Wrocław University
of Science and Technology

Faculty of Information and Communication Technology
Field of study: **ARTIFICIAL INTELLIGENCE**

Master's Thesis

REPRESENTATION LEARNING USING VARIATIONAL AUTOENCODERS

Patryk Szelewski

keywords:
representation learning, variational autoencoder, neural networks, machine learning, labor market intelligence

short summary:
The thesis contains an overview of existing representation learning methods, focusing on variational autoencoders. Furthermore, a new method for creating data representations that can be used in labor market intelligence (LMI) solutions is proposed.

Supervisor	dr hab. inż. Tomasz Kajdanowicz
	Title/degree/name and surname

The final evaluation of the thesis

Chairman of the Diploma Examination Committee	.....	.....	.....
	Title/degree/name and surname	grade	signature

For the purposes of archival thesis qualified to:*

a) category A (perpetual files)

b) category BE 50 (subject to expertise after 50 years)

* Delete as appropriate

stamp of the faculty

Wrocław 2022

Streszczenie

Ręczna ekstrakcja cech w celu poprawy rezultatów uzyskiwanych przez modele uczenia maszynowego jest czasochłonna i kosztowna, ponieważ wymaga zaangażowania ekspertów dziedzinowych. W celu zautomatyzowania procesu ekstrakcji cech z danych, rozwijana jest dziedzina sztucznej inteligencji nazywana uczeniem reprezentacji. Jedną z najbardziej przełomowych metod z ostatnich lat wykorzystywanych w uczeniu reprezentacji, jest wariacyjny autokoder, pozwalający na tworzenie niezależnych (ang. disentangled) reprezentacji danych.

Dzięki znacznemu upowszechnieniu się technik komputerowych, w ostatnich latach obserwuje się wzrost zapotrzebowania na systemy do automatycznego przetwarzania aplikacji kandydatów do pracy w celu usprawnienia procesu rekrutacyjnego.

W pracy omówiono istniejące metody uczenia reprezentacji. Ponadto przedstawiono metodę pozwalającą na tworzenie niezależnych reprezentacji profili kandydatów do podjęcia pracy *Disentangled Representation of Applicants with VAE* (DRAV). Zaprojektowano także dedykowane metryki do ewaluacji zaproponowanego modelu. Wyniki pokazują, że zaproponowana architektura może być przydatna nie tylko do tworzenia niezależnych reprezentacji, ale również do automatycznego tworzenia syntetycznych, nieistniejących w rzeczywistości profili o zdefiniowanych cechach.

Abstract

Handmade feature engineering for machine learning models is an expensive process because it requires domain knowledge about a solved problem. To automate the process of extracting features from data, a branch of artificial intelligence, called representation learning, is developed. One of the biggest representation learning breakthroughs in recent years is a technique called variational autoencoder, which is capable of creating disentangled data representation.

Due to the rapid development of computer techniques, in recent years, a growing demand for methods for automatic processing of job candidates has been observed.

In this work, we discuss currently used representation learning techniques, focusing on variational autoencoders. Furthermore, we propose a model, namely *Disentangled Representation of Applicants with VAE* (DRAV), designed to create a disentangled representation of job candidates. We also define custom metrics for model evaluation. The results show that the proposed model architecture can be useful not only to create disentangled data representation but also to generate new, synthetic descriptions of job applicants.

Contents

1 Introduction

Recent years have brought the attention of scientists to the power of neural networks. The branch of artificial intelligence neglected for many years due to the fast growing amount of processing power became one of the technologies shaping the world around us. Artificial neural networks, in addition to processing power, need a large amount of data.

Luckily, with more computers, processing power, and memory available, people store more and more data, which can be used to create complex neural models. With a large amount of data, one of the challenges became finding techniques that could capture hidden factors in the data, which can help reduce the size of the data and make it more understandable to computers. These tasks are handled by a class of machine learning approaches, namely *representation learning*, therefore, we believe that *representation learning* is one of the most challenging and promising fields of artificial intelligence (AI). In this thesis, we want to explore the possibilities of one of the most crucial representation learning techniques introduced in recent years, namely variational autoencoders (VAE) in new real-world applications.

1.1 Motivation

Multiple works in the domain of representation learning tend to describe general approaches to representation learning. The methods are tested on multiple datasets and tasks that measure their flexibility. We argue that general approaches, although they are good in theory, can be beaten by methods adjusted to the specific problem.

One of the domains in which the amount of data is growing fast is the labor market. The term Labor Market Intelligence (LMI) refers to algorithms and tools that provide useful knowledge for the development of the labor market, using artificial intelligence. LMI solutions focus mainly on the task of matching job candidates with their offers (Person–Job Fit Task), but they lack the scientific approach to the problem of creating complex representations of job applicants. We believe that proposing a method to create such representation can be beneficial to downstream tasks such as the previously mentioned Person–Job Fit Task. Furthermore, the method can be later adopted to represent similar entities, for example, job offers or employers.

1.2 Aim and scope

The main objective of this work is to propose a new method dedicated to creating a disentangled representation of job candidates. In order to achieve the objective, we define the following scope:

- An extensive literature review on deep learning (DL), representation learning, variational autoencoders (VAE) and their usage in labor market intelligence (LMI),
- proposing a new model to create disentangled representation of job candidates,
- defining metrics to compare created representations,
- testing how changing the hyperparameters and mechanisms of proposed representation learning technique affect achieved results.

1.3 Research problems

The following problems were identified and investigated in the thesis:

- How to use variational autoencoders to process natural language in an efficient way?
- How to create a disentanglement representation of job candidates?
- How can information on skills, education, and known languages be extracted from the textual description of the job candidate?
- How to control the capacity of different generative factors in the latent space?
- How to graft the competences of one job applicant's profile to the other?

1.4 Contribution

The main contribution of the author is a new model, namely *Disentangled Representation of Applicants with VAE* (DRAV). The DRAV network, in addition to creating the disentangled representation, is designed to solve two other problems, formulated for the needs of this work tasks (i) competence disentanglement control task and (ii) competence transfer task. Furthermore, we have created a custom dataset that contains multiple job candidate profiles collected from the Fiverr.com website. Finally, we propose evaluation evaluation methods for: (i) the competence disentanglement control task and (ii) measuring the general disentanglement of the job applicants' representations.

1.5 Work structure

The work has the structure as follows.

- Chapter 2 contains the basic concepts and definitions necessary to understand the scope of the thesis.
- Chapter 3 describes in detail the architecture of the proposed DRAV network.

- Chapter 4 contains the complete description of the experiments carried out: description of the custom dataset created for the needs of this project, experiment plan, description of the proposed metrics, and basic hyperparameter configuration of the DRAV model followed by results achieved.
- Chapter 5 contains the summary of the entire work and provides ideas for possible future work on the topic of representation learning using variational autoencoders.

2 Background and related work

This chapter is a summary of related research conducted in the fields of representation learning, natural language processing, and variational autoencoders.

2.1 Feature engineering

Machine learning algorithms are driven solely by data, and there is probably no easier way to provide the data to a computer than using a set of numerical features. These numerical features can be easily understood by almost anyone, for example, the temperature of the air described in Celsius, or they can represent some complex knowledge about the problem used by specialists, for example, the Specific Differential Phase (KDP) used to determine areas with the heaviest rainfalls [1].

Complex features such as the previously mentioned Specific Differential Phase are calculated on top of simpler features, for example the temperature of the air. The process of creating complex features that can benefit the solution of specific problems is called feature engineering [2].

There is no doubt that handmade feature engineering [3] is an expensive process. First, it requires both domain knowledge about a solved problem and knowledge about machine learning models for which we are going to use extracted features. One way to reduce the costs associated with manually extracting features is to use more processing power.

The term *representation learning* present in this work title refers to a kind of automated feature extraction process that is often carried out using deep neural networks and is described in more detail in Section 2.4.

2.2 Deep learning

In his book, Ian Millington states that the only way any algorithm can outperform another is by consuming more processing power or optimizing for a specific set of problems [4]. Due to the rapid increase in the amount of processing power available [5], computer engineers were increasingly prone to use ”universal keys” instead of looking for algorithms dedicated to their problem. This universal key was named deep learning.

Deep learning is a subfield of machine learning (Figure 2.1). What distinguishes deep learning from machine learning is that deep learning can use a large amount of data to extract the necessary features [6] (Figure 2.2). To better understand how deep learning models work, some basic concepts have to be defined.

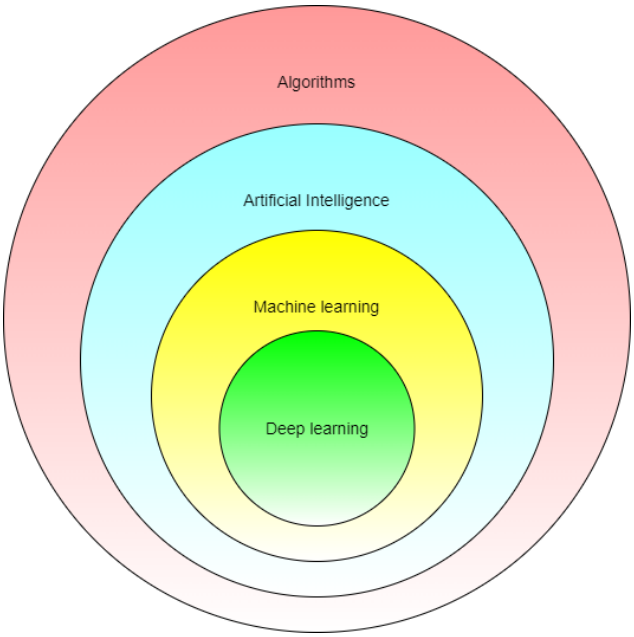


Figure 2.1: Deep learning as a subfield of Artificial Intelligence

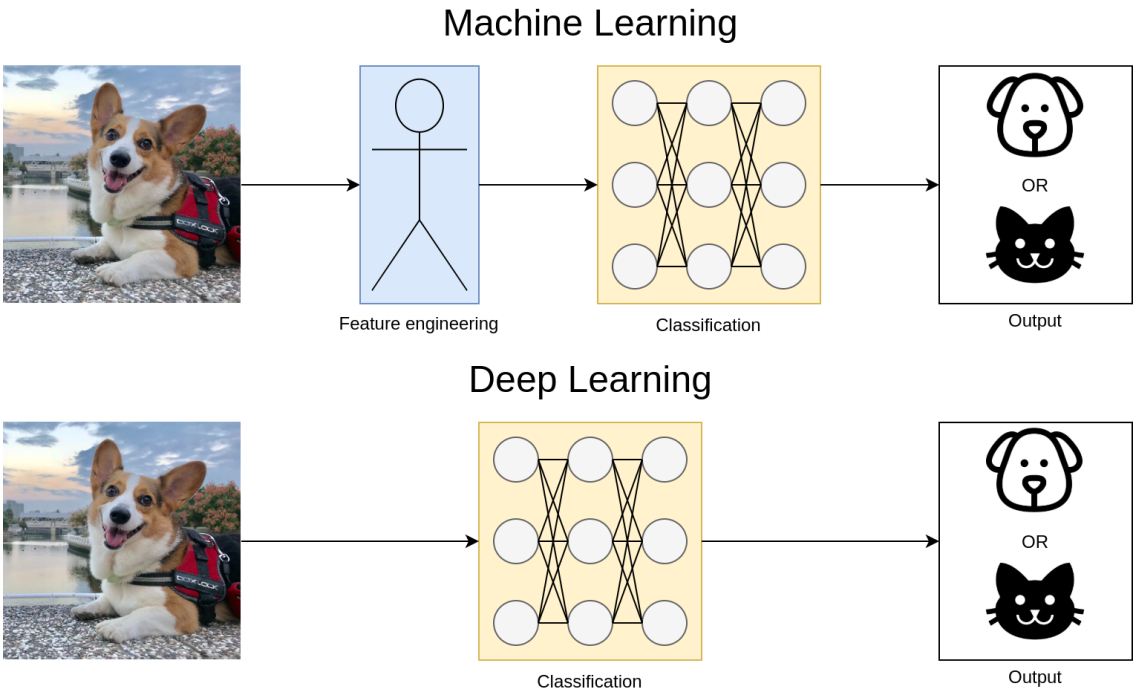


Figure 2.2: Difference between machine learning and deep learning

Multi Layer Perceptron (MLP)

