Reinforced Multiple Instance Selection for Speaker Attribute Prediction

Alireza S. Ziabari[†] **Ali Omrani**[†] **Parsa Hejabi**[†]

Preni Golazizian[†] **Brendan Kennedy**[†] **Payam Piray**[†] **Morteza Dehghani**[†]

[†] University of Southern California

{salkhord, aomrani, hejabi, golazizi, btkenned, piray, mdehghan}@usc.edu

Abstract

Language usage is related to speaker age, gender, moral concerns, political ideology, and other attributes. Current state-of-the-art methods for predicting these attributes take a speaker's utterances as input and provide a prediction per speaker attribute. Most of these approaches struggle to handle a large number of utterances per speaker. This difficulty is primarily due to the computational constraints of the models. Additionally, only a subset of speaker utterances may be relevant to specific attributes. In this paper, we formulate speaker attribute prediction as a Multiple Instance Learning (MIL) problem and propose RL-MIL, a novel approach based on Reinforcement Learning (RL) that effectively addresses both of these challenges. Our experiments demonstrate that our RL-based methodology consistently outperforms previous approaches across a range of related tasks: predicting speakers' psychographics and demographics from social media posts, and political ideologies from transcribed speeches. We create synthetic datasets and investigate the behavior of RL-MIL systematically. Our results show the success of RL-MIL in improving speaker attribute prediction by learning to select relevant speaker utterances.

1 Introduction

Examining and quantifying the connection between individuals' language usage and their psychological and demographic attributes has emerged as a focal area of research, often referred to as 'speaker attribute prediction'. Previous studies have found relationships between language usage and age in blogs (Argamon et al., 2007); personality dimensions and moral concerns in Facebook posts (Park et al., 2015; Kennedy et al., 2021); and political ideology in Twitter posts (Preoţiuc-Pietro et al., 2017) among other attributes (e.g., Farnadi et al., 2013; Borkenau et al., 2016; Moreno et al., 2021).

In datasets for predicting speaker attributes, observations typically consist of multiple instances of

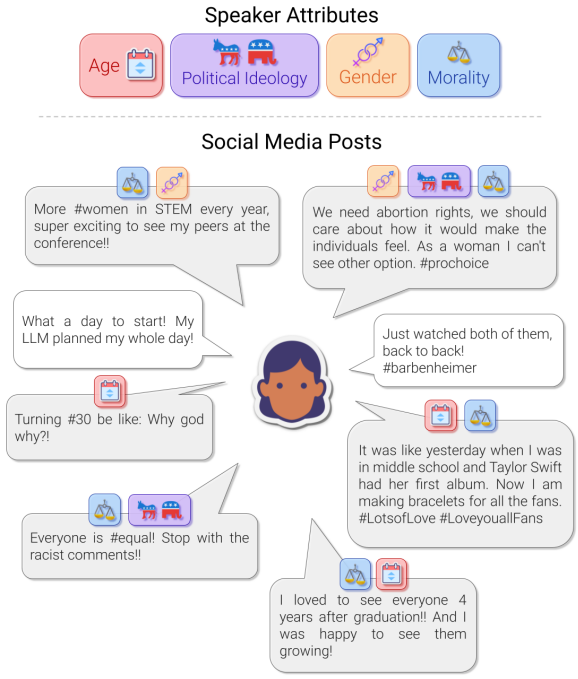

Figure 1: Illustration of the relationship between speaker attributes and social media posts, where (1) not all posts are related to a speaker attribute and (2) some are related to more than one attribute.

speaker utterances— such as, social media posts or political speeches— paired with a set of speaker attributes. While previous work combines speaker data indiscriminately into single collections of text, we propose that this nested structure makes Multi-Instance Learning (MIL; Dietterich et al., 1997) an ideal framework for speaker attribute prediction. MIL describes the setting in which labels are associated with sets of instances ("bags"). The desirable quality of MIL methods is that they combine the information of the individual instances to extract information about the bag label. However, the effectiveness of most MIL algorithms is hindered by the challenge of handling a large number of instances within a single bag. This difficulty is primarily due to the computational constraints of

Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 3307–3321

the models when dealing with bags consisting of a large number of instances. Moreover, the situation is exacerbated by the fact that not all instances may be relevant to a given speaker attribute. Prior research acknowledges these challenges and typically tackles them by down-sampling the instances in each bag, either randomly or heuristically. For example, Yuan et al. (2014) provides an instance selection algorithm inspired by the immune system to reduce the bag size before passing to MIL models. Nevertheless, these strategies often result in sparse bags (Bunescu and Mooney, 2007; Li and Sminchisescu, 2010) or bags that lack any relevant instances (Li et al., 2009).

To overcome the limitations posed by bags of many instances, we propose a novel MIL approach based on Reinforcement Learning (RL) that learns to select relevant instances for the downstream model. The RL component, responsible for the instance selection, not only reduces the number of instances fed into the models but also ensures that relevant instances to the target speaker attribute are present in the down-sampled bags.

We first formalize speaker attribute prediction as a MIL task (section 3.1) and introduce *RL-MIL*, our RL-based approach for handling large bags in MIL (section 3.4). We then show that RL-MIL consistently outperforms other MIL and Non-MIL variants on a wide range of real-world datasets for speaker attribute prediction (section 5.1). Finally, in section 5.2, we construct a synthetic dataset where we have full control over the number of instances relevant to the speaker attribute and utilize it to understand and analyze the efficacy of our RL-MIL approach. Overall, this work establishes MIL as an important framework for predicting speaker attributes and develops an RL methodology that addresses the main limitations of MIL in this context. The work in this paper can be widely applied in speaker attribute prediction across domains, and our RL method can be expanded to provide exemplar-based explanations [1].

2 Related Work

2.1 Speaker-Language Analysis

Prior approaches for speaker-language analysis typically operate by identifying word categories that are associated with a given speaker-attribute. Some approaches, termed "top–down" approaches, define these categories a priori (e.g., The Linguistic Inquiry and Word Count, Pennebaker et al., 2001). These word categories can be *content* categories (e.g., "family"), types of emotion words, or grammatical categories. Examples of this lexicon-based approach include Boyd et al. (2015), which related individuals' core values to their language. In general, regression coefficients or correlation statistics are used to relate the frequency of a given category to a speaker attribute.

Other researchers have taken a "bottom–up" approach, deriving word categories via Latent Dirichlet Allocation (LDA; Blei et al., 2003), Latent Semantic Analysis (LSA; Deerwester et al., 1990), and related techniques. These methods have been employed by Garcia and Sikström (2014), which used LSA to analyze the relationship between Facebook posts and personality traits. Moreover, Schwartz et al. (2013) and McFarland et al. (2013) applied LDA to identify linguistic trends associated with changes in personality across large samples of Facebook participants, and Eisenstein et al. (2010) developed a latent variable topic model in order to quantify geospatial differences in word usage.

2.2 Multi-Instance Learning

MIL was introduced by Dietterich et al. (1997) as the setting in which labels are paired with "bags" (i.e., sets) of instances. Some methods refer to this scenario as weak or distant supervision (Pappas and Popescu-Belis, 2017), or posit that MIL is a special case of semi-supervised learning (Zhou and Xu, 2007). The original formulation of MIL, which applied strictly to classification, took the union over a bag of instances such that a bag is positive for a class if at least one instance within the bag is positive and negative if there are no positive instances in the bag (Foulds and Frank, 2010). In our work, we take the "relaxed" version of MIL, which views instances' contribution to bag labels in an agnostic way (e.g., the bag label can be influenced by multiple instances; see Liu et al., 2012).

