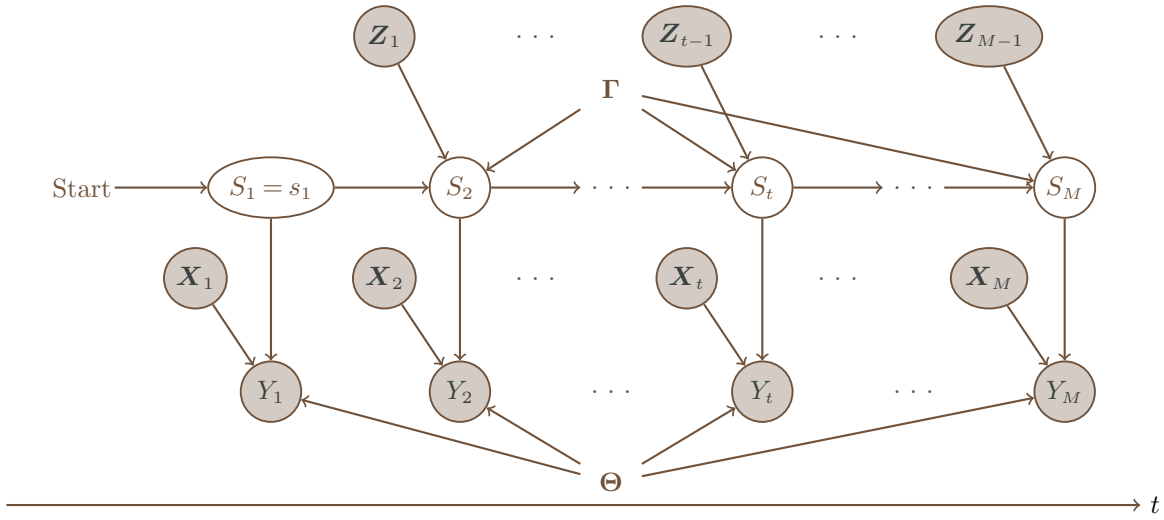
where Equation 6 estimates the right hand side. From Figure 10, the conditional probability of observing the sequence $\boldsymbol{S}_i$ is:

$$\Pr(\boldsymbol{S}_i | \boldsymbol{\Gamma}, \boldsymbol{Z}_{1:M-1}) = \boldsymbol{\pi}(S_1) \prod_{t=2}^{M} \Pr(S_{it} | S_{it-1}; \boldsymbol{\Gamma}, \boldsymbol{Z}_{t-1}) \,, \tag{21}$$

where $\boldsymbol{\pi}(S_1)$ is the the prior probability of being at state S_1. Equation 4 estimates these transition probabilities. Since the structure of the HMM imposes that every new worker lands in state s_1 ($\boldsymbol{\pi}(S_1 = s_1) = 1$), the previous equation becomes:

$$\Pr(\boldsymbol{S}_i | \boldsymbol{\Gamma}, \boldsymbol{Z}_{1:M-1}) = \prod_{t=2}^{M} \Pr(S_{it} | S_{it-1}; \boldsymbol{\Gamma}, \boldsymbol{Z}_{t-1}) \,. \tag{22}$$

Figure 10 Temporal evolution of the HMM



The structure of the latent state sequence $\boldsymbol{S}_i$, the observed sequence of outcomes $\boldsymbol{Y}_i$, the parameter vectors $\boldsymbol{\Theta}, \boldsymbol{\Gamma}$, and the sequences of input vectors $\boldsymbol{X}_{1:M}, \boldsymbol{Z}_{1:M-1}$ for a given worker i. For better readability I have dropped the worker subscript i and the competency superscript d. As with traditional probabilistic graphical models, latent states are in clear ellipses, and observed features in shaded ones (Koller and Friedman 2009).

Based on this analysis and the graph in Figure 10, the likelihood of this sequence of observations for worker i is as follows:

$$
\begin{aligned}
l(\boldsymbol{Y}_i; \boldsymbol{\Theta}, \boldsymbol{\Gamma}) &= \Pr(\boldsymbol{Y}_i | \boldsymbol{\Theta}, \boldsymbol{\Gamma}, \boldsymbol{X}_{1:M}, \boldsymbol{Z}_{1:M-1}) \\
&= \sum_{\forall \boldsymbol{S}_i} \Pr(\boldsymbol{Y}_i, \boldsymbol{S}_i | \boldsymbol{\Theta}, \boldsymbol{\Gamma}, \boldsymbol{X}_{1:M}, \boldsymbol{Z}_{1:M-1}) \\
&\overset{\text{Figure 10}}{=} \sum_{\forall \boldsymbol{S}_i} \Pr(\boldsymbol{Y}_i | \boldsymbol{S}_i; \boldsymbol{\Theta}, \boldsymbol{X}_{1:M}) \Pr(\boldsymbol{S}_i | \boldsymbol{\Gamma}, \boldsymbol{Z}_{1:M-1}) \\
&= Pr(Y_{i1} | S_{i1}; \boldsymbol{\Theta}, \boldsymbol{X}_1) \\
&\times \sum_{\forall \boldsymbol{S}_i} \prod_{t=2}^{M} \Pr(Y_{it} | S_{it}; \boldsymbol{\Theta}, \boldsymbol{X}_t) \\
&\times \Pr(S_{it} | S_{it-1}; \boldsymbol{\Gamma}, \boldsymbol{Z}_{t-1}),
\end{aligned} \tag{23}
$$

where I used the structure of the HMM to decompose the joint probability of $\Pr(\boldsymbol{Y}_i, \boldsymbol{S}_i | \boldsymbol{\Theta}, \boldsymbol{\Gamma}, \boldsymbol{X}_{1:M}, \boldsymbol{Z}_{1:M-1})$. Then, the complete likelihood for a dataset with N workers is:

$$
L(\boldsymbol{\Theta}, \boldsymbol{\Gamma}) = \prod_{i=1}^{N} l(\boldsymbol{Y_i}; \boldsymbol{\Theta}, \boldsymbol{\Gamma}) \,. \tag{24}
$$

Maximization of this likelihood estimates the parameters $\mathbf{\Theta}, \mathbf{\Gamma}$. I do this numerically through the limited memory Broyden-Fletcher-Goldfarb-Shanno (BFGS) algorithm (Byrd et al. 1995). (In practice I minimize the negative log-likelihood.)

For more conservative reputation predictions, an option is to maximize this likelihood and then smooth the framework's estimate by adding the observed accumulated reputation:

$$P_{it}(\mathcal{R}) = \frac{1}{2}(P_{it}(\mathcal{R}) + \text{Current reputation}_{t-1}) \tag{25}$$

Finally, estimation of Equation 24 is highly parallelizable, as each individual likelihood l can run independently at every iteration.

C. Comparison of alternative design choices

The three components of the framework require hyper-parameter tuning. At the same time, the justification of each design choice presented in Section 3 requires comparison with alternative approaches. In this Appendix, I discuss such alternative modeling choices for each component of the proposed framework along with the process I follow to tune all the parameters of the focal approach. For the rest of the analysis, I split the data into ten folds that consist of different workers (i.e., each worker's complete history appears only in one of the ten folds), and I use 10-fold cross-validation to estimate the performance of each alternative design approach.