The main tool used in deep learning are neural networks. The first example of a neural network was called *perceptron* [7] and was presented by Frank Rosenblatt in 1958. It was a simple network built using only input and output layers. The architecture of the perceptron is shown in Figure 2.3. For each of the n output units, the activation value is defined:

$$a(x) = \sum_{i=1}^{n} w_i x_i + b, \tag{2.1}$$

where:

- w_i is called a weight of connection to the input with index i,
- x_i in above equation is an activation value,
- b is called a bias.

To enable neural networks to approximate nonlinear functions, an activation function is defined which is applied to the activation value of a neuron. Commonly used activation functions are:

- ReLU:
$$f(a) = \begin{cases} a & \text{for } a \geq 0 \\ 0 & \text{for } a < 0 \end{cases}, \tag{2.2}$$
- sigmoid function:
$$\sigma(a) = \frac{1}{1 + e^{-a}}, \tag{2.3}$$
- hyperbolic tangent
$$tgh(a) = \frac{e^{a} - e^{-a}}{e^{a} + e^{-a}}. \tag{2.4}$$

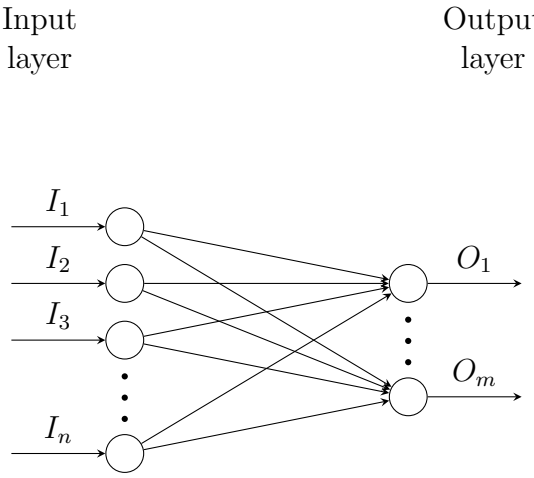


Figure 2.3: Perceptron structure

For neural networks, we can apply the universal approximation theorem. It states that multilayer feed-forward neural networks with at least one hidden layer, having finite hidden neurons with arbitrary activation function [1] can approximate any function in C(R^m):

Theorem 2.1 (universal approximation theorem [8]) *Let $\phi(.)$ be an arbitrary activation function. Let $X \subseteq R^m$ and X is compact. The space of continuous functions on X is denoted by $C(X)$. Then $\forall f \in C(X), \forall \epsilon > 0 : \exists n \in N, a_{ij}, b_i, w_i \in R, i \in \{1 \dots n\}, j \in \{1 \dots m\}$:*

$$(A_n f)(x_1 \dots x_m) = \sum_{i=1}^{n} = w_i \phi(\sum_{j=1}^{m} a_{ij} x_j + b_i). \tag{2.5}$$

$A_n f$ is an approximation of the function f using a neural network with n hidden layers. This approximation fulfills the following independence:

$$||f - A_n f|| < \epsilon. \tag{2.6}$$

Taking into account the universal approximation theorem and the difficulties in training complex neural network architectures, researchers have worked with *shallow neural networks*. The architecture of a shallow neural network with a hidden layer is shown in Figure 2.4. After a long time, it became clear that (*deep neural networks*) having multiple hidden layers can learn more abstract features and dependencies than shallow networks.

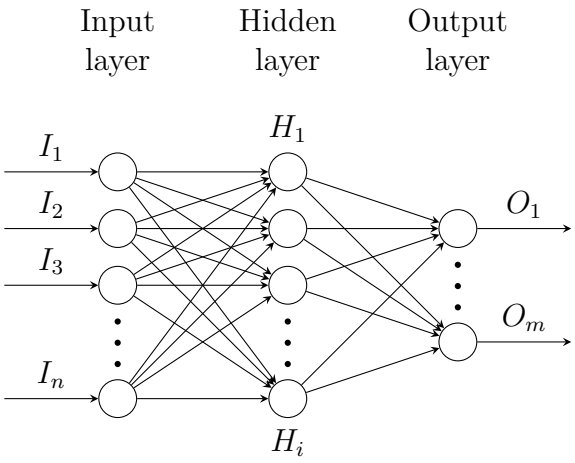


Figure 2.4: Architecture of a shallow feed-forward network

.

2.2.1 Backpropagation

Neural networks are used to approximate a function. The quality of the approximation can be improved by minimizing a function called a loss function. If the loss function and all activation functions are differentiable, differential calculus and chain rule can be used to calculate the loss function gradient for each of the model weights $\frac{\partial J}{\partial w_{i,j}^l}$. Gradients weighted

[1] ϕ is activation function if and only if, ϕ jest bounded and $\lim_{x \to +\infty} \phi(x) = a$, $\lim_{x \to -\infty} \phi(x) = b$, $a \neq b$

by learning rate λ can be used to update the weights $w_{i,j}$ of the model. This method is called backpropagation [9] and can be formulated using equation 2.7:

$$w^l_{i,j} \rightarrow w^l_{i,j} - \lambda \frac{\partial J}{\partial w^l_{i,j}}. \tag{2.7}$$

2.2.2 Convolutional neural networks

Dense neural networks with all neurons in a layer connected to all neurons in a previous layer can generalize complex dependencies at the global level, but they have problems catching local dependencies in input data. The most popular examples of these dependencies are the local features of the images. These local features are elements that together build a bigger object. For example, to identify a dog in the picture, the neural network will probably try to find two front-facing eyes, two ears, a triangle-shaped nose and a muzzle below them (Figure 2.5). It is not important whether those traits will be identified in the corner of the image or in the center.

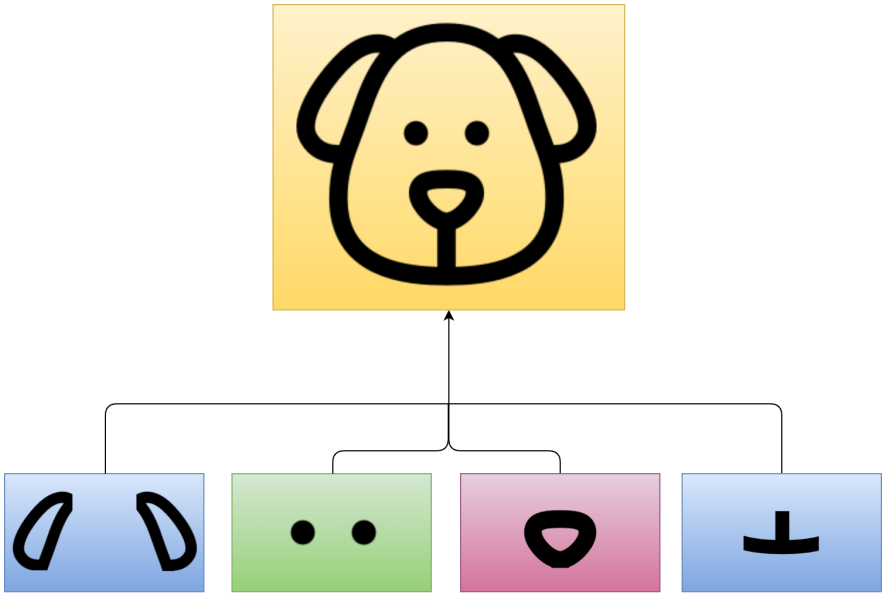


Figure 2.5: Features building a dog image: ears, eyes, nose and muzzle

Another downside of dense neural networks is their size. To train them, we have to store them in memory and update each pair of neurons in adjoining layers.

To solve the issues mentioned above, convolutional networks were introduced in 1980 [10]. The basic operation for convolutional networks is convolution, which for 2D signals is presented in the figure 2.6. Using convolution, one can extract features of an image such as edges (e.g. using the Sobel filter) or remove noise (e.g., using the Gaussian filter).

The weights of the neural networks are made up of *slices*, which are also called *leaves* and are responsible for detecting local features in the data. We can think of a slice as a convolution filter with trainable parameters.

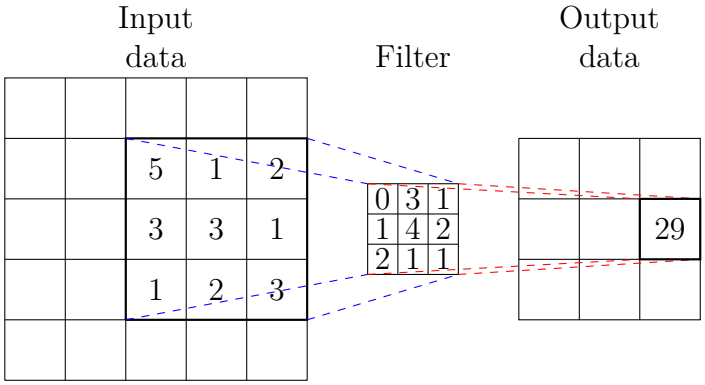


Figure 2.6: Example of the 2D convolution operation

As mentioned above, one of the advantages of convolutional networks is their small size, which can greatly improve performance when processing complex data structures. To further improve performance, convolutional layers are often connected with *max-pooling* or *average-pooling* layers. The max-pooling operation returns the maximum value of neuron activation in the area defined by the kernel size. An example of maximum pooling is presented in Figure 2.7. The average pooling layer, instead of using the max operation to determine its output, calculates an average of the values.

