The Legal and Ethical issues Workshop @LREC2022, pages 53–63
Marseille, 20 June 2022

Generating Realistic Synthetic Curricula Vitae for Machine Learning Applications under Differential Privacy

Andrea Bruera[1,3*], Francesco Aldà[2], Francesco Di Cerbo[3]
[1]Queen Mary University of London, [2]SAP SE, [3]SAP Security Research
a.bruera@qmul.ac.uk, francesco.alda@sap.com, francesco.di.cerbo@sap.com

Abstract

Applications involving machine learning in Human Resources (HR, the management of human talent in order to accomplish organizational goals) must respect the privacy of the individuals whose data is being used. This is a difficult aim, given the extremely personal nature of text data handled by HR departments, such as Curricula Vitae (CVs). We present a methodology for the generation of synthetic CVs which reflect real-world distributions of candidate attributes while providing strong privacy guarantees. These synthetic CVs can be used for training machine learning models instead of (or together with) the original data. Also, our methodology may be adapted to similar types of documents, requiring the generation of a mixture of structured data and natural language. We employ a Bayesian network to model the conditional dependencies between the candidate attributes. The structure of the underlying graph and the conditional probability distributions are learnt under differential privacy from an existing dataset. Then, we generate synthetic CVs by guiding the text generation of a Transformer-based generative language model with a manually-prepared set of prompts where the attributes sampled from the Bayesian network are plugged in. We show by way of both intrinsic (based on linguistic properties) and extrinsic (based on training a model for a classification task using the synthetic CVs) measures that our methodology can be successfully used for machine learning applications in HR, where anonymization is fundamental.

Keywords: Synthetic Data, Differential Privacy, Bayesian Network, Generative Language Model

1. Introduction

In Human Resources (HR) settings, Artificial Intelligence (AI) and Natural Language Processing (NLP) have the potential of offloading time-consuming tasks, such as selecting candidates for a position, understanding the skill set of the workforce, planning training and learning activities, from humans onto machine learning models (Ore and Sposato, 2021; Eubanks, 2022). However, machine learning models require training data; and textual data for HR applications, such as Curricula Vitae (CVs, or resumes), contain extremely sensitive pieces of personal data. These have to be protected, by means of anonymization techniques, against misuses due to the risk of identification of the individuals (Silva et al., 2020) described in the original CV dataset, making it compliant with data protection regulations active in multiple countries around the world.

With respect to anonymization, we adopt the definition provided by the General Data Protection Regulation - GDPR (Voigt and Von dem Bussche, 2017) - of the European Union, which describes anonymous information as “information which does not relate to an identified or identifiable natural person or to personal data rendered anonymous in such a manner that the data subject is not or no longer identifiable” (GDPR Recital 26). In our case, this means that a CV can be considered anonymized when it is not possible to re-identify the subject that it describes.

Some of the information contained in CVs - ‘direct identifiers’ - can be easily spotted and made anonymous by way of pre-trained Named Entity Recognition - NER (Nasar et al., 2021) - models or pattern-based (Paccosi and Aprosio, 2021) approaches (e.g. for emails, phone numbers).

However, a second type of information, called ‘indirect identifiers’, appear in textual data, which may lead to the re-identification of the individuals involved despite the absence of direct identifiers (Tucker et al., 2016). These can be especially understood in terms of the interaction of multiple pieces of information which, by themselves, would not allow re-identification, but that instead would do so when taken together, as an interconnected network. For instance, it is not easy to re-identify a male individual by simply knowing that he served as President of the United States of America. But it becomes much easier if it is known that he was born in Hawaii and obtained an undergraduate degree from Columbia University.

It is important to anonymize training datasets because the trained AI models with personal information may retain certain glimpses of personal data that can later be inferred using attacks like membership inference (Shejwalkar et al., 2021) leading to re-identification of individuals. One may argue that AI models can still be lawfully trained and used by the responsible entity for data collection and processing (the so-called Data Controller, in GDPR terms). But then these models must be subject to all requirements coming from data protection regulations (Francopoulo and Schaub, 2020).

If an individual requests the deletion of his personal information according to GDPR Article 17 “right to be forgotten”, a Data Controller has to find a way to deal

[*] Work carried out at SAP Security Research

with a text classification model trained also on that individual's data. Retraining the model for each and every deletion request would be a waste of resources in terms of time, energy and money. Anonymization allows to protect trained models from such situations.

We present an approach to avoid these issues, which consists of generating realistic synthetic CVs to be used for machine learning applications in HR. The point is not to generate CVs whose textual form would make it hard for a human reader to tell whether it was created by a computer or not; but rather, to generate CVs which capture relevant statistical properties of the attributes of the candidates, containing enough noise to ensure that re-identification is not possible, but sufficient signal to be used for machine learning applications.

This approach is increasingly common in disparate machine learning fields (Nikolenko and others, 2021), and the closest example to our case is that of healthcare and medicine (Chen et al., 2021), where data anonymization is of paramount importance. Notice that our work, despite being specific to CVs, may be in principle adapted to other kinds of documents where a mixture of structured data and raw text needs to be generated while ensuring de-identification.

An additional benefit of generating training data is that the resulting size can be as big as needed: this feature is fundamental especially for deep learning models, which require huge amounts of training data in order to learn effectively.

In our approach, we start from real samples, from which relevant attributes are extracted, and whose conditional dependencies and distributions across candidates are modelled through a Bayesian network (BN) (Niedermayer, 2008). Since the structure and the conditional probability distributions are learnt from real samples which may contain sensitive information, we make use of a differential privacy (DP) mechanism known as PrivBayes (Dwork et al., 2006; Zhang et al., 2017). This ensures that the re-identification risk for individuals can be controlled and mitigated as required.