There has been a recent resurgence on the design of MIL methods, specifically the problem of learning permutation-invariant aggregation of instance representations. In particular, Lee et al. (2019) and Ilse et al. (2018) apply "attention," a learned weighting over instances or feature dimensions, in differing ways to dynamically learn an optimal aggregation of instances into a single representation. While MIL was initially motivated by and applied to non-text datasets (e.g., predicting whether a drug

[1] Our code and experiments are available at https://github.com/AlirezaZiabari/RL-MIL

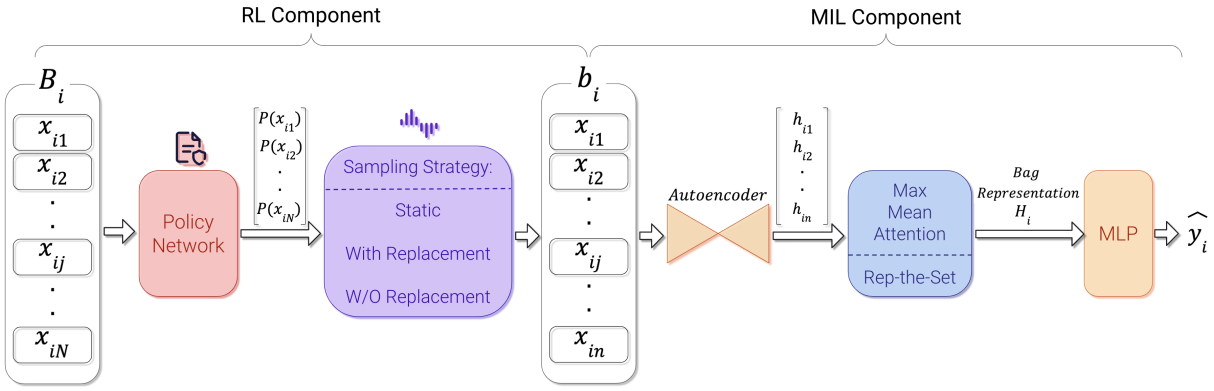

Figure 2: Overview of RL-MIL approach for selecting informative instances. On the left side is the RL component, which has two main parts: policy network and sampling. On the right side is the MIL model which has three parts: autoencoder, pooling method, and classification head(MLP).

molecule can bind well to a target protein Dietterich et al., 1997), some recent works have begun to experiment on text data. For example, Pappas and Popescu-Belis (2014) reports experiments on sentiment classification, representing each document as a set of sentences. Other approaches for this task include explicitly modeling instance relevance for predicting the bag label using a learned weighting mechanism (Pappas and Popescu-Belis, 2017) and formulating the MIL task as the propagation of bag-level labels to instance-level labels (Kotzias et al., 2015). In recent studies, Liu et al. (2022) employed MIL for offensive language detection by introducing a "mutual-att" mechanism to effectively fuse instance and bag representations bidirectionally. Zhang and Wan (2023) detoxified language models at the token-level using a pretrained MIL network.

2.3 Reinforcement Learning for Data Selection

Data selection methods can increase the efficiency of machine learning models by identifying the most informative training examples. Researchers have explored various methodologies for learning the data selection process. For instance, deep Q-network proposed by Mnih et al. (2015) has been utilized in various contexts such as active learning (Fang et al., 2017), self-training (Chen et al., 2018), and co-training (Wu et al., 2018) to boost model's performance by augmenting the training set with additional data. More recently, Ye et al. (2020) introduced the use of policy networks for self-training in the context of zero-shot text classification, and Pujari et al. (2022) used Actor-Critic Network (Konda and Tsitsiklis, 1999) to select instances from auxiliary tasks that improve a target task's performance in a multitask model. In computer vision, Zhu et al. (2022) utilized an Actor-Critic Network to construct positive/negative instances in the contrastive learning process.

3 Methods

First, we formalize speaker attribute prediction as a MIL problem and discuss our baselines including four core MIL methods. Then, we detail our proposed RL method for improved data selection, which is combined with each of the MIL methods in a model-agnostic way.

3.1 Problem Formulation

Let x_{ij} represent the j^{th} instance (e.g., a speaker's post on Twitter) of the i^{th} super-bag $B_i = \{x_{i1}, ..., x_{in}\}$ (e.g., all of a speaker's posts on Twitter) and let y_i denote the label associated with B_i (e.g., speaker's age). Given that models are limited in their input capacity, a typical MIL model f takes in a bag $b_i \subset B_i$, where $|b_i| << |B_i|$, as input and outputs a single prediction $\hat{y}_i$ for b_i.

3.2 Non-MIL Baseline

For the purpose of establishing performance baselines for MIL methods, we remove the pooling layer from the MIL methods and train on the averaged representation of a bag B instances.

3.3 Core MIL Methods

In our experiments, we explore multiple MIL approaches all designed to be "permutation invariant". Permutation invariance is crucial as without it a method would not be viable for generalizing to unseen data. Each MIL model in our experiments consists of three components: an autoencoder, a pooling layer, and a classification head. As shown in Figure 2, all MIL models begin by feeding all instances $x_{ij} \in b_i$ into the autoencoder which produces hidden representations h_{ij}. These representations are aggregated using a pooling layer, forming a single bag-level representation H_i. Afterward, H_i is passed through a classification head to obtain the bag-level label $\hat{y}_i$. We explore four different MIL approaches: *Mean* pooling, *Max* pooling, *Attention* pooling (Liu et al., 2022), and "Rep-The-Set" (Skianis et al., 2020). These approaches primarily differ in the pooling layer, which plays a crucial role in the MIL model and is responsible for transforming the instance-level representations into the bag-level representation.

Mean MIL The pooling layer averages h_{ij}s, forming a single bag-level representation $H_i \in \mathcal{R}^H$.

Max MIL In Max MIL the pooling later operates by taking the maximums of $h_{i,j}$ along each dimension to form the bag-level representation H_i.

Attention MIL We adapt the method of Ilse et al. (2018) which was originally developed and applied to image classification. Specifically, the authors importantly provided an analysis of the attention mechanism for the aggregation of instances in the MIL setting, showing it to be "permutation invariant." Formally, we aggregate h_{ij}s using attention or a learned weighted sum.

$$H_i = \sum_j \alpha_{ij} h_{ij} \tag{1}$$

α_{ij} denotes the importance of h_{ij} and α_{ij} is parameterized by a MLP with a softmax.

Rep-the-Set Lastly, we apply the method developed by Skianis et al. (2020), which learns set aggregations by computing the correspondences between the input set and *hidden sets* by "maximizing flow" through the hidden sets. We use the hidden sets from Rep-The-Set as the bag representation H_i in our experiments. Similar to other core MIL methods, H_i is followed by an MLP to predict the speaker attribute.

Algorithm 1: RL-MIL

Data: Training and validation super bags: $B = \{B_i = \{x_{ij}\}|_{j=1}^{N}\}|_{i=1}^{k}$ and $B' = \{B'_i = \{x_{ij}\}|_{j=1}^{N}\}|_{i=1}^{k'}$.

Initialize policy network with parameters θ ;
Load MIL network with parameter ϕ;
for $e = 1 \rightarrow epochs$ **do**
  Predict $P(x_{ij})|_{j=1}^{N}$ for all validation super bags in B' ;
  Select subset b'_i based on $\{P(x_{ij})|_{j=1}^{N}\}$ for all validation super bags in B';
  for $i = 1 \rightarrow k$ **do**
    Predict $P(x_{ij})|_{j=1}^{N}$ for each instance in B_i ;
    Select b_i based on $\{P(x_{ij})|_{j=1}^{N}\}$;
    Update MIL network parameters ϕ with b_i;
    Compute, and Store reward r_i on selected subset of B';
    Store $(\{\log P(x_{ij})|_{j=1}^{N}\}, r_i)$ in buffer;
  Normalize the rewards $\{r_i\}|_{i=1}^{N}$ from buffer;
  Compute $\mathcal{L}$ form $\mathcal{L}_p$ (Equation 2) and $\mathcal{L}_{reg}$ (Equation 3) ;
  Update policy network parameters θ using policy gradient with respect to $\mathcal{L}$;

3.4 Our Proposed Approach - Reinforced Multiple Instance Selection

In traditional MIL approaches, instances within a bag b_i are typically selected randomly or through heuristics from the complete set of instances in the super bag B_i. These approaches assume that all instances are equally relevant to the label, but in reality, only a few instances are usually pertinent to the task. Identifying these few relevant instances can be a non-trivial challenge. To address this, we propose RL-MIL, a data-driven solution for instance selection. RL-MIL consists of a model-agnostic RL component added prior to the MIL models that learn to select input instances with the goal of improving the performance of the MIL model. Figure 2 shows the conceptual outline of our proposed RL-MIL model.