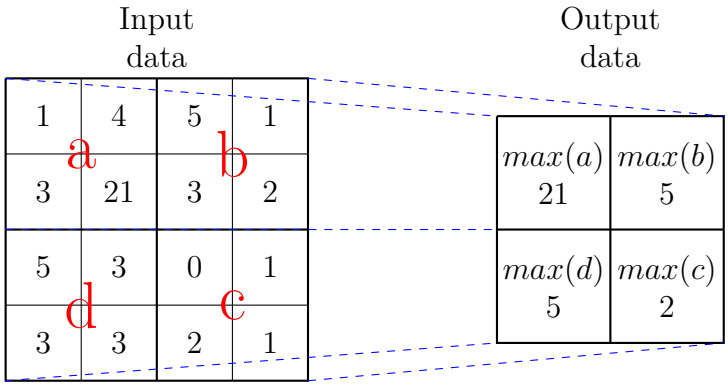


Figure 2.7: Example of maximum-pooling operation

2.2.3 Recurrent neural network

Previously mentioned neural network architectures: multilayer perceptrons and convolutional networks are examples of feedforward networks. To generate their output, they use the information from one timestamp t. To use historical data to make a prediction, *recurrent neural networks* (RNNs) [9] are used. Vanilla RNN for each of it's units makes an operation described

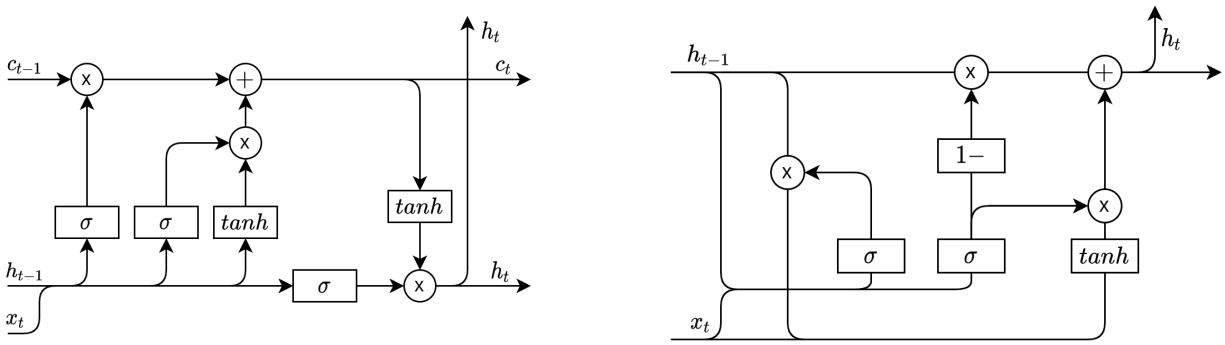
using the equation:

$$out_t = f(\sum_{i=1}^{n} w_i x_{i,t} + \sum_{j=1}^{n} u_j out_{j,(t-1)} + b), \tag{2.8}$$

where:

- out_t is an output of a neuron on time stamp t; f is the neuron's activation function,
- $w_{i,t}$ is the weight of connection between a unit and the activation value of the i-th neuron on time stamp t ($x_{i,t}$),
- u_j is a weight of connection with output j on previous timestamp $(t-1)$,
- b is a bias.

The big problem when using standard RNNs for long sequences is a problem of vanishing gradients [11]. To facilitate this issue, *Long short-term memory* (LSTM) [12] and *Gated Recurrent Units* (GRU) [13] layers were introduced. Due to the use of gates mechanism, the LSTM and GRU layers can transfer information in long sequences in a more efficient way. Both architectures are shown in Figure 2.8.



(a) LSTM network scheme

(b) GRU network scheme

Figure 2.8: LSTM and GRU networks schemas

2.3 Natural language processing

The term *natural language processing* (NLP) refers to a rapidly developing subfield of artificial intelligence. NLP connects artificial intelligence with linguistics [14] to create an interface between human language and the language used by computers.

In general, we can distinguish three main ways of learning machine learning algorithms: *supervised* – processed text is labeled with a correct answer, the algorithm is trying to predict during inference, *unsupervised* – the algorithm is trying to automatically find patterns in the data, or using *reinforcement learning* – the algorithm learns by interacting with the environment.

As examples of NLP tasks in which a supervised approach is used, we can mention Sentiment Analysis [15, 16] – detecting positive or negative sentiment in texts, Emotion Recognition [17, 18] – labeling data with emotions expressed by a writer, or SPAM detection [19, 20] – deciding whether the text is SPAM to filter it out.

Recent years have brought the attention of researchers to the idea of representing words as vectors using an unsupervised approach. These vectors are designed to store semantic information about words. Although this idea is not new and has been discussed in many works before [21, 22, 9], the year 2013 and the work of Mikolov et al. named *Word2Vec* [23] have started the rapid development of similar methods. We can distinguish *non-contextual* methods such as previously mentioned *Word2Vec* [23], *fastText* [24] or GloVe [25]. In non-contextual methods, a word vector does not depend on the context of its usage, so the word "orange" has the same word vector assigned in both sentences: "An orange fell from the tree." and "Orange is one of the largest telecom companies". There are also methods that can produce different representations of the word "orange" for each of the aforementioned examples. These methods are called *contextual models* and as one of them we can mention transformer-based methods using the attention mechanism [26] such as *BERT* [27] and its subsequent implementations such as *RoBERTa* [28] and *DistilBERT* [29].

2.4 Representation learning

As mentioned in Section 2.1, a machine learning approach that leads to automatic data transformation that improves the performance of algorithms based on transformed data is called *representation learning* [2].

The goal of representation learning is to transform the data into *latent space* that should encode as much meaningful information as possible. What makes representation learning interesting as a field of studies and is a source of motivation for this work is that latent space can encode many non-task-specific priors from the real world. In the literature [2] the following examples of real-world priors are mentioned:

- *Smoothness*: implies that if $x \approx y$ then $f(x) \approx f(y)$.
- *Multiple explanatory factors*: the data can be described using different underlying factors that can later be used in different machine learning tasks. Identifying these factors and disentangling them is the goal of a subfield of representation learning called disentangled representation learning [30] (see Section 2.4.3) and is one of the main interests of this work.
- *Manifolds*: probability mass is concentrated in regions that have a dimensionality lower than the original dimensionality of the data. It is used to encode the data in a lower-dimensional space than the original data using autoencoders (see Section 2.4.1)

What distinguishes representation learning from other machine learning tasks, such as classification or regression, is the lack of direct metrics to determine how good the representation is. The first approach is to evaluate the quality of the representation created by its application [31, 32]. For example, a representation can be evaluated by attaching a classification head to

it and testing on a classification task. We can then measure the performance of the representation using standard metrics used in classification, such as accuracy, recall, precision, and the f-1 score. Another approach is to use metrics designed for probability distributions such as *mutual information* [33].

Mutual information (MI) [34] between random variables X and Y with probability density functions, respectively $p(x)$ and $p(y)$ is defined as the Kullback-Leibler divergence (D_{KL}) between the joint density $p(x, y)$ and the product of marginal densities $p(x)p(y)$ and can be formulated using equation 2.9:

$$I(X;Y) = D_{KL}(p(x,y)||p(x)p(y)) = E_{p(x,y)}\left[log\frac{p(x,y)}{p(x)p(y)}\right]. \tag{2.9}$$

In the epoch of fast-growing excitation of the usage of deep learning for a vast amount of tasks, the representation learning domain is also dominated by deep models. To the most popular methods of learning a representation, we can count:

- Deterministic Methods – Methods that use deterministic functions to code and decode information:
  - Autoencoder [9] – (Section 2.4.1).
- generative methods – Methods that model posterior probability distribution to generate new data, similar to the input.
  - Variational autoencoder (VAE) [35] (Section 2.4.2) which along disentangled representation learning and job candidate representation is one of the main interests in this work,
  - Generative adversarial network (GAN) [36],
  - Flow-based Deep Generative Models [37]

2.4.1 Autoencoders

Autoencoders which have been first mentioned in [9] can be defined as neural networks trained in unsupervised manner, designed to reconstruct their inputs [38]. The basic autoencoder is made up of two components: *encoder* and *decoder*, and its schema is shown in Figure 2.9. If we remove all nonlinear operations from the encoder and decoder, the latent space will be identical to the one created using *Principal Component Analysis* (PCA) [39]. So, we can think of autoencoders as a generalized approach to PCA, in which instead of finding a low-dimensional hyperplane in which data lie, we are looking for a manifold that is not an affine space.
The purpose of the encoder is to transform the input into *latent space* – the informative representation of the data, which, as previously mentioned, is a generalization of PCA output, using nonlinear transformations. The role of a decoder is to create a reconstruction of the input, using the data encoded in the latent space that minimizes the value called *reconstruction loss*. We can formulate the following. The autoencoder tries to find the functions $g : \mathbb{R}^n \to \mathbb{R}^l$ and $f : \mathbb{R}^l \to \mathbb{R}^n$ that satisfy the equation:

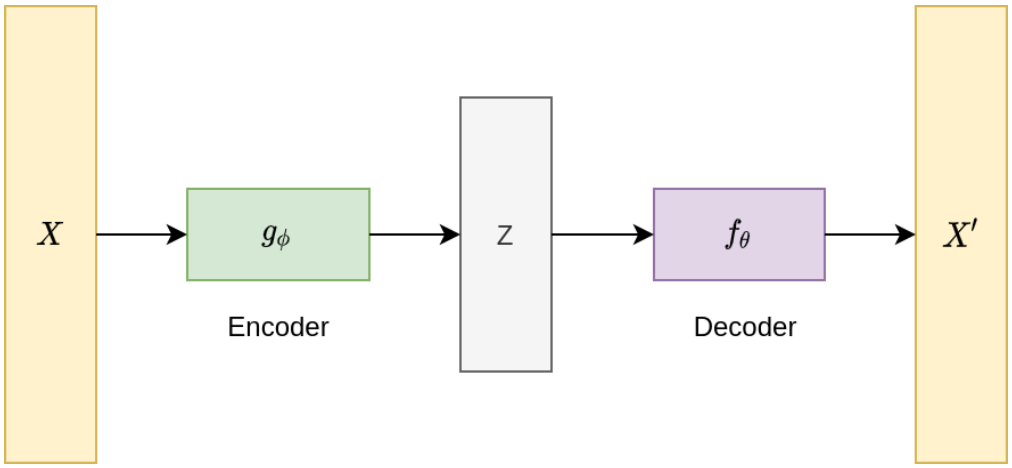


Figure 2.9: Autoencoder architecture

$$\arg\max_{g,f} E[\Delta(x, f \circ g(x))], \tag{2.10}$$

There are a variety of domains in which autoencoders are used, such as anomaly detection [40], image denoising [41], and image compression [42]. After a few modifications, autoencoders can also be used to generate new unseen data. This special type of autoencoder is called *Variational Autoencoder* [35], and its application to the task of learning the representation of job applicants is one of the main topics of this work.

2.4.2 Variational autoencoders

Introduction

The variational autoencoder (VAE) was introduced in [35] and can be defined as an autoencoder that learns not functions $g : R^n \to R^l$ and $f : R^l \to R^n$, but posterior probability distribution $q_\phi(z|x)$ and the likelihood $p_\theta(x|z)$. The architecture of the VAE network is shown in Figure 2.10.

