The Student Zipf Theory: Inferring Latent Structures in Open-Ended Student Work To Help Educators

Yunsung Kim
Stanford University
Stanford, California, USA
yunsung@stanford.edu

Chris Piech
Stanford University
Stanford, California, USA
piech@cs.stanford.edu

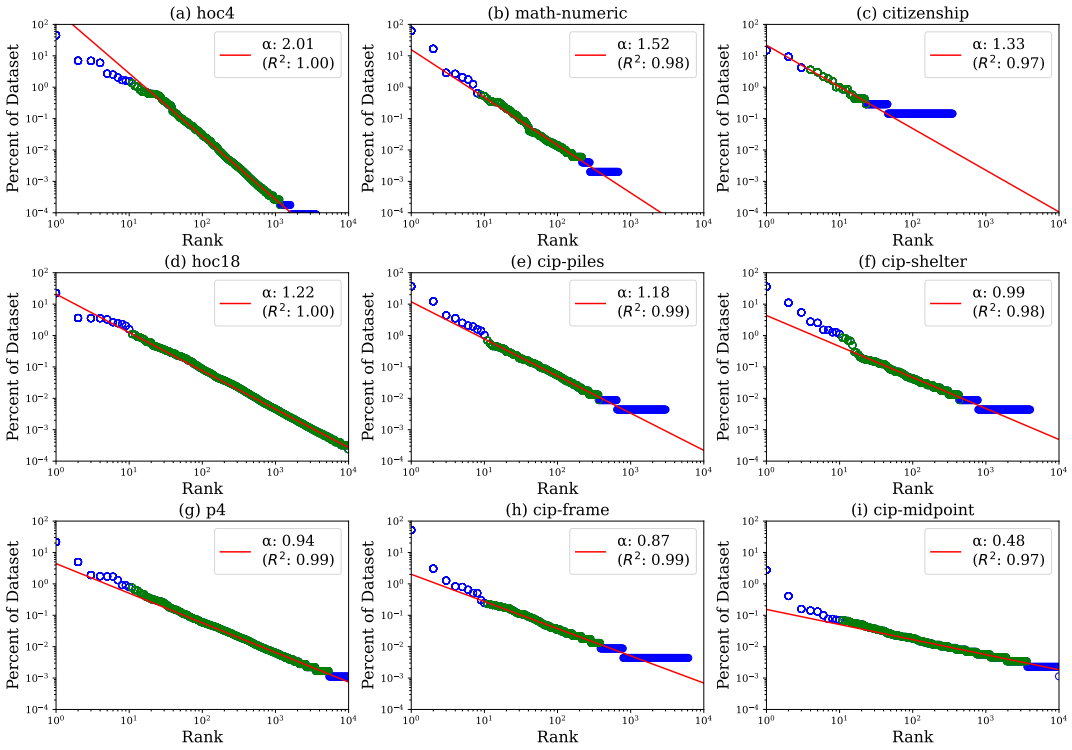

Figure 1: Rank-Frequency plots of open-ended student responses to various assignments, along with the R^2 of the linear log-rank-log-frquency fit. (See Section 3 and Section 8 for more detail.) Plots are ordered by the fitted Zipf exponent α. The Zipfian approximations were performed from the 10th most frequent response up to the infrequent tail (green).

ABSTRACT

Are there structures underlying student work that are universal across every open-ended task? We demonstrate that, across many subjects and assignment types, the probability distribution underlying student-generated open-ended work is close to Zipf's Law. Inferring this latent structure for classroom assignments can help learning analytics researchers, instruction designers, and educators understand the landscape of various student approaches, assess the complexity of assignments, and prioritise pedagogical attention. However, typical classrooms are way too small to witness even the *contour* of the Zipfian pattern, and it is generally impossible to perform inference for Zipf's law from such small number of samples. We formalise this difficult task as the *Zipf Inference Challenge*: (1) Infer the ordering of student-generated works by their underlying probabilities, and (2) Estimate the shape parameter of the underlying distribution in a typical-sized classroom. Our key insight in addressing this challenge is to leverage the densities of the student response landscapes represented by semantic similarity. We show that our "Semantic Density Estimation" method is able to do a much better job at inferring the latent Zipf shape and the probability-ordering of student responses for real world education datasets.

LAK 2023, March 13–17, 2023, Arlington, TX, USA

ACM ISBN 978-1-4503-9865-7/23/03…$15.00
https://doi.org/10.1145/3576050.3576116

CCS CONCEPTS

• **Applied computing** → **Computer-assisted instruction**; • **Computing methodologies** → Artificial intelligence.

KEYWORDS

Zipf's Law, Open-Ended Response, Constructed Response, Student Work, Probabilistic Modeling

ACM Reference Format:
Yunsung Kim and Chris Piech. 2023. The Student Zipf Theory: Inferring Latent Structures in Open-Ended Student Work To Help Educators. In *LAK23: 13th International Learning Analytics and Knowledge Conference (LAK 2023), March 13–17, 2023, Arlington, TX, USA.* ACM, New York, NY, USA, 12 pages. https://doi.org/10.1145/3576050.3576116

1 INTRODUCTION

Open-Ended Questions are a type of learning assessment in which a student is asked to freely construct an answer to a problem. The format of the responses can be diverse, ranging in complexity from numeric values and short answers with a few words, to complex essays and program codes. Responses to these types of questions often reflect rich aspects of student thinking that closed-form questions cannot afford to capture. For this reason, open-ended questions serve as vital assessment tools that offer instructors a deep understanding of each student in a classroom [5, 20].

The constructive nature of these questions often elicit a highly diversified set of responses, which is why in typical-sized classrooms, every student turns in a highly unique response. In larger classes, interesting probabilistic patterns emerge that are present but not visible in a small-sized classrooms. It has been noted in some occasions [33, 37] that the distribution of student-generated works for certain assignments in large-scale classes appeared to follow *Zipf's law*. A probability distribution is said to follow Zipf's law when the the probability of the r-th most probable outcome follows an inverse power-law with respect to r. In this case, the probability of outcome w is proportional to

$$p(w) \propto \frac{1}{\text{rank}(w)^{\alpha}} \tag{1}$$

where $\text{rank}(w)$ is the rank of the outcome w when all outcomes are ordered by their probabilities, and $\alpha > 0$ is the *exponent parameter* that determines the shape of the distribution. Graphically, this results in a linear relationship between log-rank and log-probability with slope $-\alpha$.

The Zipfian student work observation introduces many fundamental questions about the *structures* in student-constructed responses: Are the Zipf-like observations global *across different subjects and assignment types*? What *practical implications* does estimating these patterns bear for everyday classrooms? Can we *infer* these structures even when most students would submit a unique response? And lastly, *why* do such structures emerge in student responses?

And yet, the analysis and inference of Zipf's law in student responses are not an easy endeavor. Most importantly, Zipfian patterns are *impossible* to observe directly in typical sized classrooms with fewer than 100 students. This is because the heavy-tail of Zipf's law causes the majority of the responses to appear only once in a sample. Observations of these patterns has only recently been made possible with the growth of massive open online courses.

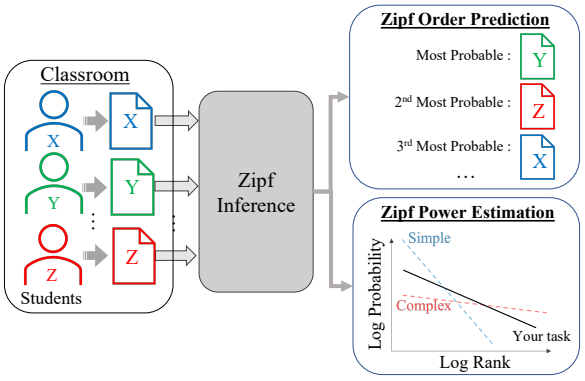

Figure 2: Zipf Inference Challenge: given open-ended student responses (1) order them by their intrinsic probability, and (2) estimate the Zipf exponent of the assignment.

