Exemplar Guided Active Learning

Jason Hartford
AI21 Labs
jasonh@cs.ubc.ca

Kevin Leyton-Brown
AI21 Labs
kevinlb@cs.ubc.ca

Hadas Raviv
AI21 Labs
hadasr@ai21.com

Dan Padnos
AI21 Labs
danp@ai21.com

Shahar Lev
AI21 Labs
shaharl@ai21.com

Barak Lenz
AI21 Labs
barakl@ai21.com

Abstract

We consider the problem of wisely using a limited budget to label a small subset of a large unlabeled dataset. We are motivated by the NLP problem of word sense disambiguation. For any word, we have a set of candidate labels from a knowledge base, but the label set is not necessarily representative of what occurs in the data: there may exist labels in the knowledge base that very rarely occur in the corpus because the sense is rare in modern English; and conversely there may exist true labels that do not exist in our knowledge base. Our aim is to obtain a classifier that performs as well as possible on examples of each "common class" that occurs with frequency above a given threshold in the unlabeled set while annotating as few examples as possible from "rare classes" whose labels occur with less than this frequency. The challenge is that we are not informed which labels are common and which are rare, and the true label distribution may exhibit extreme skew. We describe an active learning approach that (1) explicitly searches for rare classes by leveraging the contextual embedding spaces provided by modern language models, and (2) incorporates a stopping rule that ignores classes once we prove that they occur below our target threshold with high probability. We prove that our algorithm only costs logarithmically more than a hypothetical approach that knows all true label frequencies and show experimentally that incorporating automated search can significantly reduce the number of samples needed to reach target accuracy levels.

1 Introduction

We are motivated by the problem of labelling a dataset for word sense disambiguation, where we want to use a limited budget to collect annotations for a reasonable number of examples of each sense for each word. This task can be thought of as an active learning problem (Settles, 2012), but with two nonstandard challenges. First, for any given word we can get a set of candidate labels from a knowledge base such as WordNet (Fellbaum, 1998). However, this label set is not necessarily representative of what occurs in the data: there may exist labels in the knowledge base that do not occur in the corpus because the sense is rare in modern English; conversely, there may also exist true labels that do not exist in our knowledge base. For example, consider the word "bass." It is frequently used as a noun or modifier, e.g., "the *bass* and alto are good singers", or "I play the *bass* guitar". It is also commonly used to refer to a type of fish, but because music is so widely discussed online, the fish sense of the word is orders of magnitude less common than the low-frequency sound sense in internet text. The Oxford dictionary (Lexico) also notes that bass once referred to a fibrous material used in matting or chords, but that sense is not common in modern English. We want a method that collects balanced labels for the common senses, "*bass* frequencies" and "*bass* fish", and ignores sufficiently rare senses, such as "fibrous material". Second, the empirical distribution of the

34th Conference on Neural Information Processing Systems (NeurIPS 2020), Vancouver, Canada.

true labels may exhibit extreme skew: word sense usage is often power-law distributed (McCarthy et al., 2007) with frequent senses occurring orders of magnitudes more often than rare senses.

When considered individually, neither of these constraints is incompatible with existing active learning approaches: incomplete label sets do not pose a problem for any method that relies on classifier uncertainty for exploration (new classes are simply added to the classifier as they are discovered); and extreme skew in label distributions has been studied under the guided learning framework wherein annotators are asked to explicitly search for examples of rare classes rather than simply label examples presented by the system (Attenberg & Provost, 2010). But taken together, these constraints make standard approaches impractical. Search-based ideas from guided learning are far more sample efficient with a skewed label distribution, but they require both a mechanism through which annotators can search for examples and a correct label set because it is undesirable to ask annotators to find examples that do not actually occur in a corpus.

Our approach is as follows. We introduce a frequency threshold, γ, below which a sense will be deemed to be "sufficiently rare" to be ignored (i.e. for sense y, if $P(Y = y) = p_y < \gamma$ the sense is rare); otherwise it is a "common" sense of the word for which we want a balanced labeling with other common senses. Of course, we do not know p_y, so it must be estimated online. We do this by providing a stopping rule that stops searching for a given sense when we can show with high probability that it is sufficiently rare in the corpus. We automate the search for rare senses by leveraging the high-quality feature spaces provided by modern self-supervised learning approaches (Devlin et al., 2018; Radford et al., 2019; Raffel et al., 2019). We leverage the fact that one typically has access to a single example usage of each word sense[1], which enables us to search for more examples of a sense in a local neighborhood of the embedding space. This allows us to develop a hybrid guided and active learning approach that automates the guided learning search procedure. Automating the search procedure makes the method cheaper (because annotators do not have to explicitly search) and allows us to maintain an estimate of $\hat{p}_y$ by using importance-weighted samples. Once we have found examples of common classes, we switch to more standard active learning methods to find additional examples to reduce classifier uncertainty.

Overall, this paper makes two key contributions. First, we present an Exemplar Guided Active Learning (EGAL) algorithm that offers strong empirical performance under extremely skewed label distributions by leveraging exemplar embeddings. Second, we identify a stopping rule that makes EGAL robust to misspecified label sets and prove that this robustness only imposes a logarithmic cost over a hypothetical approach that knows the correct label set. Beyond these key contributions, we also present a new Reddit word sense disambiguation dataset, which is designed to evaluate active learning methods for highly skewed label distributions.

2 Related Work

Active learning under class imbalance The decision-boundary-seeking behavior of standard active learning methods which are driven by classifier uncertainty has a class balancing effect under moderately skew data (Attenberg & Ertekin, 2013). But, under extreme class imbalance, these methods may exhaust their labeling budgets before they ever encounter a single example of the rare classes. This issue is caused by an epistemic problem: the methods are driven by classifier uncertainty, but standard classifiers cannot be uncertain about classes that they have never observed. Guided learning methods (Attenberg & Provost, 2010) address this by assuming that annotators can explicitly search for rare classes using a search engine (or some other external mechanism). Search may be more expensive than annotation, but the tradeoff is worthwhile under sufficient class imbalance. However, explicit search is not realistic in our setting: search engines do not provide a mechanism for searching for a particular sense of a word and we care about recovering all classes that occur in our dataset with frequency above γ, so searching by sampling uniformly at random would require labelling $n \geq \mathcal{O}(\frac{1}{\gamma^2})$ samples[2] to find all such classes with high probability.

Active learning with extreme class imbalance has also been studied under the "active search" paradigm (Jiang et al., 2019) that seeks to find as many examples of the rare class as possible in a finite budget

[1] Example usages can be found in dictionaries or other lexical databases such as WordNet.

[2] For the probability of seeing at least one example to exceed $1 - \delta$, we need at least $n \geq \frac{\log 1/\delta}{\gamma^2}$ samples. See Lemma 3 for details.

of time, rather than minimizing classifier uncertainty. Our approach instead separates explicit search from uncertainty minimization in two different phases of the algorithm.

Active learning for word sense disambiguation Many authors have showed that active learning is a useful tool for collecting annotated examples for the word sense disambiguation task. Chen et al. (2006) showed that entropy and margin-based methods offer significant improvements over random sampling. To our knowledge, Zhu & Hovy (2007) were the first to discuss the practical aspects of highly skewed sense distributions and their effect on the active learning problems. They studied over- and under-sampling techniques which are useful once one has examples, but did not address the problem of finding initial points under extremely skewed distributions.

Zhu et al. (2008) and Dligach & Palmer (2011) respectively share the two key observations of our paper: good initializations lead to good active learning performance and language models are useful for providing a good initialization. Our work modernizes these earlier papers by leveraging recent advances in self-supervised learning. The strong generalization provided by large-scale pre-trained embeddings allow us to guide the initial search for rare classes with exemplar sentences which are not drawn from the training set. We also provide stopping rules that allow our approach to be run without the need to carefully select the target label set, which makes it practical run in an automated fashion.

Yuan et al. (2016) also leverage embeddings but they use label propagation to nearest neighbors in embedding space. This approach is similar to ours in that it also uses self-supervised learning, but we have access to ground truth through the labelling oracle which offers some protection against the possibility that senses are poorly clustered in embedding space.

Pre-trained representations for downstream NLP tasks There are a large number of recent papers showing that combining extremely large datasets with large Transformer models (Vaswani et al., 2017) and training them on simple sequence prediction objectives leads to contextual embeddings that are very useful for a variety of downstream tasks. In this paper we use contextual embeddings from BERT (Devlin et al., 2018) but because the only property we leverage is the fact that the contextual embeddings provide a useful notion of distance between word senses, the techniques described are compatible with any of the recent contextual models (e.g. Radford et al., 2019; Raffel et al., 2019).

3 Exemplar-guided active learning

We are given a large training set of unlabeled examples described by features (typically provided by an embedding function), $X_i \in \mathbb{R}^d$, and our task is to build a classifier, $f : \mathbb{R}^d \to \{1, \ldots, K\}$, that maps a given example to one of K classes. We are evaluated based on the accuracy of our trained classifiers on a *balanced* test set of the "common classes": those classes, y_i, that occur in our corpus with frequency, $p_{y_i} > \gamma$, where γ is a known threshold. Given access to some labelling oracle, $l : \mathbb{R}^d \to \{1, \ldots, K\}$, that can supply the true label of any given example at a fixed cost, we aim to spend our labelling budget on a set of training examples such that our resulting classifier minimizes the $0 - 1$ loss on the $k(\gamma) = \sum_i \mathbf{1}\left[p_{y_i} \geq \gamma\right]$ classes that exceed the threshold, $\mathcal{L} = \frac{1}{k(\gamma)} \sum_{i=1}^{K} \mathbf{1}\left[p_{y_i} \geq \gamma\right] \mathbf{E}_{X:l(X)=y_k}[\mathbf{1}(f(X) \neq y_k)]$.

That is, any label that occurs with probability at least γ in the observed data generating process will receive equal weight, $\frac{1}{k(\gamma)}$, in the test set and anything that occurs less frequently can be ignored. The task is challenging because rare classes (i.e. those which occur with frequency $\gamma < p_y \ll 1$) are unlikely to be found by random sampling, but still contribute a $\frac{1}{k(\gamma)}$ fraction of the overall loss.

Our approach leverages the guided learning insight that "search beats labelling under extreme skew", but automates the search procedure. We assume we have access to high-quality embeddings—such as those from a modern statistical language model—which gives us a way of measuring distances between word usage in different contexts. If these distances capture word senses and we have an example usage of the word for each sense, a natural search strategy is to examine the usage of the word in a neighborhood of the example sentence. This is the key idea behind our approach: we have a search phase where we sample the neighborhood of the exemplar embedding until we find at least one example of each word sense, followed by a more standard active learning phase where we seek examples that reduce classifier uncertainty.

```
Input : $\mathcal{D} = \{X_i\}_{i \in 1 \ldots n}$ a dataset of unlabeled examples
        $\phi : \text{domain}(X) \to \mathbb{R}^d$, $d : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}$ an embedding and distance function
        $l : X_i \to y_i$ a labeling operation (such as querying a human expert)
        $L$ : The total number of potential class labels
        $\gamma$ : the label-probability threshold
        $E$ the set of exemplars, $|E| = L$
        $B$ : a budget for maximum number of queries
        $b$ : batch size of queries sampled before the model is retrained

$\mathcal{A} \leftarrow \{1, \ldots, L\}$ # Set of active classes
$\mathcal{C} \leftarrow \emptyset$ # Set of completed classes
$\mathcal{D}^{(l)} \leftarrow \emptyset$ # Set of labeled examples
while $|\mathcal{D}^{(l)}| < B$ do
    $\mathcal{A}' = \mathcal{A}$ # $\mathcal{A}'$ is the target set of classes for the algorithm to find.
    while $\mathcal{A}' \neq \emptyset$ and number of collected samples $< b$ do
        Select random $i'$ from $\mathcal{A}'$ and set $\mathcal{A}' \leftarrow \mathcal{A}' \setminus \{i'\}$ and $X \leftarrow \emptyset$
        repeat
            $X \leftarrow X \cup \{x\}$ where $x$ is selected with exemplar $E_{i'}$ using either equation 1 or
              $\epsilon$-greedy sampling.
            $y \leftarrow \{l(x)$ for x in $X\}$ # Label each example in $X$
        until (Number of unique labels in $y = \lfloor b/L \rfloor$) or (Number of labeled samples = $b$);
        Update empirical means $\hat{p}_y$ and remove any classes with $\hat{p}_y + \sigma_y < \gamma$ from $\mathcal{A}$ and $\mathcal{A}'$
        $\mathcal{D}^{(l)} \leftarrow \mathcal{D}^{(l)} \cup (X, y)$
        $\mathcal{A} \leftarrow \{i \in \mathcal{A} : i$ not in unique labels in $D^{(l)}\}$ # Remove observed labels from the active
          set
    end
    Sample the remainder of the batch, $(X, y)$, using using either algorithm in equation 3
    $\mathcal{D}^{(l)} \leftarrow \mathcal{D}^{(l)} \cup (X, y)$
    Update empirical means $\hat{p}_y$ and remove any classes with $\hat{p}_y + \sigma_y < \gamma$ from $\mathcal{A}$
    $\mathcal{A} \leftarrow \{i \in \mathcal{A} : i$ not in unique labels in $D^{(l)}\}$
    Update classifier using $\mathcal{D}^{(l)}$.
end
```

Algorithm 1: EGAL: Exemplar Guided Active Learning

In the description below we denote the embedding vector associated with the target word in sentence, i, as x_i. For each sense, y, denote the embedding of an example usage as $\tilde{x}_y$. We assume that this example usage is selected from a dictionary or knowledge base so we do not include it in our classifier's training set. Full pseudocode is given in Algorithm 1.