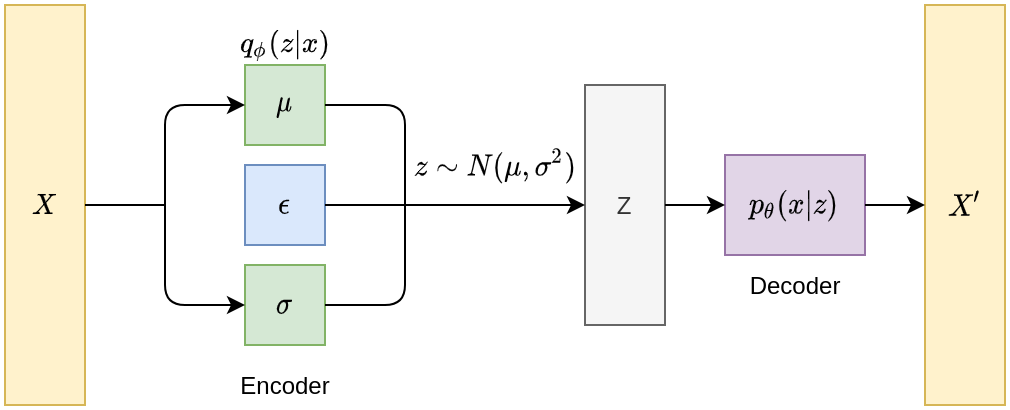


Figure 2.10: Variational autoencoder architecture

To better understand how the variational autoencoder works, some concepts must be introduced.

Variational inference

Bayes' theorem is a key to understanding the core idea behind variational autoencoders. It can be stated mathematically by equation 2.11:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}, \tag{2.11}$$

where:

- $P(A|B)$ is a conditional probability – the probability of A if B is certain,
- $P(B|A)$ is a conditional probability – the probability of B if A is certain; $P(B|A) = \mathrm{L}(A|B)$ (likelihood of A, given B),
- $P(A)$ and $P(B)$ are marginal probabilities of A and B, respectively.

Inference in Bayesian statistics refers to the calculation of the posterior distribution $p(Z|X)$, defined as the conditional probability of the parameters Z of the Bayesian model, given an observation X. The posterior probability $p(Z|X)$ can be calculated using Equation 2.12:

$$p(Z|X) = \frac{p(Z \cap X)}{p(X)}, \tag{2.12}$$

where:

- $p(Z \cap X)$ is the joint distribution of Z and X,
- $p(X)$ is the marginal distribution of observation X.

The marginal distribution $p(X)$, is received by solving the equation 2.13

$$p(x) = \int p(X|Z)p(z)dz. \tag{2.13}$$

In practice, in many cases it is not possible to compute the value of $p(x)$ using equation 2.13, due to the problem of intractability. This problem appears if complicated likelihood functions $p(x|z)$ were used, for example, a network with a hidden nonlinear layer. An approach to this issue is to find a close approximation of the true posterior. To measure the similarity between two distributions, the Kullback-Leibler divergence (KL divergence) [34] can be used. The equation for KL divergence in the discrete case can be defined using Equation 2.14:

$$D_{KL}(P||Q) = \sum_i P(x) \log \frac{P(x)}{Q(x)}, \tag{2.14}$$

where

- P and Q are discrete probability distributions.

The lower the KL divergence, the more similar the two distributions are. KL divergence is closely related to *entropy* [33] and *cross entropy* terms. The relationship of these terms can be defined using the equation 2.15:

$$\begin{aligned} D_{KL}(P||Q) &= \sum_i P(x) \log \frac{P(x)}{Q(x)} \\ &= \sum_i P(x) \log \frac{1}{Q(x)} - \sum_i P(x) \log \frac{1}{P(x)} \\ &= H(P,Q) - H(P), \end{aligned} \tag{2.15}$$

where:

- $H(P,Q)$ is the cross entropy of P and Q,
- $H(P)$ is the entropy of P.

The goal of variational inference is to find the distribution $q(Z)$ that minimizes the Kullback-Leibler divergence with respect to the true posterior $p(Z|X)$. The distribution $q(Z)$ should belong to the same family of probability distributions ψ as $p(Z|X)$. This optimization task can be stated using equation 2.16:

$$q^*(Z) = \arg\min_{q(Z)\in\psi}(D_{KL}(q(Z)||p(Z|X))), \tag{2.16}$$

where:

- $q^*(Z)$ is the optimal form of $q(Z)$.

As equation 2.16 contains the term $p(Z|X)$, already defined in equation 2.12, the value of $q^*(Z)$ cannot be calculated directly. To work around this problem, 2.16 can be rewritten to the form expressed by equation 2.17:

$$\begin{aligned} D_{KL}(q(Z)||p(Z|X)) &= \mathbb{E}_{q(Z)}[\log \frac{q(Z)}{p(Z|X)}] \\ &= \mathbb{E}_{q(Z)}[\log q(Z)] - \mathbb{E}_{q(Z)}[\log p(Z|X)] \\ &= \mathbb{E}_{q(Z)}[\log q(Z)] - \mathbb{E}_{q(Z)}[\log \frac{p(Z \cap X)}{p(X)}] \\ &= \mathbb{E}_{q(Z)}[\log q(Z)] - \mathbb{E}_{q(Z)}[p(Z \cap X)] + \mathbb{E}_{q(Z)}[\log p(X)] \end{aligned} \tag{2.17}$$

Equation 2.17 can be organized in the form:

$$\begin{aligned} \log p(X) &= [\mathbb{E}_{q(Z)}[\log p(Z \cap X)] - \mathbb{E}_{q(Z)}[\log q(Z)]] + D_{KL}(q(Z)||p(Z|X)) \\ &= [\mathbb{E}_{q(Z)}[\log \frac{p(Z \cap X)}{q(Z)}] + D_{KL}(q(Z)||p(Z|X)) \\ &= \underbrace{ELBO(q)}_{\leq \log p(X)} + \underbrace{D_{KL}(q(Z)||p(Z|X))}_{\geq 0}, \end{aligned} \tag{2.18}$$

where:

- ELBO is the evidence lower bound; log-likelihood $\log p(X) \geq ELBO(q)$ [35].

The log probability of observations $\log p(X)$ term in equation 2.18 does not depend on $q(Z)$. If $\log p(X)$ can be treated as constant with respect to $q(Z)$, maximizing ELBO also minimizes the Kullback-Leibler divergence between $q(Z)$ and the true posterior $p(Z|X)$, what is the goal stated in equation 2.16.

The ELBO can be rewritten in the form of:

$$ELBO(q) = [\mathbb{E}_{q(Z)}[\log \frac{p(Z \cap X)}{q(Z)}] = \mathbb{E}_q(Z)[\log p(X|Z)] - D_{KL}(q(Z)||p(Z)) \tag{2.19}$$

As mentioned, the goal is to maximize ELBO. As shown in equation 2.19, if doing so, the log-likelihood of the data is maximized, while the KL divergence between the approximate posterior $q(Z)$ and the prior $P(Z)$ is minimized.

VAE objective

The generative process performed by variational autoencoders can be visualized using a graphical model presented in Figure 2.11.

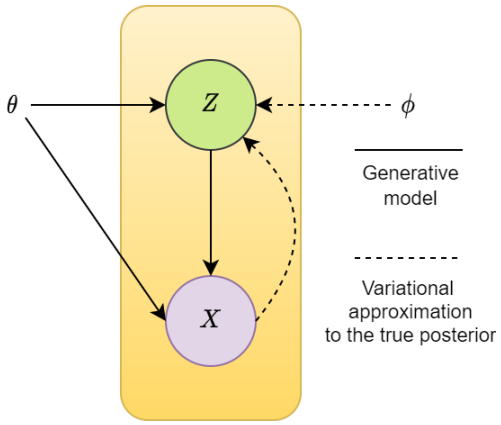


Figure 2.11: Considered directed graphical model

The problem of computing the variational approximation ($q(Z)$) to the true posterior ($p(Z|X)$) was already discussed in Section 2.4.2. In summary, to obtain the best approximation of ($p(Z|X)$), ELBO (equation 2.19) must be maximized.

One way to do this is to use two neural networks, namely the encoder $q_\phi(Z|X)$ and the decoder $p_\theta(X|Z)$, where ϕ and θ are weights and biases of the encoder and decoder networks, respectively. As the encoder network is responsible for modeling the probability distribution $q(Z)$, all occurrences of $q(Z)$ can be replaced by $q_\phi(Z|X)$. Using this notation and the Equations 2.18, 2.19, the ELBO inequality can be written down:

$$\begin{aligned} \log p_\theta(x) &\geq \mathbb{E}_{z\sim q_\phi(z|x)}[\log \frac{p_\theta(x,z)}{q_\phi(z|x)}] \\ &= \mathbb{E}_{z\sim q_\phi(z|x)}[\log p_\theta(x|z) - D_{KL}(q_\phi(z|x)||p(z))]. \end{aligned} \tag{2.20}$$

The objective of neural networks is to minimize the loss function. To encourage the encoder and decoder networks to maximize $ELBO(q)$, the loss function must be written in the form $-ELBO(q)$:

$$\begin{aligned} L(\theta, \phi, x) &= -\mathbb{E}_{z \sim q_\phi(z|x)}[\log p_\theta(x|z) + D_{KL}(q_\phi(z|x)||p(z))] \\ &= L_{rec}(\theta, \phi, x) + D_{KL}(q_\phi(z|x)||p(z))], \end{aligned} \tag{2.21}$$

where:

- L_{rec} is the expected negative log-likelihood of the data called *reconstruction loss*,
- $D_{KL}(q_\phi(z|x)||p(z))]$ is the KL divergence between the approximation of the true posterior $q_\phi(z|x)$ and the prior distribution, typically $\mathcal{N}(0, 1)$.

Both approximate posterior and the prior are assumed to be multivariate Gaussian. Therefore, it can be calculated analytically:

$$\begin{aligned} D_{KL}[\mathcal{N}(\mu_{z|x}, \Sigma_{z|x})||\,\mathcal{N}(0, 1)] &= \frac{1}{2}(\mathrm{tr}(\Sigma_{z|x}) + \mu_{z|x}^T \mu_{z|x} - k - \log\det(\Sigma_{z|x})) \\ &= \frac{1}{2}\sum_{i=1}^{k}(\sigma_i^2 + \mu_i^2 - \log\sigma_i^2 - 1), \end{aligned} \tag{2.22}$$

where:

- $\mu_{z|x}$ and $\Sigma_{z|x}$ are the parameters of the approximate posterior distribution.

It should be noted that for the loss function $L(\theta, \phi, x)$ without the $D_{KL}(q_\phi(z|x)||p(z))]$ term, the model converges to the deterministic autoencoder form already described in Section 2.4.1.

VAE training details

In the deterministic autoencoder (DAE), gradients can be propagated back from the decoder to the encoder, which can be used to adjust weights ϕ and θ using stochastic gradient descent [43]. As the latent variable Z in variational autoencoders is sampled from a probability distribution $q_\phi(z|x)$, rather than being the result of a deterministic process, gradients cannot be propagated directly back. In [35] *reparametrization trick* was introduced. Instead of sampling from the parameterized Gaussian $\mathcal{N}(\mu_{z|x}, \Sigma_{z|x})$, ϵ is sampled using a fixed distribution $\mathcal{N}(0, 1)$ and then transformed into the form of

$$z = \mu_x + \sigma_x \epsilon. \tag{2.23}$$

Thanks to this transformation, the gradients of the decoder can be propagated back through μ and σ to the encoder.