As an intermediate step, we generate synthetic candidates in the form of sets of attributes, whose conditional distributions are close enough to those of real world candidates, yet providing DP. Finally, these results are plugged into a set of linguistic prompts (Radford et al., 2019), which are presented to a generative language model that will generate each section of the synthetic CV (Schick and Schütze, 2021). We validate our approach in two ways: first, with a set of intrinsic measures (Gatt and Krahmer, 2018), looking at various linguistic properties of our generated text; secondly, with an extrinsic measurement - a candidate role classification task - where we show that our synthetic CVs, which avoid the risks of re-identification, can be successfully used as training data instead of the original, real-world, identifiable CVs, with limited loss of performance.

2. Related Work

2.1. NLP For HR

NLP is increasingly being used in HR applications, but its use for this specific aim is still considered to be limited (Strohmeier, 2022). Some examples are CV (or resume) parsing (Sinha et al., 2021), which focuses specifically on the task of extracting information about candidates from raw text data in the form of CVs; automatized procedures for candidate rating, ranking (Freire and de Castro, 2021) and selection (Kmail et al., 2015), or algorithms to match CVs and job posts (Jain et al., 2021). Notice that these approaches focus exclusively on getting the best results for each application. They do not take into consideration how to mitigate the re-identification risk involving training data independently of the task, which is instead the main interest of our work.

2.2. Differential Privacy And NLP

In recent years, differential privacy has become the de-facto standard for privacy-preserving statistical data analysis and machine learning. It provides strong, formal anonymization guarantees by enforcing that the output distribution of a randomized algorithm is not affected by small changes in its input, namely the addition (or removal) of a single data point (Dwork et al., 2006). Given its effectiveness, it has increasingly been used in NLP as a framework for anonymization (Lyu et al., 2020; Igamberdiev and Habernal, 2021).
A somewhat related approach to ours is that of Krishna et al. (2021), where the authors transform a raw text dataset by adding noise to the latent representation of a language model, before using it for a text classification application. However, Habernal (2021) shows that the sensitivity of the privacy mechanism was underestimated thus leading to an incorrect privacy analysis. In our case, instead, DP is guaranteed by the usage of PrivBayes (Zhang et al., 2017), whose robustness has been formally and empirically demonstrated, and has been adopted in other works (Ping et al., 2017).

2.3. Generation Of Synthetic Training Data For NLP

Given that deep learning models, the state of the art in most NLP tasks (Lauriola et al., 2022), require a big amount of data, which for certain linguistic phenomena can be hard to gather, recently it has become commonplace to either augment existing training data (Feng et al., 2021) with synthetic data, or employing a fully synthetic dataset, after having generated it from scratch (Schick and Schütze, 2021).
When the resulting dataset has to look like natural text, the generation process makes often use of the recently proposed generative language models based on the Transformer architecture (Vaswani et al., 2017), such as GPT (Radford et al., 2019) and CTRL (Keskar et al., 2019). These models are trained to generate realistic natural language text, word after word. The choice

of each new token is conditioned on the previous ones, with extremely realistic results.

3. Our Approach To Synthetic CV Generation

Our approach is composed of three steps: first, the extraction of candidate attributes from a dataset of real CVs, in fact transforming the CVs into structured data entries (Section 3.1); second, the creation of a differentially private Bayesian network representing the conditional dependencies between the selected attributes (Section 3.2); finally, the generation of a synthetic dataset of CVs (Section 3.3). This final stage, in turn, involves, for each CV to be generated, sampling a synthetic set of attributes for a candidate from the differentially private Bayesian network; inserting them in a series of ready-made prompts reflecting the structure of a CV; and finally feeding these filled prompts, sequentially, to the generative language model so as to create a coherent CV.

3.1. Information Extraction

The first step consists in the extraction of candidate attributes from a dataset of real CVs, in the form of raw text, using various techniques: NER and pattern heuristics to find the attributes, relation extraction (RE) to annotate the relationships holding between these attributes and the candidate (Silva et al., 2020; Paccosi and Aprosio, 2021). The output of this step is a structured dataset, containing the key attributes which constitute a candidate profile (e.g. Alma Mater, various features for education history and work experience, technical skills, spoken languages). While looking at the values of extracted attributes may still lead to the re-identification of an individual at this stage, the subsequent steps of the process will make the likelihood of such risk proportional to DP's ε. Importantly, direct personal identifiers (like name, surname, email address, social media accounts) are ignored and not included as candidate attributes.

Notice also that the goal of this phase is to retain only attributes over which distributions across candidates can be learnt, and that the breadth and scope of this phase of information extraction can vary according to each use case.

3.2. Ensuring De-Identification: Bayesian Networks

From this structured dataset of candidate attributes, we build a Bayesian network, a probabilistic graphical model which represents a set of variables and their conditional dependencies as a directed acyclic graph (Niedermayer, 2008). In our setting, the nodes of the graph are the candidate attributes and an edge between two attributes represents a cause-effect relationship between them. For example, the work experience of a candidate is naturally influenced by their education history, and edges between the corresponding attributes would represent this dependency. We provide the visualization of a possible Bayesian network for some simplified candidate attributes in Figure 1.

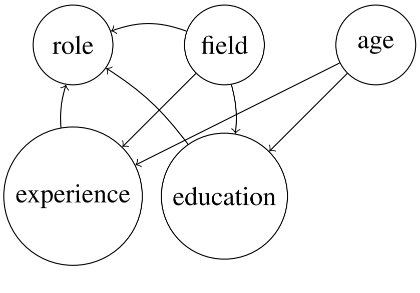

Figure 1: Toy example of a Bayesian network for candidate attributes.

Each node is associated with a function that takes as input the set of possible values for the node's parent variables, and gives as output the probability distribution on the node's values. This function constitutes a conditional probability distribution.

The structure of the graph can be learnt from data or built a priori, while the conditional probability distributions are usually learnt from data. In our case, we learn both from data. The structure of the network or the conditional probabilities may therefore leak some information on an individual in the training set. In order to provide strong privacy guarantees and minimize the re-identification risk, we leverage the notion of differential privacy.

