Learning How To Augment Data: An Application To Biomedical NER

Vincenzo Moscato^{1,†}, Marco Postiglione^{1,*,†}, Guido Secondulfo^{1,†}, Giancarlo Sperlí^{1,†} and Andrea Vignali^{1,†}

¹University of Naples Federico II, Via Claudio 21, Naples, Italy

Abstract

Traditional techniques for Named Entity Recognition (NER) need an extensive amount of labeled data in order to get accurate outcomes. However, in real-world situations, it can be difficult to find large datasets, particularly in the biomedical field, where it is challenging to retrieve the required material from which to derive the examples to be annotated and where a domain expert is required for annotations. To address this challenge, data augmentation can be used to generate synthetic data from an existing few-shot training set. However, current methods have a tendency to generate a vast amount of noise, thus hindering performance improvements. In this work, we propose a framework to refine a policy that allows the selection of the most informative examples in an augmented pool with a Policy-based Active Learning approach that employs a deep Q-network to define the selection strategy. We experimented the proposed approach on three benchmark biomedical datasets by simulating few-shot scenarios and found it to be more effective than the selected baselines in most of the cases.

Keywords

Named Entity Recognition, Data Augmentation, Active Learning, Reinforcement Learning, Few-shot learning

1. Introduction

Biomedical Named Entity Recognition (BioNER) is a crucial natural language processing task that plays a pivotal role in automatically identifying and extracting essential entities, such as diseases, chemical agents, and genes, from unstructured text data. Accurate BioNER is fundamental for numerous downstream applications, including medical question answering agents and knowledge graph building, enhancing the overall understanding of biomedical information.

Training effective Named Entity Recognition (NER) models requires significant amounts of manually annotated data, which is a time-consuming and expensive process, particularly in specialized domains, such as legal, historical, or biomedical, where domain knowledge is fundamental. Additionally, the availability of domain experts in the medical field may be limited due to their busy schedules.

To address the challenges posed by limited training

[†]These authors contributed equally.

vincenzo.moscato@unina.it (V. Moscato);

marco.postiglione@unina.it (M. Postiglione);

gu.secondulfo@studenti.unina.it (G. Secondulfo);

giancarlo.sperli@unina.it (G. Sperlí); andrea.vignali@unina.it (A. Vignali)

 2022 Copyright for this paper by its authors. Use permitted under Creative Attribution 4.0 International (CC BY 4.0).
 CEUR Workshop Proceedings (CEUR-WS.org) data, there is a growing interest in exploring *few-shot learning* techniques that enable effective NER even with a limited number of labeled examples. Such techniques offer novel approaches to data and model development, facilitating the generalization of NER models to new and unseen entities.

Among these approaches, data augmentation is a wellestablished strategy for addressing the limited amount of available training data. This technique involves increasing the size of the existing dataset by generating new samples through heuristics or external data sources. While the field of Natural Language Processing (NLP) has explored diverse data augmentation methods, including sentence perturbations [1] and generative models [2], applying these methods to NER input samples is not straightforward due to the token-level classification involved in this task. Consequently, the literature on data augmentation techniques for NER is relatively limited compared to other NLP tasks. Current methods have explored the adaptation of simple manipulation approaches [3], the use of context similarity-based criteria [4] and the imitation of language patterns from high-resource corpora [5].

Although the first attempts of NER data augmentation have shown promising results, the proposed methods of data manipulation may frequently generate a considerable amount of mislabeled and noisy samples, as the new data may not be syntactically and/or semantically accurate. For example, if we manipulate the sentence "*Hypotension* is a term that indicates low blood pressure" so as to replace the entity mention *hypotension* with another disorder (e.g. *dyspnea, hypertension*), the resulting augmented sample may be inaccurate and thus mislead

KDH@IJKAI'23: 6th International Workshop on Knowledge Discovery from Healthcare Data, August 19–20, 2023, Macao, S.A.R. *Corresponding author.

 ^{0000-0002-0754-7696 (}V. Moscato); 0000-0003-1470-8053
 (M. Postiglione); 0000-0003-4033-3777 (G. Sperlí);
 0000-0002-0273-1056 (A. Vignali)
 0202 convident for this paper by its authors. Use permitted under Creative Commons License

the model in effectively identifying mentions.