In this paper, we will explain the importance of Zipf's law and take the first steps at addressing the aforementioned fundamental questions. We will approach this seemingly impossible inference problem with the key insight of analyzing the *density* of the student response landscapes encoded in a metric space representing semantic similarity.

Concretely, our work delivers the following contributions[1]:

(1) For the first time, we observe that Zipfian patterns in student works *generalize* across various subjects and open-ended assignment types (Figure 1). We explain the practical implications this has for learning analytics researchers, instruction designers, and educators.
(2) We formally define the *Zipf Inference Challenge* (Figure 2) which consists of (1) estimating the latent probability ordering and (2) inferring the Zipf exponent from few student responses observed in regular classrooms.
(3) We provide a generic framework called *Semantic Density Estimation* for addressing Zipf inference challenge with the key intuition of using density estimation in semantic embedding space. We demonstrate the effectiveness of this framework on multiple real-world datasets and show that it outperforms the baseline methods.
(4) We take initial steps in proposing a theoretical explanation for how Zipfian patterns could emerge from the open-ended student response process.

2 WHY DOES ZIPF'S LAW MATTER IN EDUCATION?

Not only are Zipfian patterns in open-ended student work intrinsically valuable as a beautiful natural phenomenon, but these patterns may also convey many *practical* values for learning analytics researchers, learning designers, and practitioners.

[1]The code for our inference algorithm and theoretical analysis can be found at: `https://github.com/yunsungkim0908/student-zipf-theory`

First, the existence of Zipf's law has a significant consequence for the scalability of student analytics methods and algorithms - an important consideration for learning analytics and educational technology researchers. The long-tail of the response distribution implies that an overwhelming fraction of the observed responses are likely to be observed exactly once, regardless of the size of the samples. (For instance, see Figure 4). This immediately has a huge impact on *grading and feedback generation*. For example in 2014, Code.org, a massive open online course (MOOC) for programming, launched an initiative to crowdsource hundreds of thousands of instructors to provide feedback to student-generated programs, with the goal of labelling all possible answers. For short programs this seemed feasible, but it took thousands of human hours of labelling to realize that the space of unique responses was too vast for human effort to even scratch the surface. This initiative was cancelled in 2 years and hasn't been reproduced since [53]. Had it been known that the response distribution was Zipf-like, these efforts would have been reconsidered early on.

Inferring the Zipf exponent parameter α allows one to quantify such heavy-tailedness of the underlying distribution. Large α results in a highly skewed distribution where most probability is concentrated on the high-rank responses, whereas smaller α indicates a distribution that is closer to uniform and heavier-tailed. (Figure 1.) Therefore, estimating α allows one to anticipate the amount of unique responses that would appear in samples from this distribution, which can be used to schedule human efforts or anticipate the efficiency of new analytics algorithms. α may also be used to measure the complexity (or the degree of individual variability in the responses) of an assignment and compare it across different assignments, a hypothesis we develop in Section 3 based on real student response data.

Next, educators and instruction designers may benefit much from inferring the rank-ordering of the individual responses by their underlying Zipfian probabilities.[2] These probabilities indicate how often one would see the exact same response if the class size were to be much larger, and algorithmically inferring them can help educators quickly build insights into the rank-ordering of student approaches and misconceptions. This can be useful for pedagogical tasks such as noticing [27] and anticipating [34, 49] student responses, and prioritizing instructor feedback. For instance, imagine teaching an introductory probability class where we ask 50 students to write a short program and most of them turn in answers that are unique. (See Figure 3). Furthermore, assume that we were able to algorithmically determine that, if the class were to be *infinitely large* instead of 50, responses (a), (b), and (c) in Figure 3 would be the most frequently observed responses among the student responses that used a for-loop. This immediately helps instructors notice that finding the complementary probability might be a prevalent mistake in the class and is worth addressing during lecture. Also, since failing to apply an important probability concept (the product rule) is more likely among students than a minor off-by-one error, the instructors may want to change future instruction plans to dedicate more time to review this important concept, and instruction designers may want to reconsider whether the current curriculum sufficiently covers it.

[2]We leave the empirical validation of the potential benefits in real classrooms as a promising direction for future research.

The Birthday Problem. Write a function that computes the probability that, in a set of n randomly chosen people, at least two will share a birthday.

Examples of student answers using a for-loop:

(a) Rank 2: Probability of a complementary event

```
def birthday(n):
    prob = 1
    for i in range(n):
        prob *= (1 - i/365)
    return 1 - prob
```

(b) Rank 5: Misapplies a core probability concept

```
def birthday(n):
    prob = 1
    for i in range(n):
        prob -= (1 - i/365)
    return prob
```

(c) Rank 9: Makes a minor off-by-one mistake

```
def birthday(n):
    prob = 1
    for i in range(n):
        prob *= (1 - (i+1)/365)
    return prob
```

(d) Rank 46: Wrong use of programming constructs

```
def birthday(n):
    prob = 1
    for i in range(n):
        prob * 1/365
```

Figure 3: Example of an open-ended assignment, student responses, and the probability ranks of the responses within the class.

Moreover, rank-order inference will also tell us which responses are closer to the "long-tail" that are intrinsically less common. These responses would often benefit most from a prioritized follow-up from the instructor, either to address the idiosyncratic error state of the student (for instance, by suggesting a programming review for response (d)) or to analyze and address unanticipated approaches to the problem.

Similar use cases exist when piloting a large-scale course with a handful of students before it goes live. However, without algorithmically inferring the rank-ordering, human instructors would need to read, analyze, and organize these responses *manually* before they can engage in these activities and draw insights about students.

3 ZIPFIAN STRUCTURES IN STUDENT WORK

In previous works, a handful of assignments have been identified as Zipf-like. Here we gather a larger collection of open-ended assignments where there are enough submissions to observe if there is a Zipfian structure, and analyze what the fitted Zipf exponent parameters convey about the properties of the assignment.

Figure 1 plots the ranks and (relative) frequencies of student responses to 9 different assignments in log scale. These datasets were collected in the following context (See Table 1 for a detailed summary of each dataset):

Name	Total Responses	Unique Responses	Avg. Length	α	Problem Description
hoc4	1,128,916	3,630	52.4	2.08	Simple maze with 3 types of basic navigator blocks
math-numeric	49,847	680	-	1.55	Grade-2 level math question with numeric answer
citizenship	697	353	4.8	1.37	Short response question about a basic US history fact
hoc18	1,253,776	56,612	78.0	1.25	Maze with navigators plus "while" and "if" blocks
cip-piles	22,828	3,003	83.4	1.21	Move robot along a straight line and repeat a simple sub-task
p4	179,229	38,034	49.8	0.98	Draw nested shapes with "for" and "variable" blocks in 2D grid
cip-shelter	22,824	4,027	110.8	0.95	Move robot through uneven obstacles. Hints on modularization.
cip-frame	22,825	6,148	150.9	0.88	Move robot and execute nested tasks with variables. No hint
cip-midpoint	87,780	63,821	-	0.55	Intermediate progress for a robot assignment

Table 1: Summary of datasets used in our experiment, ordered by Zipf exponent α. More complex tasks have smaller exponents. The fitted exponent parameter α provides a way to compare the complexity of tasks across problem types (Section 3). Average length is the average number of program or word tokens. (Block programs were first translated into python syntax. See Section 6).

Block Programming (hoc4, hoc18, p4): *Code.org* is an online programming education platform aimed at teaching beginner programmers the elementary concepts of programming. The data we analyzed are student responses to drag-and-drop, block-based programming maze and drawing puzzles, each with different task descriptions and programming primitives allowed. These puzzles were ordered in such a way that the programming concepts involved in each puzzle were built up incrementally (hoc4 → hoc18 → p4).

General Python Syntax Programming (CIP): In 2020, Stanford University launched an introductory online programming class called *Code-in-Place* (CIP) [35], designed to deliver computing education at the level of Stanford University's introductory Python programming course to a global audience in the context of COVID-19. In several assignments, students were asked to write programs in general Python syntax that manipulate a virtual robot on a 2D maze grid and execute small tasks. Among these assignments, *piles* and *shelter* were "warm-up" problems, while *frame* was a regular assignment. Although the solution to *shelter* involved more steps, students were provided with hints on how to modularize the code with custom functions. No hints were provided for *frame*. Similar to Code.org assignments, CIP assignments were also ordered (piles → shelter → frame) according to their incremental build-up of concepts.