Guided search Given an example embedding, $\tilde{x}_y$, we could search for similar usages of the word in our corpus by iterating over corpus examples, x_i, sorted by distance, $d_i = \|x_i - \tilde{x}_y\|_2$. However, using this approach does not give us a way of maintaining an estimate of $\hat{p}_y$, the empirical frequency of the word sense in the corpus which we need for our stopping rule that stops searching for classes that are unlikely to exist in the corpus. Instead we sample each example to label, x_i, from a Boltzmann distribution over unlabeled examples,

$$x_i : i \sim \text{Cat}(q = [q_1, \ldots, q_n]), \; q_i = \frac{\exp(-d_i/\lambda_y)}{\sum_i \exp(-d_i/\lambda_y)}, \tag{1}$$

where λ_y is a temperature hyper-parameter that controls the sharpness of q.

We sample with replacement and maintain a count vector c that tracks the number of times an example has been sampled. If an example has previously been sampled, labelling does not diminish our annotation budget because we can simply look up the label, but maintaining these counts lets us maintain an unbiased estimate of p_y, the empirical frequency of the sense label and gives a way of choosing λ_y, the length scale hyper-parameter, which we describe below. We continue drawing samples until we have a batch of b labeled examples.

Optimizing the length scale The sampling procedure selects examples to label in proportion to how similar they are to our example sentence, where similarity is measured by a square exponential kernel on the distance d_i. To use this, we need to choose a length scale, λ_y, which selects how to scale distances such that most of the weight is applied to examples that are close in embedding space. One challenge is that embedding spaces may vary across words and in density around different example sentences. If λ_y is set either too large or too small, one tends to sample few examples from the target class because for extreme values of λ, q either concentrates on a single example (and is uniform over the rest) or is uniform over all examples. We address this with a heuristic that automatically selects the length scale for each example sentence x_y. We choose λ that minimizes

$$\lambda_y = \text{arg min}_\lambda \mathbb{E}\left[\frac{1}{\sum_{i \in B} w_i^2}\right];\ w_i = \frac{c_i(x_y)}{\sum_{j \in B} c_j(x_y)}. \tag{2}$$

This score measures the effective sample size that results from sampling a batch of B examples for example sentence x_y. The score is minimized when λ is set such that as much probability mass as possible is placed on a small cluster of examples. This gives the desired sampling behavior of searching a tight neighborhood of the exemplar embedding. Because the expectation can be estimated using only unlabeled examples, we can optimize this by averaging the counts from multiple runs of the sampling procedure and finding the minimal score by binary search.

Active learning The second phase of our algorithm builds on standard active learning methods. Most active learning algorithms select unlabeled points to label, ranking them by an "informativeness" score for various notions of informativeness. The most widely used scores are the *uncertainty sampling* approaches, such as *entropy* sampling, which scores examples by s_{ENT}, the entropy of the classifier predictions, and the *least confidence* heuristic s_{LC}, which selects the unlabeled example about which the classifier is least confident. They are defined as

$$s_{\text{ENT}}(x) = -\sum_i P(y_i|x;\theta)\log P(y_i|x;\theta); \quad s_{\text{LC}}(x) = -\max_y P(y|x;\theta). \tag{3}$$

Typically examples are selected to maximize these scores, $x_i = \arg\max_{x \in \mathcal{X}_{\text{pool}}} s(x)$, but again we can sample from a distribution implied by the score function to select examples $x_i : i \sim$ Cat$(q' = [q'_1, \ldots, q'_n])$ and maintain an estimate of p_y in a manner analogous to Equation 2 as $q'_i = \frac{\exp(s_{LC}(x_i)/\lambda_y)}{\sum_i \exp(s_{LC}(x_i)/\lambda_y)}$,

ϵ-greedy sampling The length scale parameters for Boltzmann sampling can be tricky to tune when applied to the active learning scores because the scores vary over the duration of training. The means that we cannot use the optimization heuristic that we applied to the guided search distribution. A simple alternative to the Boltzmann distribution sampling procedure is to sample some $u \sim$ Uniform(0,1) and select a random example whenever $u \leq \epsilon$. ϵ-greedy sampling is far simpler to tune and analyze theoretically, but has the disadvantage that one can only use the random steps to update estimates of the class frequencies. We evaluate both methods in the experiments.

Stopping conditions For every sample we draw, we estimate the empirical frequencies of the senses. We continue to search for examples of each sense as long as an upper bound on the sense frequency exceeds our threshold γ. For each sense, the algorithm remains in "importance weighted search mode" as long as $\hat{p}_y + \sigma_y > \gamma$ and we have not yet found an example of sense y in the unlabeled pool. Once we have found an example of y, we stop searching for more examples and instead let further exploration be driven by classifier uncertainty.

Because any wasted exploration is costly in the active learning setting, a key consideration for the stopping rule is choosing the confidence bound to be as tight as possible while still maintaining the required probabilistic guarantees. We use two different confidence bounds for each of the sampling strategies. When sampling using the ϵ-greedy strategy, we know that the random variable, y, obtains values in $\{0, 1\}$ so we can get tight bounds on p_y using Chernoff's bound on Bernoulli random variables. We use the following implication of the Chernoff bound (see Lattimore & Szepesvári, 2020, chapter 10),

Lemma 1 (Chernoff bound)**.** *Let y_i be a sequence of Bernoulli random variables with parameter p_y, $\hat{p}_y = \frac{1}{n}\sum_{i=1}^n y_i$ and $KL(p,q) = p\log(p/q) + (1-p)\log((1-p)/(1-q))$. For any $\delta \in (0,1)$ we* *can define the upper and lower bounds as follows,*

$$U(\delta) = \max\{x \in [0,1] : KL(\hat{p}_y, x) \leq \frac{\log(1/\delta)}{n}\},\ L(\delta) = \min\{x \in [0,1] : KL(\hat{p}_y, x) \leq \frac{\log(1/\delta)}{n}\}$$

and we have that $\mathbb{P}\left[p_y \geq U(\delta)\right] \leq \delta$ *and* $\mathbb{P}\left[p_y \leq L(\delta)\right] \leq \delta$.

There do not exist closed-form expressions for these upper and lower bounds, but they are simple bounded 1D convex optimization procedures that can be solved efficiently using a variety of optimization methods. In practice we use Scipy's (Virtanen et al., 2020) implementation of Brent's method (Brent, 1973).

When using the importance weighted approach, we have to contend with the fact that the random variables implied by the importance-weighted samples are not bounded above. This leads to wide confidence bounds because the bounds have to account for the possibility of large changes to the mean that stem from unlikely draws of extreme values. When using importance sampling, we sample points according to some distribution q and we can maintain an unbiased estimate of p_y by computing a weighted average of the indicator function, $\mathbf{1}(y_i = y)$, where each observation is weighted by its inverse propensity, which implies importance weights $w_i = \frac{1/n}{q_i}$. The resulting random variable $z_i = w_i \mathbf{1}(y_i = y)$ has expected value equal to p_y, but can potentially take on values in the range $[0, \max_i \mathbf{1}(y_i = y)\frac{1/n}{q_i}]$. Because the distribution q has probability that is inversely proportional to distance, this range has a natural interpretation in terms of the quality of the embedding space: the largest z_i is the example from our target class that is furthest from our example embedding. If the embedding space does a poor job of clustering senses around the example embedding, then it is possible that the furthest point in embedding space—which will have tiny propensity because propensity decreases exponentially with distance—shares the same label as our target class, so our bounds have to account for this possibility.

There are two ways one could tighten these bounds: either make assumptions on the distribution of senses in embedding space that imply clustering around the example embedding, or modify the sampling strategy. We choose the latter approach: we can control the range of the importance weighted samples by enforcing a minimum value, α, on our sampling distribution q such that the resulting upper bound $\max_i \frac{1/n}{q_i} = \frac{\alpha^{-1}}{n}$. In practice this can be achieved by simply adding a small constant to each q_i and renormalizing the distribution. Furthermore, we note that when the embedding space is relatively well clustered, the true distribution of z will have far lower variance than the worst case implied by the bounds. We take advantage of this by computing our confidence intervals using Maurer & Pontil (2009)'s empirical Bernstein inequality which offers tighter bounds when the empirical variance is small.

Lemma 2 (Empirical Bernstein). *Let z_i be a sequence of i.i.d. bounded random variables on the range $[0, m]$ with expected value p_y, empirical mean $\bar{z} = \frac{1}{n}\sum_i z_i$, empirical variance $V_n(Z)$. For any $\delta \in (0,1)$ we have,*

$$\mathbb{P}\left[p_y \geq \bar{z} + \sqrt{\frac{m^2 2V_n(Z)\log(2/\delta)}{n}} + \frac{7m\log(2/\delta)}{3(n-1)}\right] \leq \delta$$

Proof. Let $z_i' = \frac{z_i}{m}$ such that it is bounded on $[0,1]$. Apply Theorem 4 of Maurer & Pontil (2009) to z_i'. The result follows by rearranging to express the theorem in terms of z_i. □

The bound is symmetric so the lower bound can be obtained by subtracting the interval. Note that the width of the bound depends linearly on the width of the range. This implies a practical trade-off: making the q distribution more uniform by increasing the probability of rare events leads to tighter bounds on the class frequency which rules out rare classes more quickly; but also costs more by selecting sub-optimal points more frequently.

Unknown classes One may also want to allow for the possibility of labelling unknown classes that do not appear in the dictionary or lexical database. We use the following approach. If during sampling we discover a new class, it is treated in the same manner as the known classes, so we maintain estimates of its empirical frequency and associated bounds. This lets us optimize for classifier uncertainty over the new class and remove it from consideration if at any point the upper bound on its frequency falls below our threshold (which may occur if the unknown class simply stems from an annotator misunderstanding).

For the stronger guarantee that with probability $1-\delta$ we have found all classes above the threshold, γ, we need to collect at least $n \geq \frac{\log 1/\delta}{\gamma^2}$ uniform at random samples.