In this work, we address the issue of selecting the most informative samples from an augmented pool. Inspired by policy-based active learning [6], we do not use a fixed heuristic, but rather allow our framework to learn how to actively select data by formalizing the selection process as a reinforcement learning problem. Specifically, for each sample in the augmented pool, an agent has to decide whether to select it or not, based on its characteristics and model outputs. The selection policy is learned by means of a deep Q-network [7].

We experiment our method by simulating few-shot scenarios in BioNER applications, i.e. we use only k samples as our training data, $k \in \{10, 50, 100\}$. In such settings, we demonstrate the ability of the framework to select the most informative augmented samples first and show promising results as shown by the comparison with the selected baselines. Our approach presents a new direction for exploring the potential of data augmentation to improve the performance of NER models when the training data is scarce, and our findings reveal a considerable margin for improvement, as the data augmentation technique employed for generating the augmented pool can be readily replaced with more advanced and effective methods.

The remainder of this paper is organized as follows. In Section 2, we summarize the literature on NER data augmentation. In Section 3, we present our augmentation framework, while experimental results are reported in Section 4. We conclude our work in Section 5.

2. Related Work

Data augmentation aims to increase the amount of available training data by means of data manipulations, heuristics or external data sources. Dai and Adel [3] investigate the improvements in performance obtained by augmenting NER data with simple data manipulations, such as token replacements, mention replacements, and shuffling. However, these approaches may generate too many noisy samples which may in turn hinder the ability of the model to be effectively trained. Bartolini et al. [4] address this challenge by replacing entity mentions with the most similar entities retrieved by computing context-base similarity. Zeng et al. [8] address the poor generalization ability of few-shot systems to spurious correlations between an entity mention and its context by generating counterfactual examples. Chen et al. [5] leverage an external high-resource corpus to learn how to imitate language patterns (e.g. style, noise, abbreviations). All these works do not evaluate the impact of the noise produced by their proposed augmentation approach over models' performance.

Our approach for the selection of the less noisy samples

from an augmented pool has its foundations in policybased active learning [6]. Active learning (AL) is a wellestablished method to select the most informative unlabeled data to be annotated in order to train the best classifier, thus optimizing human efforts. AL approaches are based on heuristics: uncertainty sampling [9, 10] selects data based on the uncertainty expressed in the outputs of the model, Seung et al. [11] choose data based on the disagreement of a committee. Fang et al. [6] revise AL as a reinforcement learning problem where the selection strategy is automatically learned by an agent by means of a deep Q-network [7]. In our work, an intelligent agent automatically learns a policy to identify the most advantageous samples, from an augmented dataset, to improve the overall performance of the model. By doing so, the agent selects samples that are less likely to mislead the model without the requirement of human input or guidance.

3. Methodology

The methodological workflow of the proposed framework is illustrated in Figure 1. In this section, we will first provide a formalization of the few-shot BioNER problem, and then describe each module in-depth, from the generation of a concepts vocabulary to the reinforcement learning cycles.

3.1. Problem formulation

The input of a NER system is a sentence **s**, which can be represented as a sequence of tokens $\mathbf{s} = [t_1, t_2, ..., t_N]$. NER outputs a list of tuples $[I_s, I_e, t]$ representing named entities mentioned in **s**. Here, $I_s \in [1, N]$ and $I_e \in [1, N]$ are the indexes of start and end characters of the named entity mention, while *t* is the entity type [12].

In practice, this task is usually accomplished by producing a paired sequence of categorical values $\mathbf{y} = [y_1, y_2, ..., y_N]$ as the output of the NER model, where $y_i \in \mathcal{Y}$ indicates the entity type of the *i*-th token. Hence, a NER dataset is defined as a collection of pairwise data $\mathcal{D} = \{(\mathbf{s}_i, \mathbf{y}_i)\}_{i=1}^K$, *K* being the number of examples.