Citizenship Test This dataset has crowdsourced [4] short-answer responses to a US citizenship test. We chose the most challenging question as measured by [40]: "What is one reason the original colonists came to America?"

Grade-School Math Word Problem This is a dataset of numerical responses to an the elementary school math word problem from an online mathematics learning platform called *Beast Academy*, hosted by the Art of Problem Solving. We focus on the problem with the largest number of responses, which is a 2nd grade math word problem on digits and counting: "How many different two-digit numbers have 0 as a digit?"

For each empirical rank-frequency distribution, we found an approximation to Zipf's law by conducting linear regression between log-rank and log-frequency for samples in the "body" that ranked below the 10th most frequent and appeared at least 3 times. The quality of this approximation was measured by the coefficient of determination (R^2) for each fit. (We defer a more detailed discussion about the method of this analysis to Section 8.) The high R^2 values (> 0.96) for all datasets in Figure 1 suggest that Zipf's law does yield a good approximation to these distributions. Similar to the observations in [37], most Zipfian patterns become visibly apparent after the 10th most frequent submission and are particularly striking for a wide range of ranks leading to the tail of the dataset.

Zipf-like patterns in student work also appear to *generalize* across a wide range of domains and assignment types. Notice that the exponent parameter also inversely correlates with the "complexity" of the assignments, and is *comparable* across assignments. For instance, the fitted exponent parameters in both Code.org and CIP align with the ordering of the assignments within each curriculum and the incremental development of the associated programming concepts. Also, CIP assignments are generally more complex than assignments from Code.org because students were allowed to use any Python programming functionality, whereas students in Code.org were restricted to using block-based programs. This aligns with the fact that CIP assignments generally have smaller α than Code.org assignments. Our observations thus support the hypothesis that assignments that involve more intricate tasks and allow greater agency to students in constructing their response have smaller α.

Do responses to even more complex and individualized assignments like essay and composition questions also exhibit Zipf-like patterns? To witness the underlying response distribution for such assignments, one would need to sample magnitudes of more responses compared to the assignments in Table 1. However, grounded on the above findings, we hypothesize that the ground-truth probability distributions of student responses to such complex problems in most domains will also be well-approximated by Zipf's law. We call this the *Student Zipf Hypothesis*.

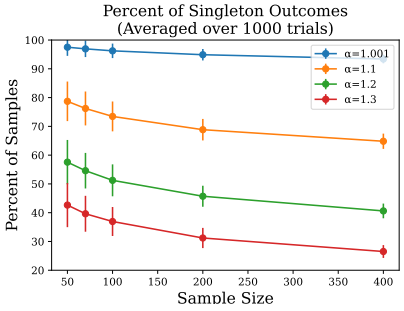

Figure 4: Percent of outcomes that appear exactly once in a sample from Zipf's law with different exponents α. Smaller α results in a higher fraction of "singletons."

4 THE ZIPF INFERENCE CHALLENGE

So far we have observed through a collection of large bodies of open-ended student responses that the underlying distribution of student work can be closely approximated by Zipf's law. We have also discussed how the parameters of this pattern can positively help the research and practice of education and learning analytics. And yet, typical classrooms are way too small to observe even the *contour* of the full distribution of responses shown in Figure 1. Taking a regular university setting as an example, it is common to have classes of fewer than a hundred students. Due to the heavy-tail property of Zipf's law, however, the majority (if not all) of the student responses will appear exactly once as in Figure 4, making it seem almost impossible to reliably estimate their underlying probability structures.

The ***Zipf Inference Challenge*** formalizes this difficult task of drawing meaningful information about the ground-truth Zipf's law of student work in the face of the small-sample, heavy-tail limitation. In this challenge, we are given a collection of student-generated responses $W = \{w_1, ..., w_n\}$ from a class of n students, which consists of m *unique* responses $U = \{u_1, ..., u_m\}$ after removing duplicates. These unique works have a hidden, ground-truth ordering (or permutation) σ^* which ranks $u_{\sigma^*(i)}$ at rank-i and orders the unique works by their probability masses. Without loss of generality, we will assume that $\sigma^*(i) = i$, so that $p(u_1) \geq p(u_2) \geq ... \geq p(u_m)$. The challenge consists of the following two sub-tasks:

Zipf Order Inference Given the collection W of student responses and the set U of unique responses, estimate the ground-truth ordering σ^* of the unique student responses by their underlying probabilities.

Zipf Exponent Estimation Given the collection of student works W, estimate the exponent α (or the "slope" of the log-rank vs log-probability relationship) in Equation 1.

4.1 Evaluating Zipf Order Inference

To measure the quality of a predicted Zipf ordering, we will use the *normalized l_1 distance of the induced log probabilities*. In this section, we will define this metric in detail and motivate our choice.

The metric for measuring the quality of the predicted ordering should be mindful of both the ordering itself and the underlying *probabilities* that are associated with each student work. Consider this worked example: we need to score the following three Zipf predictions σ_1, σ_2 and σ_3 for the rank-ordering of four unique student solutions u_1, u_2, u_3 and u_4:

$$\sigma_1 : p(u_2) \geq p(u_1) \geq p(u_3) \geq p(u_4)$$
$$\sigma_2 : p(u_1) \geq p(u_3) \geq p(u_2) \geq p(u_4)$$
$$\sigma_3 : p(u_1) \geq p(u_2) \geq p(u_4) \geq p(u_3)$$

We can score these predictions based on our knowledge of the true probabilities from the underlying distribution:

$$p(u_1) = 0.7, p(u_2) = 0.25, p(u_3) = 0.03, p(u_4) = 0.02.$$

Here, u_1 and u_2 are highly probable and together comprise 95% of the total probability mass, whereas u_3 and u_4 are much less probable compared to the other two items.

Each of these 3 orderings has exactly one discordant pair: (u_1, u_2), (u_2, u_3), and (u_3, u_4). Taking into account the ground-truth probabilities of each element, σ_1 and σ_2 should be penalized more heavily than σ_3 because the probabilities of the discordant pairs in σ_3 are comparable ($p(u_3) \approx p(u_4)$). Also, while the error in σ_1 occurs *within* a group of highly probable items, the error in σ_2 occurs *across* groups that differ by orders of magnitudes in probabilities. Misjudging a highly improbable response to be the opposite can be more critical in practice than misordering two highly likely responses, so σ_2 should be more heavily penalized than σ_1.

In this light, typical rank correlation metrics such as Kendall's τ or Spearman's ρ that only consider the relative positions within an ordering are not suitable as metric for Zipf order inference. Common ranking metrics such as Discounted Cumulative Gain, Mean Reciprocal Rank, or Mean Average Precision that are used to evaluate rankings based on the "relevance" of items to a central query are also ill-suited for a holistic evaluation of ordering.

Instead, we will view an ordering σ as inducing a probability distribution p^σ over $u_1, ..., u_m$ by assigning the i-th largest probability mass p_i^* to $u_{\sigma(i)}$, and compare this induced distribution against the ground-truth distribution p^*. In particular, we will let $p^\sigma(u_{\sigma(i)}) = p_i^*$ and use the following sum of absolute log probability ratios (which is ℓ_1 distance in log probabilities)

$$\ell(\sigma; p^*) = \sum_{i=1}^{m} \left| \log p^*(u_i) - \log p^\sigma(u_i) \right| = \sum_{i=1}^{m} \left| \log \frac{p^*(u_i)}{p^\sigma(u_i)} \right|.$$

The use of log probabilities also has the effect of preventing disproportionately large probabilities in the head from obscuring the behaviors in the body and tail. Under this metric, if σ predicts rank i for student work u_j, it is penalized more heavily when the difference in the log probabilities $\log p^*(u_i) - \log p^*(u_j)$ (or equivalently, the log probability ratio $\log \frac{p^*(u_i)}{p^*(u_j)}$) is larger in magnitude. Note that the true ordering σ^* achieves the smallest possible $\ell(\sigma^*; p) = 0$.