Lemma 3. *For any $\delta \in (0,1)$, consider an algorithm which terminates if $\sum_i \mathbf{1}(y_i = k) = 0$ after $n \geq \frac{\log 1/\delta}{\gamma^2}$ draws from an i.i.d. categorical random variable, y, with support $\{1, \ldots, K\}$ and parameters $p_1, \ldots, p_K$. Let the "bad" event, $\{B_j = 1\}$, be the event that the algorithm terminates in experiment j with parameter $p_k > \gamma$. Then $\mathbb{P}[B_j = 1] \leq \delta$, where the probability is taken across all runs of the algorithm.*

Proof sketch. The result is a direct consequence of Hoeffding's inequality. □

A lower bound of this form is unavoidable without additional knowledge, so in practice we suggest using a larger threshold parameter γ' for computing n if one is only worried about 'frequent' unknown classes. When needed, we support these unknown class guarantees by continuing to use an $\epsilon-$greedy sampling strategy until we have collected at least $n(\delta, \gamma)$ uniform at random samples without encountering an unknown class.

Regret analysis Regret measures loss with respect to some baseline strategy, typically one that is endowed with oracle access to random variables that must be estimated in practice. Here we define our baseline strategy to be that of an active learning algorithm that knows in advance which of the classes fall below the threshold γ. This choice of adversary lets us focus our analysis on the affect of searching for thresholded classes.

Algorithm 1 employs the same strategy as the optimal agent during both the search and active learning phases, but may differ in the set of classes that it considers active: in particular, at the start of execution, it considers all classes active whereas the optimal strategy only regards all classes for which $p_{y_i} > \gamma$ as active. Because of this it will potentially remain in the exemplar guided search phase of execution for longer than the optimal strategy and hence the sub-optimality of Algorithm 1 will be a function of the number of samples it takes to rule out classes that fall below the threshold γ.

Let $\Delta = \min_i |p_i - \gamma|$ denote the smallest gap between p_i and γ. Assume the classifier's performance on class y can be described by some concave utility function, $U : \mathbb{R}^{n\times d} \times [K] \to \mathbb{R}$, that measures expected generalization as a function of the number of observations it receives. This amounts to assuming that standard generalization bounds hold for classifiers trained on samples derived from the active learning procedure.

Theorem 4. *Given some finite time horizon n implied by the labelling budget, if the utility derived from additional labeled examples for each class y can be described by some concave utility function, $U : \mathbb{R}^{n\times d} \times [K] \to \mathbb{R}$ and $\Delta = \min_i |p_{y_i} - \gamma| > 0$ then Algorithm 1 has regret at most, $\mathcal{R} \leq 1 + k(\gamma)\left[\frac{2\log(n)}{\Delta^2} + \frac{2+\Delta^2}{n\Delta^2}\right]$*

The proof leverages that fact that the difference in performance is bounded by differences in the number of times Algorithm 1 selects the unthresholded classes. Full details are given in the appendix.

4 Experiments

Our experiments are designed to test whether automated search with embeddings could find examples of very rare classes and to assess the effect of different skew ratios on performance.

Reddit word sense disambiguation dataset The standard practice for evaluating active learning methods is to take existing supervised learning datasets and counterfactually evaluate the performance that would have been achieved if the data had been labeled under the proposed active learning policy. While there do exist datasets for word sense disambiguation (e.g Yuan et al., 2016; Edmonds & Cotton, 2001; Mihalcea et al., 2004), they typically either have a large number of words with few examples per word or have too few examples to test the more extreme label skew that shows the benefits of guided learning approaches. To test the effect of a skew ratio of 1:200 with 50 examples

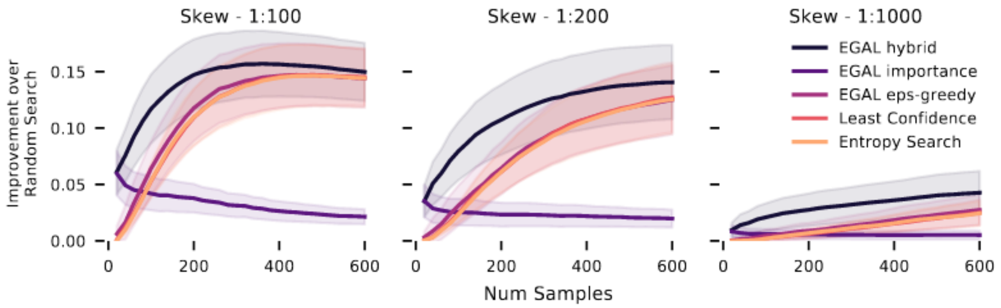

Figure 1: Average accuracy improvement over random search for all 21 words at different levels of skew. With lower levels of skew (*left*), EGAL tends to give big improvments over random search quickly as the examplars make it relatively easy to find examples of the rare classes. With larger amounts of skew (*left*), it takes longer on average before the uncertainty driven methods find examples of the rare class, so the performance difference with EGAL remains large for longer. Once skew becomes sufficiently large (*right*), EGAL still offers some benefit, but the absolute gains are smaller as the rare classes are suffiently rare that they are hard to find even with an exemplar.

of the rare class, one would need 10 000 examples of the common class; more extreme skew would require correspondingly larger datasets. The relatively small existing datasets thus limit the amount of label skew that is possible to observe, but as an artifact rather than a property of real text.

To address this, we collected a new dataset for evaluating active learning methods for word sense disambiguation. We took a large publicly-available corpus of Redditcomments (Baumgartner, 2015) and leveraged the fact that some words will exhibit a "[o]ne sense per discourse" (Gale et al., 1992) effect: discussions in different subreddits will typically use different senses of a word. Taking this assumption to the extreme, we label all applicable sentences in each subreddit with the same word sense. For example, we consider occurrences of the word “bass” to refer to the fish in the `r/fishing` subreddit, but to refer to low frequencies in the `r/guitar` subreddit. Obviously, this approach produces an imperfect labelling; e.g., it does not distinguish between nouns like “The bass in this song is amazing” and the same word’s use as an adjective as in “I prefer playing bass guitar”, and it assumes that people never discuss music in a fishing forum. Nevertheless, this approach allows us to evaluate more extreme skew in the label distribution than would otherwise have been possible. Overall, our goal was to obtain realistic statistical structure across word senses in a way that can leverage existing embeddings, not to maximize accuracy in labeling word senses.

We chose the words by listing the top 1000 most frequently used words in the top 1000 most commented subreddits, and manually looking for words whose sense clearly correlated with the particular subreddit. Table 1 lists the 21 words that we identified in this way and the associated subreddits that we used to determine labels. For each of the words, we used an example sentence from each target sense from Lexico as exemplar sentences.

Setup All experiments used Scikit Learn (Pedregosa et al., 2011)’s multi-class logistic regression classifier with default regularization parameters on top of BERT (Devlin et al., 2018) embeddings of the target word. We used Huggingface’s Transformer library (Wolf et al., 2019) to collect the `bert-large-cased` embeddings of the target word in the context of the sentence in which it was used. This embedding gives a 1024 dimensional feature vector for each example. We repeated every experiment with 100 different random seeds report the mean and 95% confidence intervals[3] on test set accuracy. The test set had an equal number of samples from each class that exceeded the threshold.

Results We compare three versions of the Exemplar-Guided Active Learning (EGAL) algorithm relative to uncertainty driven active learning methods (Least Confidence and Entropy Search) and random search. Figure 1 gives aggregate performance of the various approaches, aggregated across 100 random seeds and the 21 words. We found the hybrid of importance sampling for guided search

[3] Computed using a Normal approximation to the mean performance.

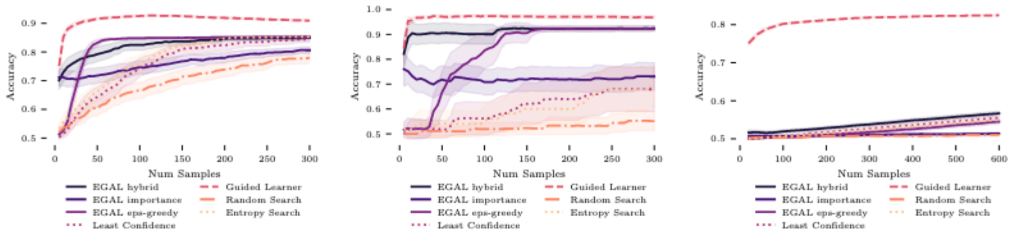

Figure 2: Accuracy vs number of samples for *bass* (left), *bank* (middle) and *fit* (right), having label skew of 1:60, 1:450 and 1:100 respectively. The word *bass* is a case where EGAL achieves significant gains with few samples; these gains are eventually evened out once the standard active learning methods gain sufficient samples. *Bank* has both a high quality exemplar and extreme skew, leading to large gains by using EGAL. *Fit* shows a failure case where EGAL's performance does not differ significantly from standard approaches.

and ϵ-greedy active learning worked best across a variety of datasets. This EGAL hybrid approach outperformed all baselines for all levels of skew, with the largest relative gains at 1:200: with 100 examples labeled, EGAL had an increase in accuracy of 11% over random search and 5% over the active learning approaches.

In Figure 2 we examine performance on three individual words and include guided learning as an oracle upper bound on the performance improve that could be achieved by a perfect exemplar search routine. On average guided learning achieved over $80 - 90\%$ accuracy on a balanced test for both tasks using less than ten samples. By contrast, random search achieved 55% and 80% accuracy on *bass* and *bank* respectively, and did not perform better than random guessing on *fit*. This suggests that the key challenge for all of these datasets is collecting balanced examples. For the first two of these three datasets, having access to an exemplar sentence gave the EGAL algorithms a significant advantage over the standard approaches; this difference was most stark on the *bank* dataset, which exhibits far more extreme skew in the label distribution. On the *fit* dataset, EGAL did not significantly improve performance, but also did not hurt performance. These trends were typical (see the appendix for all words): on two thirds of the words we tried, EGAL achieved significant improvements in accuracy, while on the remaining third EGAL offered no significant improvements but also no cost as compared to standard approaches. As with guided learning, direct comparisons between the methods are not on equal footing: the exemplar classes give EGAL more information than the standard methods have access to. However, we regard this as the key experimental point of our paper. EGAL provides a simple approach to getting potentially large improvements in performance when the label distribution is skewed, without sacrificing performance in settings where it fails to provide a benefit.

5 Conclusions

We present the Exemplar Guided Active Learning algorithm that leverages the embedding spaces of large scale language models to drastically improve active learning algorithms on skewed data. We support the empirical results with theory that shows that the method is robust to mis-specified target classes and give practical guidance on its usage. Beyond word-sense disambiguation, we are now using EGAL to collect multi-word expression data, which shares the extreme skew property.

Broader Impact

This paper presents a method for better directing an annotation budget towards rare classes, with particular application to problems in NLP. The result could be more money spent on annotation because such efforts are more worthwhile (increasing employment) or less money spent on annotation if "brute force" approaches become less necessary (reducing employment). We think the former is more likely overall, but both are possible. Better annotation could lead to better language models, with uncertain social impact: machine reading and writing technologies can help language learners and knowledge workers, increasing productivity, but can also fuel various negative trends including misinformation, bots impersonating humans on social networks, and plagiarism.

References

Attenberg, J. and Ertekin, S. Class imbalance and active learning. 2013.

Attenberg, J. and Provost, F. Why label when you can search?: Alternatives to active learning for applying human resources to build classification models under extreme class imbalance. In *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, pp. 423–432. ACM, 2010.

Baumgartner, J. *Complete Public Reddit Comments Corpus*, 2015. URL https://archive.org/details/2015_reddit_comments_corpus.

Brent, R. *Chapter 4: An Algorithm with Guaranteed Convergence for Finding a Zero of a Function*. Prentice-Hall, 1973.

Chen, J., Schein, A., Ungar, L., and Palmer, M. An empirical study of the behavior of active learning for word sense disambiguation. In *Proceedings of the main conference on human language technology conference of the north american chapter of the association of computational linguistics*, pp. 120–127. Association for Computational Linguistics, 2006.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. BERT: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Dligach, D. and Palmer, M. Good seed makes a good crop: Accelerating active learning using language modeling. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pp. 6–10, Portland, Oregon, USA, June 2011. Association for Computational Linguistics.

Edmonds, P. and Cotton, S. SENSEVAL-2: Overview. In Preiss, J. and Yarowsky, D. (eds.), *SENSEVAL@ACL*, pp. 1–5. Association for Computational Linguistics, 2001.

