Multilingual Detection of Personal Employment Status on Twitter

Manuel Tonneau[1,2,3], **Dhaval Adjodah**[1,2,4], **João Palotti**[5],
Nir Grinberg[6] and **Samuel Fraiberger**[1,2,4]
[1]The World Bank [2]New York University [3]Centre Marc Bloch
[4]Massachusetts Institute of Technology [5]Qatar Computing Research Institute
[6]Ben-Gurion University of the Negev

Abstract

Detecting disclosures of individuals' employment status on social media can provide valuable information to match job seekers with suitable vacancies, offer social protection, or measure labor market flows. However, identifying such personal disclosures is a challenging task due to their rarity in a sea of social media content and the variety of linguistic forms used to describe them. Here, we examine three Active Learning (AL) strategies in real-world settings of extreme class imbalance, and identify five types of disclosures about individuals' employment status (e.g. job loss) in three languages using BERT-based classification models. Our findings show that, even under extreme imbalance settings, a small number of AL iterations is sufficient to obtain large and significant gains in precision, recall, and diversity of results compared to a supervised baseline with the same number of labels. We also find that no AL strategy consistently outperforms the rest. Qualitative analysis suggests that AL helps focus the attention mechanism of BERT on core terms and adjust the boundaries of semantic expansion, highlighting the importance of interpretable models to provide greater control and visibility into this dynamic learning process.

1 Introduction

Up-to-date information on individuals' employment status is of tremendous value for a wide range of economic decisions, from firms filling job vacancies to governments designing social protection systems. At the aggregate level, estimates of labor market conditions are traditionally based on nationally representative surveys that are costly to produce, especially in low- and middle-income countries (Devarajan, 2013; Jerven, 2013). As social media becomes more ubiquitous all over the world, more individuals can now share their employment status with peers and unlock the social capital of their networks. This, in turn, can provide a new lens to examine the labor market and devise policy, especially in countries where traditional measures are lagging or unreliable.

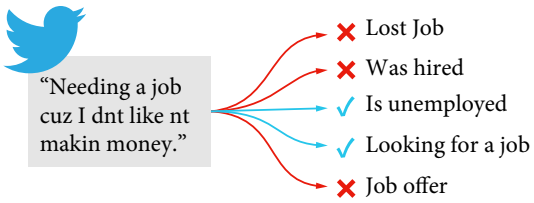

Figure 1: An example of a tweet suggestive of its author currently being unemployed and actively looking for a job.

A key challenge in using social media to identify personal disclosures of employment status is that such statements are extremely rare in an abundance of social media content – roughly one in every 10,000 posts – which renders random sampling ineffective and prohibitively costly for the development of a large labeled dataset. On the other hand, simple keyword-based approaches run the risk of providing seemingly high-accuracy classifiers while substantially missing linguistic variety used to describe events such as losing a job, looking for a job, or starting a new position (see Figure 1 for example). In the absence of a high-quality, comprehensive, and diverse ground-truth about personal employment disclosures, it is difficult to develop classification models that accurately capture the flows in and out of the labor market in any country, let alone robustly estimating it across multiple countries. Furthermore, state-of-the-art deep neural models provide little visibility into or control over the linguistic patterns captured by the model, which hampers the ability of researchers and practitioners to determine whether the model has truly learned new linguistic forms and sufficiently converged.

Active Learning (AL) is designed for settings where there is an abundance of unlabeled examples and limited labeling resources (Cohn et al., 1994). It aims to focus the learning process on the most informative samples and maximize model perfor-

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 6564 - 6587

mance for a given labeling budget. In recent years, AL proved successful in several settings, including policy-relevant tasks involving social media data (Pohl et al., 2018; Palakodety et al., 2020).

The success of pre-trained language models such as BERT (Devlin et al., 2019) in a variety of language understanding tasks has sparked interest in using AL with these models for imbalanced text classification. Yet, most research in this field has focused on artificially-generated rarity in data or imbalance that is not as extreme as the present setting (Ein-Dor et al., 2020; Schröder et al., 2021). Therefore, there is no evidence of the efficiency of AL using BERT-based models for sequence classification in real-world settings with extreme imbalance. It is unclear whether some AL strategies will perform significantly better than others in these settings, how quickly the different strategies will reach convergence (if at all), and how the different strategies will explore the linguistic space.

In this work, we leverage BERT-based models (Devlin et al., 2019) in three different AL paradigms to identify tweets that disclose an individual's employment status or change thereof. We train classifiers in English, Spanish, and Portuguese to determine whether the author of a tweet recently lost her job, was recently hired, is currently unemployed, posting to find a job, or posting a job offer. We use two standard AL strategies, Uncertainty Sampling (Lewis and Gale, 1994) and Adaptive Retrieval (Mussmann et al., 2020), and propose a novel strategy we name Exploit-Explore Retrieval that uses k-skip-n-grams (n-grams with k skipped tokens) to explore the space and provide improved interpretability. We evaluate the models both quantitatively and qualitatively across languages and AL strategies, and compare them to a supervised learning baseline with the same number of labels. Therefore, our contributions are:

- An evaluation of three AL strategies for BERT-based binary classification under extreme class imbalance using real-world data.
- A novel AL strategy for sequence classification that performs on par with other strategies, but provides additional interpretability and control over the learning process.
- A qualitative analysis of the linguistic patterns captured by BERT across AL strategies.
- A large labeled dataset of tweets about unemployment and fine-tuned models in three languages to stimulate research in this area[1].

2 Background and related work

2.1 Identifying self-disclosures on Twitter

Social media users disclose information that is valuable for public policy in a variety of areas ranging from health (Achrekar et al., 2011; Mahata et al., 2018; Klein et al., 2018) to emergency response to natural disasters (Bruns and Liang, 2012; Kryvasheyeu et al., 2016) through migration flows (Fiorio et al., 2017; Chi et al., 2020; Palotti et al., 2020). A key challenge in identifying self-disclosures on social media is the rare and varied nature of such content with a limited labeling budget. Prior work that studied self-disclosures on Twitter had either used pattern matching, which is prone to large classification errors (Antenucci et al., 2014; Proserpio et al., 2016), or focused on curated datasets (Li et al., 2014; Preoţiuc-Pietro et al., 2015; Sarker et al., 2018; Ghosh Chowdhury et al., 2019), which provide no guarantees about recall or coverage of the positive class. These issues are more severe in real-world settings of extreme imbalance, where random sampling is unlikely to retrieve any positives, let alone diverse. These challenges motivate the use of AL, as described next.

2.2 Active Learning

AL has been used successfully in various settings to maximize classification performance for a given labeling budget (see Settles (1995) for a survey). With the emergence of pre-trained language models such as BERT (Devlin et al., 2019) and their success across a number of different language tasks, recent work has studied the combination of AL and BERT, either by using BERT to enhance traditional AL methods (Yuan et al., 2020) or by applying established AL methods to improve BERT's classification performance (Zhang and Zhang, 2019; Shelmanov et al., 2019; Liu et al., 2020; Grießhaber et al., 2020; Prabhu et al., 2021; Schröder et al., 2021).

In the specific case of binary classification with moderate class imbalance, Ein-Dor et al. (2020) show that AL with BERT significantly outperforms random sampling but that no single AL strategy stands out in terms of BERT-based classification performance, both for balanced and imbalanced

[1]Labeled datasets and models can be found at https://github.com/manueltonneau/twitter-unemployment

settings. Yet, the authors only consider a relatively moderate class imbalance of 10-15% positives, and does not cover extreme imbalance, which is common in many text classification tasks. Our current research examines a considerably more extreme imbalance of about 0.01% positives, where traditional AL approaches can be ineffective (Attenberg and Provost, 2010). Under this extreme imbalance, Mussmann et al. (2020) show the potential of AL for BERT to outperform random sampling for pairwise classification. To the best of our knowledge, this work is the first to compare the performance of AL methods for BERT-based sequence classification in real-world extreme imbalance settings.

3 Experimental procedure

3.1 Data collection

Our dataset was collected from the Twitter API. It contains the timelines of the users with at least one tweet in the Twitter Decahose and with an inferred profile location in the United States, Brazil, and Mexico. In addition to the United States, we chose to focus on Brazil and Mexico as both of them are middle-income countries where Twitter's penetration rate is relatively high. For each country, we drew a random sample of 200 million tweets covering the period between January 2007 and December 2020 and excluding retweets. We then split it evenly in two mutually exclusive random samples R_e and R_s. In the following sections, we use R_e to evaluate each model's performance in a real-world setting and R_s to sample new tweets to label.

Our labeling process sought to identify four non-exclusive, binary states that workers may experience during their career: losing a job ("Lost Job"), being unemployed ("Is Unemployed"), searching for a job ("Job Search"), and finding a job ("Is Hired"). We only considered first-person disclosures as positives. For the classes "Lost Job" and "Is Hired", we only considered such events that happened in the past month as positives as we want to determine the user's current employment status. To complement the focus on workers, we also labeled tweets containing job offers ("Job Offer"). We used Amazon Mechanical Turk (MTurk) to label tweets according to these 5 classes (see Figure 1 and Section A.2 for details).

3.2 Initialization sample

As previously stated, the extreme imbalance of our classification task of one positive example for every 10,000 tweets renders random sampling ineffective and prohibitively costly. In order to build high-performing classifiers at a reasonable cost, we selected a set of 4 to 7 seed keywords that are highly specific of the positives and frequent enough for each class and country. To do so, we defined a list of candidate seeds, drawing from Antenucci et al. (2014) for the US and asking native speakers in the case of Mexico and Brazil, and individually evaluated their specificity and frequency (see Section A.1 for additional details). We then randomly sampled 150 tweets containing each seed from R_s, allowing us to produce a stratified sample L_0 of 4,524 English tweets, 2703 Portuguese tweets, and 3729 Spanish tweets respectively (Alg. 1). We then labeled each tweet using Amazon Mechanical Turk (MTurk) allowing us to construct a language-specific stratified sample that is common to the 5 classes (see Section A.3 for descriptive statistics of the stratified sample).

3.3 Models

We trained five binary classifiers to predict each of the five aforementioned labeled classes. Preliminary analysis found that BERT-based models considerably and consistently outperformed keyword-based models, static embedding models, and the combination of these models. We benchmarked several BERT-based models and found that the following models gave the best performance on our task: **Conversational BERT** for English tweets (Burtsev et al., 2018), **BERTimbau** for Brazilian Portuguese tweets (Souza et al., 2020) and **BETO** for Mexican Spanish tweets (Cañete et al., 2020) (see Section A.4 for details on model selection).

We fine-tuned each BERT-based model on a 70:30 train-test split of the labeled tweets for 20 epochs (Alg. 1). Following Dodge et al. (2020), we repeated this process for 15 different random seeds and retained the best performing model in terms of area under the ROC curve (AUROC) on the test set at or after the first epoch (see Section A.5 for details).

3.4 Model evaluation

While the standard classification performance measure in an imbalanced setting is the F1 score with a fixed classification threshold (e.g. 0.5), it is not applicable in our case for two reasons. First, we care about the performance on a large random set of tweets and the only labeled set we could compute the F1 metric from is the stratified test set

which is not representative of the extremely imbalanced random sample R_e. Second, the fact that neural networks are poorly calibrated (Guo et al., 2017) makes the choice of a predefined classification threshold somewhat arbitrary and most likely sub-optimal.

We developed an alternative threshold-setting evaluation strategy. First, we computed the predicted score of each tweet in R_e (Alg. 1), which is a random sample. Then, for each class, we labeled 200 tweets in R_e along the score distribution (see section A.7.1 for more details). We measured the performance of each classifier on R_e by computing:

- the **Average Precision** as common in information retrieval.
- the **number of predicted positives**, defined as the average rank in the confidence score distribution when the share of positives reaches 0.5.
- the **diversity**, defined as the average pairwise distance between true positives.

Details about the evaluation metrics can be found in Section A.7.

Initialization: for each seed s, sample 150 tweets containing s from R_s; have them labeled for the five classes; the resulting labeled set is the stratified sample $L_0 = S_0$; discard already sampled tweets from $R_s (R_s = R_s - L_0)$

At each iteration i and for each class:

- **Finetuning**: train-test split of S_i in 70/30; finetune 15 BERT models on the train set using different seeds; select the best model M_i^* with the highest AUROC on the test set.
- **Inference on R_e and R_s using M_i^***
- **Active Learning**: sample most informative tweets from R_s (100 per class); have them labeled for the five classes; the resulting labeled set is L_{i+1}; define $S_{i+1} = \bigcup_{j=0}^{i+1} L_j$ and $R_s = R_s - L_{i+1}$
- **Evaluation:** sample tweets along the score distribution in R_e; have them labeled; compute the average precision, number of predicted positives and diversity metrics

Algorithm 1: Experimental procedure

3.5 Active Learning strategies

Next, we used pool-based AL (Settles, 1995) in batch mode, with each class-specific fine-tuned model as the classification model, in order to query new informative tweets in R_s. We compared three different AL strategies aiming to balance the goal of improving the precision of a classifier while expanding the number and the diversity of detected positives instances:

- **Uncertainty Sampling** consists in sampling instances that a model is most uncertain about. In a binary classification problem, the standard approach is to select examples with a predicted score close to 0.5 (Settles, 2009). In practice, this rule of thumb might not always lead to identify uncertain samples when imbalance is high (Mussmann et al., 2020), especially with neural network models known to be poorly calibrated (Guo et al., 2017). To overcome this issue, we contrast a naive approach which consists in querying the 100 instances whose *uncalibrated* scores are the closest to 0.5, to an approach that uses *calibrated* scores (see Section A.9 for details).
- **Adaptive Retrieval** aims to maximize the precision of a model by querying instances for which the model is most confident of their positivity (Mussmann et al., 2020). This approach is related to certainty sampling (Attenberg et al., 2010). Here, we select the 100 tweets whose predicted score is the highest for each class.
- Our novel strategy, **Exploit-Explore Retrieval** (see Section A.8 for details), aims to maximize precision ('exploitation') while improving recall by feeding new and diverse instances at each iteration ('exploration'):
  - **Exploitation**: Randomly query 50 new tweets from the top 10^4 tweets with the highest predicted score in R_s.
  - **Exploration**: Identify the 10 k-skip-n-grams with the highest frequency of occurrences in the top 10^4 tweets, relative to their frequency in R_s. Then, randomly sample 50 new tweets containing each k-skip-n-gram (see Section A.8 for formal definition of k-skip-n-grams and a discussion on the choice of threshold).