We describe the instance selection process of RL-MIL in algorithm 1. In our experiments, we implemented the RL-component as a policy network (Sutton et al., 1999) with epsilon-greedy search,

due to its simplicity and robustness. After initialization, in the first step of training, the policy network assigns a selection probability $P(x_{ij})$ to each instance x_{ij} in the super bag B_i (line 7). Next, if the model is not in the exploration mode (line 8) , $|b_i|$ instances are selected based on one of the three different strategies; (1) *Static*: select the top n instances with the highest $P(x_{ij})$, (2) *With Replacement*: sample n instances according to $P(x_{ij})$ with replacement, and (3) *Without Replacement*: same as the second strategy but sampling is done without replacement.

To train the policy network, we first train the MIL model on b_i (line 9), then compute the reward based on the performance of the MIL model on evaluation data (line 10). We use the F_1 score as a reward for the policy network. Assume the reward for k bags in the training dataset is $\{F_1^1, F_1^2, ..., F_1^k\}$. Then the policy loss ($\mathcal{L}_p$) is calculated as follows:

$$\mathcal{L}_p = \sum_{i=1}^{k} \frac{F_1^i - \mu}{\sigma} \times \frac{-1}{n} \sum_{j=1}^{|b|} log(P(x_{ij})) \quad (2)$$

where μ and σ denote the average and standard deviation of $\{F_1^1, F_1^2, ..., F_1^k\}$. To prevent the policy network from assigning probability of 1 to all instances within a bag, we added regularization loss ($\mathcal{L}_{reg}$) as follows:

$$\mathcal{L}_{reg} = \sum_{i=1}^{k} \sum_{j=1}^{||B||} P(x_{ij}) \quad (3)$$

The total loss is a linear combination of policy loss ($\mathcal{L}_p$) and regularization loss ($\mathcal{L}_{reg}$) with hyperparameter β.

$$\mathcal{L} = \mathcal{L}_p + \beta \times \mathcal{L}_{reg} \quad (4)$$

We designed our loss to encourage the selection of bags that yield a higher than average F_1 score and discourage the selection of bags that yield a lower than average F_1 score. Note if the RL component selects a "desired" ("undesired") set of instances for a bag b_i, the normalized score ($\frac{F_1^i - \mu}{\sigma}$) for this bag will be positive (negative). This means that minimizing $\mathcal{L}_p$ would require $P(x_{ij})$ for the selected instances to increase (decrease).

4 Experiments

4.1 Datasets

In our experiments, we focus on the task of speaker-attribute prediction, by using datasets specifically designed to capture a range of attributes pertaining to the author of a collection (bag) of texts. We explore both real-world and synthetic datasets. Specifically, we test our models on predicting political ideology, gender, and age from congressional speeches, and moral concerns from Facebook posts (section 4.1.1). Furthermore, we experiment with two different variations of a synthetic dataset for detecting a toxic user where we modify the number of instances relevant to the speaker attribute (section 4.1.2). While we focus on speaker-attribute prediction, our proposed method can be applied to any problem suited for MIL.

We divide all datasets into train, test, and validation sets, adhering to an 80/10/10 split ratio, and use a bag size $|b_i| = 20$. An overview of each dataset is provided in Table 1. In the following sections, we provide a brief summary of each dataset.

4.1.1 Speaker Attribute Data

Congressional Speeches: Derived from the United States Congressional Records (Gentzkow et al., 2018), this dataset covers speeches from the 43rd to the 114th Congress and provides a comprehensive record of congressional floor speeches. In our experiments, we use speeches strating from the 108th Congress to predict speakers' gender, political ideology, and age. Notably, the original dataset does not include age data for congressional speakers. An additional dataset, detailed by Silver and Mehta (2014), was integrated to include the age of each member of Congress serving from January 1947 to February 2014. We categorized the ages of the speakers into four distinct groups: 27-40, 41-55, 56-70, and over 70 years old. For this dataset, we created super-bags with a size of 100 ($|B_i| = 100$). Both datasets are publicly available, allowing for broad research use.

Facebook + MFQ Data: The dataset introduced by Kennedy et al. (2021) originated from research on participants' Facebook posts, coupled with their responses to the Moral Foundations Questionnaire (MFQ; Graham et al., 2008). In this dataset, individuals' moral concerns are measured across five foundations: care, fairness, loyalty, authority, and purity; each measurement represents a value between zero and five, derived as an average from various questions assessing these attributes in participants. We prepared the data for classification by calculating the median for each foundation in the training set. In the dataset, a label is assigned a value of zero if it is less than its corresponding

Dataset	Citation	# Bags	Avg Bag Size	Labels
Civil Comments (Toxic-5)	cjadams et al. (2017)	1,508	50	T/NT
Civil Comments (Toxic-10)	cjadams et al. (2017)	1,592	50	T/NT
Congressional Speech	Gentzkow et al. (2019)	2,789	205.5	AGE, GEN, PI
Facebook + MFQ Data	Kennedy et al. (2021)	2,739	56.3	MF

Table 1: Overview of datasets used in the study. Labels are abbreviated as follows: T/NT (Toxic/Non-Toxic), AGE (Age Categories), GEN (Gender), PI (Political Ideology), and MF (Moral Foundations).

median in the training data, and a value of one otherwise. Similar to the Congressional Speech dataset, we formed super-bags with a size of 100 ($|B_i| = 100$). Although the dataset is private, it is accessible for research purposes upon request.

4.1.2 Synthetic Data

Toxic-10 and Toxic-5: We utilize the civil comments dataset from cjadams et al. (2017), featuring comments from Wikipedia Talk pages to construct synthetic MIL datasets. The civil comments dataset comprises 159,571 comments, of which 16,225 are identified as toxic and 143,346 are non-toxic. The labels are based on six categories: toxic, severely toxic, obscene, threat, insult, and identity hate. To align this dataset with our research objectives, we defined a binary label termed 'toxic.' A comment is labeled as toxic if it is labeled with any one of the six aforementioned categories.

Specifically, we created two distinct, balanced datasets by grouping comments into bags ($|B_i| = 50$), each containing a mix of positive (toxic) and negative (non-toxic) samples. We set the label of a bag B_i as toxic if it includes toxic instances. The first dataset (Toxic-5) consists of toxic bags with five positive instances, and the second dataset (Toxic-10) includes bags with 10 positive instances. Even though this is not considered as a typical speaker attribute dataset, in practical settings, it can still be utilized to identify whether a user employs toxic language.

4.2 Experimental Setup

RL-MIL models use two distinct advantages over MIL models: (1) access to more instances during the training and (2) the ability to learn the selection policy. To distinguish the impact of these factors, we evaluate our MIL model by randomly re-sampling input instances in each batch. This allows the model to leverage the expanded training data without explicitly learning the selection policy. We refer to these models as *Ensemble-MIL*. We conduct our experiments with four modeling categories: non-*MIL*, *MIL*, *Ensemble-MIL*, and *RL-MIL*. For each MIL approach, we evaluate four different modeling variations — Attention, Max, Mean, and RepSet described in section 3.3. We use macro-F_1 as our evaluation metric in the experiments to treat all classes equally. The RL component has three primary architectural components (Figure 2); (1) the sampling algorithm: for which we explore three different strategies, (2) the core MIL model: in other words whether MIL or Ensemble-MIL is used as the downstream model of in RL-MIL, and (3) the instance representation: for which we choose from instance representations from either the pre-trained model (x_{ij}) or after applying the autoencoder (h_{ij}). The combination of these variations can result in 12 distinct RL models.