Definition 1. *(Dwork et al., 2006) A randomized algorithm $\mathcal{M} : \mathcal{D} \rightarrow \mathcal{Z}$, i.e., the output of $\mathcal{M}$ is a random variable, is said to provide ε-differential privacy if for every $\varepsilon > 0$, for any pair of neighboring datasets $(X, X') \in \mathcal{D} \times \mathcal{D}$ that differ in one entry only, and for every measurable $Z \subseteq \mathcal{Z}$ the following holds*

$$\Pr[\mathcal{M}(X) \in Z] \leq e^{\varepsilon} \cdot \Pr[\mathcal{M}(X') \in Z]. \quad (1)$$

The privacy budget ε controls the anonymization level of the mechanism $\mathcal{M}$. The smaller the value of ε, the stronger the privacy guarantee provided, as the output distributions are pulled closer and closer.

A standard way of providing differential privacy to vector-valued functions is by adding Laplace-distributed noise to its output (Dwork et al., 2006). For functions that returns categorical values, the exponential mechanism is generally used instead (McSherry and Talwar, 2007).

These mechanisms are the main ingredients behind PrivBayes (Zhang et al., 2017), which provides a successful mechanism for learning the structure as well as the conditional probabilities of a Bayesian network under differential privacy. The following generation steps will be protected against the risks of re-identification due to the robustness of post-processing of any differentially private mechanism (Dwork et al., 2006).

Once the private Bayesian network is built, we can sample new values for all the nodes in the graph. These generated values follow the conditional dependencies of the attributes and preserve the consistency and statistical properties of the original dataset up to the noise addition which acts as a de-identification barrier. In our case, this means that we can generate a synthetic set of attributes for a realistic, but not real, candidate.

3.3. CV Generation Using Specialized Prompts And A Generative Language Model

The candidate attributes sampled from the Bayesian network together with the artificial personal details are then used to generate the text for each section of the synthetic CV.

We start from the hunch that CVs can be viewed through the lens of storytelling as strongly structured stories: in this sense, their structure, which is very similar across candidates, being relatively standardized, should reflect some degree of sequentiality, coherence and development (Popova, 2014). We therefore adapt methodologies proposed in story generation (Yao et al., 2019; Wang et al., 2020; Alhussain and Azmi, 2021) involving the use of pivotal bits of story structures. We employ them in the form of short, incomplete natural language sentences (prompts), that we provide as inputs to the generative model to direct it towards a coherent linguistic output similar to a CV.

In our approach, we exploit the attributes generated by the Bayesian network described in 3.2 as a way to control the text generation. The intuition is that the attributes will influence the probabilities of the words chosen by the model. Because of this, the resulting text will describe coherently the synthetic candidate, using a mixture of fixed attributes and real-looking text. To give an example, given two attributes {'University': 'Columbia'} and {Field of Study: 'Political Science'}, we guide the generative model to select words which are more probable when the words 'Columbia' and 'Political Science' are found in the previous context. Using a toy vocabulary {'international', 'beach', 'beer', 'law'}, higher probabilities should be assigned by the model to 'international' and 'law'.

As we were saying above, in order to make the model generate realistic text, before presenting the synthetic attributes to the generative model, we further plug them in a set of linguistic structures called prompts (Radford et al., 2019; Wang et al., 2020). Prompts are typical bits of sentences where the attribute would be found in a human language (e.g., 'I studied x at y', 'I worked as p at q for r'). Since CVs contain different sections, usually with the aim of resuming the candidate's past experience and skills in a sequentially coherent way, we previously define an ordered list of prompts, which will correspond to the various sections and will contain the relevant attributes. These prompts, with the attributes plugged in, are what will be actually presented, one after another, to the generative model as starting points for the generation of each section of the CV. Importantly, the generation works in a cyclical fashion, in a feedback loop: at each step, the model receives as input the preceding text of the synthetic CV (including text generated by the model itself), followed by the next prompt in the list as input. Its task is to generate the following natural language section in the synthetic CV. Obviously, at the beginning there is no previously generated text, but only the first prompt.

In this way, at each section we direct the model towards the creation of a new section conditioned on the previous ones, to ensure sequential coherence.

4. Model Implementation

In order to evaluate our approach, we implement in a very simple use case the full pipeline we presented. The final aim is that of generating a dataset of synthetic CVs that can be used to train a machine learning model. Starting from an existing dataset of CVs (Jiechieu and Tsopze, 2021) annotated with the candidate roles (Section 4.1), we first extract a set of candidate attributes (universities, companies, years of experience; Section 4.2), then we learn the structure and conditional probabilities of a differentially private Bayesian network modelling the conditional dependencies between the extracted attributes (Section 4.3); in parallel, we manually create a set of prompts to be filled with candidate attributes generated with the BN, which we feed sequentially to GPT-2 (Radford et al., 2019), a public generative language model (Section 4.4).

4.1. Dataset

As a starting point, we use the dataset of real-world CVs presented in (Jiechieu and Tsopze, 2021). In it, direct identifiers had already been anonymized, leaving however all indirect identifiers (see Section 1) in the text. The dataset, made of around 28000 CVs in English, was collected online from a dedicated website [1]. Each CV was automatically annotated with the role(s) provided by each applicant, to be used as its label for classification tasks; for simplicity, in the case of multiple roles, we only employ the first one. This leaves us with a set of nine possible classification labels [2]. We do not apply any particular pre-processing to the text, except for the removal of HTML markup.

4.2. Attribute Extraction

In order to extract the attributes for the candidates, we use a mixture of NER and pattern-based approaches (Paccosi and Aprosio, 2021). As a NER model, we use Spacy's Transformers[3] pre-trained NER model (without fine-tuning it), which encodes the input using RoBERTa (Liu et al., 2019) pre-trained embeddings.

[1] www.indeed.com

[2] Software Developer, Project Manager, Java Developer, Python Developer, Web Developer, Software Developer, Front-End Developer, Systems Administrator, Database Administrator, Network Administrator, Security Analyst

[3] https://spacy.io/universe/project/spacy-transformers

We focus on three types of attributes: universities, companies, and years of work. For each sentence in a CV, we first extract the spans for the named entities using Spacy; then, we only keep the organizations (labelled 'ORG'), which we further add to the candidate's attributes as universities, if the word 'University' appears in the span, and as companies otherwise. We extract the years of experience with a simple heuristics, looking for the regular expression `"(\d+)\syears"`, considering only cases where the integers are inferior to 10 (otherwise, the expression would catch also the candidate age).