C.1. Alternative modeling choices for component A

Component A of the `HMM-W2V-framework` is required in order to map skillsets into a vector space of real numbers. To achieve this, in Section 3.1, I used a W2V approach. Alternative approaches can also achieve this mapping. For instance, document embedding (D2V, Le and Mikolov 2014) can directly map each skillset into numeric vectors. Even further, simpler clustering approaches can also perform this task. For instance, a Gaussian mixture model (GMM; see Murphy 2012) can provide membership weights for each skill to any number of predefined clusters. Finally, an alternative approach could also consider the raw text from job descriptions. To test this, I implement the

proposed W2V approach (Section 3.1) on the job-description text that includes the task's required skills (W2V-D). [5]

To compare these four alternative modeling choices for component A, I follow a grid search approach. Specifically, I estimate the 10-fold cross-validated ranking correlations (Spearman ρ) of each approach for the following parameter combinations:

$$\text{Component A grid-search: } \left\{ \overbrace{\{\text{W2V, D2V, GMM, W2V-D}\}}^{\texttt{Component A}} \times \overbrace{\{5, 10, 15\}}^{|\mathcal{D}|} \right\}. \tag{26}$$

Figure 11A shows the results. The $y-$axis shows improvement over the proposed W2V approach described in Section 3.1. The results show that W2V performs significantly ($p < 0.05$) better than the alternative approaches for the focal dataset. As a result, for the main analysis of this work, I use W2V. (For other contexts, one of the alternative approaches could be more appropriate. For instance, for the restaurant review dataset described in Appendix E, D2V worked better.)

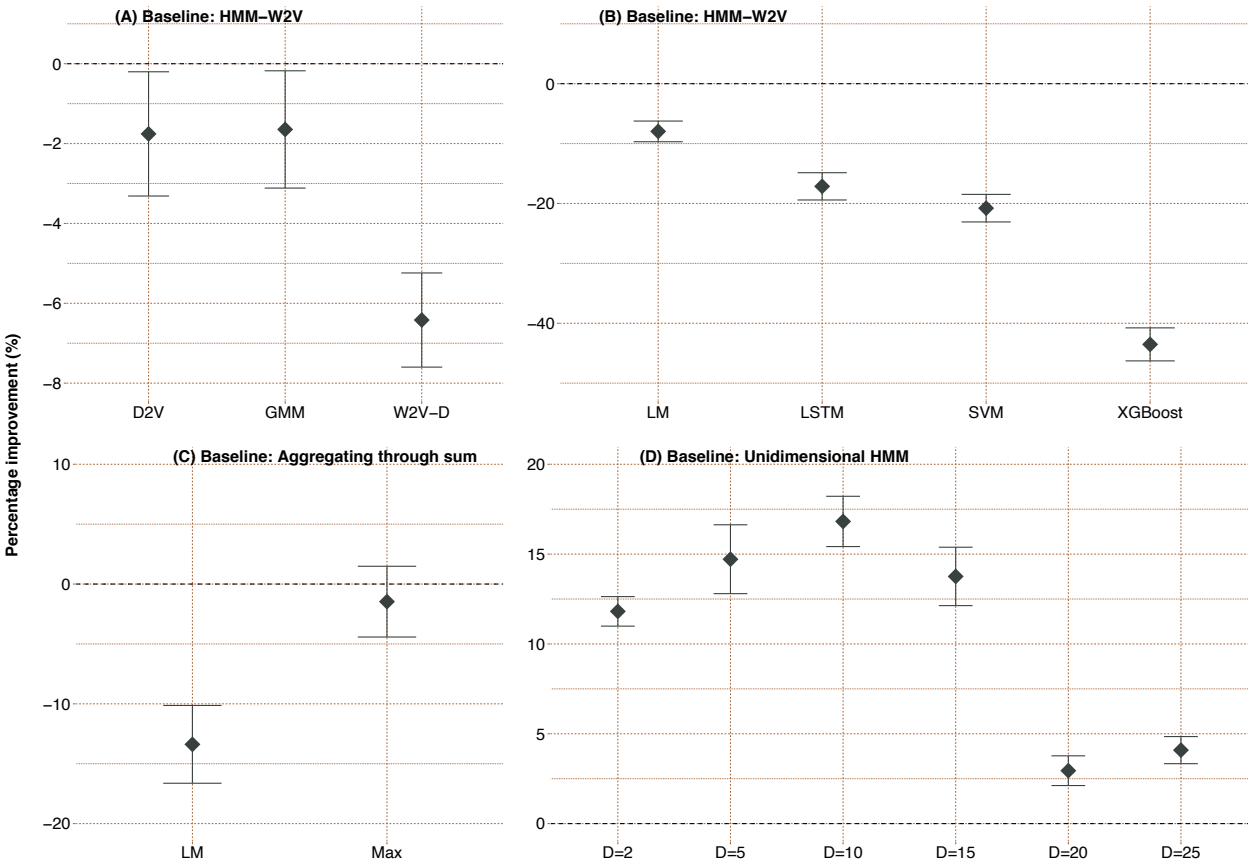
Figure 12 shows how W2V maps a randomly selected subset of skills into a reduced (through stochastic neighbor embedding, Hinton and Roweis 2003) two-dimensional space. The plot reveals hidden contextual similarities between skills. For instance, it shows that employers who request `C++` usually also request `SQLite`, while employers who request `Python` usually also request `MongoDB`. The plot hence suggests that `Python` is contextually closer to `MongoDB` than `SQLite`.

C.2. Alternative modeling choices for component B

For implementing component B of the `HMM-W2V-framework` that addresses reputation staticity, the main analysis uses an HMM. Other modeling approaches can also capture such a dynamic evolution of sequential observations. For instance, recurrent neural networks can capture such evaluations through Long Short Term Memory networks (Hochreiter and Schmidhuber 1997). In addition, I benchmark such dynamic approaches with simple (linear regression models—LM) and powerful regression approaches (SVM-regression and gradient boosting—XGBoost).

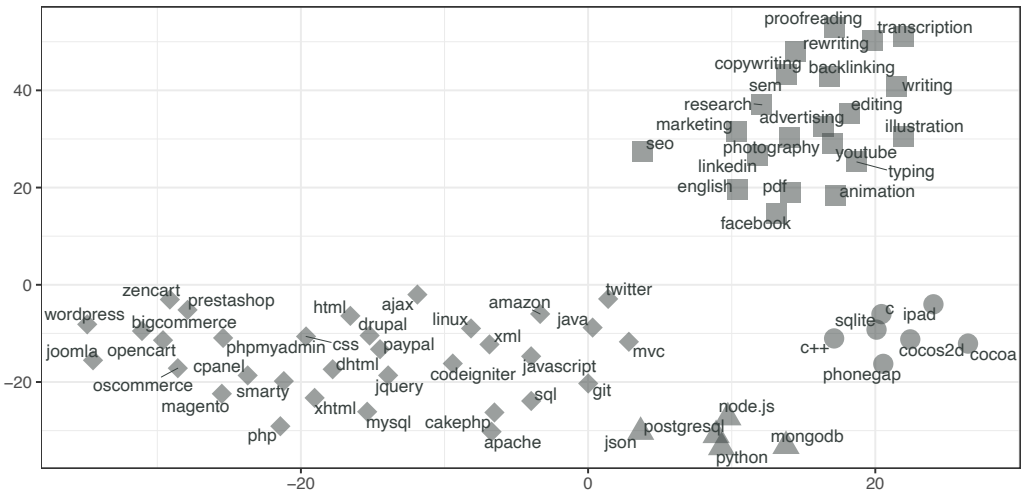
[5] I train these approaches on a separate dataset of 40,000 job-opening skillsets. I then use the pre-trained W2V, D2V and GMM models to decompose the skillsets of the focal dataset.