Implementation Details: We use the pre-trained RoBERTa-base (Liu et al., 2019) with a 768-dimension hidden state as the embedding model to compute the representation of each instance in a bag, in order to accommodate bags with fewer instances, we pad the bag with vectors of zeros. In the training stage, we use *AdamW* (Loshchilov and Hutter, 2017) as an optimizer for both MIL and RL models. In addition, PyTorch's *ReduceLROnPlateau*, with a patience of five, was implemented as an optimizer scheduler specifically for MIL. In contrast, *ExponentialLR* was adopted for Ensemble-MIL and RL-MIL. We set early stop criteria to avoid over-fitting with patience of 10 for MIL models and 100 for both Ensemble-MIL and RL-MIL models. Since RL models generally require more time to explore different states, we set higher patience for early stopping, and to have a fair comparison between the RL models and Ensemble models, we apply the same early stopping patience of 100. To have a better comparison between models, we used Bayesian hyperparameter optimization from Weights and Biases (W&B; Biewald, 2020) to find the best hyperparameter for the models, details are provided in the Appendix A.3. We use the best hyperparameters for each configuration to train 10

		Political Speeches			Facebook					
	Pooling	Age	Gender	Ideology	Authority	Care	Fairness	Loyalty	Purity	Average
	Non-MIL	0.210	0.507	0.591	0.634	0.639	0.544	**0.641**	0.630	0.549
MIL	Attention	0.306	0.732	0.738	0.649	0.588	0.569	0.597	0.603	0.650
MIL	Max	0.380	0.661	0.767	0.604	0.599	0.547	0.580	0.590	0.643
MIL	Mean	0.351	0.739	0.792	0.653	0.541	0.555	0.592	0.628	0.660
MIL	RepSet	0.350	0.739	0.806	0.660	0.590	0.405	0.591	0.649	0.652
Ensemble	Attention	0.414	0.794	0.780	0.642	0.524	0.470	0.560	0.635	0.652
Ensemble	Max	0.390	0.722	0.863	0.624	0.598	0.587	0.551	0.656	0.669
Ensemble	Mean	0.362	0.739	0.774	0.634	0.558	0.536	0.598	0.598	0.651
Ensemble	RepSet	0.364	0.645	0.831	0.620	0.579	0.577	0.564	0.606	0.652
RL-MIL	Attention	0.400	**0.816**	0.799	0.628	0.623	0.591	0.606	0.667	0.700
RL-MIL	Max	0.373	0.809	0.788	0.624	0.635	**0.602**	0.576	0.649	0.695
RL-MIL	Mean	**0.449**	0.696	0.849	0.665	**0.652**	0.568	0.612	0.657	0.696
RL-MIL	RepSet	0.362	0.765	**0.907**	**0.668**	0.636	0.587	0.586	**0.693**	**0.711**

Table 2: Macro-F_1 scores for real-world datasets: MIL in the top section, Ensemble-MIL in the middle section, and RL-MIL models in the bottom section. The table highlights the highest performances in bold.

models with varying random seeds and select the model with the highest validation performance, and report the results on the test set.

5 Results

5.1 Speaker Attributes

Table 2 shows mcaro-F_1 scores of models for predicting age, gender, and political ideology from congressional speeches and moral concerns, namely, authority, care, fairness, loyalty, and purity from Facebook posts. Comparing the MIL approaches to the non-MIL baseline, we observe an average of 11% improvement in macro-F_1 across all tasks. Importantly, MIL approaches outperform non-MIL baseline on all tasks except predicting care and loyalty. This result empirically validates our formulation of speaker attribute prediction as a MIL problem. Across all labels except loyalty, the top-performing model is always a variant of the RL-MIL. The extent of performance gain of RL-MIL model, in comparison to both Ensemble-MIL and MIL models, varies across tasks and MIL approaches. On average across all tasks, the best RL-MIL variant is 5% better than the best MIL variant and 3.1% better than the best ensemble-MIL variant in macro F_1. On average across all tasks, adding the RL component to MIL approaches results in 4.3%, 4.1%, 3.7%, and 5.1% improvement in macro F_1 on attention, max, mean, and RepSet respectively. These differences are 3.9%, 0.1%, 4.4%, and 5.2% when comparing RL-MIL to ensemble-MIL. These results clearly demonstrate the superiority of our framework, specifically that of the RL component to select instances that improve the model's prediction power in a real-world scenario where not many instances are related to the target speaker attribute.

	Pooling	Toxic-10	Toxic-5
MIL	Attention	0.899	0.821
MIL	Max	0.881	0.825
MIL	Mean	0.937	0.813
MIL	RepSet	0.906	0.826
Ensemble	Attention	0.906	0.796
Ensemble	Max	0.885	0.819
Ensemble	Mean	0.904	0.812
Ensemble	RepSet	0.918	0.810
RL-MIL	Attention	0.943	0.933
RL-MIL	Max	0.943	**0.954**
RL-MIL	Mean	0.955	0.861
RL-MIL	RepSet	**0.994**	0.919

Table 3: Classification Results for the synthetic datasets for three categories: MIL (top section), Ensemble-MIL (middle section), and RL-MIL (bottom section). Macro-F_1 is reported as a comparison metric. Highest performances are bolded in the table.

5.2 Synthetic Data

In real-world scenarios, a large subset of speaker utterances are often not related to the chosen speaker attribute. However, real-world datasets are not ideal for analyzing the relationship between frequency of related utterances and performance. This challenge arises from the inherent difficulty in accurately dis-

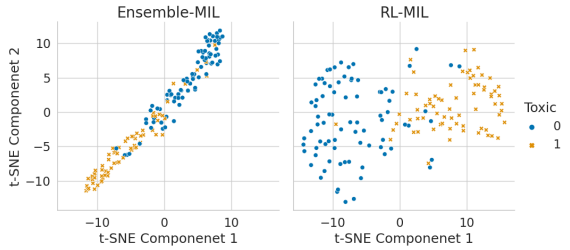

Figure 3: t-SNE visualization of bag representation after attention pooling for ensemble (left) and RL (right) models on toxic-5.

cerning which instances are associated with the specified speaker attribute. To provide more experimental control, and understand the effect of the number of informative instances in a bag on our proposed RL-MIL approach, we compare MIL categories on two synthetic datasets with varying frequencies of related instances described in section 4.1.2. Table 3 demonstrates the results of our experiments on these datasets. On both datasets, RL-MIL approaches consistently outperform both MIL and ensemble-MIL counterparts. Specifically, the top-performing RL-MIL approach outperforms all other variations by 5.7% on Toxic-10. This difference is even higher, 12.8%, on the Toxic-5 dataset. It is important to mention that the top-performing ensemble-MIL approach, despite having access to more data, does not surpass the best MIL model. This can be attributed to the random selection of instances in ensemble models. This selection strategy not only leads to bags that do not represent the ground truth label with a probability of 6.7% (the probability of choosing a bag consisting of all non-Toxic samples for Toxic-5 dataset) but also introduces instability into the learning process by frequently changing input instances. Conversely, our RL-MIL approach learns to select Toxic instances regardless of the paucity of Toxic labels in the dataset. In fact, the $12.8\% - 5.7\% = 7.1\%$ increase in performance gains of RL-MIL from Toxic-10 to Toxic-5 suggests that our approach is particularly successful when the signal for target speaker attribute (i.e., the speaker utterances that are related to the attribute) is sparse. Figure 3 shows the bag representations of the ensemble and RL model after the pooling layer. It is evident that the RL model effectively separates Toxic and Non-Toxic bags, showing its ability to learn better representations.

6 Conclusion & Discussion

In conclusion, we formulate speaker attribute prediction as a MIL problem and identify two key challenges faced by MIL models for this task due to the sparsity of signal. Speakers typically produce a large number of utterances, but only a few of these utterances are indicative of each speaker's attribute. However, MIL models are limited in their input capacity and do not readily have the ability to choose which utterances to use for prediction. To address these challenges, we proposed RL-MIL, a novel RL-based data selection framework for instance selection. We designed our framework to learn an RL model for data selection in conjunction with the MIL model responsible for speaker attribute prediction. To enable RL-MIL to discern relevant instances for each speaker attribute from a large pool of speaker utterances, we designed our data selection loss to reward the selection of bags that yield more accurate predictions. We put our method to a real-world test and experimented with predicting eight different speaker attributes, namely, gender, age, political ideology, and moral concerns from political speeches and Facebook posts. Furthermore, we created a synthetic dataset to study the success of our proposed framework and investigate the impact of signal sparsity on various RL approaches.

Our experimental results demonstrate the efficacy of RL-MIL, especially in scenarios with sparse signals. Our approach can be easily applied to other MIL problems characterized by signal sparsity, such as clinical reports that consist of multiple text reviews per patient, where not all information is pertinent to a given condition. Our code base and experimental setup provide the ground for future work to explore the potential application of our approach in such contexts.

While our work highlights the potential of RL for instance selection in MIL, due to space constraints, we did not exhaustively explore all variations of RL. Subsequent research could delve into alternative RL designs, including the RL model and loss function, for this setup. Particularly, optimizing RL in conjunction with MIL can be challenging, and future work can focus on refining this integration.