For the purposes of this work, we will be using the IOB scheme to identify entity mentions. Under this scheme, each input token is mapped to the beginning (B), inside (I) or outside (O) of an entity mention. Furthermore, we will consider inputs from biomedical domains, where the NER task is known as *Biomedical NER (BioNER)*. Due to the data scarcity that usually affects such domains, we will test our system in few-shot settings, i.e. the number of training instances *K* is small (e.g. $K \in \{10, 50, 100\}$).



Figure 1: Methodological workflow for the augmentation of BioNER datasets. First, we collect the entity mentions occurring in training data, thus building a concepts *vocabulary*, which is then used to generate an *augmented pool* of data samples with a simple data augmentation technique named *mention replacement*. A deep Q-learning based approach iteratively assigns a state to each sample in the augmented pool and decides whether to select it or not to re-train the NER model according to a policy that is updated at each cycle based on a *reward* that measures the extent to which the addition of the new samples improves the quality of the model.

3.2. Generation of a vocabulary of concepts

Based on the available training data, we extract all the entity mentions, thus building a vocabulary of concepts. In this work, we test our framework by relying solely on the input training data, but this module can be easily extended to include concepts from biomedical ontologies or guided by domain experts. For example, physicians are usually aware of the ways medical concepts can be written in clinical notes; hence, if they are interested in recognizing mentions of a particular concept, they can provide a set of aliases for our system to effectively augment the original training set.

3.3. Data augmentation via mention replacement

For each sentence in our training set, to determine whether a mention should be replaced, we employ a binomial distribution. If the outcome is affirmative, we select a replacement mention from the concepts vocabulary. Subsequently, we modify the corresponding IOB-label sequence as needed. Some examples of mention replacement are provided in Table 1.

The reason behind the choice of this augmentation technique lies in the high number of noisy samples it may generate, given the random nature of the mention replacement. This allows us to effectively test the ability of our framework to discard samples that may mislead the model. However, it is our belief that the performance of the framework can be further improved with more sophisticated augmentation methods, e.g. based on context similarity [4] or learning patterns from cross-domain data [5].

3.4. Reinforcement learning cycles

We learn how to select data from the augmented pool with a module based on reinforcement learning. Our method is built upon the foundations of Policy-based Active Learning [6], which has been previously demonstrated to be capable of automatically learning an active learning strategy from data by formulating the active learning as a reinforcement learning problem where the state corresponds to the unlabeled data selected for labeling, and their label, and the action is the selection heuristic. Specifically, we adapt the method not to work with unlabeled data and human oracles, but with the augmented pool generated in the previous step, and to learn the best strategy to select the samples that may mostly benefit the performance of the NER model. Furthermore, while Fang et al. [6] make a streaming assumption, i.e. unlabelled data arrive one by one and the agent decides

Input	Output
If untreated, hemochromatosis can cause serious illness and early death, but the disease is still substantially under- diagnosed.	If untreated, mononucleosis can cause serious illness and early death, but the disease is still substantially under- diagnosed.
When expressed in Escherichia coli, SH-PTP2 displays tyrosine-specific phosphatase activity	When expressed in Escherichia coli, PTPN6 displays tyrosine-specific phosphatase activity

Table 1

Examples of data augmentation via mention replacement. Here, entity mentions are reported in bold.

the action to take, we assume batch-based learning where the augmented pool is entirely available and the *reward* is computed on the set of actions that the agent has decided to take on the whole dataset in the *i*-th cycle.

In the remainder of this section, we provide in-depth details on the components of the reinforcement learning process.

3.4.1. States

We represent the state of each sentence **s** in the augmented pool at time *i* by taking in consideration both an embedded representation of its content and the outputs of the NER model Θ_i trained over the selected data at time *i*. Specifically, the state s_k consists in the concatenation of the three representations described in the following: content, marginals and confidence. We denote with S_i the set of states at time *i*.

Content Following Kim [13], we first encode each of the *N* tokens t_i in the sentence to produce a matrix $\mathbf{X} = {\mathbf{x}_1, \mathbf{x}_2, ..., \mathbf{x}_N}$ and then apply a convolutional neural network, which consists in a series of filters using linear transformations followed by ReLU activation functions; the last layer of the network performs a max-pooling operation that provides the representation of the sentence content $\mathbf{h_c}$.