Additionally, we compared these AL strategies to a supervised **Stratified Sampling** baseline, that consists of the same initial motifs defined in Section 3.2 and the same number of labels as available to all other AL strategies. Overall, for each strategy, each iteration and each class, we labeled 100 new tweets in R_s. We then combined the 500 new labels across classes with the existing ones to finetune and evaluate a new BERT-based model for

each class as described in Section 3.3, which we then used to select tweets for labeling for the next iteration. We considered that an AL strategy had converged when there was no significant variation of average precision, number of predicted positives and diversity for at least two iterations (see Section A.7.6 for details).

4 Results

4.1 Initial sample

At iteration 0, we fine-tuned a BERT-based classifier on a 70:30 train-test split of the initialization sample L_0 for each class and country. All the AUROC values on the test set are reported in Table 7.

We obtain very high AUROCs ranging from 0.944 to 0.993 across classes and countries. "Job Offer" has the highest AUROCs with values ranging from 0.985 for English to 0.991 for Portuguese and 0.993 for Spanish. Upon closer examination of positives for this class, we find that the linguistic structure of tweets mentioning job offers is highly repetitive, a large share of these tweets containing sentences such as "We're #hiring! Click to apply:" or naming job listing platforms (e.g: "#CareerArc"). By contrast, the most difficult class to predict is "Lost Job", with an AUROC on the test set equal to 0.959 for English and 0.944 for Spanish. This class also has the highest imbalance, with approximately 6% of positives in the stratified sample for these two languages.

Taken together, these results show that a fine-tuned BERT model can achieve very high classification performance on a stratified sample of tweets across classes and languages. However, these numbers cannot be extrapolated to directly infer the models' performance on random tweets, which we discuss in the next section.

4.2 Active Learning across languages

Next, we compared the performance of our **exploit-explore retrieval** strategy on English, Spanish and Portuguese tweets. We used exploit-explore retrieval as it provides similar results to other strategies (Section 4.3), while allowing greater visibility into selected motifs during the development process (Section 4.4). We ran 8 AL iterations for each language and report the results in Fig. 2, Fig. 5 and Table 10.

First, we observe substantial improvements in average precision (AP) across countries and classes with just one or two iterations. These improvements are especially salient in cases where precision at iteration 0 is very low. For instance, for the English "Is Unemployed" class and the Spanish "Is Hired" class, average precision goes respectively from 0.14 and 0.07 to 0.83 and 0.8 from iteration 0 to iteration 1 (Fig. 2 and Fig. 5). A notable exception to this trend is the class "Job Offer", especially for English and Portuguese. These performance differences can in part be explained by the varying quality of the initial seed list across classes. This is confirmed by the stratified sampling baseline performance discussed in 4.3. In the case of "Job Offer", an additional explanation discussed earlier in Section 4.1 is the repetitive structure of job offers in tweets which makes this class easier to detect compared to others.

Also, the class "Lost Job" has the worst performance in terms of AP across countries. One reason is that the data imbalance for this class is even higher than for other classes, as mentioned in Section 4.1. Another explanation for the low precision is the ambiguity inherent to the recency constraint, namely that an individual must have lost her job at most one month prior to posting the tweet.

Apart from the "Job Offer" class in English and Portuguese, AL consistently allows to quickly expand from iteration 0 levels with the number of predicted positives multiplied by a factor of up to 10^4 (Fig. 2). Combined with high AP values, this result means that the classifiers manage to capture *substantially* more positives compared to iteration 0. This high expansion is combined with increasing semantic diversity among true positive instances.

The class "Job Offer" stands out with little expansion and diversity changes in the English and Portuguese cases. For Spanish, expansion and diversity changes are higher. One explanation is that the structure of Mexican job offers is less repetitive, with individual companies frequently posting job offers, as opposed to job aggregators in the case of the US and Brazil.

Overall, apart from a few edge cases, we find that AL used with pre-trained language models is successful at significantly improving precision while expanding the number and the diversity of predicted positive instances in a small number of iterations across languages. Indeed, precision gains reach up to 90 percentage points from iteration 0 to the last iteration across languages and classes and the number of predicted positives is multiplied

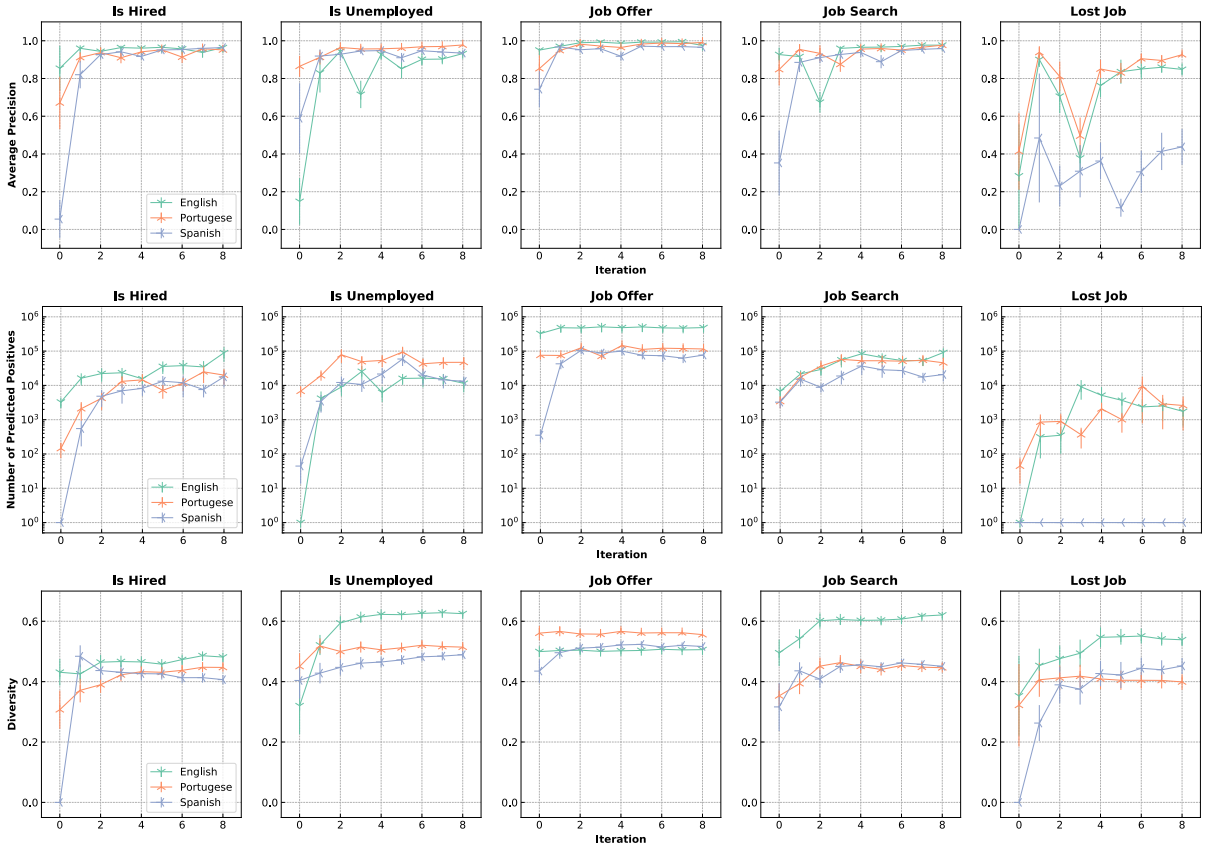

Figure 2: Average precision, number of predicted positives and diversity of true positives (in row) for each class (in column) for English (green), Portuguese (orange), and Spanish (purple). We report the standard error of the average precision and diversity estimates, and we report a lower and an upper bound for the number of predicted positives. Additional details on how the evaluation metrics are computed are reported in section A.7.

by a factor of up to 10^4. Furthermore, on average, the model converges in only 5.6 iterations across classes for English and Portuguese, and in 4.4 iterations for Spanish (see Table 10 for details).

4.3 Comparing Active Learning strategies

In this section, we evaluated on English tweets the stratified sampling baseline and the four AL strategies described in Section 3.5, namely exploit-explore retrieval, adaptive retrieval and uncertainty sampling with and without calibration. We ran five iterations for each strategy and reported the results on Figure 3 in this section as well as Table 11 and Figure 6 in Section A.10.

We find that AL brings an order of magnitude more positives and does so while preserving or improving both the precision and the diversity of results. Apart from the "Job Offer" class discussed in Section 4.2, AL consistently outperforms the stratified sampling baseline. This is especially true for the classes "Is Unemployed" and "Lost Job" where the baseline performance stagnates at a low level, suggesting a poor seed choice, but also holds for classes "Is Hired" and "Job Search" with stronger baseline performance. We also find that no AL strategy consistently dominates the rest in terms of precision, number and diversity of positives. The gains in performance are similar across AL strategies, and are particularly high for the classes "Lost Job" and "Is Unemployed", which start with a low precision. The number of predicted positives and the diversity measures also follow similar trends across classes and iterations.

We also observe occasional "drops" in average precision of more than 25% from one iteration to the next. Uncalibrated uncertainty sampling seems particularly susceptible to these drops, with at least one occurrence for each class. Upon examination of the tweets sampled for labeling by this strategy, the vast majority of tweets are negatives and when a few positives emerge, their number is not large enough to allow the model to generalize well. This variability slows down the convergence process of uncertainty sampling when it is not uncalibrated (table 11). In contrast, calibrated uncertainty sam-

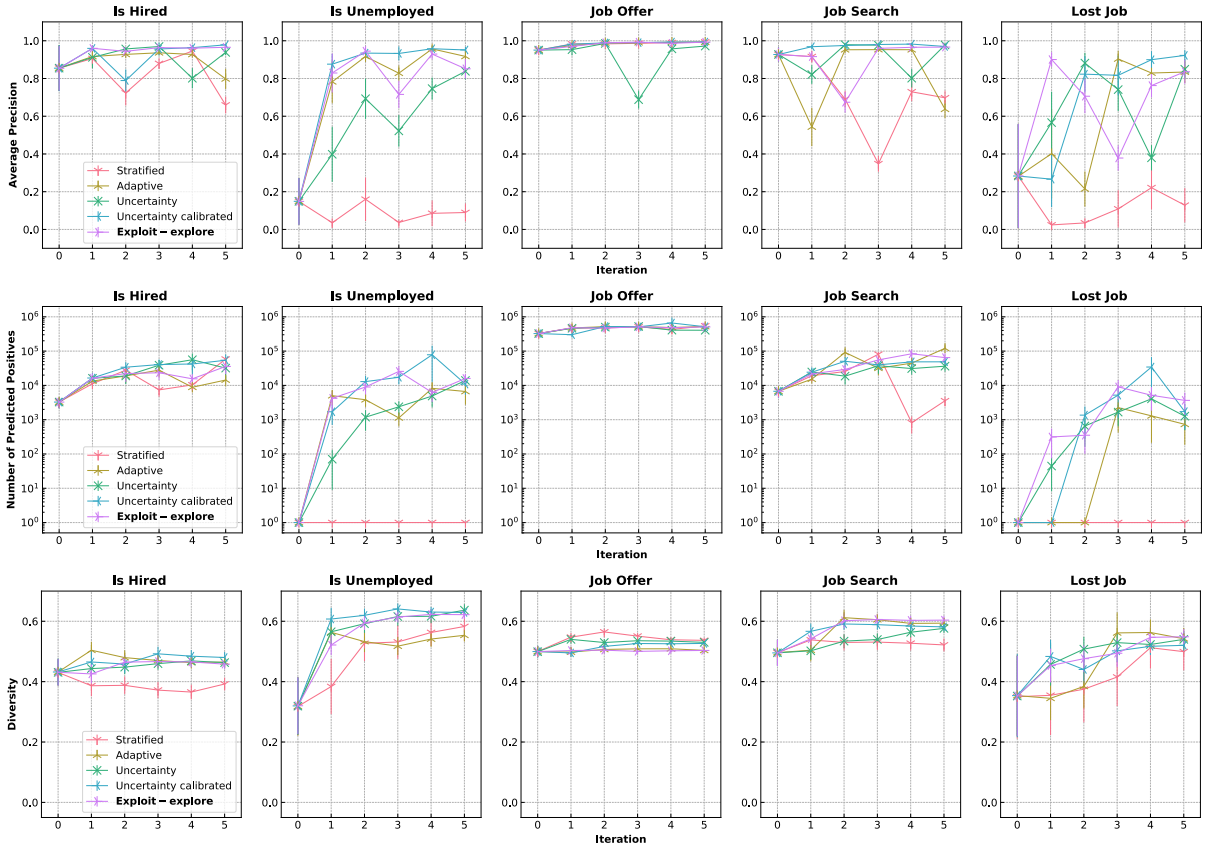

Figure 3: Average precision, number of predicted positives and diversity of true positives (in row) for each class (in column) across AL strategies. We report the standard error of the average precision and diversity estimates, and we report a lower and an upper bound for the number of predicted positives. Additional details on how the evaluation metrics are computed are reported in section A.7.

pling is less susceptible to these swings, emphasizing the importance of calibration for more "stable" convergence in settings of extreme imbalance.