2.4.3 Disentangled representation learning

In the data, we can find multiple underlying explanatory factors that can define its nature. These factors tend to change independently, and only some of them change at the time when switching to other data points. The goal of *disentangled representation learning* is to identify and extract the mentioned factors of variation [2]. An example of a disentangled representation is shown in Figure 2.12.

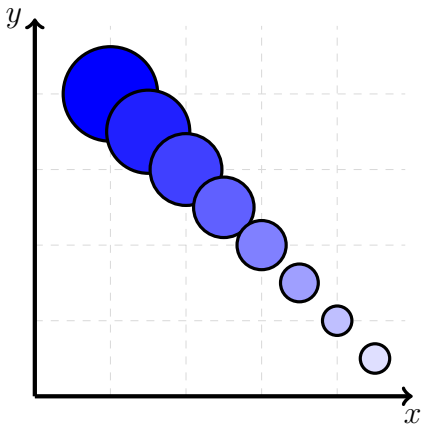


Figure 2.12: Disentangled representation

The schematic shows a representation of a blue circle with two factors of variance. We can clearly see that the Y-axis is responsible for the size of a circle: the larger the value of y, the larger the circle. On the other hand, the X-axis is responsible for the saturation: the larger the x value, the less intensive the blue color. As the features of a circle are disentangled in this representation, the features of a large blue circle $A = (x_1, y_1)$ and a small azure circle $B = (x_2, y_2)$ can be taken to create a small sky blue circle $C = (x_1, y_2)$. The process is shown in Figure 2.13.

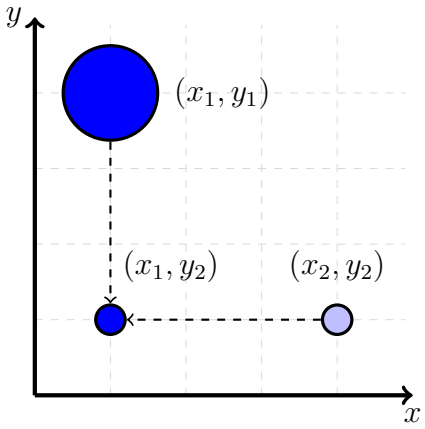
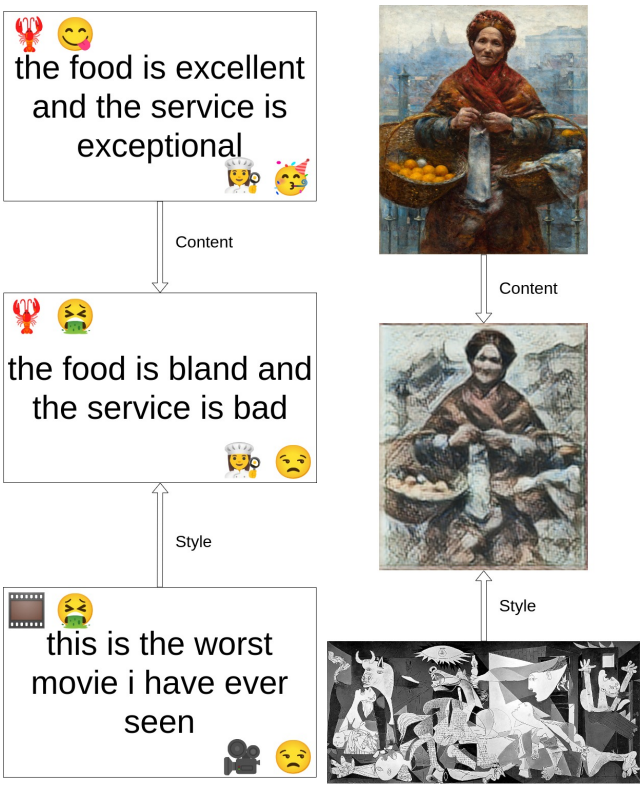


Figure 2.13: Small blue circle as a composition of features from big blue circle and small sky-blue circle

The same mechanism can be used in a technique called *style transfer* to change the style

of a content while preserving its content. This content can be text [44] or image [45], and examples of style transfer operations are shown in Figures 2.14a and 2.14b.



(a) Style transfer for text data

(b) Style transfer for graphic data

The idea of grafting features from one data point to another using disentaglement representation learning is an interesting field of studies and has not been studied so far using job applicant data. In this work, this technique is later called *competence transfer*.

Disentanglement itself, even if not previously discussed with regard to learning representation of job candidates, was very often mentioned in works concerning generative models such as generative adversarial networks (GANs) and variational autoencoders (VAEs).

InfoGAN

Generative adversarial networks are especially good at generating new unseen data. The problem with vanilla GAN [36] is the lack of interpretability of the certain dimensions in input noise vector z. To facilitate this issue, InfoGAN [46] maximizes mutual information (Equation 2.9) between a fixed subset of GAN's noise variables and observations.

β-VAE

β-VAE [47] is a simple yet powerful improvement to the Vanilla VAE (Section 2.4.2). β-VAE modifies the standard VAE loss function (Equation 2.21) by adding the coefficient β:

$$L(\theta, \phi, x) = L_{rec}(\theta, \phi, x) + \beta D_{KL}(q_\phi(z|x)||p(z)) \tag{2.24}$$

Increasing β encourages the training algorithm to focus more on identifying the factors of variation in the data [48]. The authors of this work also present an approach for quantifying disentanglement of the representation. To compute the proposed metric, inference is run on a number of samples generated by fixing the value of one generative factor of the data while randomly sampling all the others. The goal of a low-capacity linear classifier, applied on top of received latent vectors, is to determine which of the factors has been fixed in the data. The lower the accuracy of the classifier, the more entangled the representation is.

Info-VAE

In a method called Info-VAE [49], the authors propose new training objectives for the variational autoencoder. In addition to the goal of maximizing the log-likelihood of the data and minimizing the KL divergence between the true posterior and its approximation generated by the encoder, InfoVAE maximizes mutual information (Equation 2.9) between x and z in the distribution $q_\phi(x, z)$.

Non-Parallel Text Style Transfer

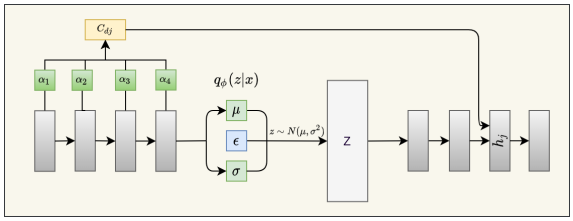
As shown in the figures 2.14a and 2.14b, style transfer can be done on both text and images. In work [50] the authors proposed a method able to disentangle the context and sentiment of the text. Thanks to that, it was possible to mix style and context of text pairs to receive new texts with desired properties. The size of the latent space dimensions responsible for the coding style and content was controlled by two classifiers, namely *multi-task clasfiers* responsible for ensuring the presence of content or style representation in given latent space dimensions. To prevent information leakage between style and content spaces, *adversarial classifiers* was added. We find the idea of using multitask and adversarial classifiers in tasks other than style transfer promising field of studies.

2.4.4 Attention mechanism

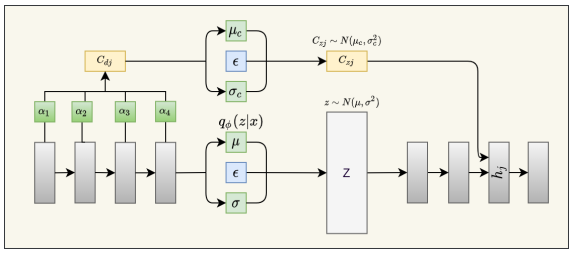
The attention mechanism was first proposed to improve the performance of encoder-decoder-based translation systems [51]. The term „attention" in its name refers to the role it plays during inference of models. In sequence-to-sequence models, it allows the decoder to pay attention to specific parts of the input sentence, rather than the whole fixed input.

The work [52] improved the attention presented by [51]. Using *general attention* instead of *local attention* enables the mechanism to focus on all hidden states of the encoder.

In work [53], the authors show that the use of deterministic attention, with direct access to the source, in VAE models can cause z to not contain much information. They call this effect *bypassing phenomenon*. To facilitate this issue, they propose a mechanism called *variational attention*. The deterministic context vector C_d instead of being fed directly to the decoder is used to model the mean μ_c and variance σ_c^2 of the Gaussian $\mathcal{N}(\mu_c, \sigma_c^2)$. The difference between deterministic and variational attention is shown in Figure 2.15



(a) Deterministic attention in a sequence-to-sequence VAE based model



(b) Variational attention in a sequence-to-sequence VAE based model

Figure 2.15: The difference between deterministic and variational attention

2.5 Job applicants representation learning

In this work, by the term job candidate, we understand a textual description of a job candidate consisting of distinguishable sections: (i) skills, (ii) education, (iii) language skills, (iv) work experience, references to social media, code repositories, and other external sources that help identify the skills and experience of a candidate, and (v) personal profile. This textual description can be provided in the form of *curriculum vitae* (CV) or be part of a user description on the freelancing platform such as *Fiverr*[2] or *Toptal*[3].

As said in Section 2.4, representation learning is a useful tool for a variety of tasks. One of them is learning to represent job applicants. According to the experienced recruiters interviewed during the process of creating this work, the most important factor determining whether the person is suitable for a job vacancy, job offers requiring senior-level proficiency, is the presence or absence of a certain set of skills. Therefore, it is not surprising that most of the works focus mainly on creating a representation of abilities. One of the ideas to do so is to use word embeddings (see Section 2.3). The idea of using word embeddings to represent the skills of job applicants was first proposed in work [54]. The authors used the evolution of the technique *word2vec* [23] to create a representation of skills to measure their similarity based on their co-occurrence in job offers. In their work, the researchers suggest that recruiters can use this technique to assess the relevance of differently named but semantically similar skills such as *html5*, *css*, and *bootstrap*, which are all technologies used in web design and should therefore be considered relevant. On the other hand, phrases such as *building recommendation engines* and *building car engines*, although they sound similar, refer to completely different abilities and should therefore be considered irrelevant.

In the work [55], the authors presented a recommendation system that helps to find suitable job offers for a person with a certain set of skills. To create such a recommendation system, researchers decided to use fine-tuned fastText word embeddings of online job vacancies (OJVs) paired with the relevance of certain skills in the context of a considered job position. This information is then stored in the graph database, which is then used to recommend a suitable job for a person represented by the word embeddings of their skills. Both described methods use word embeddings to represent skills extracted from job profiles in a multidimensional space. Although this general approach is straightforward and easy to

[2] https://www.fiverr.com
[3] https://www.toptal.com

implement, we think that creating a dedicated solution to representing job candidates can be beneficial to creating disentangled representation, performing better on downstream tasks such as matching job candidates with job offers.