Marginals Let $p_{\Theta_i}(\mathbf{y}|\mathbf{s})$ indicate the prediction outputs of the NER model given the input sentence \mathbf{s} . Another convolutional neural network is used to represent the predictive marginals, i.e. the probability distributions associated to all the tokens in \mathbf{s} . Following Fang et al. [6], the convolutional layer contains *j* filters activated with ReLU applied with a window width of 3 and height equal to the number of classes (3 in our case, i.e. I, O and B). Padding is used to endure a wide convolution, and mean pooling is used to allow the network to effectively capture the average uncertainty in each window. The final hidden layer outputs the representation of predictive marginals \mathbf{h}_m .

Confidence Following Fang et al. [6], we represent the confidence by computing the probability of

the most probable sequence of labels under the model, $C = \sqrt[n]{\max_{\mathbf{y}} p_{\Theta_i}(\mathbf{y}|\mathbf{s})}$, where *n* is the length of the sentence **s**.

3.4.2. Actions

Given the state of each input sample, an agent has to decide whether to select it or not to re-train the NER model. Thus, for each sentence \mathbf{s}_k in the augmented pool, the agent selects either to use it ($a_k = 1$) or not ($a_k = 0$). We denote the set of actions made at time *i* with \mathcal{A}_i .

3.4.3. Reward

The reward provides a feedback on the quality of the decisions made by the agent. At each step *i*, the reward is defined as the change in held-out performance:

$$\mathscr{R}_{i}(\mathscr{S}_{i-1},\mathscr{A}_{i}) = \operatorname{Performance}(\Theta_{i}) - \operatorname{Performance}(\Theta_{i-1}),$$
(1)

where Performance(·) is a measure of the model's quality. In our work, we compute the F1 score to determine rewards. Note that the value of \mathcal{R}_i could also be negative, i.e. the effect of the actions made by the agent has a detrimental effect on the performance.

3.4.4. Deep Q-Network

We adopt a deep Q-learning [7] approach where the utility of choosing the action a_k from state s_k is evaluated by the Q function $Q^{\pi}(s_k, a_k)$ according to the policy π . The Q-function is iteratively updated by the agent by considering the rewards obtained in each episode.

The deep Q-network (DQN) consists in a single hidden layer which takes the state vector of a single instance $s_k = [\mathbf{h}_c, \mathbf{h}_m, C]$ as input and uses a ReLU activation function to output two scalar values $\mathcal{Q}(s_k, a_k)$ associated to the two possible actions $a_k \in \{0, 1\}$.

The training objective is to minimize the difference between the estimated Q-value and the true Q-value for a given state-action pair. This is typically done by using a variant of the Q-learning algorithm known as the Bellman equation, which recursively defines the Q-value for a state-action pair as the immediate reward plus the