Taken together, our quantitative results show that the positive impact of AL on classification performance in an extremely imbalanced setting holds across AL strategies. Aside from a few occasional performance "drops", we find significant gains in precision, expansion and diversity across strategies. Yet, we find that no AL strategy consistently dominates the others across a range of prediction tasks for which the number and the linguistic complexity of positive instances vary widely. Next, we investigate the results qualitatively to gain deeper understanding of the learning process.

4.4 Qualitative analysis

We qualitatively examined the tweets selected for labeling by each strategy to understand better what BERT-based models capture and reflect on the quantitative results. We focused on English tweets only and took a subsample of tweets at each iteration to better understand each strategy's performance. We excluded the "Job Offer" class from this analysis since the performance, in this case, is exceptionally high, even at iteration 0.

Our analysis finds that many tweets queried by the various AL strategies capture a general "tone" that is present in tweets about unemployment, but that is not specific to one's employment status. For example, these include tweets of the form of "I'm excited to ... in two days" for the recently hired class, "I've been in a shitty mood for ..." for unemployment or "I lost my ..." for job loss. This type of false positives seems to wane down as the AL iterations progress, which suggests that a key to the success of AL is first to fine-tune the attention mechanism to focus on the core terms and not the accompanying text that is not specific to employment status. In the stratified sampling case, the focus on this unemployment "tone" remains uncorrected, explaining the poor performance for classes "Lost Job" and "Is Unemployed" and the performance drops for "Is Hired" and "Job Search".

A second theme in tweets queried by AL in-

volves the refinement of the initial motifs. Uncertainty sampling (calibrated and uncalibrated), adaptive retrieval, and the exploitation part of our exploit-explore retrieval method seem to query tweets that either directly contain a seed motif or a close variant thereof. For example, tweets for the class "Lost Job" may contain the seed motifs "laid off", "lost my job", and "just got fired". As mentioned in Section 4.2 to explain occasional drops in performance, many tweets labeled as negatives contain over-generalization of the semantic concept such as expanding to other types of losses (e.g. "lost my phone"), other types of actions (e.g. "got pissed off"), or simply miss the dependence on first-person pronouns (e.g. "@user got fired"). Many of the positively labeled tweets contain more subtle linguistic variants that do not change the core concept such as "I *really* need a job", "I really need to get a job", "I need *to find* a job", or "I need a *freaken* job". Adaptive retrieval chooses these subtle variants more heavily than other strategies with some iterations mostly populated with "I need a job" variants. Overall, these patterns are consistent with a view of the learning process, specifically the classification layer of the BERT model, as seeking to find the appropriate boundaries of the target concept.

Finally, the exploration part of the exploit-explore retrieval makes the search for new forms of expression about unemployment more explicit and interpretable. For example, the patterns explored in the first few iterations of explore-exploit retrieval include "I ... lost ... today", "quit .. my .. job", "I ... start my ... today", and "I'm ... in ... need". A detailed presentation of the explored k-skip-n-grams for US tweets can be found in Table 9 of Section A.8. While this strategy suffers from issues that also affect other AL strategies, we find that the explore part of exploit-explore retrieval is more capable of finding new terms that were not part of the seed list (e.g., quit, career) and provides the researcher with greater insight into and control over the AL process.

5 Discussion and conclusion

This work developed and evaluated BERT-based models in three languages and used three different AL strategies to identify tweets related to an individual's employment status. Our results show that AL achieves large and significant improvements in precision, expansion, and diversity over stratified sampling with only a few iterations and across languages. In most cases, AL brings an order of magnitude more positives while preserving or improving both the precision and diversity of results. Despite using fundamentally different AL strategies, we observe that no strategy consistently outperforms the rest. Within the extreme imbalance setting, this is in line with – and complements – the findings of Ein-Dor et al. (2020).

Additionally, our qualitative analysis and exploration of the exploit-explore retrieval give further insights into the performance improvements provided by AL, finding that substantial amounts of queried tweets hone the model's focus on employment rather than surrounding context and expand the variety of motifs identified as positive. This puts exploit-explore retrieval as a valuable tool for researchers to obtain greater visibility into the AL process in extreme imbalance cases without compromising on performance.

While the present work demonstrates the potential of AL for BERT-based models under extreme imbalance, an important direction for future work would be to further optimize the AL process. One could for instance study the impact on performance of the stratified sample size or the AL batch size. To overcome the poor seed quality for some classes, other seed generation approaches could be tested, such as mining online unemployment forums using topic modeling techniques to discover different ways to talk about unemployment. In terms of model training and inference, the use of multi-task learning for further performance improvement could be studied due to the fact that classes of unemployment are not mutually exclusive. We hope that our experimental results as well as the resources we make available will help bridge these gaps in the literature.

Ethics statement

We acknowledge that there is some risk, like any other technology that makes inferences at the individual level, that the technology presented here will be used for harm. However, due to the public nature of the content and the fact that the potential harm already exists using basic keyword search, we believe that the marginal risk added by our classifier is minimal.

Acknowledgements

We thank participants of the Israeli Seminar on Computational Linguistics at Ben-Gurion University of the Negev as well as the anonymous reviewers for their valuable comments. We also thank Aleister Montfort, Varnitha Kurli Reddy and Boris Sobol for their excellent research assistance. This work was supported by the SDG Partnership Fund.

References

Harshavardhan Achrekar, Avinash Gandhe, Ross Lazarus, Ssu-Hsin Yu, and Benyuan Liu. 2011. Predicting flu trends using twitter data. In *2011 IEEE conference on computer communications workshops (INFOCOM WKSHPS)*, pages 702–707. IEEE.

Dolan Antenucci, Michael Cafarella, Margaret Levenstein, Christopher Ré, and Matthew D Shapiro. 2014. Using social media to measure labor market flows. Technical report, National Bureau of Economic Research.

Josh Attenberg, Prem Melville, and Foster Provost. 2010. A unified approach to active dual supervision for labeling features and examples. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 40–55. Springer.

Josh Attenberg and Foster Provost. 2010. 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*, pages 423–432.

Francesco Barbieri, Jose Camacho-Collados, Luis Espinosa Anke, and Leonardo Neves. 2020. TweetEval: Unified benchmark and comparative evaluation for tweet classification. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1644–1650, Online. Association for Computational Linguistics.

Christos Baziotis, Nikos Pelekis, and Christos Doulkeridis. 2017. DataStories at SemEval-2017 task 4: Deep LSTM with attention for message-level and topic-based sentiment analysis. In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 747–754, Vancouver, Canada. Association for Computational Linguistics.

Richard P. Brent. 1971. An algorithm with guaranteed convergence for finding a zero of a function. *The Computer Journal*, 14(4):422–425.

Axel Bruns and Yuxian Eugene Liang. 2012. Tools and methods for capturing twitter data during natural disasters. *First Monday*.

Mikhail Burtsev, Alexander Seliverstov, Rafael Airapetyan, Mikhail Arkhipov, Dilyara Baymurzina, Nickolay Bushkov, Olga Gureenkova, Taras Khakhulin, Yuri Kuratov, Denis Kuznetsov, Alexey Litinsky, Varvara Logacheva, Alexey Lymar, Valentin Malykh, Maxim Petrov, Vadim Polulyakh, Leonid Pugachev, Alexey Sorokin, Maria Vikhreva, and Marat Zaynutdinov. 2018. DeepPavlov: Open-source library for dialogue systems. In *Proceedings of ACL 2018, System Demonstrations*, pages 122–127, Melbourne, Australia. Association for Computational Linguistics.

José Cañete, Gabriel Chaperon, Rodrigo Fuentes, Jou-Hui Ho, Hojin Kang, and Jorge Pérez. 2020. Spanish pre-trained bert model and evaluation data. In *PML4DC at ICLR 2020*.

Guanghua Chi, Fengyang Lin, Guangqing Chi, and Joshua Blumenstock. 2020. A general approach to detecting migration events in digital trace data. *PloS one*, 15(10):e0239408.

David Cohn, Les Atlas, and Richard Ladner. 1994. Improving generalization with active learning. *Machine learning*, 15(2):201–221.

Alexis Conneau, Kartikay Khandelwal, Naman Goyal, Vishrav Chaudhary, Guillaume Wenzek, Francisco Guzmán, Edouard Grave, Myle Ott, Luke Zettlemoyer, and Veselin Stoyanov. 2020. Unsupervised cross-lingual representation learning at scale. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8440–8451, Online. Association for Computational Linguistics.

Shantayanan Devarajan. 2013. Africa's statistical tragedy. *Review of Income and Wealth*, 59:S9–S15.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jesse Dodge, Gabriel Ilharco, Roy Schwartz, Ali Farhadi, Hannaneh Hajishirzi, and Noah Smith. 2020. Fine-tuning pretrained language models: Weight initializations, data orders, and early stopping. *arXiv preprint arXiv:2002.06305*.

Liat Ein-Dor, Alon Halfon, Ariel Gera, Eyal Shnarch, Lena Dankin, Leshem Choshen, Marina Danilevsky, Ranit Aharonov, Yoav Katz, and Noam Slonim. 2020. Active Learning for BERT: An Empirical Study. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 7949–7962, Online. Association for Computational Linguistics.

Lee Fiorio, Guy Abel, Jixuan Cai, Emilio Zagheni, Ingmar Weber, and Guillermo Vinué. 2017. Using twitter data to estimate the relationship between short-term mobility and long-term migration. In *Proceedings of the 2017 ACM on web science conference*, pages 103–110.

Arijit Ghosh Chowdhury, Ramit Sawhney, Puneet Mathur, Debanjan Mahata, and Rajiv Ratn Shah. 2019. Speak up, fight back! detection of social media disclosures of sexual harassment. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Student Research Workshop*, pages 136–146, Minneapolis, Minnesota. Association for Computational Linguistics.

Daniel Grießhaber, Johannes Maucher, and Ngoc Thang Vu. 2020. Fine-tuning BERT for low-resource natural language understanding via active learning. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 1158–1171, Barcelona, Spain (Online). International Committee on Computational Linguistics.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International Conference on Machine Learning*, pages 1321–1330. PMLR.

Morten Jerven. 2013. *Poor numbers: how we are misled by African development statistics and what to do about it*. Cornell University Press.

Ari Z Klein, Abeed Sarker, Haitao Cai, Davy Weissenbacher, and Graciela Gonzalez-Hernandez. 2018. Social media mining for birth defects research: a rule-based, bootstrapping approach to collecting data for rare health-related events on twitter. *Journal of biomedical informatics*, 87:68–78.

Yury Kryvasheyeu, Haohui Chen, Nick Obradovich, Esteban Moro, Pascal Van Hentenryck, James Fowler, and Manuel Cebrian. 2016. Rapid assessment of disaster damage using social media activity. *Science advances*, 2(3):e1500779.

David D Lewis and William A Gale. 1994. A sequential algorithm for training text classifiers. In *SIGIR'94*, pages 3–12. Springer.

Jiwei Li, Alan Ritter, Claire Cardie, and Eduard Hovy. 2014. Major life event extraction from Twitter based on congratulations/condolences speech acts. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1997–2007, Doha, Qatar. Association for Computational Linguistics.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A manually labelled multi-turn dialogue dataset. In *Proceedings of the Eighth International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 986–995, Taipei, Taiwan. Asian Federation of Natural Language Processing.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Mingyi Liu, Zhiying Tu, Zhongjie Wang, and Xiaofei Xu. 2020. Ltp: a new active learning strategy for bert-crf based named entity recognition. *arXiv preprint arXiv:2001.02524*.

Debanjan Mahata, Jasper Friedrichs, Rajiv Ratn Shah, and Jing Jiang. 2018. Detecting personal intake of medicine from twitter. *IEEE Intelligent Systems*, 33(4):87–95.

Stephen Mussmann, Robin Jia, and Percy Liang. 2020. On the Importance of Adaptive Data Collection for Extremely Imbalanced Pairwise Tasks. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 3400–3413, Online. Association for Computational Linguistics.

Dat Quoc Nguyen, Thanh Vu, and Anh Tuan Nguyen. 2020. BERTweet: A pre-trained language model for English tweets. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 9–14, Online. Association for Computational Linguistics.

Shriphani Palakodety, Ashiqur R KhudaBukhsh, and Jaime G Carbonell. 2020. Voice for the voiceless: Active sampling to detect comments supporting the rohingyas. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 454–462.

Joao Palotti, Natalia Adler, Alfredo Morales-Guzman, Jeffrey Villaveces, Vedran Sekara, Manuel Garcia Herranz, Musa Al-Asad, and Ingmar Weber. 2020. Monitoring of the venezuelan exodus through facebook's advertising platform. *Plos one*, 15(2):e0229175.

Daniela Pohl, Abdelhamid Bouchachia, and Hermann Hellwagner. 2018. Batch-based active learning: Application to social media data for crisis management. *Expert Systems with Applications*, 93:232–244.

Sumanth Prabhu, Moosa Mohamed, and Hemant Misra. 2021. Multi-class text classification using bert-based active learning. *arXiv preprint arXiv:2104.14289*.

Daniel Preoţiuc-Pietro, Vasileios Lampos, and Nikolaos Aletras. 2015. An analysis of the user occupational class through Twitter content. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1754–1764, Beijing, China. Association for Computational Linguistics.

Davide Proserpio, Scott Counts, and Apurv Jain. 2016. The psychology of job loss: using social media data

to characterize and predict unemployment. In *Proceedings of the 8th ACM Conference on Web Science*, pages 223–232.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Abeed Sarker, Maksim Belousov, Jasper Friedrichs, Kai Hakala, Svetlana Kiritchenko, Farrokh Mehryary, Sifei Han, Tung Tran, Anthony Rios, Ramakanth Kavuluru, et al. 2018. Data and systems for medication-related text classification and concept normalization from twitter: insights from the social media mining for health (smm4h)-2017 shared task. *Journal of the American Medical Informatics Association*, 25(10):1274–1283.

Jonathan Schler, Moshe Koppel, Shlomo Argamon, and James W Pennebaker. 2006. Effects of age and gender on blogging. In *AAAI spring symposium: Computational approaches to analyzing weblogs*, volume 6, pages 199–205.