Fellbaum, C. *WordNet: An Electronic Lexical Database*. Bradford Books, 1998.

Gale, W. A., Church, K. W., and Yarowsky, D. One sense per discourse. In *Proceedings of the workshop on Speech and Natural Language*, pp. 233–237. Association for Computational Linguistics, 1992.

Jiang, S., Garnett, R., and Moseley, B. Cost effective active search. In Wallach, H., Larochelle, H., Beygelzimer, A., d'Alché Buc, F., Fox, E., and Garnett, R. (eds.), *Advances in Neural Information Processing Systems 32*, pp. 4880–4889. Curran Associates, Inc., 2019. URL http://papers.nips.cc/paper/8734-cost-effective-active-search.pdf.

Lattimore, T. and Szepesvári, C. *Bandit Algorithms*. Cambridge University Press, 2020.

Lexico. Lexico: Bass definition. https://www.lexico.com/en/definition/bass. Accessed: 2020-02-07.

Maurer, A. and Pontil, M. Empirical Bernstein bounds and sample variance penalization, 2009.

McCarthy, D., Koeling, R., Weeds, J., and Carroll, J. Unsupervised acquisition of predominant word senses. *Computational Linguistics*, 33(4):553–590, 2007.

Mihalcea, R., Chklovski, T., and Kilgarriff, A. The SENSEVAL-3 English lexical sample task. In *Proceedings of SENSEVAL-3: Third International Workshop on the Evaluation of Systems for the Semantic Analysis of Text*, pp. 25–28, 2004.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., and Duchesnay, E. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., and Sutskever, I. Language models are unsupervised multitask learners. 2019.

Raffel, C., Shazeer, N., Roberts, A., Lee, K., Narang, S., Matena, M., Zhou, Y., Li, W., and Liu, P. J. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv e-prints*, 2019.

Settles, B. *Active learning*, pp. 1–114. Morgan & Claypool Publishers, 2012.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. Attention is all you need. In *Advances in neural information processing systems*, pp. 5998–6008, 2017.

Virtanen, P., Gommers, R., Oliphant, T. E., Haberland, M., Reddy, T., Cournapeau, D., Burovski, E., Peterson, P., Weckesser, W., Bright, J., van der Walt, S. J., Brett, M., Wilson, J., Jarrod Millman, K., Mayorov, N., Nelson, A. R. J., Jones, E., Kern, R., Larson, E., Carey, C., Polat, İ., Feng, Y., Moore, E. W., Vand erPlas, J., Laxalde, D., Perktold, J., Cimrman, R., Henriksen, I., Quintero, E. A., Harris, C. R., Archibald, A. M., Ribeiro, A. H., Pedregosa, F., van Mulbregt, P., and Contributors, S. . . SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 2020.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., and Brew, J. Huggingface's transformers: State-of-the-art natural language processing. *ArXiv*, abs/1910.03771, 2019.

Yuan, D., Richardson, J., Doherty, R., Evans, C., and Altendorf, E. Semi-supervised word sense disambiguation with neural models. *arXiv preprint arXiv:1603.07012*, 2016.

Zhu, J. and Hovy, E. Active learning for word sense disambiguation with methods for addressing the class imbalance problem. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning (EMNLP-CoNLL)*, pp. 783–790, Prague, Czech Republic, June 2007. Association for Computational Linguistics.

Zhu, J., Wang, H., Yao, T., and Tsou, B. K. Active learning with sampling by uncertainty and density for word sense disambiguation and text classification. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pp. 1137–1144, Manchester, UK, August 2008. Coling 2008 Organizing Committee.

A Appendix

A.1 Analysis

The analysis of our regret bound uses the confidence bounds implied by Hoeffding's inequality which we state below for completeness. We focus on the $\epsilon-$greedy instantiation of the algorithm which gives estimates of $\hat{p}_y$ which are bounded between 0 and 1. The tighter confidence intervals used in Algorithm 1 improve constants but at the expense of clarity.

Lemma 5 (Hoeffding's's inequality). *Let $X_1, ..., X_n$ be independent random variables bounded by the interval $[0,1] : 0 \le X_i \le 1$ with mean μ and empirical mean $\bar{X}$. Then for $\epsilon > 0$, $\mathbb{P}\left[\bar{X} - \mu \ge \epsilon\right] \le \exp(-2n\epsilon^2)$.*

Before proving Theorem 1, we give a lemma that bounds the probability that the active set of Algorithm 1 is non-empty after T steps.

Lemma 6. *Let $T_y(t)$ denote the number of times we have observed an example from class y after t steps, and $\mathcal{A}_t = \{y : T_y(t) = 0, \hat{p}_y + \sigma_y > \gamma\}$ denote the set of "active" classes with upper confidence bound $\hat{p}_y + \sigma_y$ above the threshold. Let $\Delta = \min_i |p_i - \gamma|$ and $T = \left\lceil \frac{\log(1/\delta)}{2(1-c)^2\Delta^2} \right\rceil$ for some $c \in (0,1)$ and $\delta \in (0,1)$. Then for all $t > T$, $\mathbb{P}\left[\mathcal{A}_t \ne \emptyset\right] \le \exp(-2tc^2\Delta^2)$*

Proof. By the choice of T, for all c and $t \ge T$,

$$\Delta - \sqrt{\frac{\log(1/\delta)}{2t}} \ge c\Delta. \tag{4}$$

Now,

$$\begin{aligned}
\mathbb{P}\left[\mathcal{A}_t \ne \emptyset\right] &= \mathbb{P}\Big[\exists y \quad s.t. \quad \{T_y(t) = 0\} \\
&\qquad \cap \{\hat{p}_y(t) + \sqrt{\frac{\log(1/\delta)}{2t}} > \gamma\}\Big] \\
&\le \mathbb{P}\left[\max_{y:\hat{p}_y<\gamma} \hat{p}_y(t) + \sqrt{\frac{\log(1/\delta)}{2t}} > \gamma\right] \\
&= \mathbb{P}\left[\hat{p}_{y_{\max}}(t) - p_{y_{\max}} > \Delta_{y_{\max}} - \sqrt{\frac{\log(1/\delta)}{2t}}\right] \\
&\le \mathbb{P}\left[\hat{p}_{y_{\max}}(t) - p_{y_{\max}} > c\Delta\right] \\
&\le \exp(-2tc^2\Delta^2)
\end{aligned}$$

□

The first inequality follows by dropping the $\{T_y(t) = 0\}$ event and noting that if $\{\hat{p}_y(t) + \sqrt{\frac{\log(1/\delta)}{2t}} > \gamma\}$ then the upper confidence bound of the largest $\hat{p}_y(t)$ estimate must also exceed the threshold. We denote this estimate $\hat{p}_{y_{\max}}(t)$ and its corresponding $\Delta_{y_{\max}}$ values and true means $p_{y_{\max}}$ analogously. The second inequality follows from Equation 4 the fact that $\Delta_{y_{\max}} \ge \Delta$. Finally, the third inequality applies Hoeffding's inequality (Lemma 5) with $\epsilon = c\Delta$.

A.2 Proof of Theorem 4

Proof. Because the loss function is 0 for all classes that fall below the threshold, the difference in performance between the oracle and Algorithm 1 is only a function of the difference between the number of times each algorithm samples the unthresholded classes. Let n denote a finite horizon, $T_y(t)$ denote the number of times that an example from class y is selected by Algorithm 1 after t iterations and define the event that the algorithm does not treat a class as thresholded after n iterations as, $E_y = \{\min_{t\in[n]} \hat{p}_y + \sqrt{\frac{\log(1/\delta)}{2t}} \ge \gamma\}$.

By summing over the unthresholded classes, we can decompose regret as follows,

$$\begin{aligned}\mathcal{R} = &\sum_{y:p_y \geq \gamma} \mathbb{E}\left[\mathbf{1}(E_y^c)\left[U(T_y^*(n)) - U(T_y(n))\right]\right] + \\ &\qquad \mathbb{E}\left[\mathbf{1}(E_y)\left[U(T_y^*(n)) - U(T_y(n))\right]\right] \\ \leq &\sum_{y:p_y \geq \gamma} \Big(\mathbb{P}\left[E_y^c\right] U(T_y^*(n)) + \\ &\qquad \mathbb{E}\left[U(T_y^*(n)) - U(T_y(n)) \Big| E_y\right]\Big)\end{aligned}$$

where the first term measures regret from falsely declaring the class as falling below the threshold, and the second accounts for the difference in the number of samples selected by the oracle and Algorithm 1. The inequality follows by upper-bounding the difference between the oracle and algorithm, $U(T_y^*(n)) - U(T_y(n))$, with $U(T_y^*(n))$.

To bound this regret term, we show the first term is rare and the latter results in bounded regret. First, consider the events for which the class is falsely declared as being below the threshold. We bound this by noting that for all $t \in [n]$,

$$\begin{aligned}&\mathbb{P}\left[\hat{p}_y + \sqrt{\frac{\log(1/\delta)}{2t}} < \gamma \Big| p_y > \gamma\right] \\ \leq &\mathbb{P}\left[\hat{p}_y + \sqrt{\frac{\log(1/\delta)}{2t}} < p_y \Big| p_y > \gamma\right] \leq \delta\end{aligned}$$

The first inequality uses the fact that $p_y > \gamma$ and the second applies the concentration bound.

In the complement of this event, the class is correctly maintained among the active classes so all we have to consider is how far the expected number of times that class is selected deviates from the number of times it would have been selected under the optimal policy. By assuming a concave utility function, we are assuming diminishing returns to the number of samples that you collect for each sample. This implies that we can bound $U(T_y^*(n)) - U(T_y(n)) \leq T_y^*(n) - T_y(n)$.

Let t' denote the iteration in which the last class is removed from the active set, such that $\mathcal{A}_{t'-1} \neq \emptyset$ and $\mathcal{A}_{t'} = \emptyset$. This is the step at which regret is maximized since it is the point at which the two approaches have the largest differences in the number of samples for the unthresholded states. For any subsequent step the Algorithm 1 is unconstrained in its behavior (since it no longer has to sample) and can make up some of the difference in performance with the oracle because there are diminishing returns to performance from collecting more samples. Note that for any class y with $p_y > \gamma$ that is correctly retained, we know that $\hat{p}_y - \sqrt{\frac{\log(1/\delta)}{2t'}} > \gamma$ and so it must have been selected at least $\gamma t'$ times. The optimal strategy would have selected the class at most $\frac{t'}{k(\gamma)}$ times, where $k(\gamma)$ is the number of classes that exceed the threshold, so the expected difference between $T_y(t')$ and $T_y^*(t')$ is at most $(\frac{1}{k(\gamma)} - \gamma)t' = f(\gamma)t'$ for some $f(\gamma) \in [0, 1]$ that depends only on the choice of γ.

Because of this, bounding under-sampling regret reduces to bounding, t', the number of uniform samples the algorithm draws before all class are declared inactive. To satisfy the conditions of Lemma 6, assume the algorithm draws at least $T > \frac{\log(1/\delta)}{2(1-c)^2\Delta^2}$ uniform samples, and thereafter $\mathbb{P}\left[\mathcal{A}_t \neq \emptyset\right] \leq 2\exp(-2tc^2\Delta^2)$.

$$
\begin{aligned}
\mathbb{E}\left[t'\right] &\leq T + \sum_{t=T+1}^{n} \mathbb{P}\left[\mathcal{A}_{t-1} \neq \emptyset \cap \mathcal{A}_t = \emptyset\right] \\
&\leq T + \sum_{t=T+1}^{n} \mathbb{P}\left[\mathcal{A}_{t-1} \neq \emptyset\right] \\
&\leq T + \sum_{t=T+1}^{n} \exp(-2tc^2\Delta^2) \\
&\leq T + \exp(-2c^2T\Delta^2)\frac{1}{1-\exp(-2c^2\Delta^2)} \\
&\leq T + \exp(-2c^2T\Delta^2)[\frac{1}{2c^2\Delta^2}+1]
\end{aligned}
$$

The second last inequality uses the fact that the sum is a geometric series (and takes $n \to \infty$), and the final inequality uses the identity $\frac{1}{1-\exp(-a)} \leq 1 + \frac{1}{a}$ for all a.