Shots	Dataset	Method	Precision	Recall	F1
10		Random	11.94 ± 14.87	4.81 ± 9.66	6.30 ± 12.08
	NCBI-Disease	Uncertainty	20.01 ± 9.16	36.62 ± 20.58	25.05 ± 13.07
		Ours	21.33 ± 6.82	26.31 ± 17.06	21.58 ± 13.77
		Random	7.03 ± 9.34	0.33 ± 0.35	1.02 ± 0.45
	BC2GM	Uncertainty	8.47 ± 7.95	25.38 ± 23.28	12.68 ± 11.82
		Ours	21.32 ± 5.18	30.85 ± 21.05	23.11 ± 12.11
	BC5CDR	Random	$\textbf{79.27} \pm \textbf{13.40}$	49.84 ± 25.78	55.80 ± 18.26
		Uncertainty	46.91 ± 28.45	50.22 ± 45.76	39.14 ± 35.96
		Ours	62.15 ± 11.94	74.99 ± 9.64	66.71 ± 6.92
50		Random	$\textbf{30.47} \pm \textbf{17.32}$	41.52 ± 24.53	34.82 ± 19.75
	NCBI-Disease	Uncertainty	25.24 ± 15.23	47.42 ± 27.48	32.26 ± 18.37
		Ours	29.16 ± 18.41	45.06 ± 27.31	34.37 ± 20.28
	BC2GM	Random	26.79 ± 13.92	40.17 ± 23.47	31.91 ± 17.58
		Uncertainty	25.17 ± 7.63	49.26 ± 27.83	31.28 ± 16.75
		Ours	$\textbf{32.23} \pm \textbf{2.41}$	52.70 ± 13.35	39.68 ± 5.90
	BC5CDR	Random	62.02 ± 4.62	$\textbf{86.39} \pm \textbf{4.35}$	72.02 ± 2.44
		Uncertainty	61.89 ± 5.27	85.59 ± 3.64	71.69 ± 3.32
		Ours	67.26 ± 7.16	83.38 ± 3.60	74.18 ± 4.12
100		Random	49.74 ± 4.01	68.44 ± 4.09	57.45 ± 2.26
	NCBI-Disease	Uncertainty	$\textbf{50.66} \pm \textbf{2.84}$	69.37 ± 5.36	58.39 ± 1.65
		Ours	50.25 ± 8.34	$\textbf{72.90} \pm \textbf{4.59}$	58.92 ± 4.34
	BC2GM	Random	$\textbf{37.88} \pm \textbf{3.02}$	62.90 ± 3.46	47.27 ± 3.30
		Uncertainty	35.82 ± 1.28	62.91 ± 6.38	45.60 ± 2.64
		Ours	37.02 ± 2.88	62.04 ± 7.52	46.33 ± 4.22
	BC5CDR	Random	$\textbf{67.19} \pm \textbf{4.35}$	88.65 ± 1.47	$\textbf{76.36} \pm \textbf{2.41}$
		Uncertainty	60.46 ± 3.20	$\textbf{90.44} \pm \textbf{1.42}$	72.45 ± 2.72
		Ours	64.53 ± 3.48	87.82 ± 4.54	74.28 ± 1.88

Table 2

Average results on the benchmark BioNER datasets in different k-shot scenarios, $k \in \{10, 50, 100\}$. For each method and score, we report the mean μ and standard deviation σ obtained across 5 repetitions, in the format $\mu \pm \sigma$. Results with the highest mean are reported in bold.

discounted future Q-value for the next state-action pair. 4. Experiments Mathematically, this can be expressed as:

$$\mathcal{Q}(s_i, a_i) = \mathbb{E}[r_i + \gamma \cdot \max_{a_{i+1}} \mathcal{Q}(s_{i+1}, a_{i+1})], \qquad (2)$$

where Q(s, a) is the Q-value for state *s* and action $a, \gamma \in$ [0, 1] is the discount factor, and $\max_{a_{i+1}}Q(s_{i+1}, a_{i+1})$ is the maximum *Q*-value over all possible actions in the next state.

The goal of the Q-learning algorithm is to update the Q-network weights θ to minimize the mean squared error between the estimated $\mathcal{Q}\text{-value }\mathcal{Q}(s,a;\theta)$ and the target Q-value γ :

$$\mathscr{L}(\theta) = \mathbb{E}\left[\left(y_i(r_i, s_{i+1}) - \mathcal{Q}(s_i, a_i; \theta)\right)^2\right],\tag{3}$$

where $y_i(r_i, s_{i+1}) = r_i + \gamma \cdot \max_{a_{i+1}} Q(s_{i+1}, a_{i+1}; \theta_{i-1})$ is the target Q-value based on the current parameters θ_{i-1} , and results are averaged over a minibatch of samples. Learning updates are based on stochastic gradient descent.

In this section, we provide an in-depth description of the experiments we ran to assess the performance of our system. First, we describe the experimental setup in Section 4.1; then, we discuss experimental results in Section 4.2.

4.1. Experimental setup

4.1.1. Datasets

We test our method on the three popular benchmark datasets from the biomedical field listed as follows:

- NCBI-Disease [14]: 793 abstracts from PubMed, annotated with disorders entity mentions.
- BC2GM [15]: over 20,000 abstracts from PubMed annotated with gene mentions.

• *BC5CDR* [16]: over 1,500 abstracts from PubMed annotated with diseases and chemicals. For simplicity, we consider only chemical entity mentions in our experiments.