7 Limitations

Despite the promising advancements of the RL component presented in our approach, several limitations should be acknowledged. Firstly, the pro-

posed RL approach, while adept at addressing challenges related to handling a large number of utterances per speaker, may introduce computational bottlenecks, particularly in resource-intensive applications or real-time scenarios. Secondly, we faced optimization challenges in the RL component. Addressing this challenge requires careful consideration of model hyperparameters, exploration strategies, and adaptation mechanisms to ensure that the model remains robust. Achieving stability is not only pivotal for the model's effectiveness but also essential for instilling confidence in its practical deployment. Moreover, the effectiveness of the methodology heavily relies on the quality and representativeness of the training data. Biases or inadequacies in the training set may compromise the model's ability to generalize to new and diverse datasets. Additionally, it is worth mentioning that we relied solely on English text excluded non-English speakers, and limited accessibility for a diverse global audience; therefore, the generalization of the proposed approach to diverse contexts, languages, or communication mediums remains an open question, as language intricacies can vary significantly across different settings. These limitations underscore the necessity for further research and refinement, especially when considering real-world deployment and the model's robustness in varied and dynamic environments.

Ethical Statement

In conducting research on speaker attribute selection, we recognize the ethical concerns associated with potential misuse, specifically in determining attributes and protected attributes without user consent. We emphasize that our primary objective is rooted in advancing the fundamental understanding of language science, exploring its nuances across demographics and psychographics. Our intention is to contribute to the broader knowledge base, enabling insights into the complex relationship between language and various demographic factors. It is critical to highlight that all the data, particularly Facebook data, used in our research was collected with explicit consent and consultation from the users by the authors.

Acknowledgements

We would like to thank our anonymous reviewers for their feedback. This research was supported by DARPA INCAS HR001121C0165. The views and conclusions contained herein are those of the authors and should not be interpreted as necessarily representing the official policies, either expressed or implied, of DARPA or the U.S. Government. The U.S. Government is authorized to reproduce and distribute reprints for governmental purposes notwithstanding any copyright annotation therein.

References

Shlomo Argamon, Moshe Koppel, James W Pennebaker, and Jonathan Schler. 2007. Mining the blogosphere: Age, gender and the varieties of self-expression. *First Monday*.

Lukas Biewald. 2020. Experiment tracking with weights and biases. Software available from wandb.com.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *the Journal of machine Learning research*, 3:993–1022.

Peter Borkenau, Alice Mosch, Nancy Tandler, and Annegret Wolf. 2016. Accuracy of judgments of personality based on textual information on major life domains. *Journal of Personality*, 84(2):214–224.

Ryan Boyd, Steven Wilson, James Pennebaker, Michal Kosinski, David Stillwell, and Rada Mihalcea. 2015. Values in words: Using language to evaluate and understand personal values. In *Proceedings of the International AAAI Conference on Web and Social Media*, volume 9.

Razvan C Bunescu and Raymond J Mooney. 2007. Multiple instance learning for sparse positive bags. In *Proceedings of the 24th international conference on Machine learning*, pages 105–112.

Chenhua Chen, Yue Zhang, and Yuze Gao. 2018. Learning how to self-learn: Enhancing self-training using neural reinforcement learning. In *2018 International Conference on Asian Language Processing (IALP)*, pages 25–30. IEEE.

cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. 2017. Toxic comment classification challenge.

Scott Deerwester, Susan T Dumais, George W Furnas, Thomas K Landauer, and Richard Harshman. 1990. Indexing by latent semantic analysis. *Journal of the American society for information science*, 41(6):391–407.

Thomas G Dietterich, Richard H Lathrop, and Tomás Lozano-Pérez. 1997. Solving the multiple instance problem with axis-parallel rectangles. *Artificial intelligence*, 89(1-2):31–71.

Jacob Eisenstein, Brendan O'Connor, Noah A Smith, and Eric Xing. 2010. A latent variable model for

geographic lexical variation. In *Proceedings of the 2010 conference on empirical methods in natural language processing*, pages 1277–1287.

Meng Fang, Yuan Li, and Trevor Cohn. 2017. Learning how to active learn: A deep reinforcement learning approach. *arXiv preprint arXiv:1708.02383*.

Golnoosh Farnadi, Susana Zoghbi, Marie-Francine Moens, and Martine De Cock. 2013. Recognising personality traits using facebook status updates. In *Proceedings of the international AAAI conference on web and social media*, 2, pages 14–18.

James Foulds and Eibe Frank. 2010. A review of multi-instance learning assumptions. *The knowledge engineering review*, 25(1):1–25.

Danilo Garcia and Sverker Sikström. 2014. The dark side of facebook: Semantic representations of status updates predict the dark triad of personality. *Personality and individual differences*, 67:92–96.

Matthew Gentzkow, Jesse M. Shapiro, and Matt Taddy. 2018. Congressional record for the 43rd-114th congresses: Parsed speeches and phrase counts. https://data.stanford.edu/congress_text. Accessed: 2018-01-16.

Matthew Gentzkow, Jesse M. Shapiro, and Matt Taddy. 2019. Measuring group differences in high-dimensional choices: Method and application to congressional speech. *Econometrica*, 87(4):1307–1340.

Jesse Graham, Brian A Nosek, Jonathan Haidt, Ravi Iyer, Koleva Spassena, and Peter H Ditto. 2008. Moral foundations questionnaire. *Journal of Personality and Social Psychology*.

Maximilian Ilse, Jakub Tomczak, and Max Welling. 2018. Attention-based deep multiple instance learning. In *International conference on machine learning*, pages 2127–2136. PMLR.

Brendan Kennedy, Mohammad Atari, Aida Mostafazadeh Davani, Joe Hoover, Ali Omrani, Jesse Graham, and Morteza Dehghani. 2021. Moral concerns are differentially observable in language. *Cognition*, 212:104696.

Vijay Konda and John Tsitsiklis. 1999. Actor-critic algorithms. *Advances in neural information processing systems*, 12.

Dimitrios Kotzias, Misha Denil, Nando De Freitas, and Padhraic Smyth. 2015. From group to individual labels using deep features. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 597–606.

Juho Lee, Yoonho Lee, Jungtaek Kim, Adam Kosiorek, Seungjin Choi, and Yee Whye Teh. 2019. Set transformer: A framework for attention-based permutation-invariant neural networks. In *International Conference on Machine Learning*, pages 3744–3753. PMLR.

Fuxin Li and Cristian Sminchisescu. 2010. Convex multiple-instance learning by estimating likelihood ratio. *Advances in Neural Information Processing Systems*, 23.

Wu-Jun Li et al. 2009. Mild: Multiple-instance learning via disambiguation. *Ieee transactions on knowledge and data engineering*, 22(1):76–89.

Guoqing Liu, Jianxin Wu, and Zhi-Hua Zhou. 2012. Key instance detection in multi-instance learning. In *Asian Conference on Machine Learning*, pages 253–268. PMLR.

Jiexi Liu, Dehan Kong, Longtao Huang, Dinghui Mao, and Hui Xue. 2022. Multiple instance learning for offensive language detection. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 7387–7396.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Ilya Loshchilov and Frank Hutter. 2017. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*.

Daniel A McFarland, Daniel Ramage, Jason Chuang, Jeffrey Heer, Christopher D Manning, and Daniel Jurafsky. 2013. Differentiating language usage through topic models. *Poetics*, 41(6):607–625.

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Andrei A Rusu, Joel Veness, Marc G Bellemare, Alex Graves, Martin Riedmiller, Andreas K Fidjeland, Georg Ostrovski, et al. 2015. Human-level control through deep reinforcement learning. *nature*, 518(7540):529–533.

José David Moreno, Jose A Martinez-Huertas, Ricardo Olmos, Guillermo Jorge-Botana, and Juan Botella. 2021. Can personality traits be measured analyzing written language? a meta-analytic study on computational methods. *Personality and Individual Differences*, 177:110818.

Nikolaos Pappas and Andrei Popescu-Belis. 2014. Explaining the stars: Weighted multiple-instance learning for aspect-based sentiment analysis. In *Proceedings of the 2014 Conference on Empirical Methods In Natural Language Processing (EMNLP)*, pages 455–466.

Nikolaos Pappas and Andrei Popescu-Belis. 2017. Explicit document modeling through weighted multiple-instance learning. *Journal of Artificial Intelligence Research*, 58:591–626.

Gregory Park, H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Michal Kosinski, David J Stillwell, Lyle H Ungar, and Martin EP Seligman. 2015. Automatic personality assessment through social media language. *Journal of personality and social psychology*, 108(6):934.