Figure 11 Design choices and tuning of the `HMM-W2V-framework`



The four figures compare alliterative modeling choices for each component of the `HMM-W2V-framework`. The $y-$axis shows a 10-fold cross-validated percentage improvement. W2V: Word2Vector. D2V: Doc2Vector. GMM: Gaussian mixture model. W2V-D: Word2Vector on job description text. Figure A shows that using W2V for component A of the `HMM-W2V-framework` outperforms ($p < 0.05$) D2V, GMM and W2V-D. Figure B shows that compared with alternative modeling approaches, using an HMM for component B of the `HMM-W2V-framework` better captures the dynamic nature of workers ($p < 0.001$). Figure C shows that a linear aggregation (for component C of the `HMM-W2V-framework`) performs worse than summing all dimensions, which performs on par with projecting the reputation of the dimension with the maximum weight. Figure D shows that $|\mathcal{D}| = 10$ dimensions better describe the focal dataset. In addition, it shows that configurations that include any number of dimensions (i.e., including component A of the `HMM-W2V-framework`) outperform a unidimensional HMM. Error bars represent 95% confidence intervals.

Figure 12 Skill decomposition through W2V



Visualization of mapped skills through W2V in a reduced, two-dimensional space through stochastic neighbor embedding (Hinton and Roweis 2003). Contextually similar skills appear close to each other (e.g., `C++, C, SQLite`), while contextually dissimilar skills appear far away from each other (e.g., `MongoDB` vs. `proofreading`).

To evaluate the performance of each alternative modeling of component B, I follow a grid search approach, and I estimate the 10-fold cross-validated ranking correlations (Spearman ρ) of each approach for the following parameter combinations:

$$\text{Component B grid-search: } \left\{ \overbrace{\{\text{HMM, LM, LSTM, SVM, XGBoost}\}}^{\texttt{Component B}} \times \overbrace{\{5, 10, 15\}}^{|\mathcal{D}|} \right\}. \quad (27)$$

Figure 11B shows the results. The $y-$ axis shows the percentage improvement of the HMM approach over the alternative approaches of modeling component B: It becomes clear that modeling dynamic transitions through a hidden Markov model performs significantly ($p < 0.001$) better than all alternative approaches for the focal dataset. One reason that could explain the underperformance of the LSTM approach, in particular, is the fact that the analyzed sequences are short (median completed tasks per worker is 5, Table 3). Perhaps, in larger datasets with longer sequences, LSTM could have been a more appropriate choice.

C.3. Alternative modeling choices for component C

Section 3.3 aggregates dimension-specific scores through a simple summation equation. Alternatively, an aggregation function could linearly regress each dimension (LM). Such a function could identify

dimensions that consistently provide erroneous estimates and underweight them. Another aggregation function could project the reputation score of the dimension with the maximum weight (Max). In theory, such an approach would rely on the dimension that is more relevant to the skillset at hand to make a reputation estimate.

To evaluate the performance of each alternative modeling choice of component C I follow a similar grid search approach to before, and estimate the 10-fold cross-validated ranking correlations (Spearman ρ) of each approach for the following parameter combinations:

$$\text{Component C grid-search: } \left\{ \overbrace{\{\text{Sum, LM, Max}\}}^{\texttt{Component C}} \times \overbrace{\{5, 10, 15\}}^{|\mathcal{D}|} \right\}. \tag{28}$$

Figure 11C shows the results. The $y-$axis shows the improvement of each approach compared with summing each dimension according to Equation 8. Linear modeling performs significantly ($p < 0.001$) worse compared with summing all dimensions. On the other hand, projecting the dimension with the maximum weight performs on average insignificantly ($p > 0.05$) worse than Equation 8. For the focal analysis, I choose Equation 8.

C.4. Choosing number of competency dimensions

The proposed framework requires as input the number of competency dimensions $|\mathcal{D}|$. To identify the best parameter $|\mathcal{D}|$, I use the best design choices for each component described above, and I estimate the 10-fold cross-validated ranking correlations for the following:

$$|\mathcal{D}| \text{ search: } \{1,2,5,10,15,20,25\}. \tag{29}$$

Figure 11D shows the results. The $y-$axis shows the improvement over a unidimensional HMM approach with $|\mathcal{D}| = 1$. For the uni-dimensional model, I remove component A, implicitly ignoring reputation attribution. For all $|\mathcal{D}| > 1$ the ranking correlation (Spearman ρ) is significantly ($p < 0.001$) higher than the unidimensional model. This suggests that component A itself provides a significant contribution to the performance of the `HMM-W2V-framework`. Furthermore, $|\mathcal{D}| = 10$ significantly outperforms ($p < 0.05$) $|\mathcal{D}| \in \{2, 15, 20, 25\}$. The performance of $|\mathcal{D}| = 5$ is, on average lower than $|\mathcal{D}| = 10$, but not statistically significant ($p > 0.05$). As a result, for the main analysis I use $|\mathcal{D}| = 10$.

Summary of comparisons: Overall, and based on this analysis, it becomes clear that each component of the framework contributes to the observed performance. Specifically, compared with alternative design choices, (1) using W2V for component A increases the framework's performance by up to 6%, (2) using an HMM in component B increases the framework's performance by up to 43% (Figure 11B), and (3) using Equation 8 to aggregate dimension-specific reputation estimates increases the performance of the framework by up to 13% (Figure 11C). Finally, including component A and using $|\mathcal{D}| = 10$ increases the performance of the framework by up to$\sim 17\%$ (Figure 11D).

C.5. HMM tuning

Each HMM considers various options for the transition function g^d, emission function f^d, and number of states K^d. As I mentioned earlier in Appendix A, the following continuous probability distributions can model function g^d:

$$g^d \in \{ \text{ Multinomial Logit, Ordered logit, Constrained ordered logit } \} \, \forall d \in \mathcal{D} \, . \tag{30}$$

Similarly, for the emission function f^d, the HMM considers the following set of probability distributions:

$$f^d \in \{ \text{ Beta, Truncated normal, Truncated exponential} \} \, \forall d \in \mathcal{D} \, . \tag{31}$$

Since emissions are bounded in [0,1], I truncate the normal and the exponential distribution. (The support of the Beta distribution is by default $\in [0, 1]$). Finally, I consider the following number of states:

$$K^d \in \{2, 3, ..., 6\}, \forall d \in \mathcal{D} \, . \tag{32}$$

To choose the most appropriate combination for the focal dataset, I estimate the configurations that yield the lowest 10-fold cross-validated Bayesian information criterion (BIC) scores (Schwarz 1978, Murphy 2012, Koller and Friedman 2009, Bishop 2006). Because the optimization process depends on the initialization of $\mathbf{\Theta}^d, \mathbf{\Gamma}^d$, it is prone to stuck in local maxima. To increase the

likelihood of reaching a potential global maximum, I conduct a search of 100 random initializations for each combination. Specifically, I perform the following grid-search:

$$K^d, f^d, g^d \text{ grid-search: } \Bigg\{ \overbrace{\{2,3,4,5,6\}}^{K^d} \times \overbrace{\{\text{Multinomial, Ordered logit, Constrained ordered logit}\}}^{g^d} \times \overbrace{\{\text{Beta, Truncated normal, Truncated exponential}\}}^{f^d} \times 100 \Bigg\}. \tag{33}$$

Based on the results of these searches, I pick:

- ◇ Number of states for each dimension, $K = [3,4,4,4,4,4,3,4,3,3]$.
- ◇ $f^d =$ Beta $\forall d \in \mathcal{D}$.
- ◇ $g^d =$ Multinomial logit $\forall d \in \mathcal{D}$.