To make this metric comparable across different classroom samples, we will scale it to be within $[0, 1]$ and use the following *normalized ℓ_1 distance in log probabilities*:

$$\widetilde{\ell}(\sigma; p) = \frac{\ell(\sigma; p)}{\max_{\sigma'} \ell(\sigma'; p)} \in [0, 1]. \quad (2)$$

4.2 Evaluating Zipf Exponent Estimation

Although Zipf Exponent Estimation is technically a regression task, it suffices for practical purposes to reliably *compare* the computed exponent parameters of different assignments instead of estimating them in perfect scale precision. Therefore, we can measure the quality of the estimated exponent parameter by computing the correlation between the estimated exponent and the true exponent.

5 OUR NOVEL ZIPF INFERENCE METHOD: SEMANTIC DENSITY ESTIMATION

Having formally defined Zipf Inference Challenge, we are now ready to present the first method for Zipf Inference. Our proposed method builds on the following assumption about the underlying probability distribution of responses:

"Responses that are semantically similar are also likely to be similar in probability-rank."

This will motivate us to consider the idea of *semantic distance* that measures semantic (dis)similarity, and we will explain how Zipf Inference can be done based on the *density* of the observed unique responses under this distance metric.

Many studies support the idea that an open-ended student response can be modeled by a hierarchical set of decisions and choices that were made during the response process [18, 42]. Each element in the generated response - ranging in granularity from the global outline to the specific word choices for essays or code components for programming - is traceable to one of the many such decisions. This idea of a "student decision process" (Figure 5) has previously been implemented in the form of probabilistic context-free grammars [53] or probabilistic programs [24] and has found several pedagogical use cases such as in autonomous grading or student feedback generation [24, 53].

It is then reasonable to think that a pair of responses that result from similar decisions would also be likely to be similar in probability-rank. For instance in problem-solving, the set of decisions resulting in common mistakes or misconceptions often diverge slightly (and quite predictably) from the set of decisions made in producing the most probable responses [10, 50]. On the other hand, uncommon responses tend to involve few highly improbable decisions or differ in most decisions from the "common approaches," if not both [50].

Under this assumption, if we were to have a reasonable *semantic* distance metric $d(\cdot, \cdot)$ that measures the dissimilarities in approach, decisions involved, and misconceptions or idiosyncrasies, responses with high probabilities (and thus high ranks) would be close together, and are likely to form a cluster in this "semantic" space where much of the probabilities are concentrated. This has two consequences for the observed responses. First, many observed responses will be drawn from this high-probability cluster and densely populated within the sample. This suggests finding high-rank responses by looking for responses with high sample density. Next, when the underlying distribution is closer to uniform than skewed (corresponding to a smaller Zipf exponent α), probabilities will become less concentrated on the aforementioned high-density neighborhoods. This will cause the observed responses to be more evenly spread out over the space of possible responses, and the sample densities will vary less from the densest region to the sparsest.

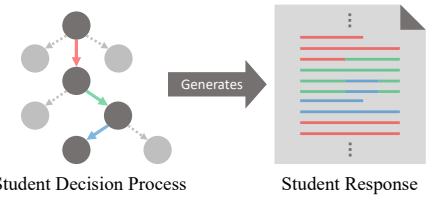

Figure 5: The student decision process. Decisions made by students (colored edges) are reflected in their responses (portions with matching colors).

Therefore, we will choose a metric of dissimilarity d and consider the *sample density* of the observed responses in the space defined by d. We now crystallize this idea into the ***Semantic Density Estimation (SDE)*** framework, inspired by the kernel density estimation method [46] used for estimating densities in vector spaces based on samples.

5.1 Kernel Density Estimation Review

Kernel Density Estimation (KDE) [46] is a popular method for estimating an unknown probability density function from samples of data in a vector space when no specific form of the density function can be appropriately assumed. Given a set X of observed samples, KDE estimates the density $f(x)$ at an input vector x using a sum of *weights*, each determined by how close x is to each of the vectors $X_i \in X$. Points that are closer to x contribute a larger weight to $f(x)$, so the resulting density estimate is larger when many of the points in the sample set are close to x.

These weights are formally determined by a *Kernel function* K, which is often a symmetric probability density function that satisfies $\int_{-\infty}^{\infty} K(x)dx = 1$. In particular, the kernel estimator with kernel K is defined as

$$f(x) = \frac{1}{nh} \sum_{i=1}^{n} K\left(\frac{\|x - X_i\|}{h}\right), \tag{3}$$

where $\|\cdot\|$ indicates vector norm, n is the size of the observed samples and h is the *bandwidth* that scales the region of influence for each sample point. Increasing the bandwidth causes more distant points to contribute to the resulting density, and thus results in a smoother density estimate overall. Two most widely used kernels are linear and Gaussian kernels. The kernel estimator may also be viewed as a mixture distribution where each mixture component is centered on X_i.

When the samples are high-dimensional or come from long-tailed distributions, having a fixed bandwidth across the entire sample as in Equation 3 tend to introduce spurious noise in samples with relatively low sample density [46]. *Variable kernel estimation* works around this issue by setting the bandwidth of the kernel centered on X_j to be proportional to the distance to its k-th nearest

neighbor $d_{j,k}$, thus adapting to the local density of each sample[3]:

$$f(x) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{hd_{j,k}}K\left(\frac{\|x - X_i\|}{hd_{j,k}}\right). \quad (4)$$

5.2 The Semantic Density Estimation (SDE) Framework

Although the objective of KDE is to estimate an unknown probability density from samples, it can also be used as a measure of empirical density within those samples. This is the core idea behind our *Semantic Density Estimation (SDE)* Framework. With the notion of semantic distance $d(\cdot,\cdot)$ mentioned earlier, we use the following variant of Equation 4 for the variable kernel estimator to estimate the within-sample density for a given student work w:

$$f(w) = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{hd_{j,k}}K\left(\frac{d(w,u_i)}{hd_{j,k}}\right), \quad (5)$$

where $u_i \in U$ indicates the i-th unique student work[4], and vector distance $\|x - X_i\|$ in Equation 4 is replaced with $d(w,u_i)$. $d(\cdot,\cdot)$ can then be defined in several ways:

Using a semantic vector-space embedding function. Any semantic embedding function ϕ that maps student work to a real vector can be used to define $d(u,v) = \|\phi(u) - \phi(v)\|$. This is equivalent to KDE in the space defined by ϕ. To the authors' knowledge, no embedding method is yet known to reliably reflect the problem-solving approaches in student-generated work in any domain. As a proxy to such model, we will use the embeddings from large-scale pre-trained semantic encoders such as CodeBERT [17] and the ℓ_2 distance metric in our experiments to demonstrate the effectiveness of SDE.

Using a generic distance metric. Even without resorting to an explicit embedding function ϕ, any metric of dissimilarity can be used as d[5]. As a demonstration of the potentials of the SDE framework, we will use *token edit distance* as our distance metric in our experiments, which further assumes that similarity in decisions also correlates with syntactic similarity.

Zipf Order Inference Method. Once we obtain the density estimates $f(u_1), ..., f(u_m)$ for each of the m unique student responses, it is straightforward to estimate the ordering: simply output the ranking $\widehat{\sigma}$ that rank-orders the density estimates, such that

$$f(u_{\widehat{\sigma}(1)}) \geq f(u_{\widehat{\sigma}(2)}) \geq ... \geq f(u_{\widehat{\sigma}(m)}). \quad (6)$$

In practice, when a response is observed multiple times in a classroom, this can be a strong signal that the response is highly probable. Therefore, in the actual ordering that we output, we will first order the duplicate items according to their multiplicity in the sample set, followed by the "singletons" ordered by the density estimates[6].