James W Pennebaker, Martha E Francis, and Roger J Booth. 2001. Linguistic inquiry and word count: Liwc 2001. *Mahway: Lawrence Erlbaum Associates*, 71(2001):2001.

Daniel Preoţiuc-Pietro, Ye Liu, Daniel Hopkins, and Lyle Ungar. 2017. Beyond binary labels: political ideology prediction of twitter users. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 729–740.

Rajkumar Pujari, Erik Oveson, Priyanka Kulkarni, and Elnaz Nouri. 2022. Reinforcement guided multi-task learning framework for low-resource stereotype detection. *arXiv preprint arXiv:2203.14349*.

H Andrew Schwartz, Johannes C Eichstaedt, Margaret L Kern, Lukasz Dziurzynski, Stephanie M Ramones, Megha Agrawal, Achal Shah, Michal Kosinski, David Stillwell, Martin EP Seligman, et al. 2013. Personality, gender, and age in the language of social media: The open-vocabulary approach. *PloS one*, 8(9):e73791.

Nate Silver and Dhruvil Mehta. 2014. Both republicans and democrats have an age problem. Accessed: September 11, 2023.

Konstantinos Skianis, Giannis Nikolentzos, Stratis Limnios, and Michalis Vazirgiannis. 2020. Rep the set: Neural networks for learning set representations. In *International conference on artificial intelligence and statistics*, pages 1410–1420. PMLR.

Richard S Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. *Advances in neural information processing systems*, 12.

Jiawei Wu, Lei Li, and William Yang Wang. 2018. Reinforced co-training. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1252–1262, New Orleans, Louisiana. Association for Computational Linguistics.

Zhiquan Ye, Yuxia Geng, Jiaoyan Chen, Jingmin Chen, Xiaoxiao Xu, SuHang Zheng, Feng Wang, Jun Zhang, and Huajun Chen. 2020. Zero-shot text classification via reinforced self-training. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3014–3024, Online. Association for Computational Linguistics.

Liming Yuan, Jiafeng Liu, and Xianglong Tang. 2014. Combining example selection with instance selection to speed up multiple-instance learning. *Neurocomputing*, 129:504–515.

Xu Zhang and Xiaojun Wan. 2023. Mil-decoding: Detoxifying language models at token-level via multiple instance learning. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 190–202.

Zhi-Hua Zhou and Jun-Ming Xu. 2007. On the relation between multi-instance learning and semi-supervised learning. In *Proceedings of the 24th international conference on Machine learning*, pages 1167–1174.

Zhonghang Zhu, Lequan Yu, Wei Wu, Rongshan Yu, Defu Zhang, and Liansheng Wang. 2022. Murcl: Multi-instance reinforcement contrastive learning for whole slide image classification. *IEEE Transactions on Medical Imaging*.

A Appendix

A.1 Detailed Results

Table 6, 4 show the average and standard deviation of macro-F_1 for 10 different random seeds for both real-world and synthetic datasets. We added Table 8, 7, 5 with more details on the RL component for comparison.

A.2 Computational Resources

All the experiments were conducted on an NVIDIA RTX A6000 with 48GB RAM. Since our model components are small, we can run 20 models at the same time on one NVIDIA RTX A6000. In total, all the experiments took around eight full days to complete.

A.3 Hyper-parameter tuning

Hyper-parameter tuning in our MIL models was facilitated using Weights and Biases (W&B) sweeps (Biewald, 2020). W&B sweeps streamline hyper-parameter optimization, offering diverse search methodologies, notably Bayesian optimization, grid search, and random search. For our work, the Bayesian optimization technique was employed, leveraging a Gaussian Process to model the performance metric as a function of hyper-parameters.

W&B sweeps support multiple distribution methods for each hyper-parameter, including but not limited to constant, categorical, continuous uniform, and discrete uniform distributions.

A.4 Hyper-parameter Tuning for MIL Models

We adopted the Bayesian optimization method, focusing on minimizing the evaluation loss. Each sweep consisted of 50 runs.

General MIL Models: Configurations shared across Attention, Max, and Mean models include:

- **Batch size**: Categorical [8, 16, 32, 64].
- **Epochs**: Integer Uniform between 50 and 400.

- **Hidden dimension**: Categorical [32, 64, 128, 256, 512].
- **Learning Rate**: Log uniform values between 1.0e-2 and 1.0e-4.
- **Dropout Probability**: Constant 0.5
- **Scheduler Patience**: Constant 5
- **Early Stopping Patience**: Constant 10

Repset Model: This model had unique configurations compared to the others:

- **Batch size**: Categorical [8, 16].
- **Epochs**: Integer Uniform between 100 and 1000.
- **Number of Hidden Sets**: Integer between 3 to 20. (Indicates the number of distinct hidden sets the network leverages for processing the input data. Each hidden set encapsulates information about certain input data features or attributes.).
- **Number of Elements**: Integer between 3 to 20. (Represents the number of elements within each hidden set).
- **Learning Rate**: Log uniform values between 1.0e-2 and 1.0e-4.
- **Dropout Probability**: Constant 0.5.
- **Scheduler Patience**: Constant 5.
- **Early Stopping Patience**: Constant 10.

A.5 Hyper-parameter Tuning for RL-MIL Models

For the RL-MIL models, the Bayesian optimization method was similarly applied, with the objective of minimizing the average deviation of model predictions from the true evaluation labels across an evaluation data pool. Each tuning sweep consisted of 50 runs.

RL-MIL Models:

- **RL Learning Rate**: Log uniform values between 1.0e-2 and 1.0e-5.
- **epsilon**: uniform between 0 and 1.
- **Regularization Coefficient**: uniform between 0 and 1.

	Pooling	Toxic-10	Toxic-5
MIL	Attention	$0.903_{(\pm 0.012)}$	$0.647_{(\pm 0.231)}$
	Max	$0.902_{(\pm 0.014)}$	$0.610_{(\pm 0.232)}$
	Mean	$0.626_{(\pm 0.306)}$	$0.789_{(\pm 0.030)}$
	RepSet	$0.863_{(\pm 0.154)}$	$0.798_{(\pm 0.035)}$
Ensemble	Attention	$0.929_{(\pm 0.036)}$	$0.759_{(\pm 0.153)}$
	Max	$0.908_{(\pm 0.019)}$	$0.806_{(\pm 0.036)}$
	Mean	$0.797_{(\pm 0.249)}$	$0.804_{(\pm 0.027)}$
	RepSet	$0.910_{(\pm 0.021)}$	$0.805_{(\pm 0.041)}$
RL-MIL	Attention	$0.931_{(\pm 0.019)}$	$0.680_{(\pm 0.210)}$
	Max	$0.914_{(\pm 0.013)}$	$0.653_{(\pm 0.212)}$
	Mean	$0.682_{(\pm 0.240)}$	$0.802_{(\pm 0.031)}$
	RepSet	$0.886_{(\pm 0.139)}$	$0.805_{(\pm 0.026)}$

Table 4: Classification Results for the synthetic datasets for classic approaches (top section), (middle section), and RL-(bottom section). Macro-F_1 is reported as a comparison metric.

- **Learning Rate**: Constant 1.0e-6.
- **Epochs**: Constant 800.
- **RL hidden dimension**: Constant 8.
- **Batch size**: Constant 128.
- **Early Stopping Patience**: Constant 100.