D. Additional comparisons

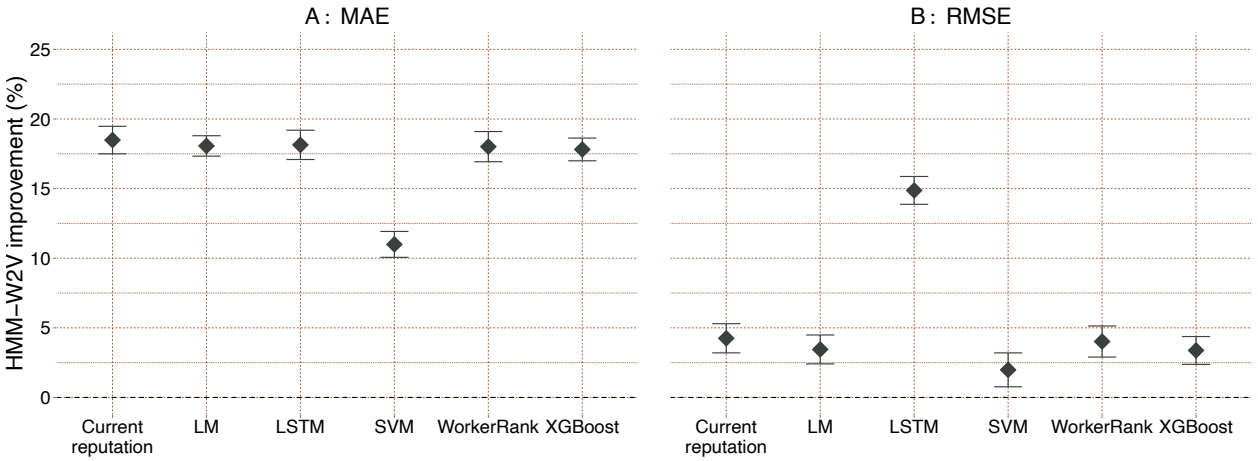
Section 5 illustrates the superiority of the `HMM-W2V-framework` in terms of ranking workers according to their likelihood of performing well, presenting accurate reputation distributions, and identifying "non-perfect" workers. Indeed, these evaluation metrics are the most relevant to benchmark a reputation system. Nevertheless, in this section I examine how the resulting HMM-W2V reputation compares to alternative reputation systems in terms of predictive performance and explanatory power.

Predictive performance: To test the predictive performance of the proposed approach, I estimate the 10-fold cross-validated mean absolute error (MAE) and the root mean squared error (RMSE) of each of the alternative reputation systems through the following linear model:

$$Y_t \sim \beta_0 + \beta Rep_t^j + \varepsilon, \tag{34}$$

where $Rep^j \in$ {HMM-W2V, Current feedback, LM, LSTM, SVM, WorkerRank, XGBoost}. Figures 13A and B show the improvement of `HMM-W2V-framework` in terms of MAE and RMSE respectively. HMM-W2V reputation is significantly ($p < 0.01$) more informative in predicting outcomes than alternative reputation approaches.

Figure 13 **Predictive performance of alternative reputation systems**



The $y-$axis shows the percentage improvement of the `HMM-W2V-framework` over the $x-$axis reputation systems in terms of MAE and RMSE. Error bars represent 95% confidence intervals.

Regression analysis: Regression analysis of multiple specifications can reveal the linear relationship of each reputation system with the observed outcomes. Table 5 shows the results of the regression analysis (Equation 34). The comparison shows that HMM-W2V reputation has higher explanatory power than the two baselines, as it yields greater R^2 than all alternative reputation systems.

Table 5 Explanatory power of alternative reputation systems

	DV: Observed performance						
	(A1)	(A2)	(A3)	(A4)	(A5)	(A6)	(A7)
Current reputation	0.025***						
	(0.00)						
LSTM		0.027***					
		(0.00)					
LM			0.039***				
			(0.00)				
SVM				0.009***			
				(0.00)			
WorkerRank					0.030***		
					(0.00)		
XGBoost						0.042***	
						(0.00)	
HMM-W2V							0.044***
							(0.00)
R^2	0.010	0.012	0.026	0.001	0.015	0.029	0.032

$^{***}p-$value < 0.001.

E. Generalizability: Application in online reputation platforms

The proposed approach, in theory, generalizes to other contexts that experience reputation inflation, reputation staticity, and reputation attribution. One such context is online reputation platforms (e.g., Yelp, TripAdvisor). These platforms experience reputation inflation (Luca 2016, Hu et al. 2017). At the same time, like contractors, venues (restaurants, hotels) in these platforms evolve. For instance, one venue might do a renovation, change menu offerings, hire a new chef, or change management. Hence, reputation staticity might also be present. Finally, because venues (like workers) are multidimensional entities, unidimensional ratings suggest the existence of reputation attribution.

It is important to highlight that some unique characteristics of online labor markets do not transfer to this alternative context. Specifically, in online labor markets, we are interested in estimating a skillset-specific reputation (Figure 1). The framework achieves this by observing skillset-specific evaluations for each worker. On the contrary, in online reputation platforms, there is no observed equivalent of skillsets. Reviewers rate their overall experience, and sometimes, they explain what they like and what they did not like in the review text. The proposed framework decomposes this text to latent dimensions, but the outcome of the framework is not "skillset-equivalent"-specific anymore. Nevertheless, the framework can be adjusted to this alternative context to provide current restaurant reputation scores. Table 6 summarizes the differences in objectives and data availability between the main and this alternative context.

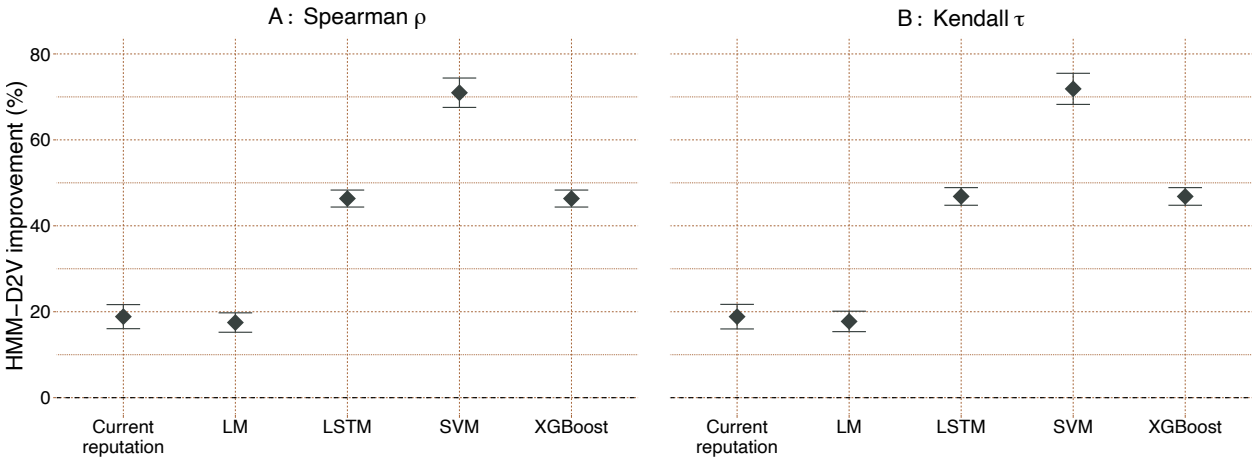
Table 6 Differences between the worker-reputation and the restaurant-reputation contexts

Context	Framework's objective	How it works
Online labor markets	Provide *current, skillset-specific* worker reputation	Decomposes observed skillset evaluations to latent dimensions. Builds dimension-specific HMMs.
Online reputation platforms	Provide *current* restaurant reputation	Decomposes review text into latent dimensions. Builds dimension-specific HMMs.