Zipf Exponent Estimation Method. As mentioned earlier, we expect samples associated with smaller Zipfian exponents α to have more consistent sample densities across the samples. In this spirit, our (un-scaled) exponent estimate $\widehat{\alpha}$ will be the *inter-quartile* difference ratio in log densities, defined as the ratio between the difference in log densities[7] and the difference in rank of the 1st and 3rd quartile:

$$\widehat{\alpha} = \frac{\log f(u_{\widehat{\sigma}(Q_1)}) - \log f(u_{\widehat{\sigma}(Q_3)})}{Q_1 - Q_3}. \quad (7)$$

Q_1 and Q_3 are the indices of the 1st and 3rd quartile. Since the main purpose of estimating Zipf exponent is to be able to compare one assignment to another, we only require that the estimated exponent is correlated with the true exponent.

5.3 Baseline Methods for Zipf Order Prediction

Although there is no well-defined baseline to compare SDE against, we can consider comparing our method against the following reasonable approaches for Zipf Order Inference[8]:

Random permutation. This elementary baseline method will randomly order the responses.

Length-based method. A common practice when skimming student responses is to look at the length of the response and deem abnormally long or short responses unlikely. This baseline reflects this real-life practice and orders responses according to the difference of its length from the average length.

Density Estimation Tree. Density estimation tree (DET) [39] is a method for estimating a probability density over a d-dimensional vector space using a piece-wise constant decision tree. We used DET to estimate the within-sample densities of the responses based on the 2-dimensional PCA projections[9] of their CodeBERT embeddings, and estimated the rank-ordering in a way similar to SDE.

6 ZIPF INFERENCE EXPERIMENTS

We now present the results of evaluating our Zipf Inference method using 6 real student datasets: hoc4, hoc18, cip-piles, p4, cip-shelter, and cip-frame[10].

Experimental Setup. For each dataset, we simulated 300 classrooms of 70 students each, by subsampling (with replacement) from the entire dataset of responses. For Zipf Order Prediction, we computed the average ℓ_1 distance in log probabilities (Equation 2) of each method (Figure 6). For Zipf Exponent Estimation, we computed the correlation between the true exponent (fitted to the entire dataset) and the average exponent estimates from SDE. We

[3] Another common adaptive method is the *generalized k-th nearest neighbor method*, which adapts the bandwidth to the distance of the point whose density is to be estimated to its k-th nearest neighbor. In our experiments, we found variable kernel estimation to perform better.

[4] We have found that using the de-duplicated set U of unique student work to compute the densities empirically yields better performance than using the possibly redundant collection W of all student work.

[5] Technically, f might not be a normalized probability density over a vector space depending on the choice of metric d. For the purposes of Zipf inference challenge, however, the density estimate need not be normalized.

[6] Ties in multiplicity were broken in a way that results in smaller L1 distance.

[7] We use log densities to mitigate for heavy-tailedness.

[8] Similar to SDE's Zipf Order Inference method, the duplicate items were first ordered according to their observed multiplicity, followed by the "singletons" sorted by each method. Ties were also broken in the same manner.

[9] This was done to avoid the curse of dimensionality.

[10] In order to use pre-trained semantic encoders and compute token edit distance in our experiments, the block codes in code.org datasets were first statically translated into Python syntax.

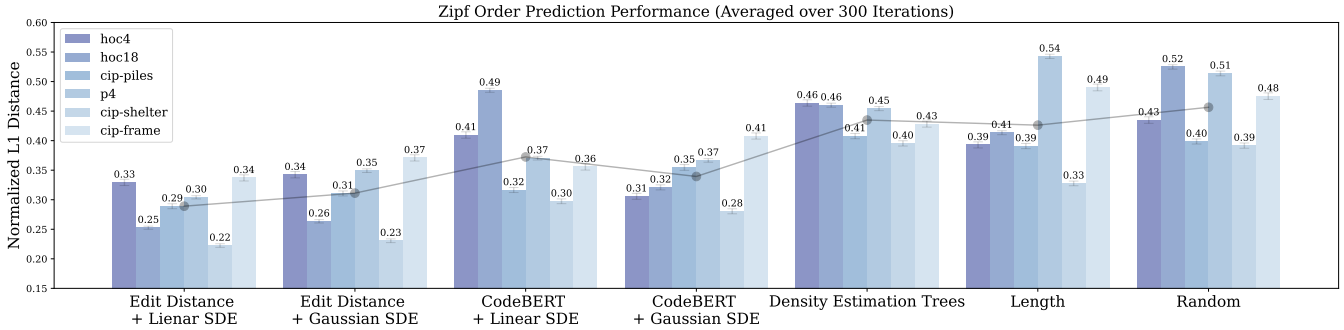

Figure 6: Average normalized ℓ_1 for Zipf Order Prediction for each method and dataset. Error bars indicate the error of the mean of ℓ_1 distance, and black dots are averages across datasets. (Small ℓ_1 distance means good performance.) SDE achieves up to 42% decrease in average ℓ_1 distance.

experimented with CodeBERT[11] and Edit Distance as the semantic distance metric, each with Gaussian and Linear kernels[12].

Zipf Inference Results. Figure 6 plots the Zipf Order Prediction performance for each method on different datasets. The datasets are shown in decreasing order of the Zipfian exponent α, starting from hoc4 with the largest α to cip-frame with the smallest. SDE outperforms all baseline methods, and among all SDE variants, Edit Distance with linear kernel achieves the best overall performance. In particular, the differences in normalized L1 distance between edit distance-based SDE and all baseline methods are statistically significant for all datasets (p-value $\ll .0001$), with SDE achieving from 15% (in hoc4) up to 42% (in hoc18) decrease in normalized ℓ_1 distance compared to the best performing baselines.

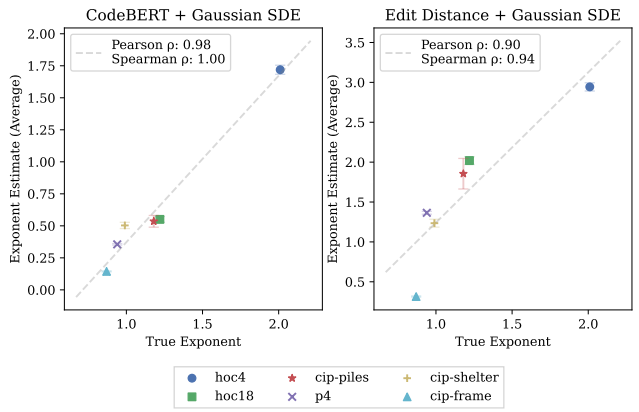

Figure 7: Average Zipf Exponent estimates against true Zipf exponents. Error bars indicate standard error of the mean.

Turning to Zipf Exponent Estimation, Figure 7 plots the true Zipf exponent α against the average of the estimates given by the best performing configuration of SDE for each semantic distance metric[13]. The average estimates of both Edit Distance and CodeBERT are highly correlated (0.90 and 0.98 Pearson correlation) with the true exponents, but CodeBERT has a particularly strong correlation and smaller variation in the estimates.

While the exponent estimates of CodeBERT SDE were overall better correlated and less noisy than the estimates from Edit Distance SDE, Edit Distance SDE consistently outperformed CodeBERT SDE in Zipf Order Prediction for all of the datasets except hoc4. We believe this is due the fact that CodeBERT was trained on a vastly large space of general-purpose programs hosted on GitHub [17]. In this space, the set of response codes written for a single assignment only comprises a *minuscule* subspace, and CodeBERT is likely not nuanced enough to tell the granular difference in semantics between a pair of student submitted codes for this reason. We believe that a semantic distance metric that better models the similarities at the level of the student's cognitive problem-solving process will achieve better Zipf Order Prediction.

7 TOWARDS A THEORETICAL EXPLANATION OF ZIPF'S LAW IN STUDENT WORK

Why do Zipf-like patterns emerge in open-ended student works? Although open-ended tasks involve a complex process that cannot fit into a single generative story, studying the mechanism behind Zipf-like structures may help researchers reveal valuable insights about the student thought process. Studies on Zipf's law provide plausible accounts on how Zipfian structures can emerge in various environments (see Section 9), but to the authors' knowledge, none of them are applicable to education and student problem-solving. Will take the first elementary step in this direction by drawing insights from the "student decision process" discussed in Section 5.