For each dataset, we have considered the original training, validation and test sets provided with their original release.

Few-shot simulations To simulate data scarcity scenarios, we randomly sample *k* sentences from the training set, $k \in \{10, 50, 100\}$. Since performance can vary greatly based on the selected training samples, we run each experiment 5 times and always report averaged results.

4.1.2. Training details

Given the data-scarcity nature of our work, we assume that data to tune hyper-parameters is not available. However, we test models on the entire test set. Hence, we choose hyperparameters based on previous work and practical considerations. Specifically, we use a pretrained biomedical Transformer network [17] and train all our models for 3 epochs with a learning rate of $2 \cdot 10^{-4}$, an AdamW optimizer, a batch size of 5 and a maximum sequence length of 256. We run our reinforcement learning framework for 5 episodes. We evaluate the quality of models in terms of precision, recall and F1 scores obtained with the seqeval¹ Python library.

4.1.3. Hardware configuration

All experiments were conducted on the platform Google Colab, using the Free tier plan, which provides a virtual machine with an NVIDIA T4 GPU with 16GB of RAM, an Intel[®] Xeon[®] processor with a frequency of 2.3GHz and 10 cores (but only one used by the VM instance), 12 GB of available memory and 78.19 GB of free disk. Due to the limits imposed by Colab's free plan, we were unable to pursue further improvements on the obtained results. Specifically, we could only run a maximum of 5 PAL episodes per experiment. Although this was sufficient to achieve the intended goals, conducting a greater number of episodes could have allowed for a more refined selection policy for the augmented instances, potentially leading to improved model performances.

4.1.4. Baselines

We compare our method with the two baselines for the selection of samples from the augmented pool described as follows:

• *Random*: we sample a random set of instances from the augmented pool.

• Uncertainty: we leverage uncertainty-based active learning [9] as an heuristic-based framework for the selection of samples, i.e. we rank augmented samples according to the uncertainty of the model in its predictions. Since model predictions are mapped to each token in the sentence, we aggregate them to obtain a single ranking value.

For each method, we always pre-train the initial model with the available training data in the simulated few-shot scenario. Then, we assess the performance of the same model when it is fine-tuned with the selected samples.

4.2. Results

Table 2 presents a comprehensive comparison performance of different baselines in various k-shot scenarios and datasets for BioNER tasks. We can observe that our method consistently shows competitive performance, achieving the highest F1 scores in several cases, indicating its effectiveness in correctly identifying entity mentions. The highest improvement over the random baseline can be seen in the 10-shot scenario on BC2GM data. Being this dataset focused on the gene entity type, and being this information usually mentioned in many heterogeneous ways (e.g. mStaf gene, OBP, primase, V8 protease, MT), a vast amount of noise can be generated by randomly replacing mentions. This negative effect of noise is higher as the data-scarcity scenario becomes stricter, because the effect of noise can be limited when we include clean training data. Furthermore, our method consistently achieves the highest precision or rivals the top performer, as the learned selection policy reliably minimizes the occurrence of false positives.

We should notice that results show high standard deviations, being the quality of the model strongly related to the sampling that generates the few-shot training set.

Figure 2 illustrates the performance trends of F1 scores as we increase the number of selected data from the augmented pool in the 50-shots setting. Our method consistently surpasses the random and uncertainty-based selection curves and the improvement is generally higher as the number of selected elements is smaller. As the number of elements increases, the three curves converge to values that are reasonably close. This trend is consistent with findings from Fang et al. [6], which demonstrated that Policy-based Active Learning achieves superior performance while using fewer selected elements. In all the three plots it can be observed that the peak performance of our method outperforms the performance obtained by the model without the augmented samples, represented by the dashed red line.

¹https://github.com/chakki-works/seqeval



Figure 2: Performance trends as the number of selected samples increases. The horizontal dashed red line is the performance of the original model without augmented samples. These results have been obtained in the 50-shots scenario.

5. Conclusion & Future Work

In this work, we proposed a novel approach for selecting informative samples from an augmented pool to improve the performance of NER models in the biomedical domain with limited training data. Our framework leverages policy-based active learning [6] to learn a selection policy that identifies the most informative augmented samples to enhance the NER model's generalization ability.