Putting this together, we get,

$$
\begin{aligned}
\mathcal{R} \leq \sum_{y:p_y \geq \gamma} &\Big(\mathbb{P}\left[E_y^c\right] U(T_y^*(n)) + \\
&\quad \mathbb{E}\left[U(T_y^*(n)) - U(T_y(n)) \Big| E_y\right]\Big) \\
\leq k(\gamma) &\left[\frac{n}{k(\gamma)}\delta + T + \exp(-2(T)c^2\Delta^2)[\frac{1}{2c^2\Delta^2}+1]\right] \\
= 1 + k(\gamma)&[\frac{2\log(n)}{\Delta^2} + \frac{2+\Delta^2}{n\Delta^2}]
\end{aligned}
$$

Where inequality uses the results above and the equality follows from substituting the expression for T, collecting like terms, setting $\delta = \frac{1}{n}$ and $c = \frac{1}{2}$, and collects like terms.

□

B Experimental details

Table 1 shows all the words used for the experiments, as well as the target subreddits, exemplar classes and the number of examples from each of the classes. We subsampled the rare classes in order to achieve target skew ratios for the datasets.

We collected the words by extracting all usages of the target words in the target subreddits from the months of January 2014 to the end of May 2015. This required parsing 450GB of Baumgartner dataset. All the embeddings were collected using a single GPU and the active learning experiments were run on the CPU nodes of a compute cluster with < 30 nodes.

C Additional experiments

C.1 Synthetic data for embedding quality

Because EGAL depends on embeddings to search the neighbourhood of a given exemplar, it is likely that performance depends on the quality of the embedding. We would expect better performance from embeddings that leads to better separated the classes because as classes become better separated, it become correspondingly more likely that a single exemplar will locate the target class. To evaluate this, we constructed a synthetic 'skew MNIST' dataset as follows: the training set of the MNIST dataset is subsampled such that the classes have empirical frequency ranging from 0.6% of the

Word	Sense 1	n	Sense 2	n	Sense 1 Exemplar	Sense 2 Exemplar
back	r/Fitness	106764	r/legaladvice	18019	He had huge shoulders and a broad back which tapered to an extraordinarily small waist.	Marcia is not interested in getting her job back, but wishes to warn others.
bank*	r/personalfinance	22408	r/canoeing, r/fishing, r/kayaking, r/rivers	113	That begs the question as to whether our money would be safer under the mattress or in a bank deposit account than invested in shares, unit trusts or pension schemes.	These figures for the most part do not include freshwater wetlands along the shores of lakes, the bank of rivers, in estuaries and along the marine coasts.
bass	r/guitar	3237	r/fishing	1816	The drums lightly tap, the second guitar plays the main melody, and the bass doubles the second guitar	They aren't as big as the Caribbean jewfish or the potato bass of the Indo-Pacific region, though
card	r/personalfinance	130794	r/buildapc	62499	However, if you use your card for a cash withdrawal you will be charged interest from day one.	Your computer audio card has great sound, so what really matters are your PC's speakers.
case	r/buildapc	128945	r/legaladvice	22966	It comes with a protective carrying case and software.	The cost of bringing the case to court meant the amount he owed had risen to £962.50.
club	r/soccer	113743	r/golf	16831	Although he has played some club matches, this will be his initial first-class game.	The key to good tempo is to keep the club speed the same during the backswing and the downswing.
drive	r/buildapc	52061	r/golf	48854	This means you can record to the hard drive for temporary storage or to DVDs for archiving.	If we're out in the car, lost in an area we've never visited before, he would rather we drive round aimlessly for hours than ask direc tions.
fit	r/Fitness	75158	r/malefashionadvice	16685	The only way to get fit is to make exercise a regularly scheduled part of every week, even every day.	The trousers were a little long in the leg but other than that the clothes fit fine.
goals	r/soccer	87831	r/politics r/economics	2486	For the record, the Brazilian Ronaldo scored two goals in that World Cup final win two years ago.	As Africa attempts to achieve ambitious millennium development goals, many critical challenges confront healthcare systems.
hard	r/buildapc	63090	r/gaming	34499	That brings me to my next point: never ever attempt to write to a hard drive that is corrupt.	It's always hard to predict exactly what the challenges will be.
hero	r/DotA2	198515	r/worldnews	5816	Axe is a more useful hero than the Night Stalker	As a nation, we ought to be thankful for the courage of this unsung hero who sacrificed much to protect society.
house	r/personalfinance	46615	r/gameofthrones	11038	Inside, the house is on three storeys, with the ground floor including a drawing room, study and dining room.	After Robert's Rebellion, House Baratheon split into three branches: Lord Robert Baratheon was crowned king and took residence at King's Landing
interest	r/personalfinance	64394	r/music	1658	The bank will not lend money, and interest payments and receipts are forbidden.	The group gig together about four times a week and have attracted considerable interest from record companies.
jobs	r/politics r/economics	42540	r/apple	3642	In Kabul, they usually have low-paying, menial jobs such as janitorial work.	Steve Jobs demonstrating the iPhone 4 to Russian President Dmitry Medvedev
magic	r/skyrim	30314	r/nba	6655	Commonly, sorcerers might carry a magic implement to store power in, so the recitation of a whole spell wouldn't be necessary.	Earvin Magic Johnson dominated the court as one of the world's best basketball players for more than a decade.
manual	r/cars	2752	r/buildapc	563	Power is handled by a five-speed manual gearbox that is beefed up, along with the clutch - make that a very heavy clutch.	What is going through my head, is that this guy is reading the instructions directly from the manual, which I can now recite by rote.
market	r/investing	23563	r/nutrition	150	It's very well established that the U.S. stock market often leads foreign markets.	I have seen dandelion leaves on sale in a French market and they make a tasty addition to salads - again they have to be young and tender.
memory	r/buildapc	21103	r/cogsci r/psychology	433	Thanks to virtual memory technology, software can use more memory than is physically present.	Their memory for both items and the associated remember or forget cues was then tested with recall and recognition.
ride	r/drums	36240	r/bicycling	5638	I learned to play on a kit with a hi-hat, a crash cymbal, and a ride cymbal.	It is, for example, a great deal easier to demonstrate how to ride a bicycle than to verbalize it.
stick	r/drums	36486	r/hockey	2865	Oskar, disgusted that the singing children are so undisciplined, pulls out his stick and begins to drum.	Though the majority of players use a one-piece stick, the curve of the blade still often requires work.
tank	r/electroniccigarette	64978	r/WorldofTanks	47572	There are temperature controlled fermenters and a storage tank, and good new oak barrels for maturation.	First, the front warhead destroys the reactive armour and then the rear warhead has free passage to penetrate the main body armour of the tank.
trade	r/investing	18399	r/pokemon	5673	This values the company, whose shares trade at 247p, at 16 times prospective profits.	Do you think she wants to trade with anyone
video	r/Music	51497	r/buildapc	23217	Besides, how can you resist a band that makes a video in which they rock their guts out while naked and flat on their backs?	You do need to make certain that your system is equipped with a good-quality video card and a sound card.

Table 1: Target words with associated subreddits and exemplar sentences. *For the "bank" word, we manually removed all comments of the form, "this won't break the bank", that we would otherwise have classified as referring to river banks.

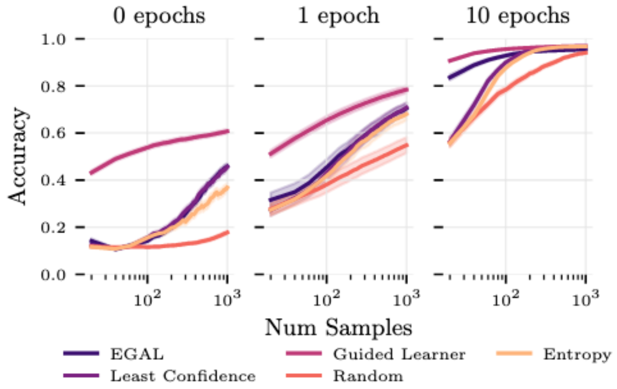

Figure 3: Accuracy for for the methods evaluated with different quality embeddings simulated by training a convolution network for 0, 1 and 10 epochs respectively. From left to right we see the difference in performance with improving quality of embedding. The left plot measures performance of logistic regression on the features from the final hidden layer of a randomly initialized convolutional network. The middle is performance after the convolutional network has been trained for a single epoch, and right is after 10 epochs. EGAL offers clear performance improvements with high quality embeddings and no degradation in performance relative to the standard approaches with poor quality embeddings.

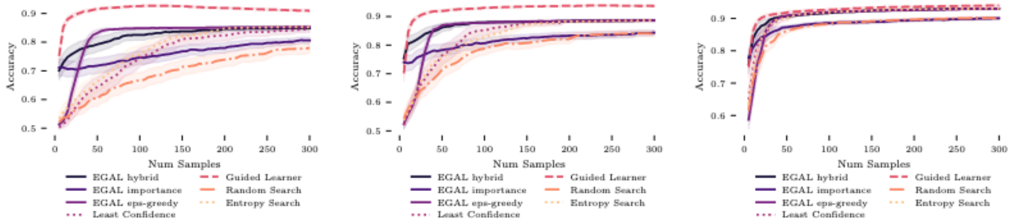

Figure 4: Accuracy as we vary the number of rare examples for the *bass*. The left plot has 50 rare examples, the middle plot has 100 and the right plot has 400. As the dataset becomes more balanced, the advantage of having access to an exemplar embedding is diminished.

data (101 training examples) to 37.4% (6501 training examples) for the most frequent class. The distribution was chosen to be as skew as possible given that the most frequent MNIST digit has 6501 training examples. The test set was kept as the original MNIST test set which is roughly balanced, so as before, the task is to transfer to a setting with balanced labels.

We evaluated the effect of different quality embeddings on performance, by using the final hidden layer of a convolution network trained for a different number of epochs on the full MNIST dataset. With zero epochs, this just corresponds to a random projection via a convolution network—so we would expect the classes to be poorly separated—but with more epochs the classes become more separable as the weights of the convolutional network are optimized to enable classification via the final layer's linear classifier. We evaluated the performance of a multi-class logistic regression classifier on embeddings from 0, 1 and 10 epochs of training.

Figure 3 shows the results. With a poor quality embedding that fails to separate classes, EGAL performed the same as the least confidence algorithm. As the quality of the embedding improved, we see the advantage of early samples from all the classes: the guided search procedure lead to class separation in very few samples, which can then be refined via the least confidence algorithm.

C.2 Performance under more moderate skew

Figure 4 shows the performance of the various approaches as the label distribution skew becomes less dramatic. The difference between the various approaches is far less pronounced in this more benign case where there was no real advantage to having access to an exemplar embedding because random sampling will quickly find the more rare examples.

C.3 Imbalance

We can gain some insight into what is driving the relative performance of the approaches by examining statistics that summarize the class imbalance over the duration of algorithms's executions. We measure imbalance using the following R^2-style statistic,

$$\text{Imbalance}(p) := 1 - \frac{\text{KL}(p, q_{\text{uniform}})}{\text{KL}(q_{\text{empirical}}, q_{\text{uniform}})}$$

where $q_{\text{empirical}}$ denotes the empirical distribution of target labels, q_{uniform} is the uniform distribution over target labels and $\text{KL}(p, q) = -\sum_x p(x) \log \frac{p(x)}{q(x)}$ is the Kullback–Leibler divergence. The score is well-defined as long as the empirical distribution of senses is not uniform, and is calibrated such that any algorithm that returns a dataset that is as imbalanced as sampling uniformly at random attains a score of zero, while algorithms that perfectly balance the data attain a score of 1.

The plots in section D.2 show this imbalance score for each of the words. There are a number of observations worth noting:

1. The algorithms with the highest accuracy in figure 2 and section D.1, also produced a more balanced distribution of senses. Guided learning was naturally the most balanced of the approaches as it has oracle access to the true labels and it explicitly optimizes for balance.
2. The importance-weighted EGAL approach over-explores leading to more imbalanced sampling that behaves like uniform sampling. This explains its relatively poor performance.
3. The standard approaches are more imbalanced because it took a large number of samples for them to see examples of the rare class; once such examples were found, they tended to produce more balanced samples as they selected samples in regions of high uncertainty.