Recall from Section 5 that a student response can be mapped to a hierarchical set of decisions. For the purposes of analyzing the Zipfian patterns, let us consider a vastly simple version of this model where a student works through a problem that involves

[11]To obtain the embeddings, we concatenated the natural language description of the problem and the student-generated response (separated by the [SEP] token), and took the embeddings of the [CLS] token as the vector representation of each submission. We then normalized the embeddings of the responses in the sample to have zero-mean and unit-norm.

[12]For each of these 4 configurations, we tried $k \in \{5, 10\}$ for the neighbhorhood size in Equation 5 and chose the value of k with better performance.

[13]Linear kernel was observed to perform comparably to Gaussian kernel for both metrics.

T decisions to be made, each having 1 correct choice and $M-1$ incorrect choices. The t-th decision, represented by the random variable X_t, can take on one of the values in $\{0, ..., M-1\}$. Let us denote $X_t = 0$ to be the "correct" choice for the t-th decision, and values 1 through $M-1$ to indicate the possible misconceptions.

Importantly, let us further assume that each student possesses a latent *ability* value $\beta \in [0, 1]$. At each step t, the student chooses the correct choice with probability β. Otherwise, one of the incorrect choices is chosen from a distribution π_t. Then, conditioned on β, X_t is a categorical random variable with distribution P_t defined as:

$$X_t|\beta \sim P_t \equiv \text{Categorical}\left(\beta, (1-\beta)\pi_{t,1}, ..., (1-\beta)\pi_{t,M-1}\right),$$

Finally, we will model a population of students with varying levels of ability by assuming β has a uniform distribution

$$\beta \sim \text{Uniform}([\varepsilon, 1-\varepsilon]), \tag{8}$$

where ε is set to avoid degenerate probabilities. Putting everything together, the probability of observing a specific decision trajectory $x = (x_1, ..., x_T)$ can be expressed as

$$P(x) = \int P(\beta) \left(\prod_{t=1}^{T} P_t(x_t|\beta) \right) d\beta$$

We will call this model the **Varied Ability Student Model (VASM)**. VASM is identical to a uniform mixture of individual **Fixed Ability Student Models (FASM)**, each of which has a fixed ability β that is constant across students.

7.1 Do Varied Ability Student Models Exhibit Zipf-like Patterns?

A theoretical result from statistical physics [32, 43] states that an exponential family latent variable model under certain conditions[14] are exactly Zipfian with $\alpha = 1$ in the infinite dimensional limit. Although VASM is an exponential family latent variable model, it does not satisfy all the conditions in [32, 43]. Yet, even in the finite-length regime, Zipf-like sub-structures do emerge in VASM. Figure 8 shows the rank-frequency plot for FASMs with $N = 10$ and $M = 5$ for a varying range of ability values, and the same plot for the corresponding VASM[15]. Since the exact rank-probability is intractable, we sampled 1 billion samples from each model and used the rank-order of relative frequencies as a surrogate to the true probability ordering.

It is noticeable that even when each individual fixed ability model doesn't exhibit Zipf-like structures, VASM has an extensive range of ranks for which rank-frequency is closer to a Zipf relationship. The effect of "mixing abilities" becomes more apparent when we look at the strength of the Zipfian patterns for varying lengths of the generated trajectories. Figure 9 shows the deviation from Zipf's law in the body of the distribution[16] for fixed and varied ability models. To isolate the effect of mixture, we simulated FASM for 7 identically-spaced ability values (0.2,...,0.8), and compared them against the version of VASM in which the ability was uniform over those 7 values. Deviation was measured by the normalized

[14]The natural parameters are the latent variables themselves.
[15]Each π_t was sampled from a uniform distribution over the set of probability distributions over $(M-2)$ elements (the $(M-2)$-dimensional probability simplex).
[16]We took this to be from rank 10^3 up to the rank at which we observed fewer than 10 occurrences.

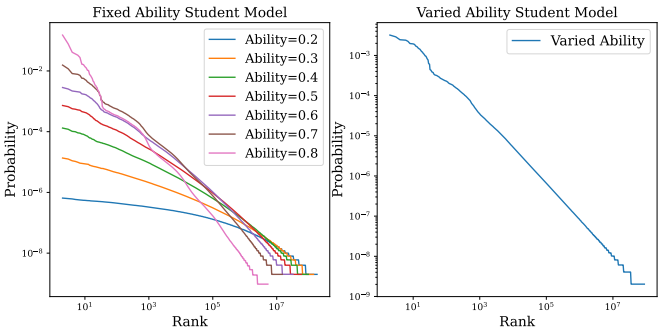

Figure 8: Rank-frequency of (Left) a fixed ability model for different ability values and (Right) Rank a varied ability model of the same size ($N = 10, M = 5$). The varied ability model has a wider range of power-law-like regions than any of the fixed ability models.

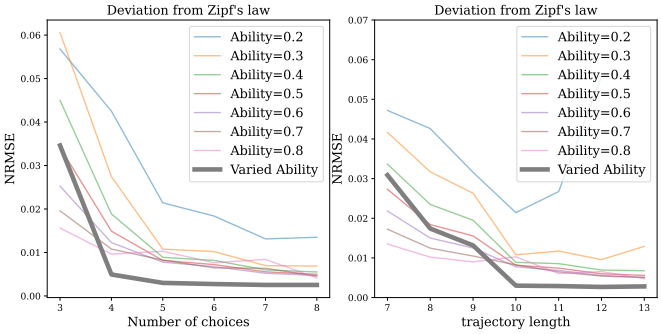

Figure 9: (Left) NRMSE for a least-squares log-rank/log-frequency plot for fixed ability models and (Right) a varied ability model for varying lengths of trajectories. $N = 10$ and $M = 5$. NRMSE for the varied ability model decreases most rapidly in small-length regions.

root-mean-squared error (NRMSE) for the least squares linear fit between log-rank and log-frequency, which allows the Zipfian fit to be compared across distributions with different log probability scales. The error of VASM is smaller than all FASMs for $M \geq 4$ and $T \geq 10$, suggesting that mixing abilities strengthened the Zipf-like patterns that emerge even for bounded-length decision sequences.

The varied ability student model is by no means a sufficient account of how Zipfian patterns emerge in student work, but it clearly suggests one possibility: a generative model that mixes over distinct *cohorts* of students (which were based on ability in our analysis) could strengthen the Zipfian patterns from each individual cohort. This possibility is further supported by similar empirical observations made in [1] where a mixture model was shown to give rise to Zipfian patterns in modelling multidimensional binary neural data.

8 DISCUSSION

Testing for Power-Law. In the past decade or so, several notable arguments[12, 15] have been made advocating for the use of maximum likelihood estimators (MLE) and bootstrapped Monte-Carlo

tests using Kolmogorov-Smirnov statistics to test for Goodness-of-Fit (GoF) to power law. Since then, several studies have used this test to demonstrate power-law fit to different datasets. Independent of these works, we have used the coefficient of determination (R^2) of a linear fit between empirical log-rank and empirical log-probability to measure how closely Zipf's law can approximate our empirical distribution in our study. We explain how our analysis differs from these works, and why the methods we employed are more suitable for our study.

First and foremost, it must be emphasized that our objective was *not* to test whether student responses *exactly follow* Zipf's Law, which is a rather strong statistical claim that the underlying distribution has a specific form[17]. Instead, our goal was to demonstrate that an *approximation* to Zipf's law is reasonable, and that this approximation can yield an insightful summary of the student response landscape to be taken up by educators. Significantly large R^2 (> 0.96) values for all 9 datasets were sufficient evidence for the quality of the approximation to this end.

Moreover, while several principled methods exist for fitting power-law distributions to data, the equivalent for fitting Zipf's law does not. This is because, although Zipf's law is a form of power-law between rank and frequency, the observed rank is *correlated* with the observed frequencies and can vary by the specific sample drawn. For this reason, MLE for power-law distributions in [12] or [15] have recently been shown to yield biased estimates [38] for Zipf's law, but estimators developed henceforth are still known to give unreliable estimates on real-world data [38]. While this has led us to opt for the most intuitive procedure, advancements in the quality of the Zipf exponent estimators will lead to a more rigorous fitting process in future research.