We evaluated our method on simulated few-shot scenarios in BioNER applications, where we demonstrated its ability to select the most informative augmented samples first and achieve promising results compared to selected baselines. Our approach presents a new direction for exploring the potential of data augmentation to improve NER models' performance in specialized domains, such as biomedical, where labeled data is scarce and domain knowledge is essential.

Future work should explore the robustness of our framework on real-world biomedical datasets and investigate the effectiveness of different data augmentation techniques on improving the performance of the proposed approach. The simple *mention replacement* approach used in the current implementation of our framework could be indeed replaced with more advanced and sophisticated approaches.

Furthermore, we intend to extend our approach to other NLP tasks beyond NER, such as relation extraction and entity linking, and compare its performance with existing state-of-the-art methods. Finally, we also plan to investigate the interpretability of the learned selection policy to gain insights into the most informative features and characteristics of the augmented data samples that contribute to the NER model's improved performance.

6. Acknowledgments

This project has been funded by PNRR MUR project PE0000013-FAIR. We also aknowledge support from NextGenerationEU via PNRR - DM 352 (CUP: E66G22000400009).

References

- M. Bayer, M. Kaufhold, C. Reuter, A survey on data augmentation for text classification, ACM Comput. Surv. 55 (2023) 146:1–146:39. URL: https: //doi.org/10.1145/3544558. doi:10.1145/3544558.
- [2] R. Liu, G. Xu, C. Jia, W. Ma, L. Wang, S. Vosoughi, Data boost: Text data augmentation through reinforcement learning guided conditional generation, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 9031–9041. URL: https://doi.org/10.18653/v1/2020.emnlp-main. 726. doi:10.18653/v1/2020.emnlp-main.726.
- [3] X. Dai, H. Adel, An analysis of simple data augmentation for named entity recognition, in: D. Scott, N. Bel, C. Zong (Eds.), Proceedings of the 28th International Conference on Computational Linguistics, COLING 2020, Barcelona, Spain (Online), December 8-13, 2020, International Committee on Computational Linguistics, 2020, pp. 3861–3867. URL: https://doi.org/10.18653/v1/2020.coling-main. 343. doi:10.18653/v1/2020.coling-main.343.
- [4] I. Bartolini, V. Moscato, M. Postiglione, G. Sperlì, A. Vignali, COSINER: context similarity data augmentation for named entity recognition, in: Simi-

larity Search and Applications - 15th International Conference, SISAP 2022, Bologna, Italy, October 5-7, 2022, Proceedings, volume 13590 of *Lecture Notes in Computer Science*, Springer, 2022, pp. 11–24. URL: https://doi.org/10.1007/978-3-031-17849-8_2. doi:10.1007/978-3-031-17849-8_2.