E.1. Comparison with alternative reputation systems

To test the proposed approach and alternative reputation systems in this different context, I analyze a set of 77,044 online reviews from a major restaurant review platform. As mentioned earlier in

Figure 14 **Ranking performance of alternative reputation systems on restaurant reputation**



The $y-$axis shows the percentage improvement of the `HMM-D2V` framework over the $x-$axis reputation system, in terms of Spearman ρ and Kendall τ. The `HMM-D2V` framework significantly ($p < 0.001$) outperforms all alternative reputation systems, with average 10-fold cross-validated improvements ranging between 20% and 70%. Error bars represent 95% confidence intervals.

Appendix C.1, D2V on review text was a better solution for implementing component A.[6] Hence, I use D2V, and allow latent restaurant dimensions of a venue to evolve as venues receive new ratings. Furthermore, I form vector $\mathbf{Z}_{t-1}$ by including the current reputation of each restaurant, the total number of reviews, and the days since its first review. Similarly, vector $\mathbf{X}_t$ includes the restaurant's current reputation.

Figure 14 shows the results. The $y-$axis shows the improvement of the proposed approach over alternative reputation systems, in terms of 10-fold cross-validated ranking correlations. The HMM-D2V framework significantly ($p < 0.001$) outperforms all alternative reputation systems, providing more accurate and current rankings of the listed venues.

[6] D2V likely works better in this context because review text is unstructured and unique. Conceptually, summing up weights of seemingly random words found in reviews through Equation 1 will generate noisy representations. On the contrary, summing up mappings of structured skill terms through W2V in the online labor market context should generate more informative representations, as each individual single skill contains crucial information.

E.2. Comparison with recommender systems

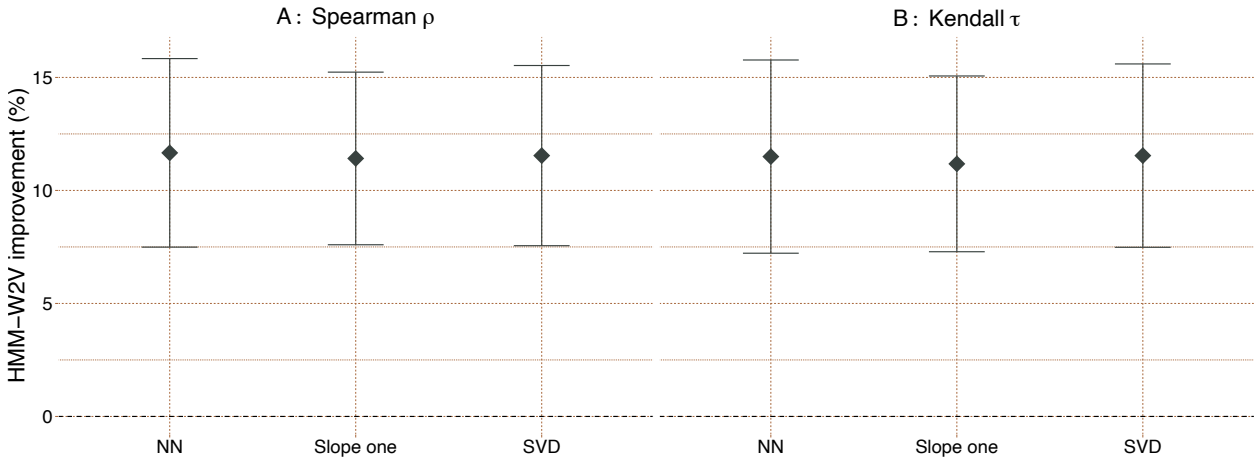
The restaurant-reputation context is closer to recommender systems than the focal worker-reputation context. Specifically, in this context, a mapping for recommender systems that could provide restaurant reputation is:

- Reviewer $\mapsto$ user.
- Restaurant rating $\mapsto$ rating.
- Restaurant $\mapsto$ item.

Compared with the main context, because in this context there is no objective restaurant equivalent of "skillset-specific" reputation (Table 6), restaurants map directly to items. In addition, reviewers rarely rate the same restaurant twice; hence, compared with online labor markets where workers get rated for the same skills repeatedly and I had to average their skillset-specific ratings (Table 2), restaurant ratings map directly to the recommender systems ratings. Finally, in online reputation platforms, it is reasonable to assume that reviewers are the recommender system's users, as every user's objective is to dine at a restaurant; instead, in online labor markets, it is not as reasonable to recommend workers to employers without considering the type of jobs and required skills that employers are looking for.

Given this more straightforward fit of recommender systems, I expect them to perform better compared with their performance in predicting worker-reputation scores (Section 5.4). Figure 15 shows the results. Indeed, explicit-feedback recommender systems perform better in this context than in online labor markets (Figure 8). Yet, and similar to online labor markets, the `HMM-W2V-framework` significantly ($p < 0.001$) outperforms these approaches, as it better captures restaurant evolution through the multi-dimensional HMM modeling (standard collaborative filtering approaches do not model item (restaurant) evolution). Finally, note that the proposed framework outperforms next-item recommenders (CNN) on average by 400% ($p < 0.001$), as the necessary encoding (Section 5.4.1) introduces significant noise in the prediction of restaurant reputation. I omit this comparison from Figure 8 for better presentation clarity.

Figure 15 Comparison with recommender systems



The `HMM-W2V-framework` significantly outperforms adaptations of recommender systems in this alternative context. As a result, it provides more accurate restaurant-reputation scores. The improvement of the `HMM-W2V-framework` over the deep learning recommender framework (CNN) is over 400%. I omit this point for presentation clarity. Error bars represent 95% confidence intervals.

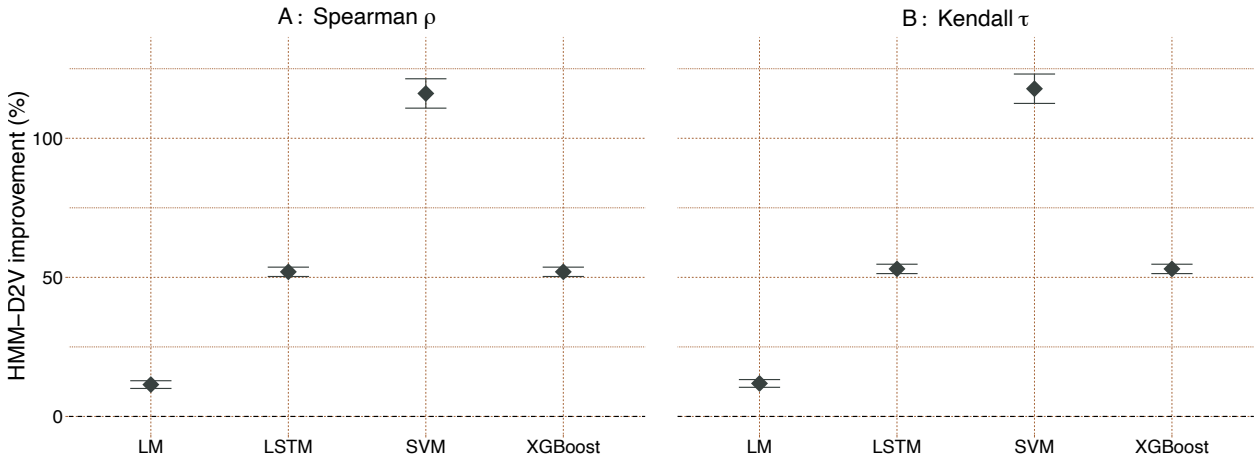
E.3. Comparison with human-rated dimensions