Ideally, each CV should contain mentions of both universities and companies - however, we find that this is not the case. Therefore, we filter the dataset keeping only the entries where we could find at least one university or one company, and one attribute for years of experience. This reduces the size of the dataset to around 7000 CVs. We further assume that the levels of education corresponding to each university follow gradually (one university: Bachelor's, two universities: Master's, three universities: Ph.D.), and we add these accordingly as attributes to the candidate profile.

An important issue is to keep only the essential amount of data points and attributes, in order to limit the computational strain when using the Bayesian network. To do so, first of all, we keep only the first three companies (and their matched years of experience) and universities.[4]

Then, to reduce the presence of attributes over which no generalization is possible, and to keep under control the time required to learn the Bayesian network, we set a frequency threshold for the extracted universities and companies. We only keep the original mentions for attributes appearing at least 5 times (leaving us with 792 universities and 1048 companies), and we substitute the entities filtered out with two generic spans ('Other University', 'Generic IT Company') just to use them as dummy features for the generation of CVs. Finally, we also include among the attributes the applicant role, extracted as described in Section 4.1.

Regarding direct identifiers (e.g., name, address, email, social media links, etc.), for each realistic candidate, we generate them as purely fake data using an existing Python library[5]: the aim is just that of providing realistic prompts to the generative language model. Since all these values are generated artificially and independently of the original dataset, no data privacy is compromised at this step.

Finally, for training the Bayesian network, we create a separate set, containing only the candidates having at least one company and one university, leaving us with a set of around 1500 CVs.

4.3. Bayesian Network

For the Bayesian network, we use the Python package developed by the authors of Ping et al. (2017). As introduced in 3.2, we consider the following attributes in our experiments: the applicant role; up to three universities and education titles; up to three job experiences, with their length in years; the total number of years of work. Regarding the conditional dependencies among the nodes, these are learnt under differential privacy using PrivBayes (Zhang et al., 2017; Ping et al., 2017), where we limit the maximum number of parent nodes to 3.

The conditional probabilities are learnt from the reduced set of around 1500 CVs described in Section 4.2, adding Laplace noise to ensure differential privacy, as described in Zhang et al. (2017). As presented in Section 3.2, the privacy budget ε controls the anonymization guarantees. The smaller the value of this parameter, the higher the noise injected and hence the privacy guarantees provided. When learning the Bayesian network, we experiment with different values of the privacy budget ε in order to investigate its effect on our downstream classification task: 0.1 (which, following Zhang et al. (2017), ensures strong DP); 1; 10; 10000.

4.4. Prompting The Generative Model

To generate the synthetic CVs with GPT-2, we manually define a template CV structure reflecting standard versions of CVs, consisting of:

1. An introductory fake personal information part (see Section 4.2), followed immediately by a short summary of the candidate's skills;
2. Education;
3. Work experience;
4. Linguistic skills;
5. Hobbies.

For each section, we write a set of two to five possible prompts to randomly sample from at generation time, so as to ensure variability. These prompts are common ways of introducing the corresponding CV sections (e.g. for education, 'I studied *x* at *y*', 'I attended *y*, where I studied *x*').

Notice that not all the sections involve attributes generated by the Bayesian network: in our case, only sections 1 (fake personal information, candidate role), 2 (universities and titles), 3 (companies and years of work) do. In the other cases, prompts are just generic bits of sentences (e.g. for hobbies, 'In my spare time, I') which are meant to nevertheless drive the generation towards a coherent profile. In the case of sections 2 and 3, where multiple universities and companies are present, prompts will be generated sequentially multiple times (i.e. first for the Bachelor's, then for the Master's, etc).

[4] When not enough years of work could be extracted, we randomly generated an integer, ranging between 0 and the minimum between 0 and the biggest number of extracted years of experience.

[5] `https://faker.readthedocs.io`

As per GPT-2 [6], we use the English pre-trained Medium model (Radford et al., 2019) available within Huggingface's Transformers library (Wolf et al., 2020). We chose English as the language for our experiments because this is the language of the dataset we used for the extrinsic task (see Sections 4.1 and 5.2). The generation, as discussed in section 3.3, works in a cyclical way, so as to enforce coherence among the various sections: generation starts from the prompt for section 1, and then is stopped after 30 words (slightly longer than the average length of a sentence in English (Sigurd et al., 2004)); the prompt for section 2 is appended to the result of the generation, and this whole text is fed back as a new prompt to GPT-2, which again generates no more than 30 words; and so on until the end of all the sections is reached, and the CV is ready.

We generate in this way a set of 4000 synthetic CVs, matching the training dataset of real CVs (see Section 5.2), that we will use in the following evaluation steps.

5. Empirical Evaluation

We evaluate both the linguistic quality of the CVs, using a set of dedicated metrics (Section 5.1), and their downstream usability for machine learning, through a classification task (Section 5.2). Results show that, despite some loss in terms of performance, the generated data which guarantee privacy can be successfully used for machine learning, opening to a wide range of HR applications.

5.1. Intrinsic Evaluation: Linguistic Features

Intrinsic evaluations of generated texts look at the linguistic properties of the results, independently of their effect on performance on a given NLP task (Gatt and Krahmer, 2018). A common way of evaluating generated text is to obtain a matched set of real sentences starting from the same input, comparing the two (an overview of such metrics can be found at Gatt and Krahmer (2018)) - but, in our case, this is not possible. Another one is that of asking humans to evaluate the generated texts on a range of criteria. However, this approach has been subject to scrutiny for its arbitrariness (Howcroft et al., 2020) and, most importantly, it does not apply to our case, since we are not interested in fooling people into believing that a synthetic CV is actually real. We nevertheless report some examples of CVs generated with our methodology in the Appendix.