- [5] S. Chen, G. Aguilar, L. Neves, T. Solorio, Data augmentation for cross-domain named entity recognition, in: M. Moens, X. Huang, L. Specia, S. W. Yih (Eds.), Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing, EMNLP 2021, Virtual Event / Punta Cana, Dominican Republic, 7-11 November, 2021, Association for Computational Linguistics, 2021, pp. 5346–5356. URL: https://doi.org/10.18653/v1/2021.emnlp-main. 434. doi:10.18653/v1/2021.emnlp-main.434.
- [6] M. Fang, Y. Li, T. Cohn, Learning how to active learn: A deep reinforcement learning approach, in: M. Palmer, R. Hwa, S. Riedel (Eds.), Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, EMNLP 2017, Copenhagen, Denmark, September 9-11, 2017, Association for Computational Linguistics, 2017, pp. 595–605. URL: https://doi.org/10.18653/v1/d17-1063. doi:10. 18653/v1/d17-1063.
- [7] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. A. Riedmiller, A. Fidjeland, G. Ostrovski, S. Petersen, C. Beattie, A. Sadik, I. Antonoglou, H. King, D. Kumaran, D. Wierstra, S. Legg, D. Hassabis, Human-level control through deep reinforcement learning, Nat. 518 (2015) 529–533. URL: https://doi.org/10.1038/nature14236. doi:10.1038/nature14236.
- [8] X. Zeng, Y. Li, Y. Zhai, Y. Zhang, Counterfactual generator: A weakly-supervised method for named entity recognition, in: B. Webber, T. Cohn, Y. He, Y. Liu (Eds.), Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing, EMNLP 2020, Online, November 16-20, 2020, Association for Computational Linguistics, 2020, pp. 7270–7280. URL: https://doi.org/10.18653/ v1/2020.emnlp-main.590. doi:10.18653/v1/2020.
- [9] D. D. Lewis, A sequential algorithm for training text classifiers: Corrigendum and additional data, SI-GIR Forum 29 (1995) 13–19. URL: https://doi.org/10. 1145/219587.219592. doi:10.1145/219587.219592.
- [10] S. Tong, D. Koller, Support vector machine active learning with applications to text classification, J. Mach. Learn. Res. 2 (2001) 45–66. URL: http://jmlr. org/papers/v2/tong01a.html.
- [11] H. S. Seung, M. Opper, H. Sompolinsky, Query by committee, in: D. Haussler (Ed.), Proceedings of the Fifth Annual ACM Conference on Computational Learning Theory, COLT 1992, Pittsburgh,

PA, USA, July 27-29, 1992, ACM, 1992, pp. 287–294. URL: https://doi.org/10.1145/130385.130417. doi:10.1145/130385.130417.

- [12] J. Li, A. Sun, J. Han, C. Li, A survey on deep learning for named entity recognition, IEEE Trans. Knowl. Data Eng. 34 (2022) 50–70. URL: https:// doi.org/10.1109/TKDE.2020.2981314. doi:10.1109/ TKDE.2020.2981314.
- [13] Y. Kim, Convolutional neural networks for sentence classification, in: A. Moschitti, B. Pang, W. Daelemans (Eds.), Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing, EMNLP 2014, October 25-29, 2014, Doha, Qatar, A meeting of SIGDAT, a Special Interest Group of the ACL, ACL, 2014, pp. 1746–1751. URL: https://doi.org/10.3115/v1/d14-1181. doi:10.3115/ v1/d14-1181.
- [14] R. I. Dogan, R. Leaman, Z. Lu, NCBI disease corpus: A resource for disease name recognition and concept normalization, J. Biomed. Informatics 47 (2014) 1–10. URL: https://doi.org/10.1016/j.jbi.2013.12.006. doi:10.1016/j.jbi.2013.12.006.
- [15] L. L. Smith, L. K. Tanabe, R. Ando, C.-J. Kuo, I.-F. Chung, C.-N. Hsu, Y.-S. Lin, R. Klinger, C. Friedrich, K. Ganchev, M. Torii, H. Liu, B. Haddow, C. A. Struble, R. J. Povinelli, A. Vlachos, W. A. Baumgartner, L. E. Hunter, B. Carpenter, R. T.-H. Tsai, H.-J. Dai, F. Liu, Y. Chen, C. Sun, S. Katrenko, P. W. Adriaans, C. Blaschke, R. Torres, M. L. Neves, P. Nakov, A. Divoli, M. Maña-López, J. Mata, W. J. Wilbur, Overview of biocreative ii gene mention recognition, Genome Biology 9 (2008) S2 – S2.
- [16] J. Li, Y. Sun, R. J. Johnson, D. Sciaky, C. Wei, R. Leaman, A. P. Davis, C. J. Mattingly, T. C. Wiegers, Z. Lu, Biocreative V CDR task corpus: a resource for chemical disease relation extraction, Database J. Biol. Databases Curation 2016 (2016). URL: https: //doi.org/10.1093/database/baw068. doi:10.1093/ database/baw068.
- [17] J. Lee, W. Yoon, S. Kim, D. Kim, S. Kim, C. H. So, J. Kang, BioBERT: a pre-trained biomedical language representation model for biomedical text mining, Bioinformatics (2019). URL: https://doi.org/ 10.1093%2Fbioinformatics%2Fbtz682. doi:10.1093/ bioinformatics/btz682.