One concern of the proposed approach is that platforms are already developing multidimensional reputation systems, and as a result, the `HMM-W2V-framework` might be only recovering noisy information from such humanly rated dimensions. The focal restaurant review platform is an ideal context to test this, as it allows customers to rate venues across four dimensions: "Food," "Atmosphere," "Value," and "Service." Figure 16 compares the `HMM-W2V-framework` with alternative approaches that use these human-inputted multidimensional reputation scores as input. Specifically, these approaches estimate the following specification:

$$Y_t \sim G(\boldsymbol{Z}_{t-1}, \boldsymbol{X}_t, \bar{\text{Food}}_{t-1}, \bar{\text{Atmosphere}}_{t-1}, \bar{\text{Value}}_{t-1}, \bar{\text{Service}}_{t-1}) \,, \tag{35}$$

where $G \in \{\text{LM, LSTM, SVM, XGBoost}\}$ and $\bar{M}_{it-1}$ is the average reputation score of dimension M up to time $t-1$, $M \in \{$'Food", "Atmosphere," "Value," "Service"$\}$. The proposed approach significantly ($p < 0.001$) outperforms these alternative specifications. This suggests that the latent dimensions captured by the focal framework contain different information than the observed multidimensional feedback scores.

Figure 16 **Comparison with human-inputted multidimensional reputation**



The $y-$axis shows the percentage improvement of the `HMM-D2V` framework over the $x-$axis reputation system in terms of Spearman ρ and Kendall τ. Each one of the alternative reputation systems uses the four additional humanly-inputted reputation dimensions (Equation 35). The `HMM-D2V` framework significantly ($p < 0.001$) outperforms all alternative reputation systems, with average 10-fold cross-validated improvements ranging between 15% and 120%. Error bars represent 95% confidence intervals.

F. Parameter tuning

The alternative reputation systems discussed in Section 5 require hyperparameter tuning. This Appendix discusses the grid search approach I follow to tune the parameters of Gradient boosting and the dynamic LSTM network.

For Gradient boosting I use the Python package `xgboost`. I tune four hyperparameters: the number of trees to fit ("`n_estimators`"), the maximum tree depth ("`max_depth`"), the boosting learning rate ("`learning_rate`") and the subsample ratio of the training instance ("`subsample`"). I estimate the 10-fold cross-validated ranking correlation scores for the following combinations:

$$\text{XGBoost combinations:} \left\{ \overbrace{\{50, 100, 500\}}^{\texttt{n\_estimators}} \times \overbrace{\{3, 5, 10\}}^{\texttt{max\_depth}} \times \overbrace{\{0.8, 0.9, 1\}}^{\texttt{subsample}} \times \overbrace{\{0.001, 0.005, 0.01\}}^{\texttt{learning\_rate}} \right\}. \quad (36)$$

For building LSTM networks, I use the Python packages `keras.models.Sequential` and `keras.layers.LSTM`. I use adaptive learning rate optimization (Kingma and Ba 2014) to minimize the Mean Absolute Error (MAE) of the model. I tune three hyperparameters: the dimensionality of the output space ("`units`"), the number of "`epochs`" to train the model, and the number of samples

per gradient update "`batch_size`." I estimate the 10-fold cross-validated ranking correlation scores for the following combinations:

$$\text{LSTM combinations:} \left\{ \overbrace{\{20, 30, 40\}}^{\texttt{units}} \times \overbrace{\{10, 20, 30\}}^{\texttt{epochs}} \times \overbrace{\{32, 64, 128\}}^{\texttt{batch\_size}} \right\}. \tag{37}$$

Based on the resulting 10-fold cross-validated scores, I choose the following configurations for the main analysis:

- XGBoost: `learning_rate`: 0.01, `max_depth`: 2 , `n_estimators`: 100, `subsample`: 1.
- LSTM: `units`: 30, `epochs`: 20 , `batch_size`: 64.

G. Predictive features for job-applicant recommendations

To form vector $\mathbf{W}_p$ and build the job-applicant recommender systems described in Section 5.4.2, I use the same set of predictive variables as in Kokkodis et al. (2015). These are:

1. Years of experience: the self-reported job-applicant's experience (numeric).
2. Education: the self-reported education level of the job applicant (categorical).
3. Work-hours: the number of hours that the job-applicant has worked on the platform (numeric).
4. Rehire: whether or not the job-applicant has worked with the employer (numeric).
5. Current reputation: the accumulated reputation of the job applicant (numeric).
6. Certifications: the number of certification tests of the job applicant (numeric)
7. Bid: the bid price of the job applicant (numeric).
8. Completed jobs: the total number of the job-applicant's completed jobs (numeric).
9. Applicant-employer countries' PMI: the pairwise mutual information between the job-applicant's country and the employer's country (numeric; see Equation 38).
10. Certifications inner product: the inner product between the job-applicant certifications and the required skills by the opening (numeric).
11. Skills inner product: the inner product between the skillset of the job-applicant and the required skills by the opening (numeric).

Most of these variables are self-explained. The pairwise mutual information (PMI) of the job-applicant and employer countries is:

$$\text{Applicant-employer countries' PMI}(C_a, C_e) = \log \frac{\Pr(C_a, C_e)}{\Pr(C_a)\Pr(C_e)}, \tag{38}$$

where C_a is the country of the job-applicant, and C_e is the country of the employer.

H. Data-driven managerial insights

Section 3 describes the mapping process of skillsets to latent competency dimensions (W2V). The actual competencies' representations remain hidden, as the framework's primary goal is to provide worker-reputation scores. However, the decomposition of skillsets could provide interesting managerial insights. To illustrate, in this appendix, I examine each competency and try to extract market information that could guide managerial interventions.