The approach we use here, instead, is that of defining a set of automatized ways of measuring intrinsic linguistic properties of the generated texts along a number of dimensions (Roemmele et al., 2017; See et al., 2019). In doing this, we exclusively want to investigate the quality of the generated by GPT-2 following our prompts. Therefore, we assume that the features produced by the Bayesian network should have no effect on this, and we report the intrinsic evaluation scores obtained from the training set for $\varepsilon = 0.1$, which ensures the highest level of differential privacy.

More specifically, we are interested in measuring, on the one hand, the lexical diversity and refinement of the generated texts, and on the other, their syntactic complexity. We want to do so because the prompts for the type of text we are generating, CVs, are less open-ended than prompts for other genres. We suspect that this may negatively impact the generation abilities of GPT-2, making it turn towards repetitive, oversimplified, shallow output.

We therefore adopt a set of measures from Roemmele et al. (2017). First, given that high-quality writing has been associated with the presence of more diverse words and phrases (Pitler and Nenkova, 2008), we report the **type-token ratio (TTR)**, both for bi-grams and uni-grams, computing it within each CVs and then averaging the results.

Second, since lower frequency words indicate a more advanced output (Crossley et al., 2011), we compute the **average word frequency** of the generated words, using as frequency estimates, token occurrencies from a dump of the English version of Wikipedia, considering only words appearing at least 10 times in the whole corpus (Roemmele et al., 2017).

Finally, we turn to noun phrases (NPs) and verb constructions as indicators of syntactic complexity, and therefore richer text (McNamara et al., 2010). We look at the **average ratio of NPs and verbs** over sentence length, and at the **average number of tokens contained in each type of phrase or construction** (in the case of verbs, we measure the length in tokens of the subtree in the dependency parse), again divided by sentence length. To single out NPs, verbs and their dependency parse subtrees, we use the pre-trained Spacy Transformers model.

To provide a comparison with real-world text, we compute the same metrics on a random sample of 4000 real CVs (that we call 'Real') taken from the dataset of (Jiechieu and Tsopze, 2021).

Results are reported in Table 1: in general, they indicate that the generated text mirrors closely enough the intrinsic linguistic properties of real world CVs, with a few trade-offs between the two.

GPT-2, through prompting, generates a higher number of token types (higher TTR uni-gram), but tends to repeat bi-grams (lower TTR bi-gram) slightly more often than real candidates do. In a parallel fashion, the NPs produced by GPT-2 are more frequent (higher NP ratio), but slightly shorter (smaller NP average length) than those of real CVs; and the opposite is true of verb constructions (verbs ratio), whose longer average length indicate higher complexity for GPT-2 than for real candidates. Finally, the average corpus token frequency of generated and real CVs are quite close, with GPT-2 preferring slightly more common words. This provides an initial sanity check of our approach to the

[6] We did not use GPT-3 (Brown et al., 2020), the most recent version of GPT, as its pretrained weights were not publicly available at the time of the work.

generation of synthetic CVs.

	Generated	Real
TTR uni-gram	0.071	0.043
TTR bi-gram	0.249	0.334
Average word frequency	11.42	11.03
NP ratio	0.296	0.249
NP average length	0.067	0.0827
Verbs ratio	0.085	0.093
Verb-subtree average length	0.582	0.411

Table 1: Results for the intrinsic evaluation tests

5.2. Extrinsic Evaluation: Applicant Role Classification

5.2.1. Methodology

To evaluate to what extent our synthetic CVs can be used for downstream machine learning in HR applications, we exploit the labeled dataset that we obtained at the end of the process described in Section 4.2.

Remember that each CV comes with the role of the candidate provided by the candidate themselves. This will be the label for our classification task, which we call **Candidate Role Classification**, and that can be considered as an automatized recruitment task, similarly to CV-job description matching or candidate recommender (Zaroor et al., 2017; Lamba et al., 2020). As introduced in Section 4.1, there are nine labels in total.

We randomly split the 7000 real CVs containing at least either one university or one company, and one explicit mention of the years of work (see Section 4.2) into a train set of 4000 CVs and a test set of 1000 CVs (equivalent to a 80/20 split), leaving 2000 CVs on the side as a potential development set, that eventually we do not use.

We do not apply any pre-processing to the generated text, except for the removal of the direct identifiers - the fake personal information (cf. Section 4.2) - as they are just noise: they are purely random tokens and have no relation with the classification label.

Since the aim of our work is not obtaining the highest score possible, but rather validating our approach within a machine learning framework, we train and test two different general-purpose classifiers based on word embeddings, widely used in the field of NLP.

The first one is the **fastText** (Joulin et al., 2017) classifier, which builds upon the CBOW model of fastText (Bojanowski et al., 2017), employing both uni-grams and n-grams to efficiently learn to perform text classification. We train the model for 100 epochs using default parameters.

The second one is instead based on **BERT** (Devlin et al., 2019), a pre-trained contextualized language model which has been shown to excel at a wide range of NLP tasks (Rogers et al., 2020). We fine-tune the BERT large cased model for text classification with Huggingface's Transformers library, for 10 epochs, with default parameters.

5.2.2. Results

We report the results in Table 2. The table shows the weighted F1 scores obtained against the real CV test data using three different training data - real, generated, and augmented (merging generated and real) CVs.

The first case, that of real CVs, constitutes an upper bound on the classification performance, given that train and test have similar, human-generated, linguistic form. Instead, in the second case, where the training set is fully synthetic, the model faces a greater challenge, in that it has to learn to abstract from the surface form of the synthetic CVs, which is different from that of the real ones, in order to be able to learn.

	BERT	fastText
Real CVs		
	0.88	0.81
Generated		
$\varepsilon = 10000$	0.71	**0.75**
$\varepsilon = 10$	0.71	0.75
$\varepsilon = 1$	0.73	0.74
$\varepsilon = 0.1$	0.73	0.68
Augmented		
	0.89	0.81

Table 2: Results for the extrinsic evaluation on Candidate Role Classification

Despite a certain loss in performance against the upper bound, which is to be expected, CVs generated with our approach can provide good performance (remember that there are nine possible classes - random baselines, at around 0.11, are reported in Figures 3 and 4). Most importantly, they do so while providing differential privacy, which is an extremely important added value, if not a necessary requirement, in the case of HR applications for NLP models.