Outside the Zipfian range. In more than half of the assignments in our dataset, the fitted Zipf exponents were less than 1. Theoretically, Zipf exponent cannot be less than 1 unless Zipf's law holds for only a finite region. This means that, while the Zipfian range for such assignments can still be arbitrarily large, this range has to be finite and the probability of the remaining ranks will be closer to exponential decay than Zipf's law. Where the Zipfian pattern begins, where it ends, and how the responses are distributed outside the Zipfian range are interesting topics for future research.

Also, both in the real-student datasets and in the simulation of the Varied Ability Student Model, the high-probability "head" of the distribution had quite visibly different shapes from the rest of the distribution. Explaining and characterizing this difference remains an open question.

9 RELATED WORK

Zipf's law. Many natural phenomena obey Zipf's law, and decades-old studies have provided explanations for them in network structures [3, 8, 21], biological evolution [54], distribution of income [11], word frequencies in natural language [25, 47], and city populations [47]. We refer the reader to [31] for an excellent overview. Yet, none of the aforementioned models can be adapted to yield a satisfying account for the Zipfian patterns in student work, which is inherently a *hierarchical* decision process.

[17] The authors have found no evidence that student responses exactly follow Zipf's law.

Certain sequence models have also been proven to give rise to Zipf's law. The simplest and the most well-known is the distribution of space-delimited words generated from a randomly typed keyboard [7, 13, 30]. These results have also been further generalized to random trajectories on homogeneous finite-state Markov chains [6]. Yet, these models are too simple to be applicable to student work, which are results of a more complex, heterogeneous process. To the authors' knowledge, no explanation exists for the emergence of Zipf's law in cognitive models of thinking.

In the domain of statistical physics, [43] recently discovered that multivariate latent variable exponential family models with priors over the natural parameters converge to Zipf's law with $\alpha = 1$ in the limit of infinite-dimensional observations. Similar result also holds in finite dimensions when the range of frequencies is broad and the conditional distributions for each fixed latent variable is pairwise disjoint [1]. [1] also shows an empirical example of neural data where the disjointness assumption may not hold but Zipf's law does.

Patterns in large-scale student work. The initial observation of Zipfian patterns in large-scale student work was made in [33], which observed that constituent parts of code submissions ("code phrases") in a massive online course obey Zipf's law. [37] further observed that student-submitted programs in some massive programming classrooms appeared to follow Zipf's law. We further observe that the Zipf observation in student-constructed responses is generalizable across subjects and assignment types, and explain how these patterns can be inferred and used in education.

Previous studies have analyzed different constructive patterns within the space of student solutions [36, 41]. Recently, [24, 53] developed methods for mapping student misconceptions to their corresponding response outputs in order to generate synthetic, rubric-annotated responses to train an auto-grader.

Computer-assisted pedagogical interventions. Most advances in computer-assisted pedagogical interventions for open-ended questions have focused on efficient feedback generation and scalable grading. Subjects of these studies include short answers [9, 23], essays [44, 45], math problems [16, 22], and programming [24, 53].

Several works have addressed the problem of early predicting academic performance of students [29, 52] and promptly detecting students in need of pedagogical attention [2, 14, 26, 51, 54]. These methods use patterns in students' study behaviors and learning interactions over the span of the course.

Studies have suggested that structured organization of student submissions can greatly enhance the quality and efficiency of grading. For instance, grouping has been applied in the context of grading multiple choice questions [48], short answer questions [4], and propositional logic [28]. [19] claims that quality of grading student submissions could increase when they are organized by similarity. Similarly, we believe organizing student work by their underlying probabilities could lead to highly efficient grading and feedback.

10 CONCLUSION

We observed that, across different subjects and assignment types, the underlying probability distribution of open-ended student work

can be well-approximated by Zipf's law. We explained how inferring this latent structure in typical classrooms can practically benefit learning analytics researchers, educators, and instruction designers. We then explained why it is difficult to infer these latent structures in typical classrooms, formalized the novel "Zipf Inference Challenge" to address it, and designed the first inference method called "Semantic Density Estimation" using the notion of density in semantic distance space. We also discussed a potential theoretical cause of the student Zipf pattern.

The amount of open-ended student artifacts available to the learning analytics community and the power of analytics tools are both growing at an unprecedented pace. Studying the patterns emergent in these massive bodies of student artifacts will increasingly yield actionable insights about the mechanism behind student work, which could lead to tangible impacts in everyday classrooms. We hope our work further ignites research in this exciting direction.

ACKNOWLEDGMENTS

The authors would like to thank Stanford Institute for Human-Centered AI (Hoffman-Yee Research Grant) and Kwanjeong Educational Foundation for their generous support.

REFERENCES

[1] Laurence Aitchison, Nicola Corradi, and Peter E Latham. 2016. Zipf's law arises naturally when there are underlying, unobserved variables. *PLoS computational biology* 12, 12 (2016), e1005110.

[2] Gökhan Akçapınar, Mohammad Nehal Hasnine, Rwitajit Majumdar, Brendan Flanagan, and Hiroaki Ogata. 2019. Developing an early-warning system for spotting at-risk students by using eBook interaction logs. *Smart Learning Environments* 6, 1 (2019), 1–15.

[3] Albert-László Barabási, Réka Albert, and Hawoong Jeong. 1999. Mean-field theory for scale-free random networks. *Physica A: Statistical Mechanics and its Applications* 272, 1-2 (1999), 173–187.

[4] Sumit Basu, Chuck Jacobs, and Lucy Vanderwende. 2013. Powergrading: a clustering approach to amplify human effort for short answer grading. *Transactions of the Association for Computational Linguistics* 1 (2013), 391–402.

[5] Menucha Birenbaum and Kikumi K Tatsuoka. 1987. Open-ended versus multiple-choice response formats—it does make a difference for diagnostic purposes. *Applied Psychological Measurement* 11, 4 (1987), 385–395.

[6] Vladimir V Bochkarev and Eduard Yu Lerner. 2012. Zipf and non-Zipf laws for homogeneous Markov chain. *arXiv preprint arXiv:1207.1872* (2012).

[7] Vladimir V Bochkarev and Eduard Yu Lerner. 2016. The exact power law and Pascal pyramid. *arXiv preprint arXiv:1605.09052* (2016).

[8] Andrei Broder, Ravi Kumar, Farzin Maghoul, Prabhakar Raghavan, Sridhar Rajagopalan, Raymie Stata, Andrew Tomkins, and Janet Wiener. 2000. Graph structure in the web. *Computer networks* 33, 1-6 (2000), 309–320.

[9] Michael Brooks, Sumit Basu, Charles Jacobs, and Lucy Vanderwende. 2014. Divide and correct: using clusters to grade short answers at scale. In *Proceedings of the first ACM conference on Learning@ scale conference*. 89–98.

[10] John Seely Brown and Kurt VanLehn. 1980. Repair theory: A generative theory of bugs in procedural skills. *Cognitive science* 4, 4 (1980), 379–426.

[11] David G Champernowne. 1953. A model of income distribution. *The Economic Journal* 63, 250 (1953), 318–351.

[12] Aaron Clauset, Cosma Rohilla Shalizi, and Mark EJ Newman. 2009. Power-law distributions in empirical data. *SIAM review* 51, 4 (2009), 661–703.

[13] Brian Conrad and Michael Mitzenmacher. 2004. Power laws for monkeys typing randomly: the case of unequal probabilities. *IEEE Transactions on information theory* 50, 7 (2004), 1403–1414.

[14] Evandro B Costa, Baldoino Fonseca, Marcelo Almeida Santana, Fabrísia Ferreira de Araújo, and Joilson Rego. 2017. Evaluating the effectiveness of educational data mining techniques for early prediction of students' academic failure in introductory programming courses. *Computers in human behavior* 73 (2017), 247–256.

[15] Anna Deluca and Álvaro Corral. 2013. Fitting and goodness-of-fit test of non-truncated and truncated power-law distributions. *Acta Geophysica* 61, 6 (2013), 1351–1394.