Christopher Schröder, Andreas Niekler, and Martin Potthast. 2021. Uncertainty-based query strategies for active learning with transformers. *arXiv preprint arXiv:2107.05687*.

Burr Settles. 1995. Active learning literature survey. *Science*, 10(3):237–304.

Burr Settles. 2009. Active learning literature survey. Technical report, University of Wisconsin-Madison Department of Computer Sciences.

Artem Shelmanov, Vadim Liventsev, Danil Kireev, Nikita Khromov, Alexander Panchenko, Irina Fedulova, and Dmitry V Dylov. 2019. Active learning with deep pre-trained models for sequence tagging of clinical and biomedical texts. In *2019 IEEE International Conference on Bioinformatics and Biomedicine (BIBM)*, pages 482–489. IEEE.

Fábio Souza, Rodrigo Nogueira, and Roberto Lotufo. 2020. BERTimbau: pretrained BERT models for Brazilian Portuguese. In *9th Brazilian Conference on Intelligent Systems, BRACIS, Rio Grande do Sul, Brazil, October 20-23 (to appear)*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

Jorge A. Wagner Filho, Rodrigo Wilkens, Marco Idiart, and Aline Villavicencio. 2018. The brWaC corpus: A new open resource for Brazilian Portuguese. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, Miyazaki, Japan. European Language Resources Association (ELRA).

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, and Alexander Rush. 2020. Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 38–45, Online. Association for Computational Linguistics.

Michelle Yuan, Hsuan-Tien Lin, and Jordan Boyd-Graber. 2020. Cold-start active learning through self-supervised language modeling. In *Empirical Methods in Natural Language Processing*.

Justine Zhang, Ravi Kumar, Sujith Ravi, and Cristian Danescu-Niculescu-Mizil. 2016. Conversational flow in Oxford-style debates. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 136–141, San Diego, California. Association for Computational Linguistics.

Leihan Zhang and Le Zhang. 2019. An ensemble deep active learning method for intent classification. In *Proceedings of the 2019 3rd International Conference on Computer Science and Artificial Intelligence*, pages 107–111.

A Experimental details

A.1 Stratified sampling

We define seed motifs as either strings (e.g. "just got fired"), 2-grams (e.g. ("just", "hired")) or regexes (e.g. ("`(^|\W)looking[\w\s\d]* gig[\W]`").

To select initial seed motifs, we used the list of initial motifs elaborated by Antenucci et al. (2014). We also imposed extra requirements on additional motifs, such as the presence of first-person pronouns (e.g. "I got fired" for the "Lost Job" class), as we restricted the analysis to the author's own labor market situation. We also used adverbs such as "just" to take into account the temporal constraint for classes "Lost Job" and "Is Hired". For Mexican Spanish and Brazilian Portuguese motifs, we both translated the English motifs and asked native speakers to confirm the relevance of the translations and add new seeds (e.g. "chamba" is a Mexican Spanish slang word for "work"). We then ran a similar selection process.

For each of the candidate seed motif, we computed specificity and frequency on the random set R_e. For each class χ, we defined specificity for a given motif M as the share of positives for class χ in a random sample of 20 tweets from R_e that contain M. The frequency of motif M is defined as the share of tweets in R_e that contain M.

In order to have motifs that are both frequent and specific enough, we defined the following selection rule: we only retained motifs that have a specificity of or over 1% and for which the product of specificity and frequency is above 1.10^{-7}.

In total , we evaluated a total of 54 seeds for the US, 101 for Mexico and 42 for Brazil. After evaluation, we retained 26 seeds for the US, 26 for MX and 21 for Brazil. We report the retained motifs in Table 1.

A.2 Data labeling

To label unemployment-related tweets, we used the crowdsourcing platform Amazon Mechanical Turk. This platform has the advantage of having an international workforce speaking several languages, including Spanish and Brazilian Portuguese on top of English.

For each tweet to label, turkers were asked the five questions listed in Table 2. Each turker was presented with a list of 50 tweets and each labeled tweet was evaluated by at least two turkers. A turker could choose to answer either *yes*, *no* or, *I am not sure*. We included two attention check questions to exclude low-quality answers. Regarding the attention checks, we had the two following sentences labeled: "I lost my job today", which is a positive for class "Lost Job" and "Is Unemployed" and negative for the other classes, and "I got hired today", which is a positive for the class "Is Hired" and a negative for the other classes. We discarded answers of workers who didn't give the five correct labels for each quality check. To create a label for a given tweet, we required that at least two workers provided the same answer. A *yes* was then converted to a positive label, a *no* to a negative label, a tweet labeled by two workers as *unsure* was dropped from the sample.

During this labeling process, all workers were paid with an hourly income above the minimum wage in their respective countries. For a labeling task of approximately 15 minutes, turkers from the US, Mexico and Brazil received respectively 5USD, 5USD and 3USD.

A.3 Dataset description

A.3.1 Share of positives per class

We provide descriptive statistics on the share of positives per class in the stratified sample for each language in Table 3.

A.3.2 Class co-occurence

In this section, we provide an analysis of the extent to which each class is mutually exclusive. For this, we focus on the English initial stratified sample.

First, the classes "Is Unemployed", "Lost Job" and "Job Search" are non-mutually exclusive in many cases. As expected, the class "Lost Job" is highly correlated with the class "Is Unemployed" with 95% of Lost Job positives being also positives for "Is Unemployed" in the US initial stratified sample (e.g. "i lost my job on monday so i'm hoping something would help.", "as of today, for the first time in two years.....i am officially unemployed"). There are a few exceptions where users get hired quickly after being fired (e.g. "tfw you find a new job 11 days after getting laid off "). "Job Search" is also correlated with "Is Unemployed" (e.g. "I need a job, anyone hiring?"), though less than Lost Job, with 43% of positives being also positives for "Is Unemployed" in the initial stratified sample. Cases where users are looking for a job but are not unemployed include looking for a second job (e.g. "need a second job asap.") or looking for a better job while working (e.g. "tryna find a better

Class	English motifs	S_{EN}	F_{EN}	Spanish motifs	S_{SP}	F_{SP}	Portuguese motifs	S_{PT}	F_{PT}
Is Unemployed	• (i, unemployed)	0.45	9.6e-6	• estoy desempleado	0.75	1.6e-6	• estou desempregad[o/a]	0.65	6e-6
	• unemployed	0.15	7.4e-5	• sin empleo	0.05	1.4e-5	• (eu, sem, emprego)	0.15	3.6e-6
	• (i, jobless)	0.45	2.4e-6	• sin chamba	0.15	1e-5			
	• jobless	0.15	3.2e-5	• nini	0.15	4.9e-4			
	• unemployment	0.1	9e-5	• no tengo trabajo/ chamba/empleo	0.5	8.6e-6			
Lost Job	• (i, fired)	0.05	4.9e-5	• me despidieron	0.2	2.6e-6	• (perdi, emprego)	0.35	3e-6
	• i got fired	0.25	3.3e-6	• perdí mi trabajo	0.2	5.3e-7	• (perdi, trampo)	0.15	1.6e-6
	• just got fired	0.2	2e-6	• me corrieron	0.1	1.1e-5	• fui demitido	0.75	2.9e-6
	• laid off	0.2	1.2e-5	• me quedé sin trabajo/ /chamba/empleo	0.4	2.4e-6	• me demitiram	0.5	2.8e-7
	• lost my job	0.35	1.9e-6				• me mandaram embora	0.25	6.7e-7
				• ya no tengo trabajo/ /chamba/empleo	0.55	9.8e-7			
Job Search	• (anyone, hiring)	0.45	2e-6	• (necesito, trabajo)	0.7	2.5e-5	• (gostaria, emprego)	0.2	9.5e-7
	• (wish, job)	0.2	1.3e-5	• (necesito, empleo)	0.9	3.2e-6	• (queria, emprego)	0.45	1.5e-5
	• (need, job)	0.55	5.5e-5	• (busco, trabajo)	0.5	9e-6	• (preciso, emprego)	0.5	3.6e-5
	• (searching, job)	0.15	1.7e-6	• (buscando, trabajo)	0.45	1.7e-5	• (procurando, emprego)	0.25	1.5e-5
	• (looking, gig)	0.3	3.4e-6	• (alguien, trabajo)	0.1	3e-5			
	• (applying, position)	0.35	1.2e-6						
	• (find, job)	0.3	8.9e-5						
Is Hired	• (found, job)	0.25	6.2e-6	• (conseguí, empleo)	0.55	2.5e-5	• (consegui, emprego)	0.15	3e-5
	• (just, hired)	0.15	9.4e-6	• nuevo trabajo	0.75	3.4e-5	• fui contratad[o/a]	0.45	2.6e-6
	• i got hired	0.6	2e-6	• nueva chamba	0.45	3.3e-6	• (começo, emprego)	0.4	2.1e-6
	• (got, job)	0.45	7.6e-5	• (encontré, trabajo)	0.25	4.7e-6	• (novo, emprego/trampo)	0.25	4.1e-5
	• new job	0.25	8e-5	• (empiezo, trabajar)	0.4	4.5e-6	• primeiro dia de trabalho	0.65	1.3e-5
				• primer día de trabajo	0.55	2.3e-5			
Job Offer	• job	0.1	3e-3	• empleo	0.15	8.6e-4	• (enviar, curr[i/í]culo)	0.65	1.4e-5
	• hiring	0.2	5e-4	• contratando	0.35	2.9e-5	• (envie, curr[i/í]culo)	0.7	8e-6
	• opportunity	0.4	9.6e-4	• empleo nuevo	0.55	8.8e-7	• (oportunidade, emprego)	0.5	1.6e-5
	• apply	0.15	6.7e-4	• vacante	0.55	2e-4	• (temos, vagas)	0.45	1.5e-5
				• estamos contratando	0.9	9.7e-6			

Table 1: Initial motifs for each language and class. The use of parentheses indicate regexes matching all strings containing the words in the parentheses in the order in which they are indicated. A slash separating several words indicates that the regex will match any of the candidate words separated by slashes. For each motif M in country c, S_c and F_c are respectively M's specificity and frequency in the evaluation random sample R_e.

job"). There are also a few ambiguous cases where users mention that they are looking for a job but it is not clear whether they are unemployed (e.g. "job hunting") as well as edge cases where users just got hired but already are looking for another job (e.g. "i got hired at [company] but i don't like the environment any other suggestions for jobs ?"). For the class "Is Unemployed", mutually exclusive examples are cases where the user only mentions her unemployment, without mentioning a recent job loss or the fact that she is looking for a job (e.g. "well i'm jobless so there's that").

Second, the classes "Is Hired" and "Job Offer" are essentially orthogonal from one another and from the other classes. The class "Is Hired" (e.g. "good morning all. started my new job yesterday. everyone was awesome.") is almost always uncorrelated with the other classes apart from a few edge cases mentioned above. The class "Job Offer" (e.g. "we are #hiring process control/automation engineer job in atlanta, ga in atlanta, ga #jobs #atlanta") is almost always orthogonal to the other classes apart from a few exceptions. For instance, it can happen that a user who just got hired mentions job offers in her new company (e.g. "if you guys haven't been to a place called top golf i suggest you to go there or apply they are literally the best people ever i'm so happy i got hired").

We detail the class co-occurrence in the US initial stratified sample in Table 4.

A.3.3 Additional descriptive statistics

In this section, we include additional information about the US initial stratified sample. Table 5 contains information on average character length and most frequent tokens per class. Table 6 describes the Part-of-speech tag distribution in positives across classes.

A.4 Pre-trained language model characteristics

To classify tweets in different languages and as mentioned in Section 3.3, we used the following

Class	Question
Is Unemployed	Does the tweet indicate that the person who wrote the tweet is currently (at the time of tweeting) unemployed? For example, tweeting "Now I am unemployed", or "I just quit my job" is likely to indicate that the person who tweeted is currently unemployed.
Lost Job	Does this tweet indicate that the person who wrote the tweet became unemployed within the last month? For example, tweeting "I lost my job today", or "I was fired earlier this week" is likely to indicate that the person who tweeted became unemployed within the last month.
Job Search	Does this tweet indicate that the person who wrote the tweet is currently searching for a job? For example, tweeting "I am looking for a job", or "I am searching for a new position" is likely to indicate that the person who tweeted is currently searching for a job.
Is Hired	Does this tweet indicate that the person who wrote the tweet was hired within the last month? For example, tweeting "I just found a job", or "I got hired today" is likely to indicate that the person who tweeted was hired within the last month.
Job Offer	Does this tweet contain a job offer? For example, tweeting "Looking for a new position?", or "Here is a job opportunity you might be interested in" is likely to indicate that the tweet contains a job offer.

Table 2: List of questions asked to the Amazon Turkers when labelling each tweet

pre-trained language models from the Hugging Face model hub (Wolf et al., 2020):

– **Conversational BERT**[2] for English tweets, trained and released by Deep Pavlov (Burtsev et al., 2018). This model was initialized with BERT base cased weights and shares the same configuration. It was then further pre-trained using a masked language modeling objective on an English corpus containing social media data (Twitter and Reddit), dialogues (Li et al., 2017), debate transcripts (Zhang et al., 2016), movie subtitles (Lison and Tiedemann, 2016) as well as blog posts (Schler et al., 2006).

– **BETO** for Spanish tweets (Cañete et al., 2020). This model has a BERT-base architecture and was pre-trained from scratch on a Spanish corpus derived from Wikipedia and the Spanish part of the OPUS project (Tiedemann, 2012).

– **BERTimbau** for Brazilian Portuguese tweets (Souza et al., 2020). This model also has a BERT-base architecture and was pre-trained from scratch on a large multi-domain Brazilian Portuguese corpus called brWaC (Wagner Filho et al., 2018).

All three language models have 110 million parameters.