H.1. Competency-specific skillsets

Which are the most representative skillsets in each competency? Recall that all skillsets decompose to all competencies (Equation 1). The skillsets that decompose to high weights within each competency are the ones that are more representative and allow workers to transition to higher competency-specific quality states. This is modeled explicitly in in the definition of the HMM observations (i.e., $Y_t^d = w_{\mathcal{R}}^d Y_t$). Hence, I can identify the skillset-specific workers' qualities that each competency captures by extracting the skillsets with the largest weights. Specifically, for each of the ten competencies I consider, the top five weighted skillsets include the following skills:

- Competency-0: {video-production, final-cut-pro, voice-over, video-postediting, video-editing, youtube-marketing}
- Competency-1: {brochure-design, print-design, print-layout-design, illustration}
- Competency-2: {manual-testing, software-testing, software-qa-testing, functional-testing}
- Competency-3: {google-adwords, google-analytics, google-adsense, ppc-advertising, sem}
- Competency-4: {android-app-development, 3d-rendering, unity-3d, rest, node.js, objective-c, angularjs}

- Competency-5: {data-scraping, google-spreadsheet, format-and-layout, data-entry, web-scraping, pdf-conversion, microsoft-excel}
- Competency-6: {portuguese, chinese, german, dutch, japanese, french, spanish, russian, italian}
- Competency-7: {business-analysis, internet-research, data-entry, market-research, business-plans}
- Competency-8: {translation-english-vietnamese, translation-english-italian, translation-english-portuguese, translation-english-korean, translation-english-malay }
- Competency-9: {medical-writing, writing, creative-writing, writing-slang-style, recipe-writing, article-writing, content-writing}

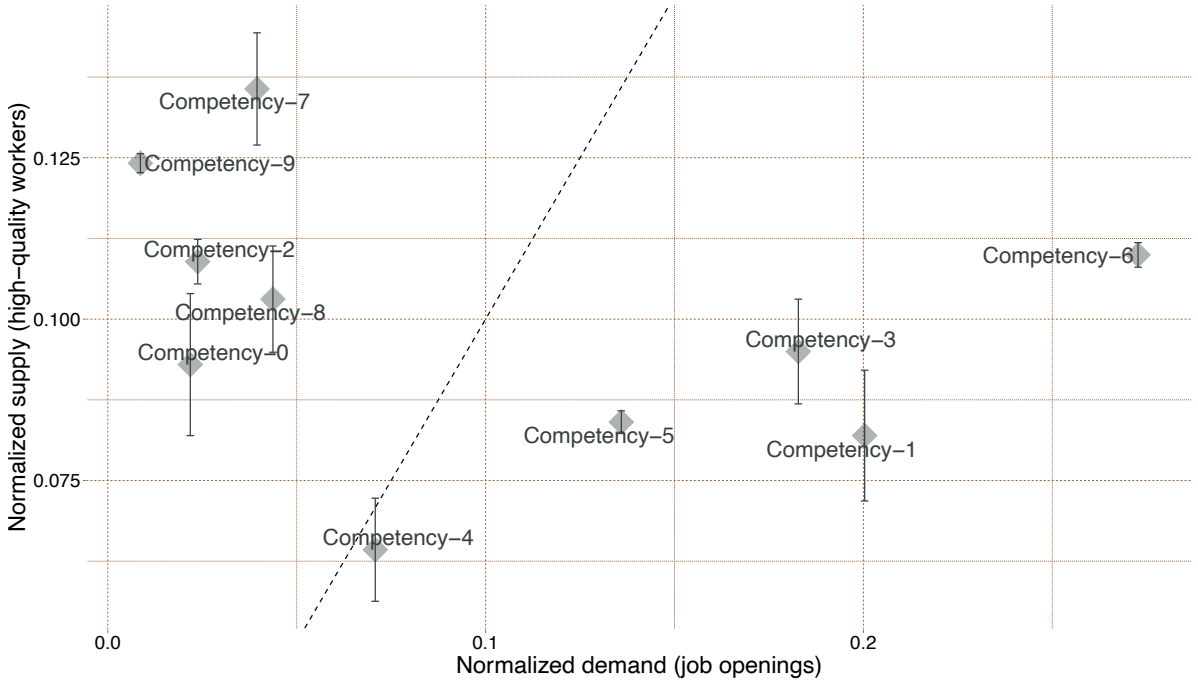
Through Equation 1, each skillset maps to all competencies. As a result, the framework implicitly identifies relationships (correlations) between the available competencies. For instance, when a worker receives a rating that primarily evaluates the worker's performance on a given skillset (e.g., video-production and final-cut-pro, competency-0), the rating maps to all competencies, allowing the framework to estimate an effect of this rating in seemingly uncorrelated competencies (e.g., competency-8). This unique characteristic of the proposed skillset-mapping approach demonstrates that it goes beyond any future human-rated dimensions, such as the ones discussed in Appendix E.3.

H.2. Competency-specific demand and supply

Once I know the skillset representation for each competency, I can identify competencies for which demand is higher than the supply of good workers. In particular, I first identify (through simulations) how many workers exist in the highest-quality state within each competency. This is a proxy of the supply of capable competency-specific workers. Then, I estimate the number of available job openings within each competency. This is a proxy of the competency-specific demand.

Figure 17 shows the scatterplot of the normalized competency-specific demand with the normalized competency-specific supply. The dashed line is the diagonal (slope = 45 degrees). Points below the diagonal represent high demand and low supply of good-quality workers. Points above the diagonal represent high supply of good workers and lower demand. Based on this analysis, the most

Figure 17 **Competency-specific demand and supply of high-quality workers**



The $y-$axis shows the normalized supply of high-quality workers (i.e., workers in the highest-quality state in each competency). The $x-$axis shows the normalized competency-specific demand (i.e., job openings). The dashed line is the diagonal (slope = 45 degrees). Error bars represent 95% confidence intervals of simulated worker paths.

in-demand competency that lacks high-quality workers is competency-6, which includes language-related skillsets. Platform managers can look into the origins of this demand-supply discrepancy and try to attract (new) workers with foreign language skills. On the other hand, competency-7 seems to provide an oversupply of high-quality workers with expertise in business analysis, and internet research. Hence, managers do not need to target workers in this competency at this moment.

To summarize, the `HMM-W2V-framework` allows managers to get deeper market insights through a formal latent-competency and hidden-state analysis, without the need to manually combine skillsets, identify thresholds of high-quality workers, and make assumptions of the current quality of each worker.

I. Discussion of worker competency-specific transitions

Figure 18A shows the cross-competency 10-fold cross-validated improvement of each alternative transition function over the constrained ordered logit function in terms of ranking correlations

(Spearman ρ). As mentioned in Appendix C.5 and is evident in Figure 18A, multinomial transitions perform significantly ($p < 0.001$) better than both ordered logit and constrained ordered logit transitions.

Of particular interest is the comparison between the constrained and unconstrained ordered logit functions. Figure 18A shows that an ordered logit without constrains performs slightly ($p < 0.05$) better. In other words, empirical evidence suggests that restricting transitions to higher-quality states hurts the performance of the model. This is in line with the expected behavior of the `HMM-W2V-framework`, as it continuously uses new evidence to update its quality estimates for each worker (Section 6.4).

To examine the frequency with which workers transition to lower-quality states, I simulate worker paths on the learned competency-specific HMMs and count the times that workers transition to a lower-quality state. Figure 18B shows that, on average, one out of five observed transitions is to a lower-quality state. This observation further highlights the need for the proposed framework to allow lower-state transitions in order to dynamically adjust and update its quality estimations for each worker.