[16] John A Erickson, Anthony F Botelho, Steven McAteer, Ashvini Varatharaj, and Neil T Heffernan. 2020. The automated grading of student open responses in mathematics. In *Proceedings of the Tenth International Conference on Learning Analytics & Knowledge*. 615–624.

[17] Zhangyin Feng, Daya Guo, Duyu Tang, Nan Duan, Xiaocheng Feng, Ming Gong, Linjun Shou, Bing Qin, Ting Liu, Daxin Jiang, et al. 2020. CodeBERT: A Pre-Trained Model for Programming and Natural Languages. In *Findings of the Association for Computational Linguistics: EMNLP 2020*. 1536–1547.

[18] Linda Flower and John R Hayes. 1981. A cognitive process theory of writing. *College composition and communication* 32, 4 (1981), 365–387.

[19] Sonja Johnson-Yu, Nicholas Bowman, Mehran Sahami, and Chris Piech. [n.d.]. SimGrade: Using Code Similarity Measures for More Accurate Human Grading. ([n. d.]).

[20] William L Kuechler and Mark G Simkin. 2010. Why is performance on multiple-choice tests and constructed-response tests not more closely related? Theory and an empirical test. *Decision Sciences Journal of Innovative Education* 8, 1 (2010), 55–73.

[21] Ravi Kumar, Prabhakar Raghavan, Sridhar Rajagopalan, D Sivakumar, Andrew Tomkins, and Eli Upfal. 2000. Stochastic models for the web graph. In *Proceedings 41st Annual Symposium on Foundations of Computer Science*. IEEE, 57–65.

[22] Andrew S Lan, Divyanshu Vats, Andrew E Waters, and Richard G Baraniuk. 2015. Mathematical language processing: Automatic grading and feedback for open response mathematical questions. In *Proceedings of the second (2015) ACM conference on learning@ scale*. 167–176.

[23] Claudia Leacock and Martin Chodorow. 2003. C-rater: Automated scoring of short-answer questions. *Computers and the Humanities* 37, 4 (2003), 389–405.

[24] Ali Malik, Mike Wu, Vrinda Vasavada, Jinpeng Song, John Mitchell, Noah Goodman, and Chris Piech. 2019. Generative Grading: Neural Approximate Parsing for Automated Student Feedback. *arXiv preprint arXiv:1905.09916* (2019).

[25] Benoit B Mandelbrot. 2013. *Fractals and scaling in finance: Discontinuity, concentration, risk. Selecta volume E*. Springer Science & Business Media.

[26] Farshid Marbouti, Heidi A Diefes-Dux, and Krishna Madhavan. 2016. Models for early prediction of at-risk students in a course using standards-based grading. *Computers & Education* 103 (2016), 1–15.

[27] John Mason. 2002. *Researching your own practice: The discipline of noticing*. Routledge.

[28] Agathe Merceron and Kalina Yacef. 2004. Clustering students to help evaluate learning. In *IFIP World Computer Congress, TC 3*. Springer, 31–42.

[29] Vera L Miguéis, Ana Freitas, Paulo JV Garcia, and André Silva. 2018. Early segmentation of students according to their academic performance: A predictive modelling approach. *Decision Support Systems* 115 (2018), 36–51.

[30] George A Miller. 1957. Some effects of intermittent silence. *The American journal of psychology* 70, 2 (1957), 311–314.

[31] Michael Mitzenmacher. 2004. A brief history of generative models for power law and lognormal distributions. *Internet mathematics* 1, 2 (2004), 226–251.

[32] Thierry Mora and William Bialek. 2011. Are biological systems poised at criticality? *Journal of Statistical Physics* 144, 2 (2011), 268–302.

[33] Andy Nguyen, Christopher Piech, Jonathan Huang, and Leonidas Guibas. 2014. Codewebs: scalable homework search for massive open online programming courses. In *Proceedings of the 23rd international conference on World wide web*. 491–502.

[34] National Council of Teachers of Mathematics. 2014. Principles to Actions: Ensuring Mathematical Success for All, Author.

[35] Christopher Piech, Ali Malik, Kylie Jue, and Mehran Sahami. 2021. Code in Place: Online Section Leading for Scalable Human-Centered Learning. In *Proceedings of the 52nd ACM Technical Symposium on Computer Science Education*. 973–979.

[36] Chris Piech, Mehran Sahami, Daphne Koller, Steve Cooper, and Paulo Blikstein. 2012. Modeling how students learn to program. In *Proceedings of the 43rd ACM technical symposium on Computer Science Education*. 153–160.

[37] Christopher James Piech. 2016. *Uncovering patterns in student work: Machine learning to understand human learning*. Stanford University.

[38] Charlie Pilgrim and Thomas T Hills. 2021. Bias in Zipf's law estimators. *Scientific reports* 11, 1 (2021), 1–11.

[39] Parikshit Ram and Alexander G Gray. 2011. Density estimation trees. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. 627–635.

[40] Brian Riordan, Andrea Horbach, Aoife Cahill, Torsten Zesch, and Chungmin Lee. 2017. Investigating neural architectures for short answer scoring. In *Proceedings of the 12th workshop on innovative use of NLP for building educational applications*. 159–168.

[41] Kelly Rivers and Kenneth R Koedinger. 2014. Automating hint generation with solution space path construction. In *International Conference on Intelligent Tutoring Systems*. Springer, 329–339.

[42] Dale H Schunk. 2012. *Learning theories an educational perspective sixth edition*. pearson.

[43] David J Schwab, Ilya Nemenman, and Pankaj Mehta. 2014. Zipf's law and criticality in multivariate data without fine-tuning. *Physical review letters* 113, 6 (2014), 068102.

[44] Mark D Shermis and Jill Burstein. 2013. Handbook of automated essay evaluation. *NY: Routledge* (2013).

[45] Mark D Shermis and Ben Hamner. 2012. Contrasting state-of-the-art automated scoring of essays: Analysis. In *Annual national council on measurement in education meeting*. National Council on Measurement in Education Vancouver, BC, Canada, 14–16.

[46] Bernard W Silverman. 1986. *Density Estimation for Statistics and Data Analysis*. Vol. 26. CRC Press.

[47] Herbert A Simon. 1955. On a class of skew distribution functions. *Biometrika* 42, 3/4 (1955), 425–440.

[48] Arjun Singh, Sergey Karayev, Kevin Gutowski, and Pieter Abbeel. 2017. Gradescope: a fast, flexible, and fair system for scalable assessment of handwritten work. In *Proceedings of the fourth (2017) acm conference on learning@ scale*. 81–88.

[49] Margaret S Smith and Mary Kay Stein. 2018. 5 Practices for Orchestrating Productive Mathematics Discussions. In *5 Practices for Orchestrating Productive Mathematics Discussions*. The National Council of Teachers of Mathematics, Inc.

[50] Kurt VanLehn. 1982. Bugs are not enough: Empirical studies of bugs, impasses and repairs in procedural skills. *The Journal of Mathematical Behavior* (1982).

[51] Carlos J Villagrá-Arnedo, Francisco J Gallego-Durán, Faraón Llorens-Largo, Patricia Compañ-Rosique, Rosana Satorre-Cuerda, and Rafael Molina-Carmona. 2017. Improving the expressiveness of black-box models for predicting student performance. *Computers in Human Behavior* 72 (2017), 621–631.

[52] Hajra Waheed, Saeed-Ul Hassan, Naif Radi Aljohani, Julie Hardman, Salem Alelyani, and Raheel Nawaz. 2020. Predicting academic performance of students from VLE big data using deep learning models. *Computers in Human behavior* 104 (2020), 106189.

[53] Mike Wu, Milan Mosse, Noah Goodman, and Chris Piech. 2019. Zero shot learning for code education: Rubric sampling with deep learning inference. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 33. 782–790.

[54] George Udny Yule. 1925. II.—A mathematical theory of evolution, based on the conclusions of Dr. JC Willis, FR S. *Philosophical transactions of the Royal Society of London. Series B, containing papers of a biological character* 213, 402-410 (1925), 21–87.