When it comes to the choice of language models for each language, the emerging literature considering language model pre-training on tweets to improve downstream tasks in the Twitter context gave us several potential candidates for English tweet classification. On top of Conversational BERT, we experimented with BERTweet (Nguyen et al., 2020), which is the leader on the TweetEval leaderboard[3] as of March 2022 (Barbieri et al., 2020). We also tested the performance of renowned pre-trained language models such as BERT base and RoBERTa base. We found that both Conversational BERT and BERTweet outperformed these well-known models for our task. Also, while BERTweet usually slightly outperformed Conversational BERT on the test set from the stratified sample in terms of AUROC, it had a worse performance on the random set R_e. This is why we chose Conversational BERT for English tweets.

For Spanish and Brazilian Portuguese tweets, in the absence of Twitter-specialized language models, we opted for the best performing pre-trained language models as of Fall 2020 for these languages, namely BETO for Spanish and BERTimbau for Brazilian Portuguese. We also experimented with multilingual language models, such as XLM-RoBERTa (Conneau et al., 2020), but the monolingual approaches for Spanish and Brazilian Portuguese were performing better, both on the test set from the stratified sample and on the random set.

A.5 Fine-tuning and evaluation

As mentioned in 3.3 and following Dodge et al. (2020), we fine-tuned each BERT-based model with 15 different seeds and for 20 epochs. We evaluated the models 10 times per epoch and use early stopping with a patience of 11. We used a training and evaluation batch size of 8. The best model is defined as the best performing model in terms of area under the ROC curve (AUROC) on the evaluation set, at or after the first epoch.

As described in Algorithm 1, we then ran the inference of the best model on both random sets R_e and R_s. To speed up this inference process, we converted the PyTorch models to ONNX.

In terms of computing infrastructure, we used either V100 (32GB) or RTX8000 (48GB) GPUs for the fine-tuning and parallelize inference over 2000

[2]Available at https://huggingface.co/DeepPavlov/bert-base-cased-conversational

[3]The current leaderboard can be found here: https://github.com/cardiffnlp/tweeteval

Language	Label	Class				
		Lost Job	Is Hired	Is Unemployed	Job Offer	Job Search
English	yes	270	334	796	600	524
	no	4239	4181	3710	3918	3993
	unsure	15	9	18	6	7
Spanish	yes	213	388	1116	515	659
	no	3488	3331	2579	3210	3059
	unsure	28	10	34	4	11
Portuguese	yes	175	422	925	485	614
	no	2514	2272	1761	2215	2084
	unsure	14	9	17	3	5

Table 3: Label distribution on the stratified sample for each country and class

Class	Share of positives per class (in %)				
	Is Unemployed	Lost Job	Job Search	Is Hired	Job Offer
Is Unemployed	100	32	28	1.3	0
Lost Job	95	100	10	4	0
Job Search	43	5	100	2	0
Is Hired	3.2	3.2	2.9	100	2.3
Job Offer	0	0	0	1.3	100

Table 4: Class co-occurrence in the US initial stratified sample. It reads as follows: out of all positives for the Is Unemployed class, 32% are positives for Lost Job.

Class	Average length	Top 10 most common tokens									
		1	2	3	4	5	6	7	8	9	10
Is Unemployed	105	i	job	a	to	my	and	the	for	fired	got
Lost Job	103	i	my	got	fired	job	just	a	to	and	the
Job Search	96	i	a	job	for	to	the	anyone	and	hiring	in
Is Hired	99	i	job	a	got	the	my	and	new	hired	to
Job Offer	128	job	a	for	in	jobs	hiring	to	at	the	##q

Table 5: Average character length and top 10 most frequent tokens for each class in the initial US stratified sample

CPU nodes. The average runtime for fine-tuning and evaluation on the one hand and inference on the other hand is respectively of 45 minutes and 3 hours.

A.6 Performance at iteration 0

We report detailed AUROC results on the test set from the stratified sample in Table 7.

A.7 Evaluation metrics

In this section, we detail the evaluation process. The values of each metric across iterations for each language and each method can respectively be found in Table 10 and 11.

A.7.1 Sampling for evaluation

As mentioned in Section 3.4, for each country, AL strategy, iteration and class, we labeled 200 tweets along the BERT confidence score distribution. This tweet selection overweighted the top of the score distribution. Specifically, we retained tweets with the following ranks in the score distribution: 1-20; 101-110; 317-326; 1,001-1,010; 2,155-2,164; 4,642-4,651; 10,001-10,010; 17,783-17,792; 31,623-31,632; 56,235-56,244; 100,001-100,010; 158,490-158,499; 251,189-251,198; 398,108-398,117; 630,958-630,967; 1,000,001-1,000,010.

POS tag	Share per class (in %)				
	Is Unemployed	Lost Job	Job Search	Is Hired	Job Offer
ADJ	7.18	6.21	6.92	7.95	8.15
ADP	7.43	7.61	8.27	7.67	9.46
ADV	6.85	8.47	6.13	6.78	3.75
AUX	8.60	9.52	6.96	7.86	5.30
CCONJ	4.23	4.02	3.50	4.58	2.89
DET	6.71	5.82	8.95	7.86	6.81
INTJ	1.85	2.26	1.53	1.45	0.72
NOUN	9.73	9.64	10.61	10.22	11.00
NUM	2.07	2.07	1.66	2.24	2.74
PART	4.54	4.06	3.96	3.85	2.61
PRON	9.97	10.07	9.90	9.40	5.56
PROPN	5.00	5.19	4.31	5.96	8.37
PUNCT	8.54	8.59	9.55	8.36	10.39
SCONJ	4.01	3.01	3.69	2.40	1.38
SPACE	1.13	1.05	1.04	1.14	2.19
SYM	1.27	1.29	1.37	1.07	5.89
VERB	10.02	10.23	10.82	10.22	10.28
X	0.85	0.90	0.83	0.98	2.52

Table 6: Part-of-Speech (POS) tag distribution among positives of each class from the initial US stratified sample. The definition of the acronyms can be found here.

Language	Model	Class				
		Lost Job	Is Hired	Is Unemployed	Job Offer	Job Search
English	Conversational BERT	0.959	0.976	0.965	0.985	0.98
Spanish	BETO	0.944	0.98	0.949	0.993	0.959
Portuguese	BERTimbau	0.978	0.973	0.949	0.991	0.971

Table 7: AUROC results on the evaluation set at iteration 0.

A.7.2 Average Precision

With the retained tweets, we computed the Average Precision (AP) at each iteration and for each class and language. We used the standard definition of AP in information retrieval and defined AP at iteration i for class c and method m as:

$$AP_{i,c,m} = \frac{\sum_{r \in R_{i,c,m}} P(r) \times pos(r)}{N_{i,c,m}}$$

where:

- $R_{i,c,m}$ is the ensemble of ranks in the confidence score distribution of class c at iteration i and for method m of all tweets sampled for evaluation and labeled for class c and method m both at iteration i and preceding iterations
- $P(r)$ is the share of positives in sampled tweets with rank at iteration i and for class c inferior or equal to r
- $pos(r)$ is equal to 1 if tweet ranked r for iteration i and class c is positive and 0 otherwise
- $N_{i,c,m}$ is the number of tweets sampled and labeled for class c and method m both at iteration i and preceding iterations

A.7.3 Number of predicted positives

We defined the number of predicted positives E as the average rank in the confidence score distribution when the share of positives reaches 0.5. In practice, for each iteration i and class c and the related BERT model M, we first ranked the evaluation set R_e according the prediction scores from M. We then binned the evaluation labels of each

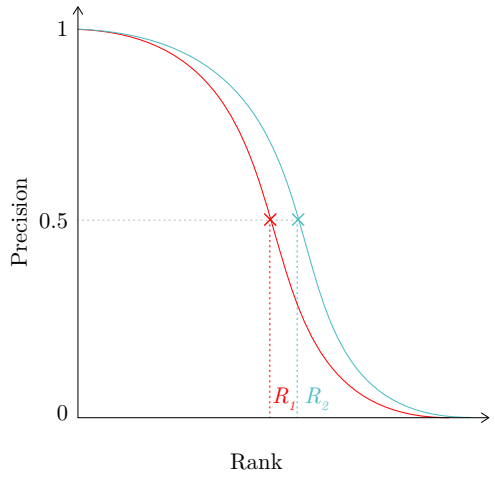

Figure 4: Illustration of the procedure used to determine the number of predicted positives. In this example, the number of predicted positives is R_1 for iteration 1 and R_2 for iteration 2.

iteration until i into 20 bins of equal size, and we estimated the proportion of positives in each bin and the average rank of each bin. We then identified the first bin for which the proportion of positive labels reaches 0.5. We estimated an upper and a lower bound for E by taking the average rank of tweets included in the bin above and below the 0.5 cutoff respectively, and we estimated E as the midpoint between its lower bound and its upper bound estimate. For each round, we report E as well as its lower and upper bound estimates. We provide an illustration of this procedure in Figure 4.

By convention, the number of predicted positives is equal to 1 when the proportion of positive labels sampled from the evaluation set remains below 0.5 for all ranks.

A.7.4 Diversity of true positives

To compute diversity for a given iteration i and class c, we first encoded all positive tweets sampled for the evaluation of class c at iteration i as well as preceding iterations into sentence embeddings (Reimers and Gurevych, 2019). To do so, we used the "all-mpnet-base-v2" model for English and the "paraphrase-multilingual-mpnet-base-v2" model for Spanish and Portuguese (Reimers and Gurevych, 2020). These models are in open source access on the sentence-transformers GitHub repository[4].

After computing the embeddings, we defined the diversity rate in a set of positive tweets as the mean pairwise distance between all possible pairs in this set. The pairwise distance between tweet A and B is defined as $1 - sim(E_A, E_B)$ where sim is a cosine similarity function and E_A and E_B are the sentence embeddings for tweets A and B. By convention, diversity is equal to 0 when there is no more than 1 positive label.

A.7.5 Standard error computation

For average precision and diversity, we derived standard errors by using bootstrap samples on the pool of N tweets used to compute the metric. We sampled with replacement N tweets in this pool and repeated the process 1000 times. We then computed the metric for each of these samples and finally computed the mean and the standard error.

For the number of predicted positives, our method does not allow to directly use bootstrap. We therefore computed the upper and lower bound as described in Section A.7.3.

A.7.6 Convergence

As stated in Section 3.5, we considered that an AL strategy had converged when there was no significant variation of average precision, number of predicted positives and diversity for at least two iterations.

To determine whether there is a significant variation in average precision and diversity from one iteration to the next, we performed t-tests. For the number of predicted positives, since we could only estimate an upper and lower bound, we considered that there was no significant variation from one iteration to the next if the interval between the lower

[4] https://github.com/UKPLab/sentence-transformers

bound and the upper bound overlapped from one iteration to the next.

We report in bold the metric values at convergence in Table 10 and 11.

A.8 Exploit-explore retrieval algorithm

In this section, we detail the functioning of the new AL strategy we coin exploit-explore retrieval in Algorithm 2.

We define the k-skip-n-grams used in this approach as follows: for a given text sequence T, the set of k-skip-n-grams, with k a positive integer and n in $\{2; 3\}$, is made of all the ordered combinations of n words in T. For instance, for $T =$ "I am very happy", the set of k-skip-2grams is: { (I, am), (I, very), (I, happy), (am, very), (am, happy), (very, happy)}. The k blanks do not need to be successive. To define the k-skip-n-grams contained in tweets, each tweet was tokenized using the *ekphrasis* package (Baziotis et al., 2017).

To decide on the 10^4 threshold for top tweets, we estimated the base rate for each class and country. We defined the base rate for a given class as the share of positives for this class in the whole sample of tweets. To estimate this base rate for each class and country, we computed the specificity and frequency of each initial motif (listed in Table 1) and defined the base rate estimate as the sum over each motif of the motif's frequency weighted by its specificity. We detail the estimation results in Table 8.

The base ranks in our random sample of 100 million tweets R_e (ie: base rate multiplied by 10^8) ranged from 10^2 to 10^5 with a majority below 10^4 in Mexico and Brazil. We tried $T = 10^3$, $T = 10^4$ and $T = 10^5$ as candidate thresholds for the top tweets and they gave very similar results for the k-skip-n-grams used in the exploration step. We finally chose 10^4 to balance between higher base ranks in the US and lower base ranks elsewhere. Our choice for the other hyperparameters were dictated by our budget constraint.

For illustration of the exploration part of this method, we detail the top-lift k-skip-n-grams selected from US tweets, for each iteration and for each class, in Table 9.

A.9 Calibration for uncertainty sampling

In order to calibrate the BERT confidence scores to do uncertainty sampling, we proceeded in the following way.

For each country, AL strategy and class, we used the 200 tweets we retained along the confidence score distribution on R_s and labeled for evaluation. From this labeled set, we built 10.000 balanced bootstrap samples and fit a logistic regression to each of these samples. We therefore obtained a set of 10.000 logistic regression parameter pairs $((\beta_{0,i}, \beta_{1,i}))_{i\in[1,10.000]}$. We then used this set of parameters to find the BERT confidence score x^* for which its calibrated version is equal to 0.5. To do so, we used Brent's method (Brent, 1971) and defined x^* as the root of the following function:

$$\frac{\sum_{i=1}^{10.000} \sigma(\beta_{0,i} + \beta_{1,i}x)}{10.000} - 0.5$$

where σ is a standard logistic function.

Knowing x^*, we were then able to perform uncertainty sampling by sampling tweets with confidence scores around x^*.

A.10 Additional experimental results

In this section, we report additional experimental results on precision and average precision.

We report precision for the exploit-explore retrieval strategy across countries in Figure 5 and for the four AL stategies on English tweets in Figure 6.

Also, we detail the evaluation results for the exploit-explore retrieval strategy across countries in Table 10 and for the four AL stategies on English tweets in Figure 11.