Also, augmenting the dataset of real CVs with synthetic CVs gives a marginal advantage to the model. This seems to suggest that our methodology for the creation of training data may be of particular interest in cases where a big training set needs to be bootstrapped from a small dataset of CVs, and where there are no strong constraints on differential privacy. In such cases, the real and the synthetic sources of training data can be used together.

By closely inspecting the results, however, no clear decreasing trend emerges as more noise is added through the ε parameter. This is surprising, as one would expect that the gradual addition of noise, pushing further apart the distributions of the attributes across training and test set, should negatively impact classification performance. We interpret this as suggesting that the lion's share of successful classification is due to the prompts and GPT-2, and not so much to the features generated by the Bayesian network - at least for our current classification task, and for the attributes we have chosen.

	BERT masked	BERT random
Real CVs		
	0.65 (-.23)	0.12
Generated		
$\varepsilon = 10000$	**0.55** (-.16)	0.11
$\varepsilon = 10$	0.56 (-.15)	0.1
$\varepsilon = 1$	0.54 (-.19)	0.09
$\varepsilon = 0.1$	0.57 (-.16)	0.08
Augmented		
	0.64 (-.25)	0.13

Table 3: Further analyses for Candidate Role Classification with BERT: providing an empirical random baseline, and measuring the effect of removing explicit mentions of the candidate roles in text. We report the scores, together with the loss in performance from the original setting within brackets.

	fastText masked	fastText random
Real CVs		
	0.75 (-.06)	0.11
Generated		
$\varepsilon = 10000$	0.58 (-.17)	0.1
$\varepsilon = 10$	0.58 (-.17)	0.18
$\varepsilon = 1$	0.55 (-.19)	0.13
$\varepsilon = 0.1$	**0.62** (-.06)	0.2
Augmented		
	0.71 (-.10)	0.12

Table 4: Further analyses for Candidate Role Classification with fastText (same table structure as Figure 3).

However, we reckon that if the overall generated text, and not simply the mentions of the attributes, is the most important part of the training procedure, classifier performance could be driven by simple heuristics, as is sometimes the case in NLP tasks (Rosenman et al., 2020). In our case, in particular, the models may be simply looking for explicit mention of the candidate role in the generated text.

In order to investigate whether this is the case, we perform an additional ablation-style analysis, where we first remove from the training set explicit mentions of a CV's label, and then re-run the classification. More specifically, we mask explicit mentions of a CV candidate role by a generic mention 'worker' in the generated training sets (e.g. instead of 'I was employed as a Java Developer at', the CV would appear as 'I was employed as a worker at'). Results are reported in Table 3 and Table 4, under the mention 'masked'.

To show that performance is well above chance, we also report empirical random baselines ('random' columns in Figures 3 and 4), which fluctuate around the theoretical random baseline of $1/9 = 0.11$. They were computed by averaging the results of 100 train/test runs obtained after randomly permuting the nine labels of the training set.

When masking mentions of candidate roles, scores decrease in all cases. This indicates that both BERT and fastText classification models make use of the explicit mention of the class. BERT is affected more (average -0.19). fastText, instead, seems to be slightly more robust to our ablation-style manipulation (overall average -.13). Despite this loss in performance, however, scores remain well above the random baseline reported.

This validates our approach: it confirms, by looking at the cases where the synthetic CVs are involved (Generated and Augmented), that our generation procedure can create training data which encodes semantic information which is coherent with the candidate profile and role. Also, the robustness of our approach with respect to the noise injected in the probability distributions seems to promise that strong privacy constraints can be respected.

6. Conclusion

We have presented and empirically validated a methodology for the generation of synthetic CVs which reflect real-world distributions of candidate attributes while providing anonymization.

Synthetic CVs are interesting from two points of view. First, they are relevant for developing HR applications in compliance with personal data protection regulations, especially when powered by machine learning models. Secondly, they are a type of document that, in order to be generated, requires both structured data and raw text: therefore we expect that work on the generation of CVs could be adapted in principle to other similar types of text.

Our approach makes use of three stages: application of NLP techniques to extract candidate attributes; using Bayesian networks in order to learn the conditional dependencies between the attributes under differential privacy; and finally generating synthetic CVs by driving the generation of a Transformer-based generative language model through a manually-prepared set of prompts where the attribute sampled from the Bayesian network are plugged.

Evaluations based on linguistic properties indicate that the generated CVs have good-enough linguistic quality, and a machine learning evaluation (training a model for a classification task using the synthetic CVs instead of the real ones) shows that our approach, which provides differential privacy and a potentially unlimited amount of training data, offers promising performances for machine learning applications in HR.

Acknowledgement

This work was partially funded by the European Union's Horizon 2020 research and innovation programme under project TESTABLE, grant agreement No 101019206.

Bibliographical References

Alhussain, A. I. and Azmi, A. M. (2021). Automatic story generation: a survey of approaches. *ACM Computing Surveys (CSUR)*, 54(5):1–38.

Bojanowski, P., Grave, E., Joulin, A., and Mikolov, T. (2017). Enriching word vectors with subword information. *Transactions of the association for computational linguistics*, 5:135–146.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al. (2020). Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chen, R. J., Lu, M. Y., Chen, T. Y., Williamson, D. F., and Mahmood, F. (2021). Synthetic data in machine learning for medicine and healthcare. *Nature Biomedical Engineering*, 5(6):493–497.

Crossley, S. A., Weston, J. L., McLain Sullivan, S. T., and McNamara, D. S. (2011). The development of writing proficiency as a function of grade level: A linguistic analysis. *Written Communication*, 28(3):282–311.