Figure 18 **Constraining transitions to higher-quality states hurts the performance of the `HMM-W2V-framework`**

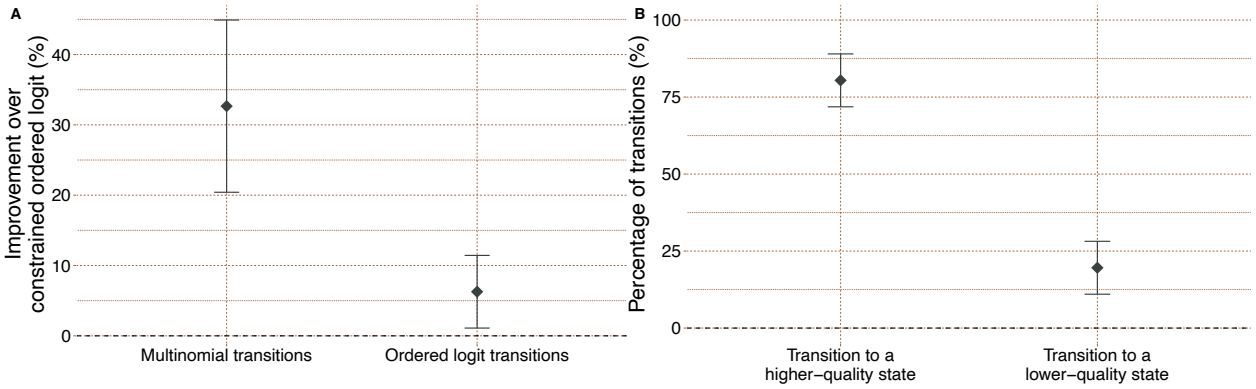


Figure A shows that multinomial and ordered logit transitions yield significantly ($p < 0.05$) better results than constrained ordered logit (Appendix A). Figure B shows that one out of five transitions across the competency-specific HMMs is to a lower-quality state. This highlights the need for the HMMs to adjust and correct their worker-quality estimates in the presence of new evidence. Error bars represent 95% confidence intervals.

J. Markov assumption

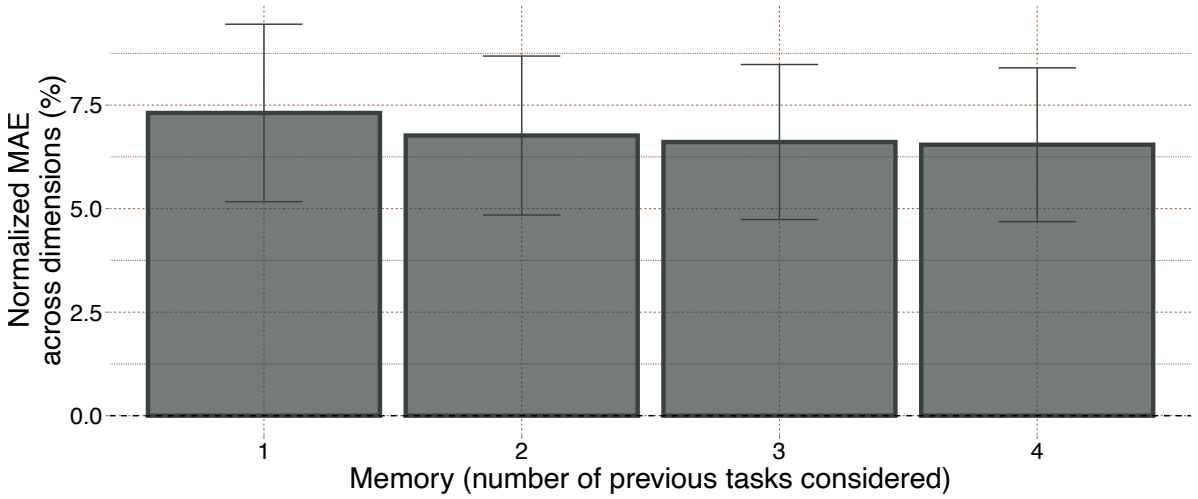
The HMM approach assumes that the next state of each worker depends on the current state of the worker and the set of observed characteristics $\boldsymbol{Z}_{it-1}, \boldsymbol{X}_t$. The presented results in Sections 5, 2.3 and Appendix D show that this assumption does not significantly hurt the performance of the framework.

There are multiple reasons that explain this. First, even though the Markovian assumption suggests that the next state depends on the current state, reaching a current state accumulates effects from the complete sequence of observations up to that point (Murphy 2012, Zucchini et al. 2017, Sahoo et al. 2012). In other words, the current state essentially encapsulates the likelihood of observing the sequence of states up to that point. Second, the fact that I consider up-to-date observed characteristics to affect both emissions and transitions controls for accumulated information up to that point for each worker (e.g., total number of jobs, accumulated feedback scores). Finally, given that each dimension-specific HMM captures dimension-specific reputation, it is reasonable to assume that the last update of a worker's performance in each dimension is likely the most current representation of the worker's dimension-specific quality.

To test this last argument, I estimate the predictive performance of varying memory windows for each dimensions $d \in \mathcal{D}$. Figure 19 shows the results. The $y-$axis shows the normalized MAE across all dimensions. Increasing the memory window to higher orders (2,3,4) appears to not have a significant effect on the predictive performance ($p > 0.05$).

Finally, even in scenarios where the Markovian assumption does not seem to fit the data, application of the Viterbi (Forney 1973) algorithm would likely solve this issue, as the Viterbi algorithm can estimate the current state of each user by maximizing the likelihood of the sequence of observations up to that point.

Figure 19 **Predictive performance of alternative memory windows**



There is no significant ($p > 0.05$) predictive improvement when considering more than one completed tasks. Error bars represent 95% confidence intervals.

References

Bishop, M. Christopher. 2006. *Pattern Recognition and Machine Learning*. Springer.

Byrd, H. Richard, Peihuang Lu, Jorge Nocedal, Ciyou Zhu. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing* **16** 1190–1208.

Forney, G. David. 1973. The viterbi algorithm. *IEEE* **61** 268–278.

Ghose, Anindya, Param Vir Singh, Vilma Todri. 2017. Got annoyed? examining the advertising effectiveness and annoyance dynamics. *International Conference on Information Systems*.

Ghose, Anindya, Vilma Todri. 2016. Towards a digital attribution model: Measuring the impact of display advertising on online consumer behavior. *MIS Quarterly* **40** 889–910.

Hinton, Geoffrey E., Sam T. Roweis. 2003. Stochastic neighbor embedding. *Advances in Neural Information Processing Systems*. 857–864.

Hochreiter, Sepp, Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation* **9** 1735–1780.

Hu, Nan, Paul A. Pavlou, Jie Zhang. 2017. On self-selection biases in online product reviews. *MIS Quarterly* **41** 449–471.

Kingma, Diederik P, Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint* 1412.6980.

Kokkodis, Marios, Panagiotis Papadimitriou, Panagiotis G. Ipeirotis. 2015. Hiring behavior models for online labor markets. *International Conference on Web Search and Data Mining*. 223–232.

Koller, Daphne, Nir Friedman. 2009. *Probabilistic graphical models: Principles and techniques*. MIT press.

Le, Quoc, Tomas Mikolov. 2014. Distributed representations of sentences and documents. *International Conference on Machine Learning*. 1188–1196.

Luca, Michael. 2016. Reviews, reputation, and revenue: The case of yelp.com. (Working Paper).

Murphy, Kevin P. 2012. *Machine learning: A probabilistic perspective*. The MIT Press.

Sahoo, Nachiketa, Param Vir Singh, Tridas Mukhopadhyay. 2012. A hidden Markov model for collaborative filtering. *MIS Quarterly* **36** 1329–1356.

Schwarz, G. 1978. Estimating the dimension of a model. *The Annals of Statistics* **6** 461–464.

Todri, Vilma, Anindya Ghose, Param Vir Singh. 2020. Trade-offs in online advertising: Advertising effectiveness and annoyance dynamics across the purchase funnel. *Information Systems Research* **31** 102–125.

Wooldridge, M. Jeffrey. 2010. *Econometric analysis of cross section and panel data*. MIT Press.

Zucchini, Walter, Iain L. MacDonald, Roland Langrock. 2017. *Hidden Markov models for time series: An introduction using R*. CRC press.