Initialization:
$\forall k \in \mathbb{N}^*$ and n=2,3, determine all ordered k-skip-n-grams in the random set R_s used to sample tweets for labelling. This results in a set of 2-grams S_2 and 3-grams S_3;
For n=2,3, discard all k-skip-n-grams from S_n that:
- contain one-grams made of at least one subtoken that is not in the BERT model vocabulary
- contain at least one repetition (e.g. (i, i, job))
- that have a frequency lower than 1 in 100K

At each iteration i:
Discard tweets that were sampled and labeled at iteration $i-1$ from R_s ;
For each class χ:

- Run inference on R_s with the best BERT-based classifier for class χ
- **Exploitation**: sample 50 tweets from the set of top 10,000 tweets in terms of confidence score assigned by the BERT-based classifier
- **Exploration**: for n=2,3,
  - Compute lift for each k-skip-n-gram in S_n
  - Discard all k-skip-n-grams from S_n that
    (1) were used to sample tweets for class χ at iteration $i-1$ and/or
    (2) have at least one one-gram in common with another k-skip-n-gram.
    Only the k-skip-n-gram with the highest lift is kept.
  - Select 5 top-lift k-skip-n-grams in S_n
  - For each retained top-lift k-skip-n-gram, sample 5 tweets in R_s containing this motif

Label sampled tweets for each class;
Add new sampled tweets to the set of all labels;
Perform new train-test split on this set and use this split to train and evaluate the classifier for the next iteration;

Algorithm 2: Exploit-explore retrieval

Language	Class	Base rate
English	Is Hired	3.03×10^{-4}
English	Is Unemployed	2.16×10^{-4}
English	Job Offer	5.38×10^{-3}
English	Job Search	4.8×10^{-4}
English	Lost Job	2.04×10^{-5}
Spanish	Is Hired	5.64×10^{-5}
Spanish	Is Unemployed	8.16×10^{-5}
Spanish	Job Offer	2.58×10^{-4}
Spanish	Job Search	3.55×10^{-5}
Spanish	Lost Job	1.46×10^{-5}
Portuguese	Is Hired	4.82×10^{-5}
Portuguese	Is Unemployed	7.51×10^{-5}
Portuguese	Job Offer	4.59×10^{-5}
Portuguese	Job Search	7.57×10^{-5}
Portuguese	Lost Job	3.91×10^{-6}

Table 8: Estimated base rate for each country and class.

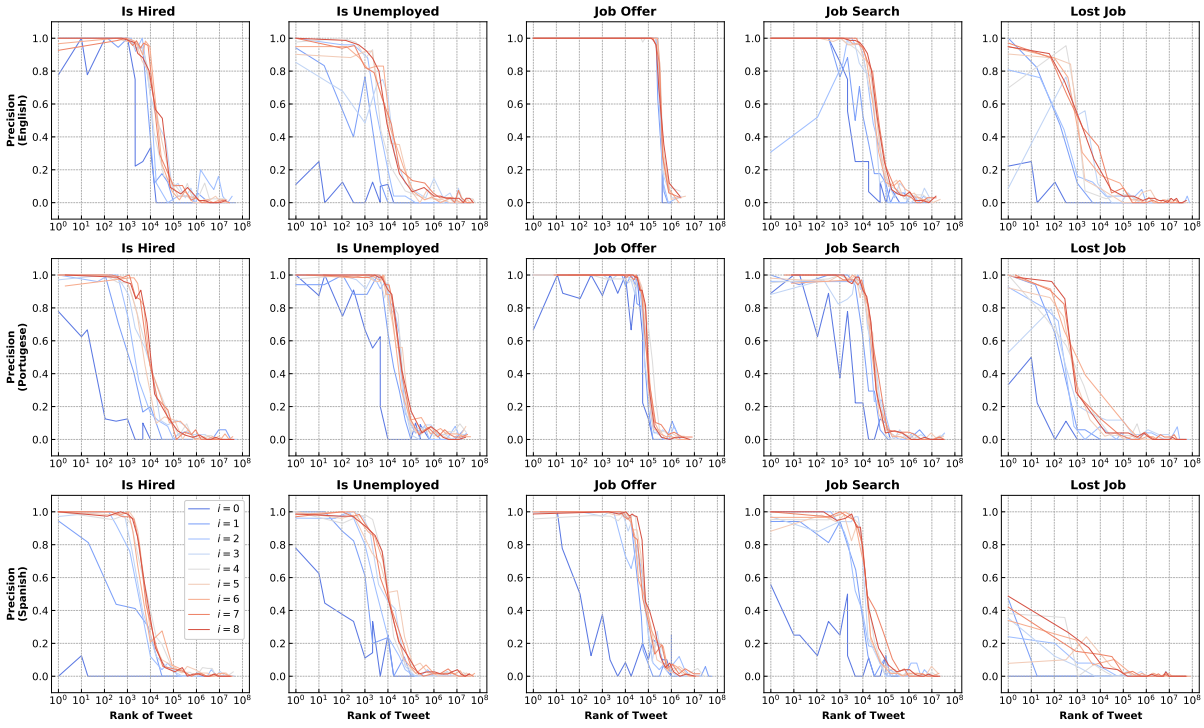

Figure 5: Precision (y-axis) as a function of tweet rank based on confidence score (i.e. positive label probability output by the model) (x-axis). For each language (in row) and class (in column), we ranked the tweets from the evaluation random set R_e by their confidence score assigned by the BERT-based classifiers in descending order. We then sampled tweets along the rank distribution and labeled them. Each marker corresponds to a sample of 10 labeled tweets. Colors encode successive iterations of AL from 0 (blue) to 8 (red).

Class	Iteration 1	Iteration 2	Iteration 3	Iteration 4
Is Unemployed	• (got, headache) • (having, breakdown) • (lost, voice) • (im, losing) • (m, depressed) • (got, a, headache) • (am, having, attack) • (i, lost, phone) • (im, in, need) • (losing, my, mind)	• (am, homeless) • (lost, job) • (need, broke) • (unemployed, a) • (been, single, for) • (homeless, in, to) • (i, am, unemployed) • (really, need, job)	• (got, fired) • (in, desperately) • (job, hunting) • (laid, i) • (unemployed, and) • (got, fired, the) • (have, no, life) • (laid, off, to) • (lost, my, up) • (need, job, i)	• (am, clueless) • (i, homeless) • (im, broke) • (in, limbo) • (unemployed, to) • (i, am, homeless) • (laid, off, and) • (m, broke, to • (need, job ,can) • (strong, have, been)
Lost Job	• (broke, today) • (fell, bed) • (got, hospital) • (just, kicked) • (lost, yesterday) • (got, kicked, of) • (i, lost, today) • (just, pulled, over) • (out, the, hospital) • (phone, last, night)	• (fired, me) • (got, laid) • (unfollowed, checked) • ([, by,]) • (am, sick, again) • (and, me, checked) • (quit, my, job) • (to, i, fired)	• (been, sick) • (fired, my) • (got, banned) • (just, cancelled) • (worked, days) • (been, sick, for) • (don, have, weekend) • (i, fired, my) • (today, bad, day)	• (just, fired) • (now, pissed) • (today, sucked) • (unemployed, for) • (i, was, fired) • (just, went, from) • (my, job, today) • (now, am, pissed)
Job Search	• (any, places) • (job, asap) • (know, hiring) • (need, second) • (new, suggestions) • (if, anyone, knows) • (am, for, jobs) • (hiring, i, a) • (need, new, job) • (something, do, tonight)	• (any, jobs) • (interview, wish) • (job, anyone) • (knows, let) • (need, hiring) • (a, second, job) • (am, any, suggestions) • (got, an, interview) • (i, looking, anyone) • (knows, me, how)	• (applying, i) • (interview, tomorrow) • (job, luck) • (please, pls) • (anyone, knows, of) • (have, wish, luck) • (job, need, i) • (places, that, are) • (to, interview, me)	• (applying, for) • (interview, get) • (need, paying) • (second, job) • (that, hiring) • (got, a, interview) • (hope, get, job) • (i, need, second)
Is Hired	• (first, nervous) • (got, hired) • (job, excited) • (new, woot) • (start, tomorrow) • (finally, a, phone) • (i, hired, and) • (start, my, new) • (the, job, got) • (tomorrow, first, day)	• (got, accepted) • (hired, at) • (start, job) • (started, weeks) • (tomorrow, nervous) • (first, at, new) • (job, i, got) • (start, my, tomorrow) • (started, a, ago)	• (excited, job) • (hired, on) • (start, new) • (first, at, day) • (hired, to, a) • (i, job, got) • (start, tomorrow, and) • (starting, my, new)	• (got, $$) • (hired, for) • (i, promoted) • (job, tomorrow) • (first, day, new) • (it, can, oh) • (just, call, from) • (start, my, job)
Job Offer	• (apply, arc) • (click, jobs) • (recommend, career) • (anyone, retail) • (technician, hiring) • (click, apply, job) • (now, developer, in) • (recommend, anyone, this) • (we, jobs, career)	• (hiring, view) • (it, analyst) • (job, details) • (manager, apply) • (position, open) • (hiring, it, details) • (is, apply, jobs) • (job, analyst, view) • (now, opportunities, in) • (we, are, assistant)	• (apply, career) • (click, job) • (hiring, hospitality) • (view, details) • (we, arc) • (hiring, to, career) • (it, view, details) • (now, manager, in) • (we, apply, job)	• (apply, retail) • (are, hospitality) • (click, arc) • (job, circle) • (needed, hiring) • (are, apply, career) • (click, job, jobs) • (manager, new, york) • (now, hiring, circle) • (we, to, arc)

Table 9: Top-lift k-skip-n-grams for each class and iteration of the Explore-Exploit Retrieval on US tweets. The fact that not all (class, iteration) pair have 10 k-skip-n-grams is explained by the fact that some set of tweets containing a top-lift k-skip-n-gram could not be labeled because of disagreement between crowdworkers on the right label to assign.

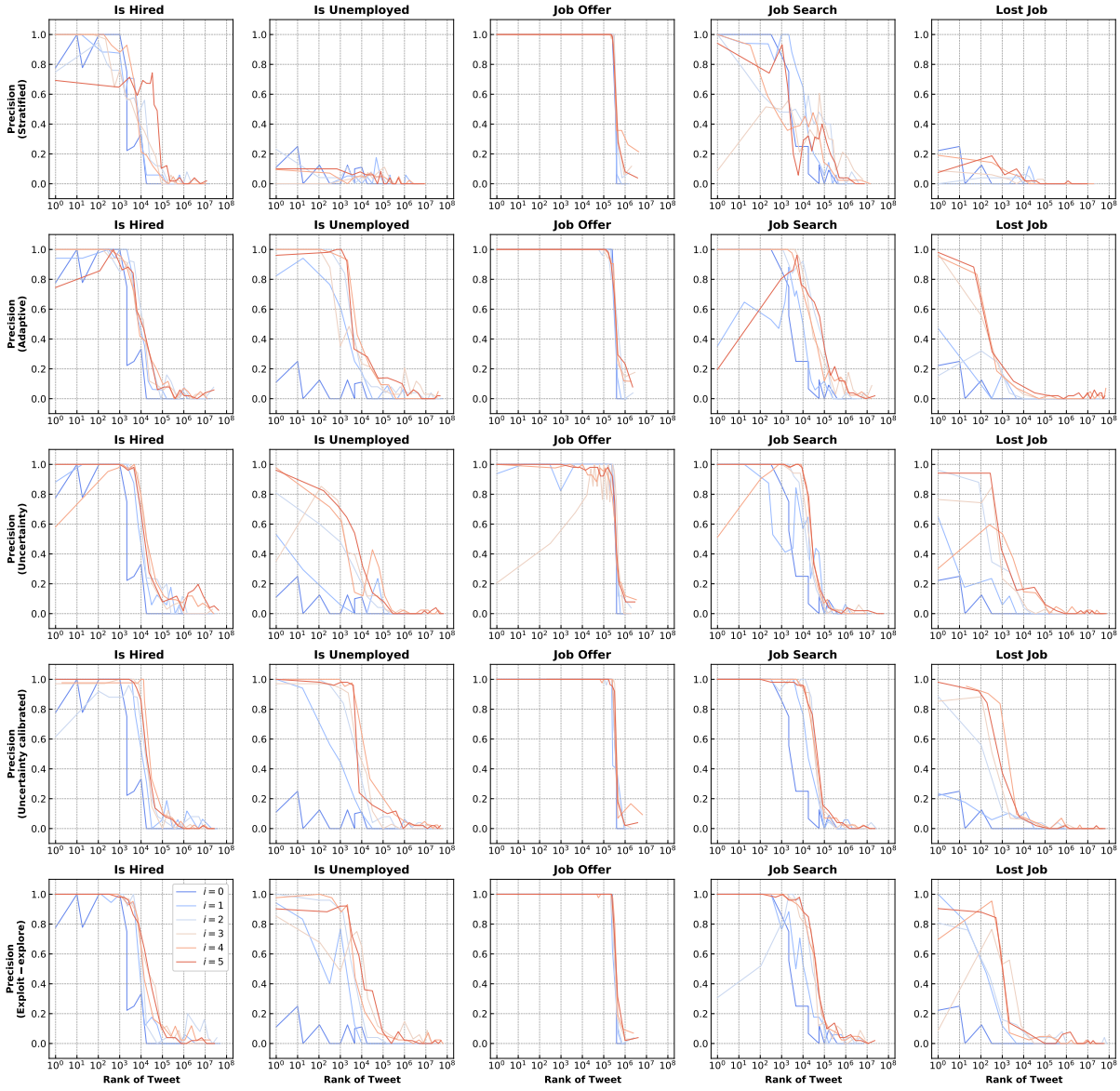

Figure 6: Precision (y-axis) as a function of tweet rank based on confidence score (i.e. positive label probability output by the model (x-axis)). For each AL strategy (in row) and class (in column), we ran the same process as the one described in Figure 5. Colors encode successive iterations of AL from 0 (blue) to 5 (red).