Devlin, J., Chang, M.-W., Lee, K., and Toutanova, K. (2019). Bert: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Dwork, C., McSherry, F., Nissim, K., and Smith, A. (2006). Calibrating noise to sensitivity in private data analysis. In *Theory of cryptography conference*, pages 265–284. Springer.

Eubanks, B. (2022). *Artificial intelligence for HR: Use AI to support and develop a successful workforce*. Kogan Page Publishers.

Feng, S. Y., Gangal, V., Wei, J., Chandar, S., Vosoughi, S., Mitamura, T., and Hovy, E. (2021). A survey of data augmentation approaches for nlp. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 968–988.

Francopoulo, G. and Schaub, L.-P. (2020). Anonymization for the gdpr in the context of citizen and customer relationship management and nlp. In *workshop on Legal and Ethical Issues (Legal2020)*, pages 9–14. ELRA.

Freire, M. N. and de Castro, L. N. (2021). e-recruitment recommender systems: a systematic review. *Knowledge and Information Systems*, 63(1):1–20.

Gatt, A. and Krahmer, E. (2018). Survey of the state of the art in natural language generation: Core tasks, applications and evaluation. *Journal of Artificial Intelligence Research*, 61:65–170.

Habernal, I. (2021). When differential privacy meets nlp: The devil is in the detail. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1522–1528.

Howcroft, D. M., Belz, A., Clinciu, M.-A., Gkatzia, D., Hasan, S. A., Mahamood, S., Mille, S., Van Miltenburg, E., Santhanam, S., and Rieser, V. (2020). Twenty years of confusion in human evaluation: Nlg needs evaluation sheets and standardised definitions. In *Proceedings of the 13th International Conference on Natural Language Generation*, pages 169–182.

Igamberdiev, T. and Habernal, I. (2021). Privacy-preserving graph convolutional networks for text classification. *arXiv preprint arXiv:2102.09604*.

Jain, L., Vardhan, H., Kathiresan, G., and Narayan, A. (2021). Optimizing people sourcing through semantic matching of job description documents and candidate profile using improved topic modelling techniques. In *Advances in Artificial Intelligence and Data Engineering*, pages 899–908. Springer.

Jiechieu, K. F. F. and Tsopze, N. (2021). Skills prediction based on multi-label resume classification using cnn with model predictions explanation. *Neural Computing and Applications*, 33(10):5069–5087.

Joulin, A., Grave, É., Bojanowski, P., and Mikolov, T. (2017). Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431.

Keskar, N. S., McCann, B., Varshney, L. R., Xiong, C., and Socher, R. (2019). Ctrl: A conditional transformer language model for controllable generation. *arXiv preprint arXiv:1909.05858*.

Kmail, A. B., Maree, M., Belkhatir, M., and Alhashmi, S. M. (2015). An automatic online recruitment system based on exploiting multiple semantic resources and concept-relatedness measures. In *2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 620–627. IEEE.

Krishna, S., Gupta, R., and Dupuy, C. (2021). Adept: Auto-encoder based differentially private text transformation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2435–2439.

Lamba, D., Goyal, S., Chitresh, V., and Gupta, N. (2020). An integrated system for occupational category classification based on resume and job matching. In *Proceedings of the International Conference on Innovative Computing & Communications (ICICC)*.

Lauriola, I., Lavelli, A., and Aiolli, F. (2022). An introduction to deep learning in natural language processing: models, techniques, and tools. *Neurocomputing*, 470:443–456.

Liu, Y., Ott, M., Goyal, N., Du, J., Joshi, M., Chen, D., Levy, O., Lewis, M., Zettlemoyer, L., and Stoyanov, V. (2019). Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Lyu, L., He, X., and Li, Y. (2020). Differentially private representation for nlp: Formal guarantee and an empirical study on privacy and fairness. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2355–2365.

McNamara, D. S., Crossley, S. A., and McCarthy, P. M. (2010). Linguistic features of writing quality. *Written communication*, 27(1):57–86.

McSherry, F. and Talwar, K. (2007). Mechanism design via differential privacy. In *48th Annual IEEE Symposium on Foundations of Computer Science (FOCS'07)*, pages 94–103. IEEE.

Nasar, Z., Jaffry, S. W., and Malik, M. K. (2021). Named entity recognition and relation extraction: State-of-the-art. *ACM Computing Surveys (CSUR)*, 54(1):1–39.

Niedermayer, D. (2008). An introduction to bayesian networks and their contemporary applications. In *Innovations in Bayesian networks*, pages 117–130. Springer.

Nikolenko, S. I. et al. (2021). *Synthetic data for deep learning*. Springer.

Ore, O. and Sposato, M. (2021). Opportunities and risks of artificial intelligence in recruitment and selection. *International Journal of Organizational Analysis*, pages 1–12.

Paccosi, T. and Aprosio, A. P. (2021). Redit: A tool and dataset for extraction of personal data in documents of the public administration domain. In *CLiC-it 2021 Italian Conference on Computational Linguistics*.

Ping, H., Stoyanovich, J., and Howe, B. (2017). Datasynthesizer: Privacy-preserving synthetic datasets. In *Proceedings of the 29th International Conference on Scientific and Statistical Database Management*, pages 1–5.

Pitler, E. and Nenkova, A. (2008). Revisiting readability: A unified framework for predicting text quality. In *Proceedings of the 2008 conference on empirical methods in natural language processing*, pages 186–195.

Popova, Y. B. (2014). Narrativity and enaction: the social nature of literary narrative understanding. *Frontiers in psychology*, 5:895.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., et al. (2019). Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Roemmele, M., Gordon, A. S., and Swanson, R. (2017). Evaluating story generation systems using automated linguistic analyses. In *SIGKDD 2017 Workshop on Machine Learning for Creativity*, pages 13–17.

Rogers, A., Kovaleva, O., and Rumshisky, A. (2020). A primer in bertology: What we know about how bert works. *Transactions of the Association for Computational Linguistics*, 8:842–866.

Rosenman, S., Jacovi, A., and Goldberg, Y. (2020). Exposing shallow heuristics of relation extraction models with challenge data. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 3702–3710.