C.4 Class coverage

The fact that the standard approaches needed a large number of samples to observe rare classes can also be seen in the plots in section D.3 which show the proportion of the unthresholded classes ($p_y \geq \gamma$) with at least one example after a given number of samples have been drawn by the respective algorithms. Again, we see that the standard approaches have to draw a far more samples before observing the rare classes because they don't have an exemplar embedding to guide search. These plots also give some insight into the failure cases. For example, if we compare the word `manual` to the word `goal` in figure 6 and figure 16 (which show accuracy and coverage respectively), we see that the exemplar for the word `goal` resulted in rare classes being found more quickly than in `manual`. This difference is reflected in a large early improvement in accuracy for EGAL methods over the standard approaches. This suggests that—as one would expect—EGAL offers significant gains when the exemplar leads to fast discovery of rare classes.

D All individual words

D.1 Accuracy

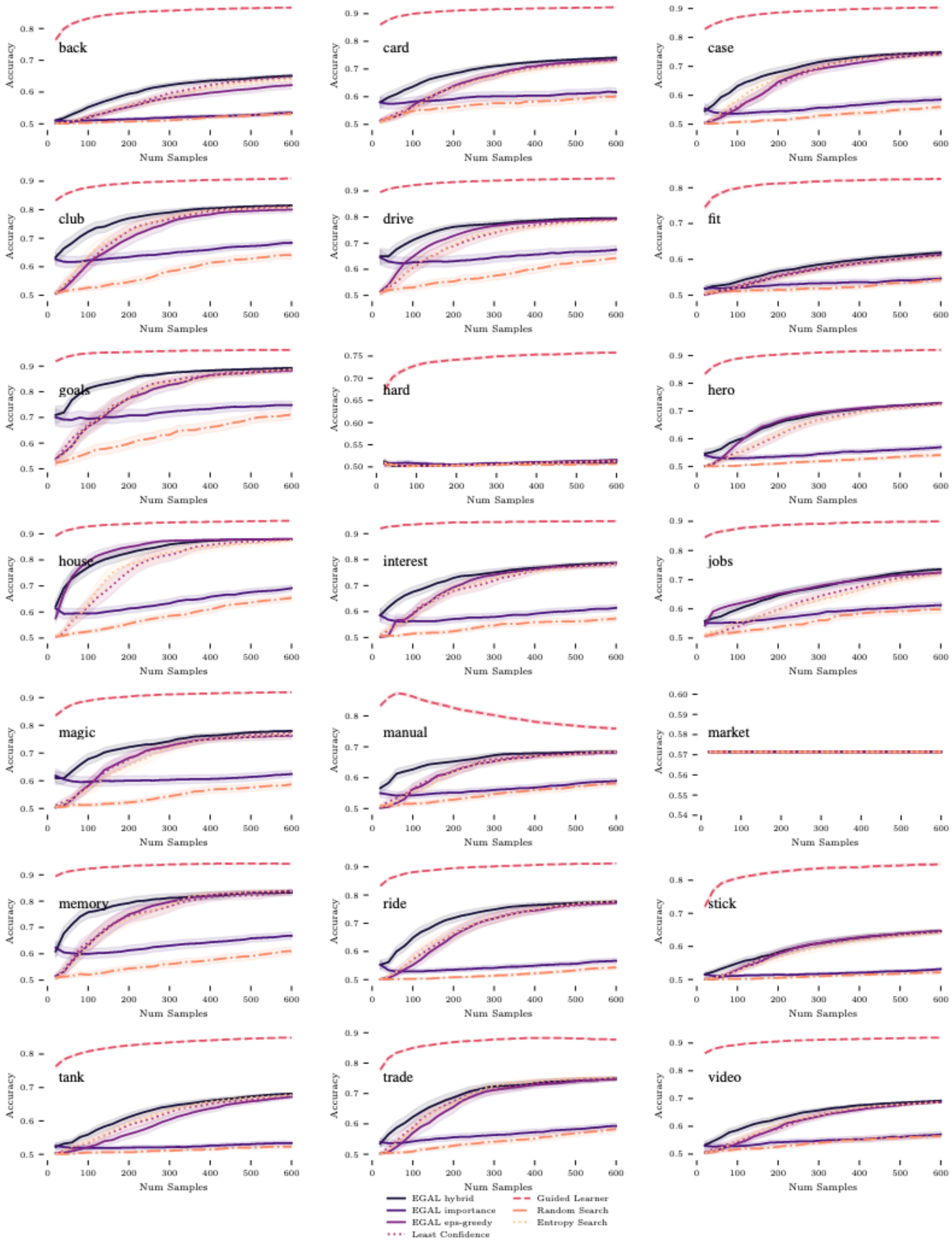

Figure 5: Average accuracy for each of the individual words with a ratio of frequent to rare class of 1:100. The rare class is randomly sub-sampled to achieved the desired skew level; each experiments is repeated 100 times with a different sub-sample drawn for each random seed.

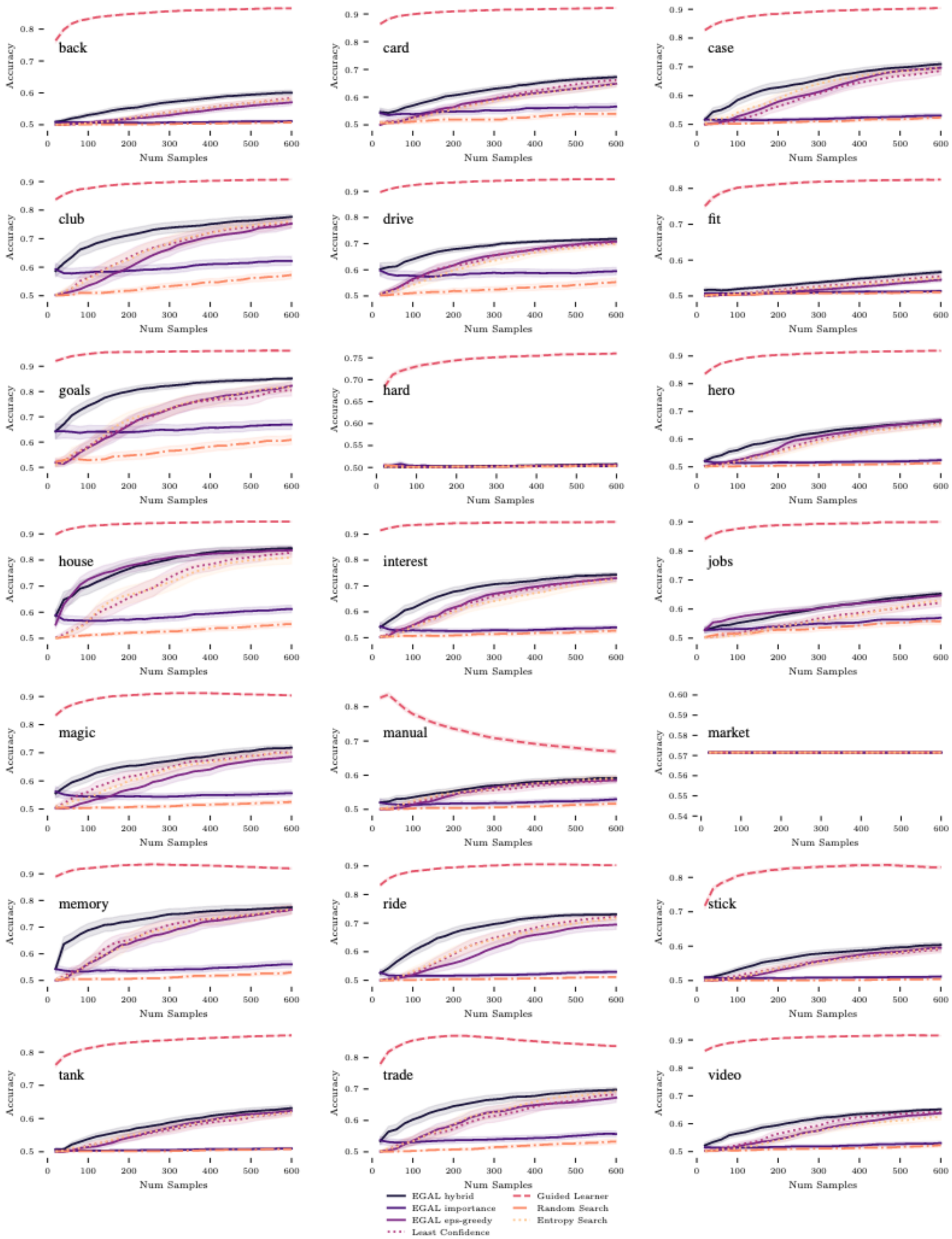

Figure 6: Average accuracy for each of the individual words with a ratio of frequent to rare class of 1:200.

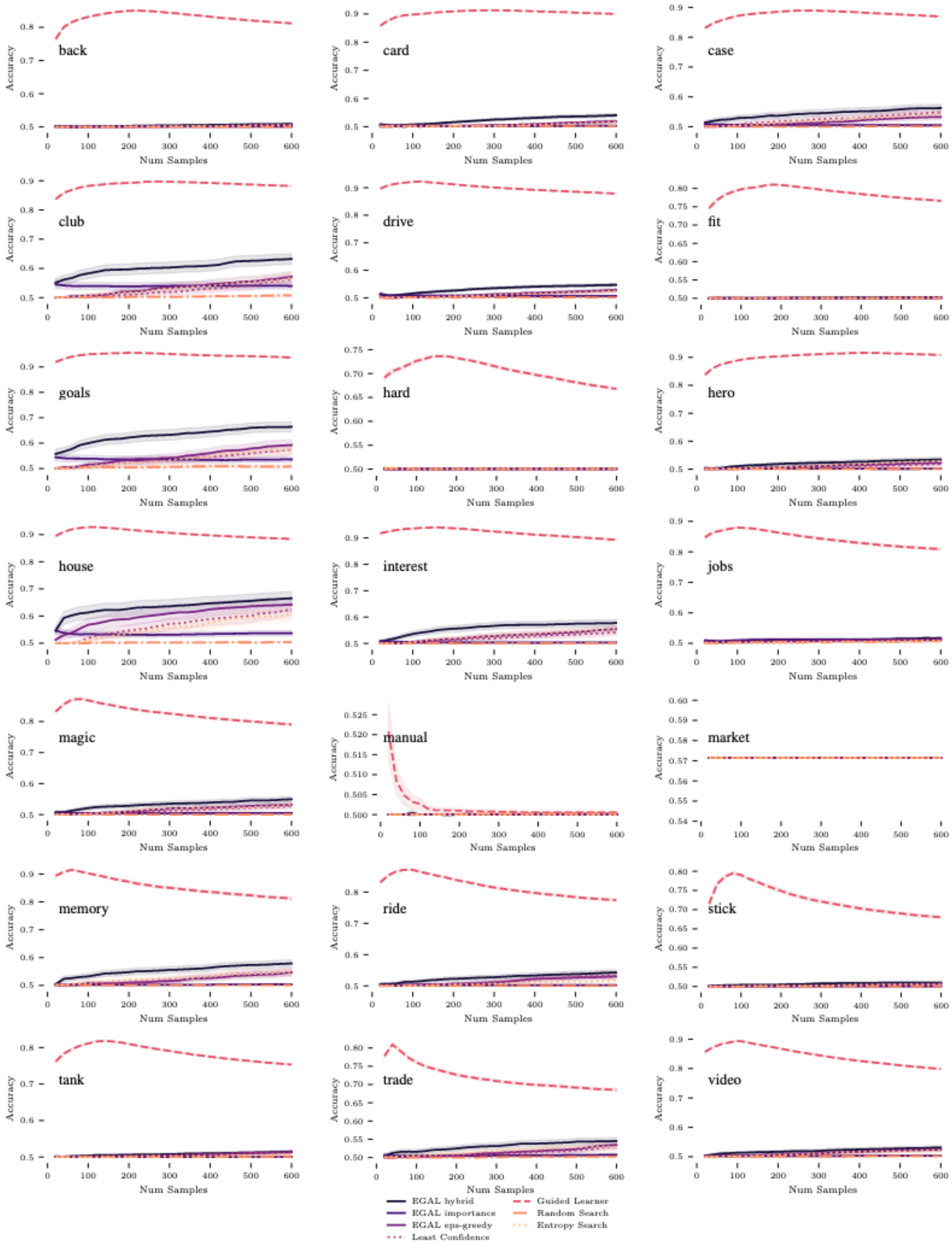

Figure 7: Average accuracy for each of the individual words with a ratio of frequent to rare class of 1:1000.