			i=0	i=1	i=2	i=3	i=4	i=5	i=6	i=7	i=8
IH	P	EN	83.0 (3.8)	96.1 (0.7)	94.3 (1.1)	**96.3 (0.6)**	96.0 (0.7)	96.5 (0.5)	95.7 (0.9)	93.8 (1.3)	96.2 (0.4)
IH	P	PT	68.2 (7.6)	90.6 (1.7)	93.7 (1.2)	91.0 (1.8)	93.9 (0.8)	**94.9 (0.7)**	91.5 (1.6)	95.8 (0.6)	95.3 (0.6)
IH	P	ES	7.1 (4.7)	80.4 (5.1)	92.4 (1.4)	93.8 (1.7)	**91.5 (1.7)**	94.6 (0.9)	95.4 (0.5)	96.2 (0.4)	96.3 (0.7)
IH	E	EN	[2.2e3, 4.4e3]	[1.0e4, 2.2e4]	[1.3e4, 3.2e4]	**[1.5e4, 3.2e4]**	[9.9e3, 2.1e4]	[2.2e4, 5.0e4]	[2.1e4, 5.4e4]	[1.8e4, 5.1e4]	[4.9e4, 1.3e5]
IH	E	PT	[7.5e1, 2.1e2]	[9.2e2, 3.3e3]	[1.9e3, 6.6e3]	[8.6e3, 1.8e4]	[7.7e3, 2.1e4]	**[4.1e3, 1.0e4]**	[6.9e3, 1.7e4]	[1.2e4, 3.8e4]	[1.0e4, 3.0e4]
IH	E	ES	[1, 1]	[1.7e2, 9.4e2]	[2.3e3, 7.3e3]	[2.9e3, 1.1e4]	**[4.9e3, 1.1e4]**	[7.7e3, 1.9e4]	[4.6e3, 1.9e4]	[4.5e3, 1.1e4]	[8.7e3, 2.7e4]
IH	D	EN	43.2 (2.1)	42.6 (1.6)	46.5 (1.2)	**46.8 (1.0)**	46.5 (0.9)	45.8 (0.9)	47.3 (0.9)	48.6 (0.7)	48.1 (0.7)
IH	D	PT	30.8 (3.2)	37.2 (1.9)	38.9 (1.3)	42.2 (1.1)	43.2 (1.0)	**43.1 (0.9)**	43.7 (0.9)	44.7 (0.8)	44.7 (0.7)
IH	D	ES	0.0 (0.0)	48.4 (1.7)	43.8 (1.6)	43.1 (1.3)	**42.6 (1.1)**	42.6 (1.0)	41.4 (0.9)	41.3 (0.9)	40.7 (0.8)
IU	P	EN	14.5 (5.9)	82.7 (4.5)	94.2 (1.4)	70.3 (2.7)	92.8 (1.3)	85.4 (2.0)	90.2 (1.4)	**89.8 (1.2)**	93.2 (1.0)
IU	P	PT	86.8 (3.2)	91.4 (2.7)	96.4 (0.6)	95.6 (0.6)	**95.5 (0.7)**	95.9 (0.5)	96.8 (0.5)	96.8 (0.5)	97.7 (0.2)
IU	P	ES	59.8 (8.0)	91.5 (1.9)	93.2 (2.3)	94.0 (1.2)	**95.1 (1.3)**	90.9 (1.4)	94.7 (0.9)	94.2 (0.9)	93.7 (0.9)
IU	E	EN	[1, 1]	[1.8e3, 6.7e3]	[4.8e3, 1.3e4]	[1.5e4, 3.7e4]	[3.3e3, 9.2e3]	[1.0e4, 2.2e4]	[8.7e3, 2.4e4]	**[9.5e3, 2.2e4]**	[6.2e3, 1.7e4]
IU	E	PT	[4.6e3, 8.9e3]	[1.4e4, 2.5e4]	[4.5e4, 1.1e5]	[2.9e4, 7.0e4]	**[3.4e4, 7.3e4]**	[6.1e4, 1.3e5]	[2.6e4, 6.0e4]	[3.0e4, 6.3e4]	[2.9e4, 6.5e4]
IU	E	ES	[1.4e1, 7.5e1]	[1.6e3, 5.3e3]	[6.2e3, 1.8e4]	[6.5e3, 1.5e4]	**[1.1e4, 3.2e4]**	[3.7e4, 8.1e4]	[1.0e4, 3.0e4]	[8.2e3, 2.1e4]	[8.2e3, 1.8e4]
IU	D	EN	32.2 (4.9)	52.1 (1.7)	59.4 (1.1)	61.4 (0.9)	62.3 (0.8)	62.2 (0.7)	62.6 (0.6)	**62.8 (0.6)**	62.5 (0.6)
IU	D	PT	45.1 (2.3)	51.8 (1.2)	50.0 (1.1)	51.5 (0.9)	**50.5 (0.9)**	51.2 (0.8)	52.0 (0.6)	51.6 (0.6)	51.4 (0.6)
IU	D	ES	40.1 (3.0)	42.8 (1.7)	44.7 (1.3)	46.2 (1.0)	**46.5 (0.8)**	47.2 (0.8)	48.2 (0.8)	48.5 (0.7)	48.9 (0.6)
JO	P	EN	95.1 (0.5)	97.0 (0.3)	98.9 (0.1)	99.3 (0.1)	98.6 (0.1)	99.3 (0.1)	99.4 (0.1)	**99.5 (0.1)**	97.5 (0.1)
JO	P	PT	83.4 (4.3)	95.0 (0.5)	98.3 (0.3)	97.3 (0.4)	96.3 (0.5)	98.3 (0.2)	98.7 (0.1)	**98.5 (0.2)**	99.1 (0.1)
JO	P	ES	73.4 (4.6)	96.6 (0.5)	94.8 (0.9)	95.6 (0.6)	91.9 (1.3)	96.9 (0.5)	96.9 (0.4)	**96.8 (0.3)**	96.5 (0.5)
JO	E	EN	[2.5e5, 4.0e5]	[3.6e5, 6.0e5]	[3.5e5, 5.9e5]	[3.5e5, 6.7e5]	[3.8e5, 5.9e5]	[3.6e5, 6.5e5]	[3.6e5, 5.9e5]	**[3.6e5, 5.8e5]**	[3.6e5, 6.1e5]
JO	E	PT	[5.6e4, 9.5e4]	[6.1e4, 8.8e4]	[9.9e4, 1.5e5]	[5.6e4, 8.6e4]	[1.2e5, 1.7e5]	[9.0e4, 1.3e5]	[9.4e4, 1.5e5]	**[8.7e4, 1.5e5]**	[9.0e4, 1.4e5]
JO	E	ES	[2.1e2, 4.9e2]	[3.0e4, 5.6e4]	[8.2e4, 1.3e5]	[6.5e4, 1.1e5]	[7.5e4, 1.3e5]	[5.8e4, 9.3e4]	[4.9e4, 9.8e4]	**[4.5e4, 8.0e4]**	[6.1e4, 9.2e4]
JO	D	EN	49.9 (0.9)	50.3 (0.6)	50.4 (0.5)	50.1 (0.5)	50.2 (0.5)	50.3 (0.4)	50.7 (0.4)	**50.5 (0.3)**	50.6 (0.3)
JO	D	PT	56.1 (1.2)	56.6 (0.8)	55.8 (0.6)	55.7 (0.5)	56.7 (0.5)	56.1 (0.5)	56.2 (0.4)	**56.2 (0.4)**	55.6 (0.4)
JO	D	ES	43.4 (2.1)	49.6 (1.0)	51.1 (0.8)	51.4 (0.7)	52.2 (0.6)	52.4 (0.5)	51.5 (0.5)	**52.0 (0.5)**	51.6 (0.5)
JS	P	EN	92.4 (1.2)	90.8 (1.6)	67.3 (3.1)	95.7 (0.5)	96.3 (0.5)	**96.6 (0.4)**	97.0 (0.3)	97.6 (0.2)	97.7 (0.2)
JS	P	PT	84.1 (3.9)	95.2 (1.2)	92.9 (2.2)	86.9 (2.3)	95.4 (1.0)	95.8 (1.0)	**94.7 (1.3)**	96.6 (0.5)	97.7 (0.3)
JS	P	ES	38.6 (7.3)	88.4 (2.5)	91.5 (1.6)	**93.0 (1.7)**	93.9 (1.6)	89.0 (1.9)	94.8 (1.1)	95.6 (0.7)	96.0 (0.6)
JS	E	EN	[4.4e3, 9.1e3]	[1.6e4, 2.7e4]	[1.9e4, 3.9e4]	[3.3e4, 7.7e4]	[5.4e4, 1.1e5]	**[4.3e4, 8.6e4]**	[3.6e4, 7.0e4]	[3.7e4, 6.9e4]	[5.9e4, 1.2e5]
JS	E	PT	[2.2e3, 4.6e3]	[1.2e4, 2.3e4]	[2.6e4, 4.7e4]	[3.8e4, 7.6e4]	[3.5e4, 7.0e4]	[3.4e4, 7.0e4]	**[3.3e4, 6.9e4]**	[3.7e4, 7.2e4]	[2.9e4, 6.3e4]
JS	E	ES	[2.2e3, 4.3e3]	[9.3e3, 2.1e4]	[6.2e3, 1.1e4]	**[1.1e4, 2.7e4]**	[2.0e4, 5.3e4]	[1.7e4, 4.0e4]	[1.6e4, 3.8e4]	[1.2e4, 2.3e4]	[1.3e4, 2.9e4]
JS	D	EN	49.6 (2.2)	54.2 (1.6)	60.2 (1.3)	60.6 (0.9)	60.3 (0.8)	**60.4 (0.7)**	60.7 (0.6)	61.7 (0.5)	62.1 (0.5)
JS	D	PT	35.3 (2.0)	39.5 (1.7)	45.2 (1.4)	46.3 (1.3)	45.1 (1.1)	44.0 (1.0)	**45.4 (0.9)**	44.8 (0.8)	44.6 (0.8)
JS	D	ES	31.5 (3.9)	43.6 (1.5)	40.8 (1.4)	**45.0 (1.1)**	45.5 (0.8)	44.9 (0.7)	46.3 (0.7)	45.7 (0.7)	45.0 (0.6)
LJ	P	EN	35.1 (13.7)	90.7 (3.0)	70.0 (5.7)	38.8 (5.0)	75.2 (4.8)	83.7 (2.9)	**84.3 (2.3)**	85.9 (1.6)	84.5 (2.2)
LJ	P	PT	44.8 (11.7)	93.3 (2.1)	82.7 (3.6)	49.5 (4.4)	85.2 (1.9)	82.8 (3.0)	**90.9 (1.1)**	88.7 (1.5)	92.1 (1.3)
LJ	P	ES	0.0 (0.0)	49.1 (14.5)	21.4 (5.8)	28.7 (7.1)	**36.0 (6.3)**	11.6 (2.0)	32.3 (4.8)	40.7 (5.3)	42.8 (5.2)
LJ	E	EN	[1, 1]	[7.4e1, 5.5e2]	[1.0e2, 6.0e2]	[3.8e3, 1.4e4]	[1.3e3, 9.0e3]	[1.1e3, 6.2e3]	**[8.4e2, 3.9e3]**	[9.0e2, 4.1e3]	[6.4e2, 2.8e3]
LJ	E	PT	[1.4e1, 7.5e1]	[2.8e2, 1.4e3]	[3.0e2, 1.5e3]	[1.4e2, 5.8e2]	[1.0e3, 3.1e3]	[4.2e2, 1.6e3]	**[7.8e2, 1.8e4]**	[5.3e2, 5.3e3]	[4.8e2, 4.7e3]
LJ	E	ES	[1, 1]	[1, 1]	[1, 1]	[1, 1]	**[1, 1]**	[1, 1]	[1, 1]	[1, 1]	[1, 1]
LJ	D	EN	35.5 (6.8)	45.3 (2.8)	47.7 (2.2)	49.5 (2.3)	54.7 (1.6)	54.9 (1.3)	**55.1 (1.2)**	54.2 (1.1)	53.8 (1.0)
LJ	D	PT	32.1 (6.6)	40.7 (2.8)	41.3 (2.0)	41.8 (1.9)	40.9 (1.7)	40.5 (1.6)	**40.4 (1.4)**	40.4 (1.3)	39.8 (1.2)
LJ	D	ES	0.0 (0.0)	26.3 (3.0)	38.9 (2.9)	37.4 (2.6)	**42.7 (2.2)**	42.1 (2.1)	44.3 (1.8)	44.0 (1.6)	45.2 (1.4)

Table 10: Evaluation results using the exploit-explore retrieval active learning method. The results are reported across languages – English ('EN'), Portugese ('PT'), Spanish ('ES') – performance metrics – average precision ('P'), number of predicted positives ('E'), diversity ('D') – and classes – is hired ('IH'), is unemployed ('IU'), job offer ('JO'), job search ('JS'), job loss ('LJ'). Standard errors for P and D are shown in parentheses, and we report a lower bound and an upper bound for E. Bold values indicate the iteration at which a model converges.