The task of matching the job offer with the right job candidates, in the literature also called the Person–Job Fit task, is time-consuming for human experts. The reason behind this is that there are no generally accepted standards and rules that will guarantee satisfaction for both sides of the process. However, in the past, there have been some attempts to isolate factors that can be used to systematize the procedure. In the book [56] John Holland presented *personality–job fit theory*. The author presented six personality types: *realistic*, *investigative*, *artistic*, *social*, *enterprising*, *conventional*, and proposed that satisfaction and the tendency to quit a job depend on how well one's personalities match a certain job type. The key points of Holland's theory are that there are differences in personality of people applying for a job, there are different job types, and if we match a person with a job benefiting from its personality, it will increase satisfaction and decrease propensity to leave a position. Referring this theory to the aforementioned works, we can come to a conclusion that there are other factors different from applicant's skills, such as person's personality, that can be useful for the recruiters in order to assess an application.

In work [57] the authors created a system using 1D convolutional neural networks that processes job offers (represented by a set of requirements) and resumes (represented by a set of work experience entries) to create a latent space for both. Matching is then performed by finding the closest job offer to a resume by minimizing the cosine similarity metric. As a baseline, the researchers decided to use different classifiers used in machine learning, such as *logistic regression*, *decision trees*, *naive bayes*, *support vector machines*, *AdaBoost*, *random forests*, *gradient-boosting decision trees*, *linear discriminant analysis*, *and quadratic discriminant analysis*. Compared to baseline methods, the use of convolutional networks to create a joint latent space greatly improved the results achieved. It should be noted that other sections other than *experience*, such as *personal profile* and *education* were completely ignored during the training and evaluation process. The work [58] presents a different approach to the problem of matching job offers with applicants called the Ability–Aware Person–Job Fit Neural Network (APJFNN). Instead of convolutional neural networks, bidirectional LSTM (*BiLSTM*) networks were used. Furthermore, instead of focusing on the entire job experience description, the researchers decided to create a system capable of detecting and weighing skills and requirements in resumes and job offers, respectively. To do so, the model was constructed using three modules: word representation level, hierarchical ability-aware representation made of four attention mechanisms [26] and finally the classification part consisting of densely connected layers. As a baseline, a subset of baseline methods from the aforementioned work [57] was used and consisted of *logistic regression*, *decision trees*, *AdaBoost*, *Random Forests* and *gradient boosting decision trees*. To verify whether the proposed attention mechanisms actually help the model make predictions, the researchers evaluated the version that omits them. The experiments showed that APJFNN outperforms all other methods tested.

There were also attempts to use autoencoders and variational autoencoders to represent job applications. In work [59] the authors used data on activity on various job search websites to create a representation using classic autoencoders (AE), denoising autoencoders (DAE) [60], and variational autoencoders (VAE). The results suggest that VAE outperforms other approaches in the task examined.

In summary, there are multiple approaches to learning the representation of job candidates. Some of them use methods used in various NLP tasks, for example, learning word embeddings [54, 55]. There are also works that use more domain-specific approaches [57, 58, 60, 59]. However, the idea of using variational autoencoders to create disentangled representations of job candidates based on resumes has not yet been examined.

2.6 Summary

The rapid growth of computer technology in recent years has allowed researchers to develop advanced methods for processing various types of data. One of these methods is the architecture called *variational autoencoder*, which can produce a disentangled representation of the data. There are no prior works that describe the application of variational autoencoders to the task of creating disentangled representations of job candidates' profiles. Furthermore, in the literature, no attempts have been made to solve the task of *competence transfer*.

3 Proposed solution

We propose a novel approach to the task of representing job applicants, namely *Disentangled Representation of Applicants with VAE* (DRAV)[1].

3.1 Motivation

Variational autoencoders [35] can be used successfully in the training of sequence-to-sequence models (seq2seq) [61]. Furthermore, due to their probabilistic properties, they are especially good at creating disentangled representations [48]. The downside of representations created by VAE is that it is not explicitly defined which factors of variance are coded in which dimensions of the created latent space. The ability to control the disentanglement for selected generative factors can be for example used to generate new data points with certain defined properties, while randomizing others. In terms of representation of job applicants, disentanglement with controlled position and capacity of representation with regard to the generative factors can be used:

- for applying a technique named *competence transfer* – transferring competences between different persons job candidates,
- to generate synthetic descriptions of job candidates,
- to improve the accuracy of algorithms designed to parse skills, education, and language skills from the textual description.

3.2 Factors of variance in job candidates representation learning

Most of the works that discuss the topic of creating disentangled representations use images as input. As shown in Figure 2.13, identifying generative factors in images, especially when using synthetic data sets, is simple and intuitive. In the case of NLP tasks, they are more difficult to determine and fuzzy. One of the most commonly used factors of variation in texts are: sentiment [16, 44], emotions [62], and the presence of hate speech [63].

In the task of learning the representation of job candidates, the mentioned factors cannot be applied, as the analyzed texts tend to have positive or neutral sentiments and do not tend to contain extreme emotions or hate speech. We state that in this case distinguishable sections in a textual description of a job candidate, described in Section 2.5, can be used

[1]source code available here: https://github.com/pszelew/DRAV

as generative factors. In this work, we focus on four of them: education, language skills, general skills, and personal profile (the personal profile in this work is also called *content* or *description*).

3.3 DRAV architecture overview

DRAV is a sequence-to-sequence model equipped with a variational autoencoder. As of one the main goals of the DRAV network is creating disentangled representation of the data. We propose two methods to ensure the disentanglement of the created representation.

The architecture of the first DRAV network architecture is shown in Figure 3.1, it is later called *Adversarial Disentangled Representation of Applicants with VAE* (ADRAV). The other approach is Multi-LSTM Disentangled Representation of Applicants with VAE (MLDRAV), which architecture is shown in Figure 3.2.

The main goal during the training phase of both DRAV network architectures is to recreate the textual input, which is the description of the job candidate. To guide the ADRAV network to create disentangled generative factors in specified latent dimensions, multitask and adversarial classifiers are introduced. MLDRAV uses multiple LSTM networks to code the necessary information in the desired subspaces of the latent space. The ADRAV and MLDRAV architectures are described in detail in the following sections.

3.3.1 Word embeddings

The input of the DRAV network is a textual description of the candidate. To process text data using recurrent neural networks, consecutive words in an input sequence must be represented by vectors. We propose three methods to vectorize textual descriptions.

The first of the proposed methods is the use of an embedding layer. It can be understood as a linear layer that transforms a one-hot encoded word into a fixed-length continuous vector with the dimension of the LSTM network used in the encoder module. The weights of the embedding layer are trained together with the other parameters of the DRAV network.

The different approach uses pre-trained fastText [24] word vectors. fastText model can be pre-trained on textual data used later in the training process of DRAV or on the large data corpus, for example, texts from Wikipedia[2] or Common Crawl[3]. The advantage of fastText method over embedding layer is that fastText model can be used to produce embeddings of unseen words thanks to the analysis of character n-grams and representing the words as the sum of these n-grams.

The last of the proposed approaches to produce word embeddings is the use of transformer-based methods. Transformer-based models such as RoBERTa, Roberta, or DistilBERT compute embeddings using local context, in contrast to embedding layer and fastText models, which are lookup tables. This approach helps transformer-based methods produce better quality results at the expense of performance. To produce accurate word embeddings, we can simply use models already pre-trained on a large dataset, as well as finetune them using

[2] https://www.wikipedia.org
[3] https://commoncrawl.org

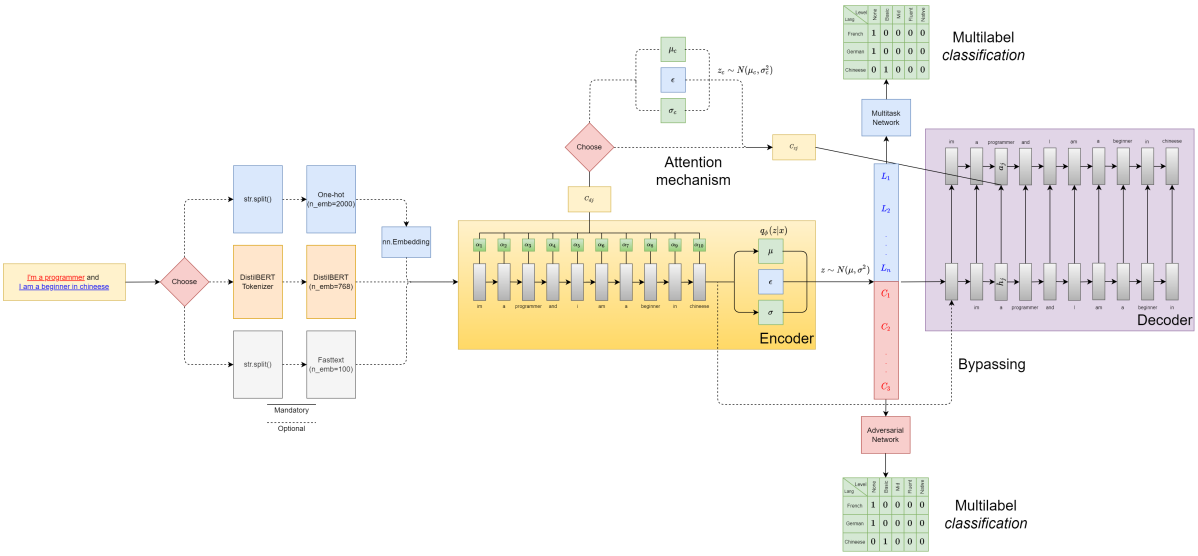


Figure 3.1: ADRAV architecture schema. To simplify the schema, it shows only two factors of variance out of four discussed in this work: language skills and content.