D.2 Imbalance

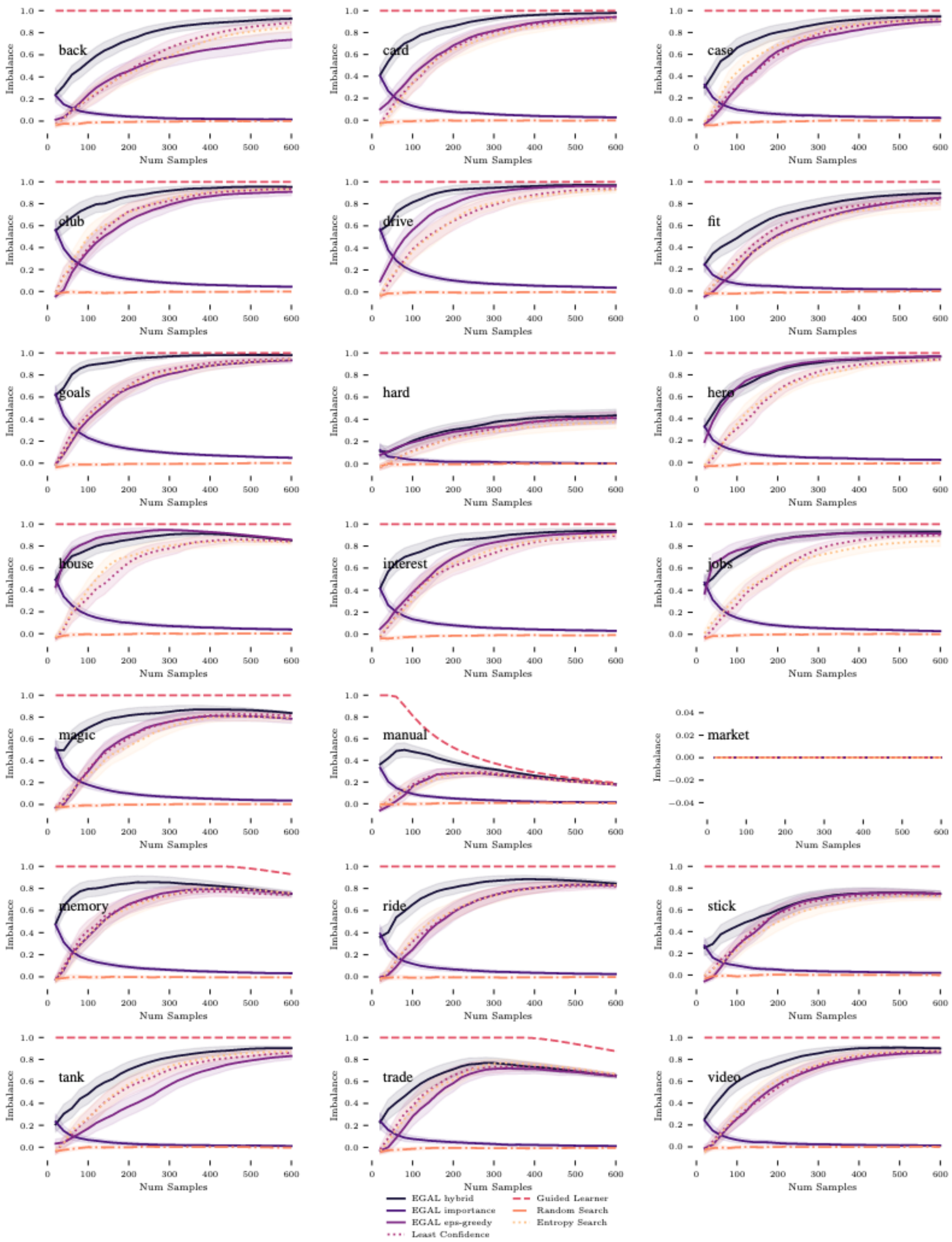

Figure 8: Average imbalance for each of the individual words with a ratio of frequent to rare class of 1:100.

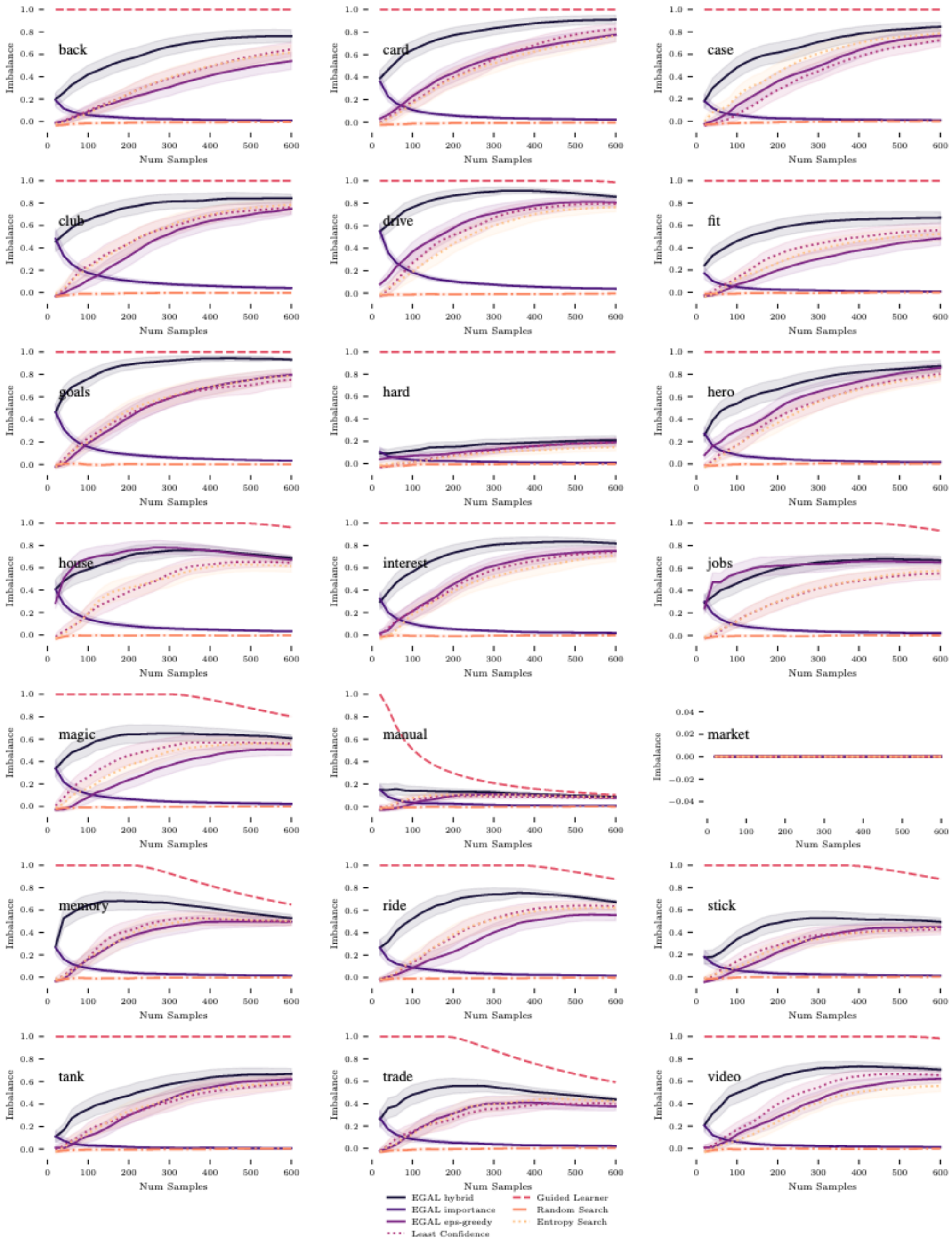

Figure 9: Average imbalance for each of the individual words with a ratio of frequent to rare class of 1:200.

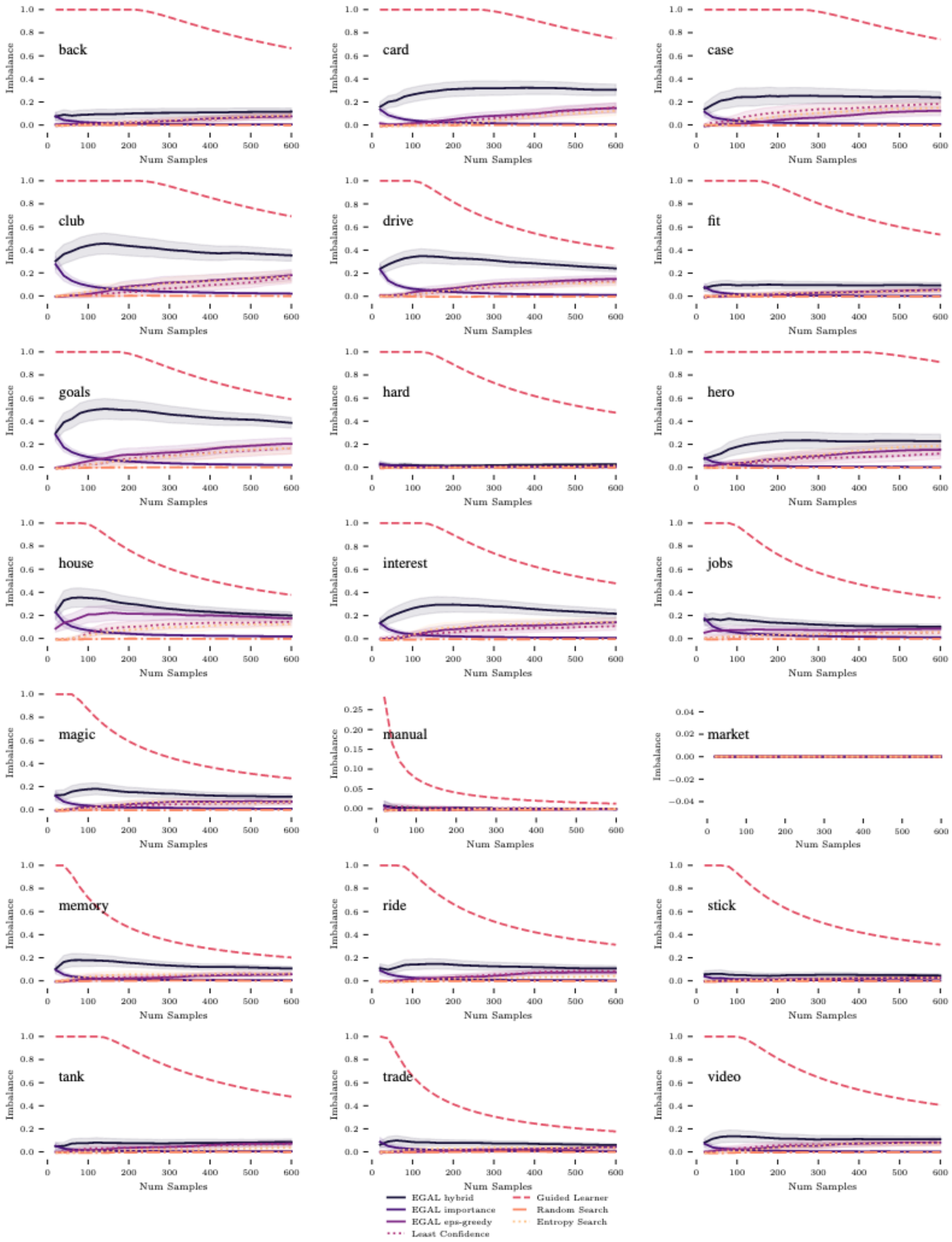

Figure 10: Average imbalance for each of the individual words with a ratio of frequent to rare class of 1:1000.