			i=0	i=1	i=2	i=3	i=4	i=5
IH	P	SS	83.1 (3.8)	89.9 (1.6)	71.4 (3.5)	87.8 (1.3)	94.3 (0.8)	65.4 (2.6)
IH	P	AR	83.1 (3.7)	91.6 (2.7)	92.9 (1.3)	**92.9 (0.8)**	92.7 (0.7)	80.3 (2.5)
IH	P	UU	83.1 (3.8)	91.5 (3.3)	95.5 (0.7)	**97.0 (0.5)**	80.1 (2.5)	93.5 (0.6)
IH	P	UC	83.1 (3.8)	95.8 (0.8)	78.8 (3.8)	95.7 (1.8)	96.2 (1.0)	**97.7 (0.3)**
IH	P	EE	83.0 (3.8)	96.1 (0.7)	94.3 (1.1)	**96.3 (0.6)**	96.0 (0.7)	96.5 (0.5)
IH	E	SS	[2.2e3, 4.4e3]	[7.7e3, 1.5e4]	[1.9e4, 3.7e4]	[4.7e3, 1.0e4]	[6.7e3, 1.4e4]	[4.9e4, 6.8e4]
IH	E	AR	[2.2e3, 4.4e3]	[9.2e3, 1.8e4]	[1.2e4, 2.6e4]	**[1.8e4, 3.8e4]**	[6.2e3, 1.1e4]	[9.4e3, 1.9e4]
IH	E	UU	[2.2e3, 4.4e3]	[9.9e3, 2.2e4]	[1.1e4, 2.7e4]	**[2.2e4, 5.5e4]**	[2.7e4, 8.5e4]	[1.4e4, 4.9e4]
IH	E	UC	[2.2e3, 4.4e3]	[9.7e3, 2.3e4]	[2.0e4, 4.8e4]	[2.4e4, 5.7e4]	[2.1e4, 6.4e4]	**[2.8e4, 8.1e4]**
IH	E	EE	[2.2e3, 4.4e3]	[1.0e4, 2.2e4]	[1.3e4, 3.2e4]	**[1.5e4, 3.2e4]**	[9.9e3, 2.1e4]	[2.2e4, 5.0e4]
IH	D	SS	43.1 (2.1)	38.7 (1.8)	38.8 (1.5)	37.2 (1.3)	36.5 (1.2)	39.2 (1.0)
IH	D	AR	43.0 (2.1)	50.5 (1.4)	47.9 (1.2)	**46.9 (1.1)**	46.3 (1.0)	46.5 (0.9)
IH	D	UU	43.0 (2.2)	44.3 (1.3)	44.9 (1.2)	**46.0 (1.1)**	46.9 (1.0)	46.3 (0.9)
IH	D	UC	43.1 (2.2)	46.5 (1.8)	45.8 (1.4)	49.3 (1.1)	48.4 (1.0)	**47.9 (0.9)**
IH	D	EE	43.2 (2.1)	42.6 (1.6)	46.5 (1.2)	**46.8 (1.0)**	46.5 (0.9)	45.8 (0.9)
IU	P	SS	14.5 (5.9)	3.6 (1.1)	15.7 (4.0)	3.9 (0.9)	8.8 (3.5)	**9.1 (2.1)**
IU	P	AR	14.5 (5.9)	78.3 (5.3)	91.4 (1.5)	81.8 (2.6)	95.6 (0.9)	91.7 (1.7)
IU	P	UU	14.4 (5.9)	40.9 (10.4)	69.3 (5.7)	52.7 (5.8)	74.9 (4.2)	84.0 (2.5)
IU	P	UC	14.4 (5.9)	86.5 (3.0)	92.9 (1.1)	93.3 (1.7)	**95.7 (0.6)**	95.0 (0.8)
IU	P	EE	14.5 (5.9)	82.7 (4.5)	94.2 (1.4)	70.3 (2.7)	92.8 (1.3)	85.4 (2.0)
IU	E	SS	[1, 1]	[1, 1]	[1, 1]	[1, 1]	[1, 1]	**[1, 1]**
IU	E	AR	[1, 1]	[2.7e3, 7.4e3]	[2.4e3, 5.3e3]	[6.4e2, 1.6e3]	[3.8e3, 1.3e4]	[2.7e3, 1.1e4]
IU	E	UU	[1, 1]	[9.0e0, 1.3e2]	[4.8e2, 1.9e3]	[1.3e3, 3.4e3]	[2.3e3, 7.5e3]	[7.3e3, 2.1e4]
IU	E	UC	[1, 1]	[7.1e2, 2.7e3]	[6.9e3, 1.9e4]	[1.1e4, 2.4e4]	**[1.2e4, 1.4e5]**	[5.3e3, 1.6e4]
IU	E	EE	[1, 1]	[1.8e3, 6.7e3]	[4.8e3, 1.3e4]	[1.5e4, 3.7e4]	[3.3e3, 9.2e3]	[1.0e4, 2.2e4]
IU	D	SS	32.2 (4.4)	38.7 (4.4)	52.6 (3.1)	53.0 (2.7)	56.4 (2.2)	**58.2 (2.3)**
IU	D	AR	32.0 (4.7)	56.3 (1.9)	53.2 (1.8)	51.7 (1.5)	54.1 (1.3)	55.4 (1.0)
IU	D	UU	31.9 (4.8)	56.4 (2.4)	59.2 (1.6)	61.7 (1.5)	61.7 (1.1)	63.7 (1.0)
IU	D	UC	32.0 (4.8)	60.8 (1.8)	62.0 (1.2)	64.1 (0.9)	**63.1 (0.8)**	62.9 (0.7)
IU	D	EE	32.2 (4.9)	52.1 (1.7)	59.4 (1.1)	61.4 (0.9)	62.3 (0.8)	62.2 (0.7)
JO	P	SS	95.1 (0.5)	98.2 (0.2)	98.4 (0.1)	**98.6 (0.1)**	99.0 (0.1)	99.4 (0.1)
JO	P	AR	95.1 (0.5)	97.4 (0.3)	98.8 (0.1)	99.0 (0.1)	**99.2 (0.1)**	99.2 (0.1)
JO	P	UU	95.1 (0.5)	95.3 (1.2)	98.6 (0.2)	69.6 (2.2)	95.6 (0.8)	97.2 (0.5)
JO	P	UC	95.1 (0.5)	98.1 (0.3)	98.9 (0.1)	99.0 (0.1)	**99.3 (0.1)**	99.3 (0.1)
JO	P	EE	95.1 (0.5)	97.0 (0.3)	98.9 (0.1)	99.3 (0.1)	98.6 (0.1)	99.3 (0.1)
JO	E	SS	[2.5e5, 4.0e5]	[3.6e5, 5.8e5]	[3.9e5, 5.6e5]	**[4.1e5, 6.1e5]**	[3.2e5, 5.4e5]	[3.8e5, 7.1e5]
JO	E	AR	[2.5e5, 4.0e5]	[3.4e5, 6.1e5]	[3.6e5, 6.7e5]	[3.6e5, 6.7e5]	**[3.5e5, 6.3e5]**	[3.8e5, 6.6e5]
JO	E	UU	[2.5e5, 4.0e5]	[3.7e5, 5.5e5]	[4.0e5, 6.0e5]	[4.2e5, 6.2e5]	[3.7e5, 4.4e5]	[3.6e5, 4.5e5]
JO	E	UC	[2.5e5, 4.0e5]	[2.5e5, 3.5e5]	[3.6e5, 6.9e5]	[4.0e5, 6.2e5]	**[3.6e5, 9.6e5]**	[3.7e5, 6.5e5]
JO	E	EE	[2.5e5, 4.0e5]	[3.6e5, 6.0e5]	[3.5e5, 5.9e5]	[3.5e5, 6.7e5]	[3.8e5, 5.9e5]	[3.6e5, 6.5e5]
JO	D	SS	49.9 (0.9)	54.7 (0.7)	56.5 (0.6)	**55.1 (0.5)**	53.8 (0.5)	53.7 (0.4)
JO	D	AR	50.0 (0.9)	49.8 (0.7)	50.7 (0.6)	50.8 (0.6)	**50.9 (0.6)**	50.4 (0.5)
JO	D	UU	49.9 (0.9)	54.0 (0.7)	52.9 (0.6)	53.5 (0.6)	53.4 (0.5)	53.0 (0.5)
JO	D	UC	50.0 (0.9)	49.5 (0.7)	51.7 (0.6)	52.7 (0.5)	**52.5 (0.4)**	52.7 (0.4)
JO	D	EE	49.9 (0.9)	50.3 (0.6)	50.4 (0.5)	50.1 (0.5)	50.2 (0.5)	50.3 (0.4)
JS	P	SS	92.5 (1.2)	91.8 (1.9)	68.4 (2.8)	34.9 (2.0)	73.4 (1.7)	69.3 (1.7)
JS	P	AR	92.6 (1.1)	53.8 (5.6)	95.1 (0.9)	95.4 (0.4)	95.1 (0.5)	62.7 (2.0)
JS	P	UU	92.5 (1.3)	80.4 (2.1)	97.7 (0.5)	97.4 (0.4)	80.3 (2.6)	97.6 (0.3)
JS	P	UC	92.5 (1.1)	97.0 (0.5)	97.7 (0.6)	**97.9 (0.4)**	98.3 (0.3)	97.0 (0.5)
JS	P	EE	92.4 (1.2)	90.8 (1.6)	67.3 (3.1)	95.7 (0.5)	96.3 (0.5)	**96.6 (0.4)**
JS	E	SS	[4.4e3, 9.1e3]	[1.3e4, 2.5e4]	[2.1e4, 3.0e4]	[6.9e4, 9.4e4]	[3.9e2, 1.2e3]	[2.4e3, 4.7e3]
JS	E	AR	[4.4e3, 9.1e3]	[1.1e4, 1.9e4]	[6.4e4, 1.2e5]	[2.0e4, 4.4e4]	[2.9e4, 6.1e4]	[8.8e4, 1.5e5]
JS	E	UU	[4.4e3, 9.1e3]	[1.8e4, 3.1e4]	[1.1e4, 2.7e4]	[2.2e4, 5.3e4]	[2.1e4, 4.1e4]	[2.5e4, 4.7e4]
JS	E	UC	[4.4e3, 9.1e3]	[1.5e4, 3.4e4]	[3.3e4, 6.8e4]	**[2.9e4, 5.1e4]**	[3.1e4, 6.6e4]	[3.5e4, 6.3e4]
JS	E	EE	[4.4e3, 9.1e3]	[1.6e4, 2.7e4]	[1.9e4, 3.9e4]	[3.3e4, 7.7e4]	[5.4e4, 1.1e5]	**[4.3e4, 8.6e4]**
JS	D	SS	49.6 (2.1)	53.7 (1.4)	53.0 (1.3)	53.0 (1.3)	52.7 (1.2)	52.1 (1.1)
JS	D	AR	49.6 (2.1)	50.4 (1.8)	61.2 (1.3)	60.5 (1.0)	59.3 (0.9)	59.3 (0.9)
JS	D	UU	49.7 (2.1)	50.2 (1.4)	53.4 (1.2)	54.0 (1.0)	56.4 (0.9)	57.8 (0.8)
JS	D	UC	49.4 (2.2)	56.6 (1.3)	59.1 (0.9)	**58.9 (0.8)**	58.4 (0.8)	58.2 (0.7)
JS	D	EE	49.6 (2.2)	54.2 (1.6)	60.2 (1.3)	60.6 (0.9)	60.3 (0.8)	**60.4 (0.7)**
LJ	P	SS	35.2 (13.7)	2.6 (1.1)	3.8 (1.3)	**11.8 (5.6)**	22.2 (6.0)	11.8 (3.3)
LJ	P	AR	35.0 (13.5)	40.5 (10.5)	20.1 (4.0)	91.1 (1.9)	**82.8 (3.8)**	83.0 (2.4)
LJ	P	UU	35.2 (13.7)	54.4 (8.2)	89.2 (2.1)	74.1 (4.7)	37.7 (4.2)	84.6 (2.8)
LJ	P	UC	35.2 (13.6)	25.8 (8.3)	82.3 (4.0)	80.5 (5.7)	**90.9 (2.7)**	91.3 (1.7)
LJ	P	EE	35.1 (13.7)	90.7 (3.0)	70.0 (5.7)	38.8 (5.0)	75.2 (4.8)	83.7 (2.9)
LJ	E	SS	[1, 1]	[1, 1]	[1, 1]	**[1, 1]**	[1, 1]	[1, 1]
LJ	E	AR	[1, 1]	[1, 1]	[1, 1]	[4.2e2, 4.1e3]	**[2.1e2, 2.4e3]**	[1.9e2, 1.3e3]
LJ	E	UU	[1, 1]	[8.2e0, 8.1e1]	[1.6e2, 1.1e3]	[6.4e2, 2.7e3]	[1.9e3, 6.2e3]	[5.0e2, 2.0e3]
LJ	E	UC	[1, 1]	[1, 1]	[2.8e2, 2.4e3]	[1.1e3, 9.3e3]	**[1.6e3, 6.7e4]**	[5.2e2, 2.9e3]
LJ	E	EE	[1, 1]	[7.4e1, 5.5e2]	[1.0e2, 6.0e2]	[3.8e3, 1.4e4]	[1.3e3, 9.0e3]	[1.1e3, 6.2e3]
LJ	D	SS	35.4 (6.7)	35.4 (7.0)	37.4 (5.4)	**41.4 (4.7)**	51.2 (3.5)	49.8 (3.0)
LJ	D	AR	35.6 (6.7)	34.5 (3.6)	38.5 (3.4)	56.2 (3.2)	**56.1 (2.1)**	54.2 (1.8)
LJ	D	UU	35.5 (6.6)	45.9 (3.4)	50.8 (2.0)	53.0 (1.6)	52.3 (1.5)	54.0 (1.3)
LJ	D	UC	35.2 (7.0)	48.2 (2.9)	44.0 (2.4)	50.3 (2.0)	**51.8 (1.4)**	52.1 (1.2)
LJ	D	EE	35.5 (6.8)	45.3 (2.8)	47.7 (2.2)	49.5 (2.3)	54.7 (1.6)	54.9 (1.3)

Table 11: Evaluation results on English tweets reported across active learning methods – stratified sampling ('SS'), adaptive retrieval ('AR'), uncertainty uncalibrated ('UU'), uncertainty calibrated ('UC'), exploit-explore retrieval ('EE') – performance metrics – average precision ('P'), number of predicted positives ('E'), diversity ('D') – and classes – is hired ('IH'), is unemployed ('IU'), job offer ('JO'), job search ('JS'), job loss ('LJ'). Standard errors for P and D are shown in parentheses, and we report a lower bound and an upper bound for E. Bold values indicate the iteration at which a model converges.