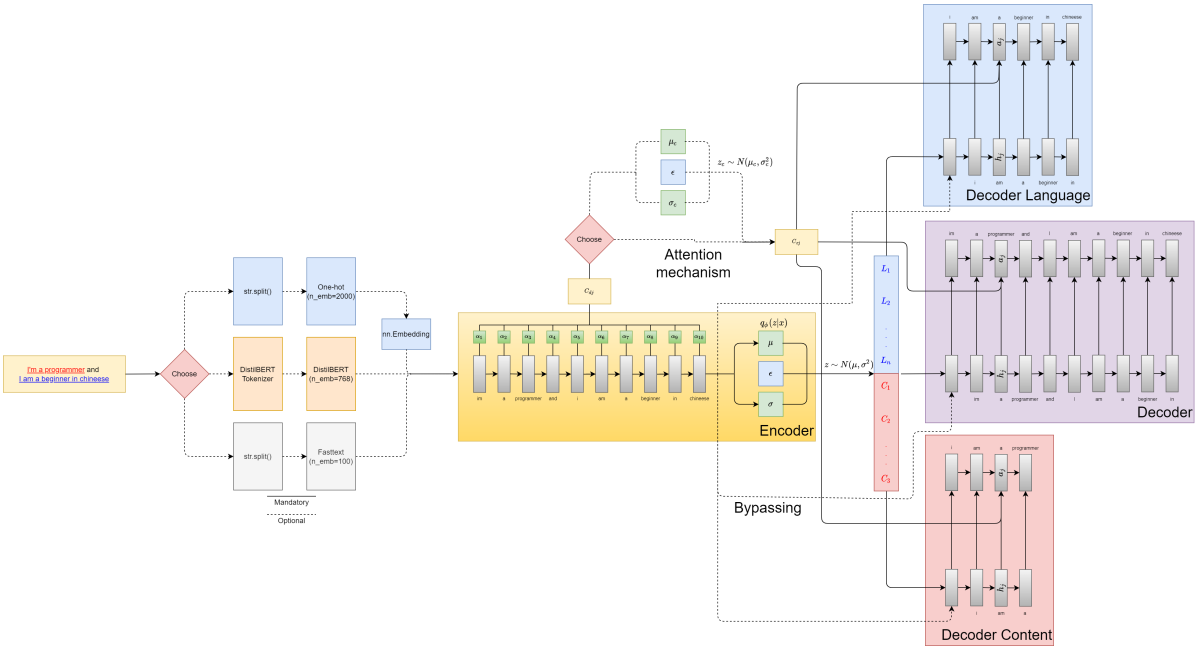


Figure 3.2: MLDRAV architecture schema. To simplify the schema, it shows only two factors of variance out of four discussed in this work: language skills and content.

custom job candidate dataset. Due to the sizes of large models such as BERT (*bert-base* has $110M$ parameters) and Roberta (*roberta-base* has $123M$ parameters), in this work, significantly smaller DistilBERT is much faster, but still accurate option to consider (*distilbert-base-uncased* has $66M$ parameters).

3.3.2 Encoder network

The core of the encoding part of the DRAV network is a bidirectional multilayer LSTM recurrent neural network. The LSTM network takes as input a sequence of word embeddings. The LSTM is followed by a series of linear layers, called hidden layers, with a dropout and the ReLU activation function. Subsequently, fully connected layers – l_μ and l_σ – are applied to generate the multivariate Gaussian distribution parameters. The latent vector z is sampled from the generated distribution using *reparametrization tricked* described in Section 2.4.2.

3.3.3 Variational attention

In Section 2.4.4 deterministic attention mechanism (DAttn) and its modification: variational attention (VAttn) have been described. We use the global attention proposed in [52] with the *dot* score function, producing the deterministic context vector c_d. To prevent deterministic leakage of information called *bypassing phenomenon* [53], we apply the idea of sampling the context vector c_z from a Gaussian parameterized by μ_a and σ_a, namely variational attention. For a mean μ_a, we apply an identity transformation, so $\mu_a \equiv c_d$. The logarithm of variance $\log \sigma_a$ is calculated using the network made up of a neural layer with the activation function *tanh* (Equation 2.4) followed by a linear layer.

To train variational attention, an additional loss function is introduced. As the prior is assumed to be $N(0, 1)$, the loss function for the variational attention mechanism can be calculated by adjusting Equation 2.22:

$$\begin{aligned} L_{\text{vattn}}(\phi) &= \sum_{j=1}^{S} D_{KL}(q_\phi^{(a)}(a_j|x)||p(a_j)) \\ &= \sum_{j=1}^{S} \left[\frac{1}{2} \sum_{i=1}^{k} (\sigma_{a,i,k}^2 + \mu_{a,i,k}^2 - \log \sigma_{a,i,k}^2 - 1) \right], \end{aligned} \tag{3.1}$$

where:

- S is the length of a sequence generated by the decoder,
- $q_\phi^{(a)}(a_j|x)$ is the posterior attention distribution for the timestamp t_j.

3.3.4 Decoder network

The goal of the decoder network is to recreate the input sequence using the information encoded in the latent vector z. The main component of the decoder network is the unidirectional multilayer LSTM network. The output of the LSTM network is passed to the attention layer described in Section 3.3.3. The context vector c_z is concatenated with the raw LSTM output.

Afterwards, the tanh function is applied to the concatenation, and its result is passed through the series of fully connected layers with ReLU activation function. The output of the last layer is passed through the softmax function. The output vector has the length equal to the size of the language dictionary. The process of creating the language dictionary is described in Section 4.1.4. The DRAV reconstruction loss $L_{\text{rec}}(\theta, \phi, x)$, is calculated by averaging the negative log-likelihood loss of all subsequent word predictions.

As shown in Figure 2.8a, the LSTM cell takes three inputs: input vector x_t, hidden state state h_{t-1} and cell state c_{t-1} from the previous timestamp. In a starting timestamp t_0, the initial values of all values must be set.

Input vector initialization

We initialize the input vector x_t with the embedding of the "start of the sequence" token `<SOS>`, concatenated with the latent vector z.

Hidden state initialization

There are works using models similar to the DRAV model, where the hidden state of the LSTM decoder networks is initialized with the last hidden state of the LSTM encoder network [64]. In [53] the authors suggest that this technique, namely bypassing, can prevent the latent variable z from capturing useful information, as it will be encoded in a bypassed hidden state. The proposed solution to this problem is to initialize the hidden state h_0 and the cell state c_0 as zero vector.

Passing the latent information to the decoder network

After passing the sequence of the word embeddings through the encoder network thanks to the reparametrization trick, we generate the latent vector z. We have experimented with multiple ways of passing the z vector to the decoder network:

- passing z through the sequence of fully connected layers. In the result, we receive the vector of word embeddings size, which can be used as the first input x_0 of the LSTM network,
- concatenating the z with the output of the current step in the decoding process; z concatenated with x_t is passed to the output layer,
- concatenating the z with the output of the previous step in the decoding process; z concatenated with x_{t-1} is passed as an input of the LSTM cell; the method was proposed in [65].

We have decided to use the last option, which is concatenating the z with the output of the previous decoding step. It allows the decoding LSTM to use full latent space information in the decoding process

3.3.5 Disentanglement control mechanisms

One of the contributions of this thesis is a mechanism, namely *comptence disentanglement control*, which guides the latent space to separate the generative factors into the indicated subspaces. The method used in this mechanism is the main difference between the ADRAV and MLDRAV architectures.

The competence disentanglement control mechanism applied in the ADRAV architecture in Figure 3.1 can be divided into two subtasks:

- ensuring that the information referring to a given factor of variance is coded in the desired subspace using multitask networks,
- ensuring that the information referring to a given factor of variance does not leak to other subspaces of latent space using the adversarial networks.

The mechanism applied in the MLDRAV architecture only ensures that the information referring to a given factor of variance is coded in the desired subspace.

Multitask networks

Ensuring that the information about a given generative factor is coded in the desired subspace in the ADRAV architecture is done by multitask networks, namely *skills multitask network* (parameterized by γ_s), *education multitask network* (parameterized by γ_e), *languages multitask network* (parameterized by γ_l). Their goal is to predict a probability distribution encoded in the train targets. It should be noted that we do not apply the multitask and adversarial networks for the content, on purpose, as it is significantly longer, and less informative hence harder to predict accurately. All multitask networks consist of a linear layer, and their output can be defined as follows:

$$y_{(\text{mul},N)} = W_{\gamma_N} \cdot z_N + b_{\gamma_N}, \tag{3.2}$$

where:

- N is the selector of the generative factor (s, e or l),
- z_N is the fragment of the latent space, designed to encode the generative factor N,
- W_{γ_N} is the weight matrix,
- b_{γ_N} is the bias vector.

Networks are trained along the encoder network using the cross-entropy loss function, which can be defined using the equation:

$$L_{(\text{mul},N)}(\phi, \gamma_N) = -\sum_{c=1}^{C_N} \left[\log \frac{\exp y_{(\text{mul},N,c)}}{\sum_{i=1}^{C_N} \exp y_{(\text{mul},N,i)}} \cdot y_{(\text{target},N,c)} \right], \tag{3.3}$$

where:

- N is the selector of the generative factor (s, e or l),

- ϕ are the parameters of the encoder network,
- $y_{(\text{mul})}$ is the output of the multitask network,
- $y_{(\text{target})}$ is the target distribution,
- C_N is the dimension of the desired probability distribution.

Adversarial networks

The objective of the second subtask of the competence disentanglement control mechanism in ADRAV was to prevent the leakage of information from the indicated for a factor of variance outside the given subspace. We handle this issue by introducing adversarial networks, namely *skills adversarial network* (parameterized by δ_s), *education adversarial network* (parameterized by δ_e) and *languages adversarial network* (parameterized by δ_l). The output of the adversarial network can be defined using the following equation.

$$y_{(\text{adv},N)} = W_{\delta_N} \cdot z_N^c + b_{\delta_N}, \tag{3.4}$$

where:

- N is the selector of the generative factor (s, e or l),
- z_N^c is the latent space without the dimensions desired to encode the generative factor N,
- W_{δ_N} is the weight matrix,
- b_{δ_N} is the bias vector.

Adversarial networks are trained using a loss function similar to the one used in the training of multitask classifiers:

$$L_{(\text{dis},N)}(\delta_N) = -\sum_{c=1}^{C_N} \left[\log \frac{\exp y_{(\text{adv},N,c)}}{\sum_{i=1}^{C_N} \exp y_{(\text{adv},N,i)}} \cdot y_{(\text{target},N,c)} \right], \tag{3.5}$$

where:

- N is the selector of the generative factor (s, e or l),
- $y_{(adv)}$ is the output of the multitask network,
- $y_{(target)}$ is the target distribution,
- C_N is the dimension of the desired probability distribution.

It should be noted that the loss $L_{(\text{dis},N)}(\delta_N)$ is not propagated back to the encoder network during the training phase. We want the encoder network weights to maximize the adversary's entropy, which can be formulated using the following equation:

$$L_{(\text{adv},N)}(\phi) = -\sum_{c=1}^{C_N} \left[\frac{\exp y_{(adv,N,c)}}{\sum_{i=1}^{C_N} \exp y_{(adv,N,i)}} \cdot \log \frac{\exp y_{(adv,N,c)}}{\sum_{i=1}^{C_N} \exp y_{(adv,N,i)}} \right], \tag{3.6}$$

where:

- N is the selector of the generative factor (s, e or l),
- $y_{(\text{adv})}$ is the output of the multitask network,
- C_N is the dimension of the desired probability distribution.

LSTM multitask networks

The LSTM multitask networks used in the MLDRAV architecture are almost identical to the main decoder network described in Section 3.3.4, and we can think of them as four additional decoder networks. The difference between them and the main decoder network is that their goal is not to recreate the entire input text, but only the following parts of it: skills, education, language skills, and description. Similarly to simpler multitask networks, LSTM multitask networks have access to only a certain part of the latent space.

3.3.6 Competence transfer

One of the goals of the DRAV network is to separate the generative factors in the latent space using the techniques described in Section 3.3.5. If the generative factors are well separated in the latent space, we can use it to generate new text samples with the desired properties. If we take the latent subspace responsible for coding one generative factor, for example, the skills section extracted from one seller profile, namely the source, and graft it into the latent space received from the second profile, namely the target, we expect the decoder to be able to produce a new description consisting of the skills section of the source profile and education, language, and content of the target profile. We call this procedure *competence transfer*, and its schematic is shown in Figure 3.3.