	RL variation	Toxic-10	Toxic-5
Attention	static	$0.491_{(\pm 0.238)}$	$0.455_{(\pm 0.191)}$
	static (AE)	$0.696_{(\pm 0.222)}$	$0.599_{(\pm 0.258)}$
	w replacement	$0.898_{(\pm 0.023)}$	$0.669_{(\pm 0.198)}$
	w replacement (AE)	$0.897_{(\pm 0.025)}$	$0.659_{(\pm 0.191)}$
	wo replacement	$0.931_{(\pm 0.019)}$	$0.680_{(\pm 0.210)}$
	wo replacement (AE)	$0.903_{(\pm 0.019)}$	$0.682_{(\pm 0.208)}$
Max	static	$0.604_{(\pm 0.224)}$	$0.493_{(\pm 0.184)}$
	static (AE)	$0.828_{(\pm 0.207)}$	$0.596_{(\pm 0.242)}$
	w replacement	$0.888_{(\pm 0.027)}$	$0.647_{(\pm 0.197)}$
	w replacement (AE)	$0.883_{(\pm 0.034)}$	$0.607_{(\pm 0.220)}$
	wo replacement	$0.914_{(\pm 0.013)}$	$0.653_{(\pm 0.212)}$
	wo replacement (AE)	$0.913_{(\pm 0.019)}$	$0.659_{(\pm 0.206)}$
Mean	static	$0.368_{(\pm 0.131)}$	$0.600_{(\pm 0.218)}$
	static (AE)	$0.469_{(\pm 0.166)}$	$0.541_{(\pm 0.252)}$
	w replacement	$0.608_{(\pm 0.273)}$	$0.769_{(\pm 0.030)}$
	w replacement (AE)	$0.682_{(\pm 0.240)}$	$0.764_{(\pm 0.031)}$
	wo replacement	$0.606_{(\pm 0.247)}$	$0.802_{(\pm 0.031)}$
	wo replacement (AE)	$0.648_{(\pm 0.233)}$	$0.797_{(\pm 0.025)}$
RepSet	static	$0.479_{(\pm 0.187)}$	$0.602_{(\pm 0.225)}$
	static (AE)	$0.761_{(\pm 0.281)}$	$0.589_{(\pm 0.249)}$
	w replacement	$0.870_{(\pm 0.102)}$	$0.783_{(\pm 0.020)}$
	w replacement (AE)	$0.849_{(\pm 0.100)}$	$0.780_{(\pm 0.026)}$
	wo replacement	$0.886_{(\pm 0.139)}$	$0.805_{(\pm 0.026)}$
	wo replacement (AE)	$0.878_{(\pm 0.136)}$	$0.806_{(\pm 0.020)}$

Table 5: Different RL components variations results for Synthetic datasets. "(AE)" means the representation after the autoencoder is passed to the Policy Network to predict the selection probability.

		Facebook					Political Speeches		
	Pooling	Authority	Care	Fairness	Loyalty	Purity	Age	Gender	Ideology
	Non-MIL	$0.627_{(\pm 0.036)}$	$0.558_{(\pm 0.066)}$	$0.552_{(\pm 0.068)}$	$0.578_{(\pm 0.032)}$	$0.586_{(\pm 0.066)}$	$0.212_{(\pm 0.014)}$	$0.480_{(\pm 0.022)}$	$0.522_{(\pm 0.052)}$
MIL	Attention	$0.652_{(\pm 0.018)}$	$0.499_{(\pm 0.095)}$	$0.464_{(\pm 0.080)}$	$0.600_{(\pm 0.031)}$	$0.656_{(\pm 0.043)}$	$0.298_{(\pm 0.031)}$	$0.652_{(\pm 0.036)}$	$0.752_{(\pm 0.024)}$
MIL	Max	$0.647_{(\pm 0.025)}$	$0.517_{(\pm 0.105)}$	$0.497_{(\pm 0.084)}$	$0.569_{(\pm 0.085)}$	$0.638_{(\pm 0.038)}$	$0.309_{(\pm 0.039)}$	$0.654_{(\pm 0.042)}$	$0.587_{(\pm 0.207)}$
MIL	Mean	$0.659_{(\pm 0.016)}$	$0.551_{(\pm 0.075)}$	$0.515_{(\pm 0.060)}$	$0.558_{(\pm 0.109)}$	$0.647_{(\pm 0.038)}$	$0.294_{(\pm 0.032)}$	$0.624_{(\pm 0.121)}$	$0.781_{(\pm 0.027)}$
MIL	RepSet	$0.641_{(\pm 0.044)}$	$0.477_{(\pm 0.131)}$	$0.369_{(\pm 0.017)}$	$0.588_{(\pm 0.036)}$	$0.576_{(\pm 0.137)}$	$0.327_{(\pm 0.035)}$	$0.664_{(\pm 0.056)}$	$0.777_{(\pm 0.027)}$
Ensemble	Attention	$0.649_{(\pm 0.022)}$	$0.595_{(\pm 0.040)}$	$0.457_{(\pm 0.057)}$	$0.590_{(\pm 0.028)}$	$0.636_{(\pm 0.027)}$	$0.376_{(\pm 0.028)}$	$0.757_{(\pm 0.050)}$	$0.794_{(\pm 0.040)}$
Ensemble	Max	$0.631_{(\pm 0.030)}$	$0.595_{(\pm 0.040)}$	$0.541_{(\pm 0.031)}$	$0.594_{(\pm 0.025)}$	$0.627_{(\pm 0.028)}$	$0.388_{(\pm 0.041)}$	$0.731_{(\pm 0.048)}$	$0.783_{(\pm 0.040)}$
Ensemble	Mean	$0.641_{(\pm 0.025)}$	$0.594_{(\pm 0.035)}$	$0.553_{(\pm 0.039)}$	$0.598_{(\pm 0.027)}$	$0.649_{(\pm 0.048)}$	$0.371_{(\pm 0.037)}$	$0.665_{(\pm 0.122)}$	$0.796_{(\pm 0.031)}$
Ensemble	RepSet	$0.643_{(\pm 0.023)}$	$0.503_{(\pm 0.118)}$	$0.501_{(\pm 0.082)}$	$0.593_{(\pm 0.037)}$	$0.592_{(\pm 0.108)}$	$0.384_{(\pm 0.024)}$	$0.712_{(\pm 0.056)}$	$0.816_{(\pm 0.025)}$
RL-MIL	Attention	$0.521_{(\pm 0.105)}$	$0.521_{(\pm 0.096)}$	$0.446_{(\pm 0.100)}$	$0.532_{(\pm 0.116)}$	$0.525_{(\pm 0.140)}$	$0.374_{(\pm 0.026)}$	$0.731_{(\pm 0.038)}$	$0.791_{(\pm 0.021)}$
RL-MIL	Max	$0.602_{(\pm 0.079)}$	$0.545_{(\pm 0.064)}$	$0.477_{(\pm 0.088)}$	$0.518_{(\pm 0.113)}$	$0.590_{(\pm 0.083)}$	$0.384_{(\pm 0.031)}$	$0.739_{(\pm 0.048)}$	$0.773_{(\pm 0.036)}$
RL-MIL	Mean	$0.570_{(\pm 0.124)}$	$0.524_{(\pm 0.104)}$	$0.490_{(\pm 0.080)}$	$0.509_{(\pm 0.100)}$	$0.566_{(\pm 0.112)}$	$0.381_{(\pm 0.042)}$	$0.640_{(\pm 0.106)}$	$0.807_{(\pm 0.026)}$
RL-MIL	RepSet	$0.565_{(\pm 0.109)}$	$0.431_{(\pm 0.104)}$	$0.450_{(\pm 0.087)}$	$0.539_{(\pm 0.101)}$	$0.517_{(\pm 0.121)}$	$0.359_{(\pm 0.033)}$	$0.701_{(\pm 0.042)}$	$0.815_{(\pm 0.018)}$

Table 6: Classification macro-F_1 for the real-world datasets. The results are averaged with their standard deviation between 10 runs.