D.3 Coverage

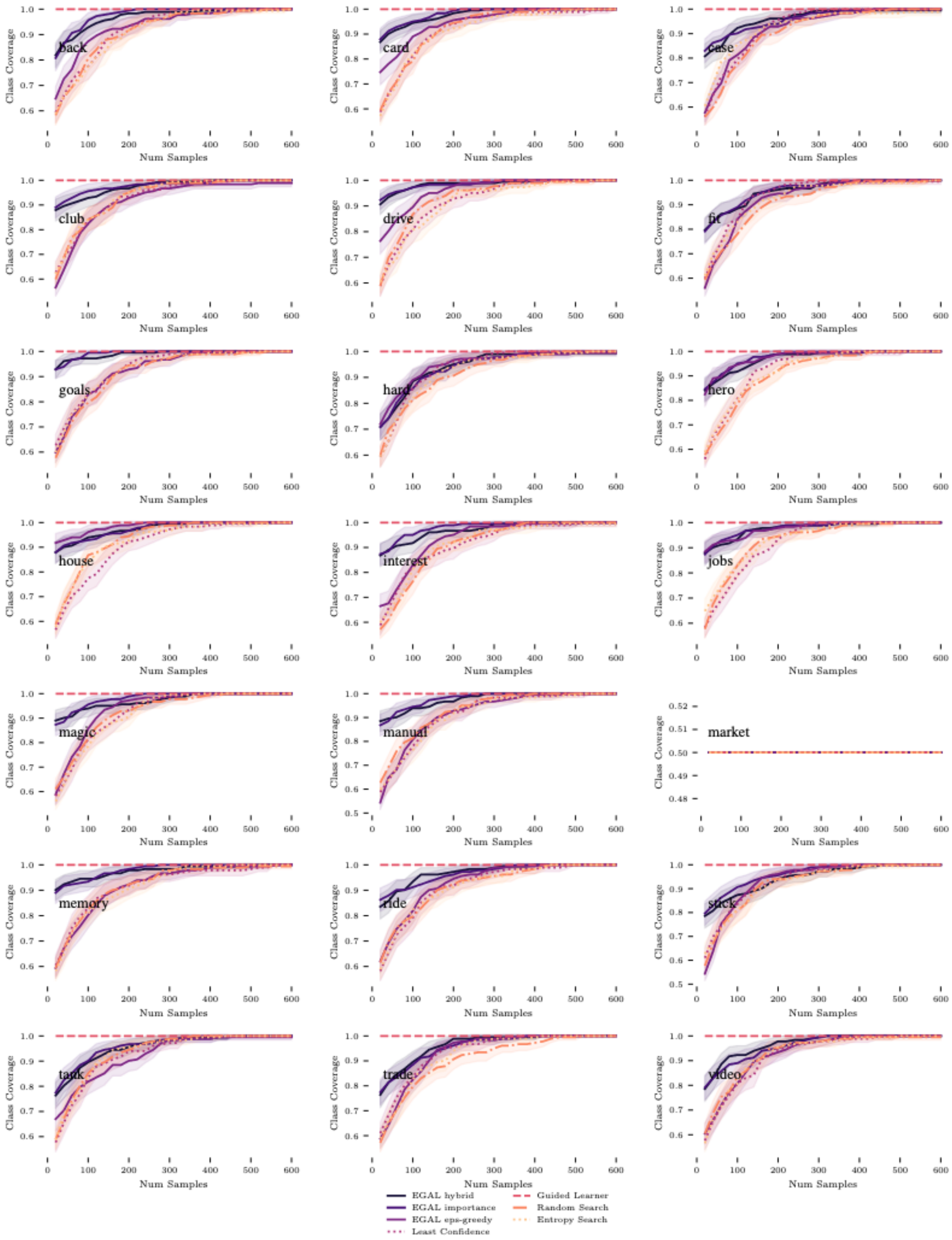

Figure 11: Average class coverage for each of the individual words with a ratio of frequent to rare class of 1:100.

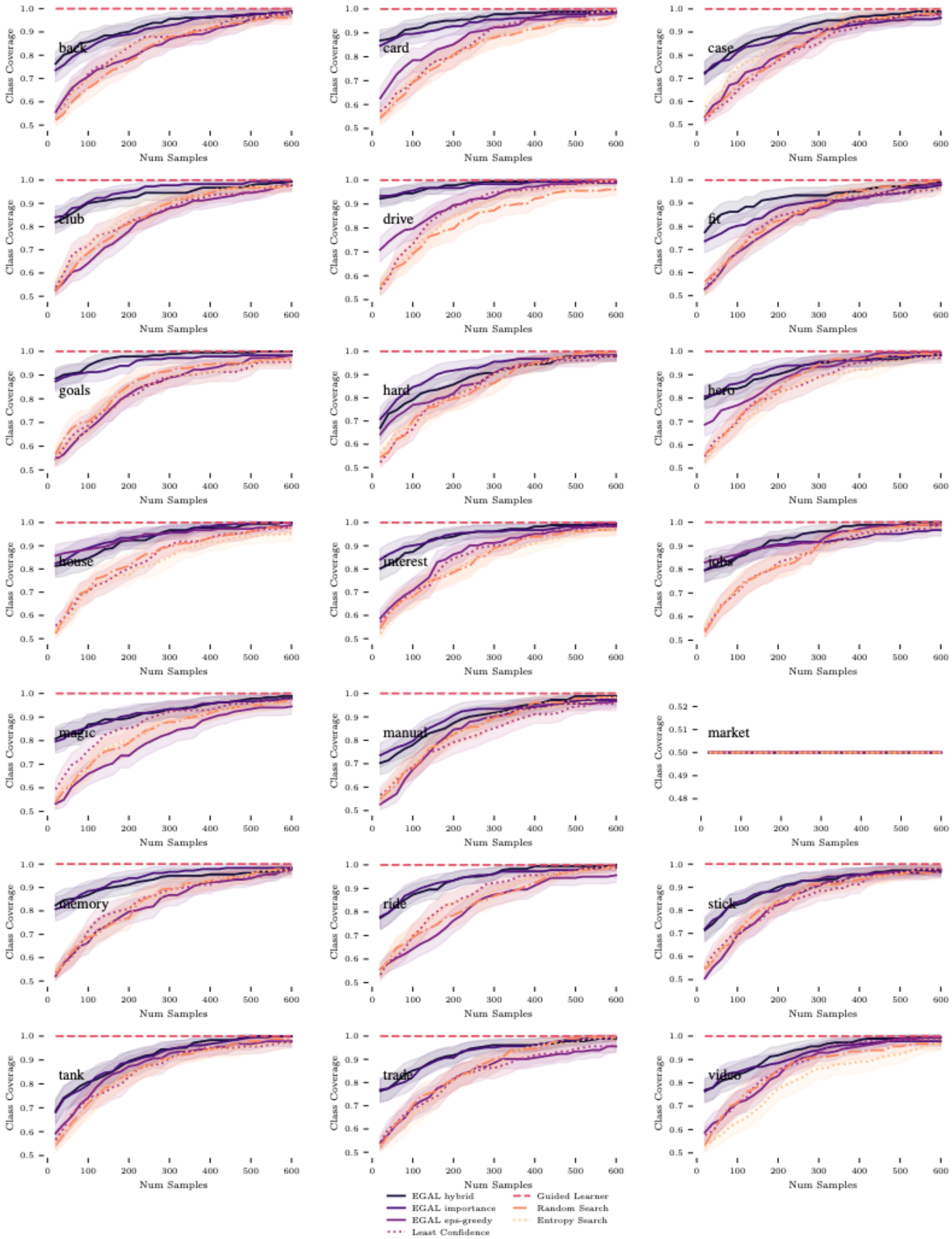

Figure 12: Average class coverage for each of the individual words with a ratio of frequent to rare class of 1:200.

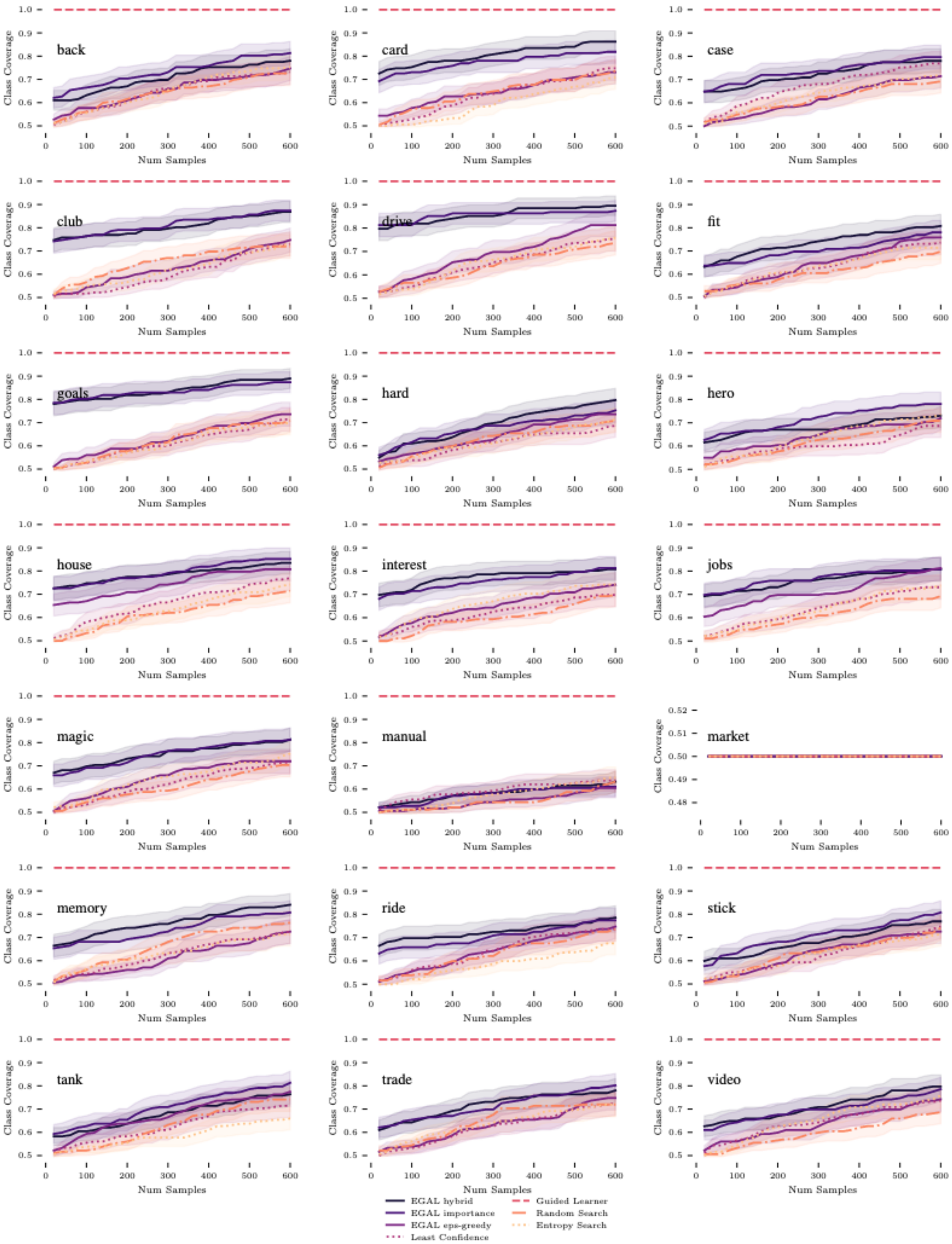

Figure 13: Average class coverage for each of the individual words with a ratio of frequent to rare class of 1:1000.