Schick, T. and Schütze, H. (2021). Generating datasets with pretrained language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6943–6951.

See, A., Pappu, A., Saxena, R., Yerukola, A., and Manning, C. D. (2019). Do massively pretrained language models make better storytellers? In *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, pages 843–861.

Shejwalkar, V., Inan, H. A., Houmansadr, A., and Sim, R. (2021). Membership inference attacks against nlp classification models. In *NeurIPS 2021 Workshop Privacy in Machine Learning*.

Sigurd, B., Eeg-Olofsson, M., and Van Weijer, J. (2004). Word length, sentence length and frequency–zipf revisited. *Studia linguistica*, 58(1):37–52.

Silva, P., Gonçalves, C., Godinho, C., Antunes, N., and Curado, M. (2020). Using nlp and machine learning to detect data privacy violations. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 972–977. IEEE.

Sinha, A. K., Akhtar, A. K., Kumar, A., et al. (2021). Resume screening using natural language processing and machine learning: A systematic review. *Machine Learning and Information Processing*, pages 207–214.

Strohmeier, S. (2022). Artificial intelligence in human resources-an introduction. In *Handbook of Research on Artificial Intelligence in Human Resource Management*. Edward Elgar Publishing.

Tucker, K., Branson, J., Dilleen, M., Hollis, S., Loughlin, P., Nixon, M. J., and Williams, Z. (2016). Protecting patient privacy when sharing patient-level data from clinical trials. *BMC Medical Research Methodology*, 16(1):5–14.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, Ł., and Polosukhin, I. (2017). Attention is all you need. *Advances in neural information processing systems*, 30.

Voigt, P. and Von dem Bussche, A. (2017). The eu general data protection regulation (gdpr). *A Practical Guide, 1st Ed., Cham: Springer International Publishing*, 10(3152676):10–5555.

Wang, S., Durrett, G., and Erk, K. (2020). Narrative interpolation for generating and understanding stories. *arXiv preprint arXiv:2008.07466*.

Wolf, T., Debut, L., Sanh, V., Chaumond, J., Delangue, C., Moi, A., Cistac, P., Rault, T., Louf, R., Funtowicz, M., et al. (2020). Transformers: State-of-the-art natural language processing. In *Proceedings of the 2020 conference on empirical methods in natural language processing: system demonstrations*, pages 38–45.

Yao, L., Peng, N., Weischedel, R., Knight, K., Zhao, D., and Yan, R. (2019). Plan-and-write: Towards better automatic storytelling. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7378–7385.

Zaroor, A., Maree, M., and Sabha, M. (2017). Jrc: a job post and resume classification system for online recruitment. In *2017 IEEE 29th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 780–787. IEEE.

Zhang, J., Cormode, G., Procopiuc, C. M., Srivastava, D., and Xiao, X. (2017). Privbayes: Private data release via bayesian networks. *ACM Transactions on Database Systems (TODS)*, 42(4):1–41.

Appendix: Examples of Generated CVs

We report three examples of synthetic CVs, generated using the methodology described above. To distinguish them from the text generated by GPT-2, manually written prompts (see Section 3.3) are reported in grey italics. The CVs were generated using a value of $\varepsilon = 0.1$, which ensures the highest privacy guarantees (see Section 3.2).

1. *Personal Details Name: Mariola Keudel Nationality: DE Address: Holtstr. 4/9 78091 Sebnitz Country of residence: DE Phone number: 08775 002138 Email: mkeudel@hofmann.de Field of expertise: IT - current position: Java Developer*

   Summary Dedicated and focused Java Developer who excels at helping companies gain control of their data. Work Experience and Career In my role as Java Developer at SCCM, I have worked on a number of IT projects.

   Studies and Education In my Ph.D. in Computer Science at Texas A&M University I specialized in the analysis of network traffic to understand how clients use network resources.

   Computer and Technical Skills My technical skills include Java, Java EE, SQL, Java EE 7, Spring and Spring Database.

   Languages I speak several languages, including French, Dutch, German, Italian and English.

   Hobbies and Interests My interests include learning new things about programming and software development, being a fun geek and a role model, and my desire to live in a big city and help others with their coding and programming skills."

2. *Personal Details Name: Fidel Peiró Nationality: ES Address: Sahin-Kreusel-Allee 2/4 40411 Neustadtner Waldnaab Country of residence: DE Phone number: 0293336135 Email: fpeir0̆0f3@quiros.net Field of expertise: IT - current position: Database Administrator*

   Summary With 5 years of experience working as Database Administrator, I have worked at several companies, such as: Microsoft, IBM, HP, Dell and HP Zellers.

   Work Experience and Career My responsabilities as Database Administrator at IBM include: Provide troubleshooting for database backups.

   Studies and Education I pursued a B.A. in Computer Science at University of Cincinnati studying Computer Science and Engineering.

   Computer and Technical Skills I have experience with Windows 8 and Microsoft Office Suite 2015.

   Languages I speak several languages, including Spanish, Italian, French, German and Japanese.

   Hobbies and Interests When I am not working, I like to go fishing, hiking, spending time with my family, playing golf and listening to music.

3. *Personal Details Name: Christopher Carter Nationality: US Address: Pasadizo de María Del Carmen Terrón 9 Puerta 5 Palencia, 11174 Country of residence: ES Phone number: +34720 46 84 32 Email: christopher.carter@chandler-garcia.net Field of expertise: IT - current position: Java Developer*

   Summary Highly motivated Java Developer who enjoys writing, networking and programming.

   Work Experience and Career In my role as Java Developer at Citrix, I was responsible for the entire Java development cycle from the very early days until the early years of its release in Java EE 6.

   Studies and Education I pursued a B.A. in Computer Science at Cornell University studying System Administration and Computer Systems.

   Computer and Technical Skills My technical skills include: programming, assembly language, Linux, C and C++ programming, Java/Ruby scripting and web development.

   Languages Aside from my native language, I speak English and German.

   Hobbies and Interests I enjoy hiking, reading books about languages and exploring my environment.