	RL variation	authority	care	fairness	loyalty	purity
Attention	static	$0.522_{(\pm 0.100)}$	$0.443_{(\pm 0.101)}$	$0.405_{(\pm 0.089)}$	$0.488_{(\pm 0.105)}$	$0.525_{(\pm 0.140)}$
Attention	static (AE)	$0.521_{(\pm 0.105)}$	$0.412_{(\pm 0.094)}$	$0.446_{(\pm 0.100)}$	$0.532_{(\pm 0.116)}$	$0.519_{(\pm 0.127)}$
Attention	ensemble + static	$0.544_{(\pm 0.118)}$	$0.521_{(\pm 0.096)}$	$0.435_{(\pm 0.094)}$	$0.520_{(\pm 0.110)}$	$0.463_{(\pm 0.147)}$
Attention	ensemble + static (AE)	$0.546_{(\pm 0.075)}$	$0.463_{(\pm 0.117)}$	$0.395_{(\pm 0.043)}$	$0.498_{(\pm 0.117)}$	$0.422_{(\pm 0.101)}$
Max	static	$0.566_{(\pm 0.065)}$	$0.512_{(\pm 0.089)}$	$0.455_{(\pm 0.092)}$	$0.518_{(\pm 0.113)}$	$0.525_{(\pm 0.123)}$
Max	static (AE)	$0.602_{(\pm 0.079)}$	$0.501_{(\pm 0.091)}$	$0.441_{(\pm 0.068)}$	$0.437_{(\pm 0.056)}$	$0.547_{(\pm 0.117)}$
Max	ensemble + static	$0.534_{(\pm 0.075)}$	$0.545_{(\pm 0.064)}$	$0.467_{(\pm 0.095)}$	$0.498_{(\pm 0.101)}$	$0.517_{(\pm 0.096)}$
Max	ensemble + static (AE)	$0.534_{(\pm 0.107)}$	$0.516_{(\pm 0.076)}$	$0.477_{(\pm 0.088)}$	$0.496_{(\pm 0.110)}$	$0.590_{(\pm 0.083)}$
Mean	static	$0.465_{(\pm 0.118)}$	$0.497_{(\pm 0.098)}$	$0.401_{(\pm 0.051)}$	$0.421_{(\pm 0.098)}$	$0.534_{(\pm 0.138)}$
Mean	static (AE)	$0.570_{(\pm 0.124)}$	$0.524_{(\pm 0.104)}$	$0.490_{(\pm 0.080)}$	$0.428_{(\pm 0.098)}$	$0.538_{(\pm 0.154)}$
Mean	ensemble + static	$0.523_{(\pm 0.115)}$	$0.441_{(\pm 0.078)}$	$0.438_{(\pm 0.103)}$	$0.468_{(\pm 0.100)}$	$0.535_{(\pm 0.136)}$
Mean	ensemble + static (AE)	$0.497_{(\pm 0.112)}$	$0.491_{(\pm 0.101)}$	$0.487_{(\pm 0.065)}$	$0.509_{(\pm 0.100)}$	$0.566_{(\pm 0.112)}$
RepSet	static	$0.533_{(\pm 0.112)}$	$0.431_{(\pm 0.104)}$	$0.378_{(\pm 0.041)}$	$0.509_{(\pm 0.090)}$	$0.478_{(\pm 0.133)}$
RepSet	static (AE)	$0.565_{(\pm 0.109)}$	$0.377_{(\pm 0.059)}$	$0.365_{(\pm 0.013)}$	$0.472_{(\pm 0.084)}$	$0.517_{(\pm 0.121)}$
RepSet	ensemble + static	$0.416_{(\pm 0.073)}$	$0.422_{(\pm 0.099)}$	$0.450_{(\pm 0.087)}$	$0.468_{(\pm 0.097)}$	$0.434_{(\pm 0.077)}$
RepSet	ensemble + static (AE)	$0.499_{(\pm 0.094)}$	$0.410_{(\pm 0.087)}$	$0.437_{(\pm 0.068)}$	$0.539_{(\pm 0.101)}$	$0.515_{(\pm 0.143)}$

Table 7: Different RL components variations results for Facebook datasets. “(AE)” means the representation after the autoencoder is passed to the Policy Network to predict the selection probability. ”ensemble + “ means using the ensemble model as a core model for the RL-MIL.

	RL variation	age	gender	party
Attention	static	$0.225_{(\pm 0.048)}$	$0.580_{(\pm 0.106)}$	$0.501_{(\pm 0.113)}$
	static (AE)	$0.260_{(\pm 0.052)}$	$0.605_{(\pm 0.108)}$	$0.540_{(\pm 0.140)}$
	ensemble + static	$0.291_{(\pm 0.053)}$	$0.611_{(\pm 0.076)}$	$0.650_{(\pm 0.125)}$
	ensemble + static (AE)	$0.334_{(\pm 0.036)}$	$0.688_{(\pm 0.079)}$	$0.731_{(\pm 0.103)}$
	ensemble + w replacement	$0.380_{(\pm 0.019)}$	$0.731_{(\pm 0.038)}$	$0.791_{(\pm 0.037)}$
	ensemble + w replacement (AE)	$0.381_{(\pm 0.033)}$	$0.744_{(\pm 0.030)}$	$0.789_{(\pm 0.040)}$
	ensemble + wo replacement	$0.374_{(\pm 0.026)}$	$0.772_{(\pm 0.040)}$	$0.791_{(\pm 0.021)}$
	ensemble + wo replacement (AE)	$0.369_{(\pm 0.032)}$	$0.759_{(\pm 0.021)}$	$0.787_{(\pm 0.021)}$
Max	static	$0.269_{(\pm 0.048)}$	$0.628_{(\pm 0.068)}$	$0.507_{(\pm 0.121)}$
	static (AE)	$0.268_{(\pm 0.061)}$	$0.611_{(\pm 0.089)}$	$0.608_{(\pm 0.092)}$
	ensemble + static	$0.351_{(\pm 0.027)}$	$0.664_{(\pm 0.108)}$	$0.692_{(\pm 0.115)}$
	ensemble + static (AE)	$0.347_{(\pm 0.037)}$	$0.739_{(\pm 0.048)}$	$0.684_{(\pm 0.071)}$
	ensemble + w replacement	$0.384_{(\pm 0.034)}$	$0.728_{(\pm 0.055)}$	$0.777_{(\pm 0.039)}$
	ensemble + w replacement (AE)	$0.384_{(\pm 0.031)}$	$0.721_{(\pm 0.053)}$	$0.765_{(\pm 0.042)}$
	ensemble + wo replacement	$0.384_{(\pm 0.028)}$	$0.731_{(\pm 0.028)}$	$0.773_{(\pm 0.036)}$
	ensemble + wo replacement (AE)	$0.392_{(\pm 0.024)}$	$0.732_{(\pm 0.027)}$	$0.769_{(\pm 0.042)}$
Mean	static	$0.236_{(\pm 0.052)}$	$0.595_{(\pm 0.117)}$	$0.593_{(\pm 0.158)}$
	static (AE)	$0.277_{(\pm 0.064)}$	$0.478_{(\pm 0.051)}$	$0.659_{(\pm 0.096)}$
	ensemble + static	$0.311_{(\pm 0.101)}$	$0.518_{(\pm 0.098)}$	$0.671_{(\pm 0.162)}$
	ensemble + static (AE)	$0.348_{(\pm 0.059)}$	$0.608_{(\pm 0.112)}$	$0.684_{(\pm 0.143)}$
	ensemble + w replacement	$0.380_{(\pm 0.021)}$	$0.640_{(\pm 0.106)}$	$0.785_{(\pm 0.015)}$
	ensemble + w replacement (AE)	$0.379_{(\pm 0.016)}$	$0.661_{(\pm 0.121)}$	$0.781_{(\pm 0.024)}$
	ensemble + wo replacement	$0.386_{(\pm 0.044)}$	$0.647_{(\pm 0.106)}$	$0.807_{(\pm 0.026)}$
	ensemble + wo replacement (AE)	$0.381_{(\pm 0.042)}$	$0.664_{(\pm 0.116)}$	$0.801_{(\pm 0.025)}$
RepSet	static	$0.299_{(\pm 0.060)}$	$0.349_{(\pm 0.151)}$	$0.645_{(\pm 0.102)}$
	static (AE)	$0.299_{(\pm 0.047)}$	$0.425_{(\pm 0.179)}$	$0.711_{(\pm 0.067)}$
	ensemble + static	$0.345_{(\pm 0.058)}$	$0.597_{(\pm 0.134)}$	$0.715_{(\pm 0.118)}$
	ensemble + static (AE)	$0.326_{(\pm 0.037)}$	$0.620_{(\pm 0.121)}$	$0.736_{(\pm 0.057)}$
	ensemble + w replacement	$0.359_{(\pm 0.033)}$	$0.701_{(\pm 0.042)}$	$0.808_{(\pm 0.030)}$
	ensemble + w replacement (AE)	$0.356_{(\pm 0.028)}$	$0.695_{(\pm 0.034)}$	$0.807_{(\pm 0.027)}$
	ensemble + wo replacement	$0.365_{(\pm 0.030)}$	$0.714_{(\pm 0.043)}$	$0.813_{(\pm 0.022)}$
	ensemble + wo replacement (AE)	$0.366_{(\pm 0.026)}$	$0.712_{(\pm 0.044)}$	$0.815_{(\pm 0.018)}$

Table 8: Different RL components variations results for Congressional Speeches dataset. "(AE)" means the representation after the autoencoder is passed to the Policy Network to predict the selection probability. "ensemble +" means using the ensemble model as a core model for the RL-MIL.