3.3.7 Composed loss function

After describing all components of the DRAV model, the final objective functions can be defined.

ADRAV

ADRAV tries to minimize the loss function that can be defined using the following equation:

$$\begin{aligned} L_{\text{ADRAV}} = & L_{\text{rec}}(\theta, \phi, x) \\ & + \beta \cdot [\lambda_{(\text{kl},s)} \cdot D_{KL}(q_\phi(z_s|x)||p(z_s) + \lambda_{(\text{kl},e)} \cdot D_{KL}(q_\phi(z_e|x)||p(z_e) \\ & + \lambda_{(\text{kl},l)} \cdot D_{KL}(q_\phi(z_l|x)||p(z_l) + \lambda_{(\text{kl},c)} \cdot D_{KL}(q_\phi(z_c|x)||p(z_c) + \psi \cdot L_{\text{vattn}}(\phi)] \\ & + \lambda_{(\text{mul},s)} \cdot L_{(\text{mul},s)}(\phi, \gamma_s) + \lambda_{(\text{mul},e)} \cdot L_{(\text{mul},e)}(\phi, \gamma_e) + \lambda_{(\text{mul},l)} \cdot L_{(\text{mul},l)}(\phi, \gamma_l) \\ & - \lambda_{(\text{adv},s)} \cdot L_{(\text{adv},s)}(\phi) - \lambda_{(\text{adv},e)} \cdot L_{(\text{adv},e)}(\phi) - \lambda_{(\text{adv},l)} \cdot L_{(\text{adv},l)}(\phi), \end{aligned} \tag{3.7}$$

where:

- $L_{\text{rec}}(\theta, \phi, x)$ is the reconstruction loss of the decoder,

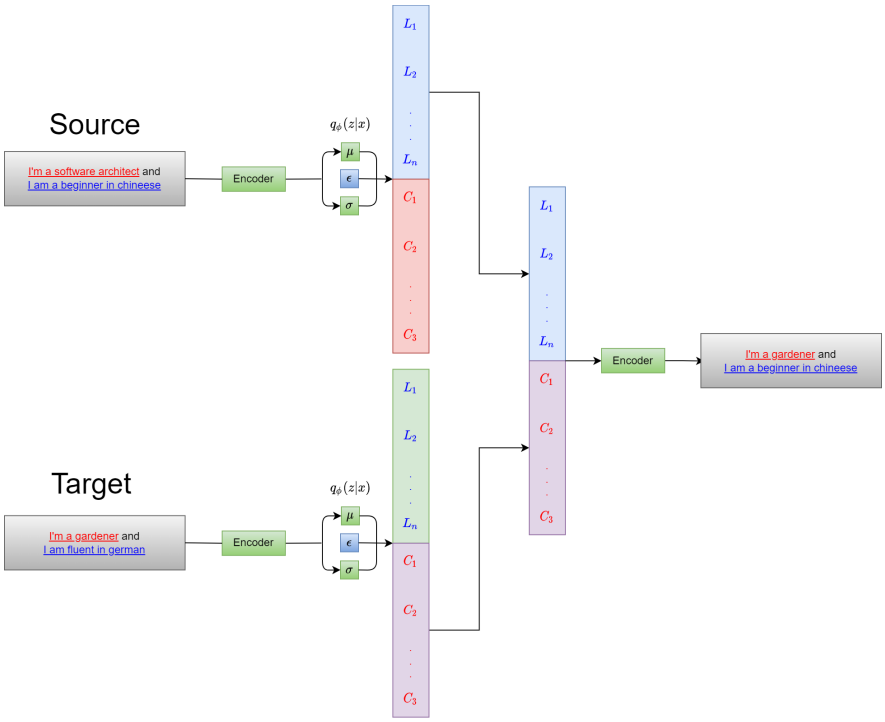


Figure 3.3: Competence transfer operation

- β is the parameter proposed in β-VAE method, described in detail in Section 2.4.3,
- $\lambda_{(\text{kl},s)}$, $\lambda_{(\text{kl},e)}$, $\lambda_{(\text{kl},l)}$ and $\lambda_{(\text{kl},c)}$ are the weights for the D_{KL} losses of skills, education, language proficiency and content, respectively,
- ψ is the weight that adjusts the influence of variational attention loss,
- $\lambda_{(\text{mul},s)}$, $\lambda_{(\text{mul},e)}$ and $\lambda_{(\text{mul},l)}$ are the weights for the losses of multitask skills, education and language proficiency networks, respectively,
- $\lambda_{(\text{adv},s)}$, $\lambda_{(\text{adv},e)}$ and $\lambda_{(\text{adv},l)}$ are the weights for adversarial losses of skills, education and language proficiency networks, respectively.

It should be emphasized that, as we want to maximize entropy $L_{(\text{adv},N)}(\phi)$, so instead of adding the entropy to the final loss, we add its negation.

MLRAV

DRAV network in MLRAV architecture variant tries to minimize the loss function that can be defined using the following equation:

$$\begin{aligned} L_{\text{MLDRAV}} = {} & L_{\text{rec}}(\theta,\phi,x) + L_{(\text{rec},s)}(\theta,\phi,x) + L_{(\text{rec},e)}(\theta,\phi,x) + L_{(\text{rec},l)}(\theta,\phi,x) + L_{(\text{rec},c)}(\theta,\phi,x) \\ & + \beta \cdot [\lambda_{(\text{kl},s)} \cdot D_{KL}(q_\phi(z_s|x)||p(z_s) + \lambda_{(\text{kl},e)} \cdot D_{KL}(q_\phi(z_e|x)||p(z_e) \\ & + \lambda_{(\text{kl},l)} \cdot D_{KL}(q_\phi(z_l|x)||p(z_l) + \lambda_{(\text{kl},c)} \cdot D_{KL}(q_\phi(z_c|x)||p(z_c) + \psi \cdot L_{\text{vattn}}(\phi)], \end{aligned} \tag{3.8}$$

where:

- $L_{\text{rec}}(\theta, \phi, x)$ is the reconstruction loss of the decoder,
- β is the parameter proposed in β-VAE method, described in detail in Section 2.4.3,
- $L_{(\text{rec},s)}$, $L_{(\text{rec},e)}$, $L_{(\text{rec},l)}$, $L_{(\text{rec},c)}$ are the reconstruction losses of the LSTM multitask networks,
- $\lambda_{(\text{kl},s)}$, $\lambda_{(\text{kl},e)}$, $\lambda_{(\text{kl},l)}$ and $\lambda_{(\text{kl},c)}$ are the weights for the D_{KL} losses of skills, education, language proficiency and content, respectively,
- ψ is the weight that adjusts the influence of variational attention loss,

4 Experiments

In this chapter, the process of assessing the effectiveness of the proposed DRAV network is discussed.

4.1 Data

The dataset consists of 56,083 seller profiles on the Fiverr[1] freelance services marketplace. Each seller is asked to provide a brief description, language proficiency, a set of skills, and a history of education. An example of the profile can be shown in Figure 4.1. Profiles were collected from the three categories: *graphics & design*, *digital marketing*, and *writing & translation*.

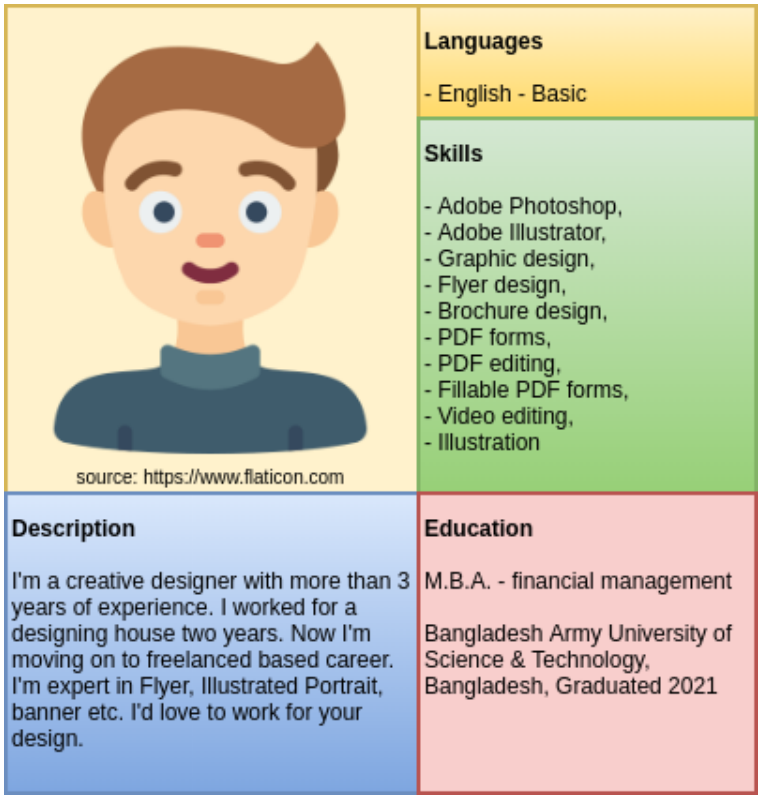


Figure 4.1: An example of a description of a job candidate in Fiverr

[1] https://www.fiverr.com

4.1.1 Copyrights

The data from Fiverr was gathered using web scraping techniques. Profiles were available without the need to log into the Fiverr website. They do not contain sensitive personal data. The dataset was collected only for the research process presented in this thesis and will not be used for commercial purposes.

4.1.2 Exploratory data analysis

The collected texts are multilingual. As this work does not consider the topic of creating multilingual data representations, we limit the dataset to profiles written in English (98.6% of the dataset). The detailed distribution of languages in the dataset is presented in Appendix A.1.

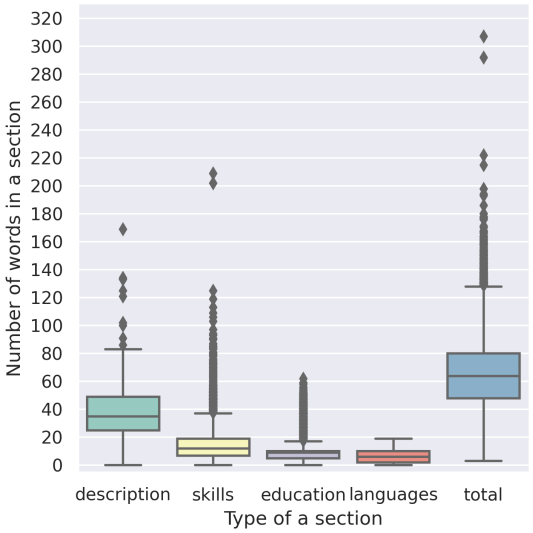
The DRAV network was created to process the sequences of words. To know how long the input sentences are in the dataset, we have created a visualization shown in Figure 4.2a. It is not a surprise that the longest section is the description. The median of the total number of words is 60 words, but the longest description in the dataset consists of more than 300 words. We have decided to keep long descriptions as well. First, because the model architecture does not limit the maximum input sequence length. The second reason is that we want to follow the trends set by popular language models such as *BERT* or *DistilBERT*, which were designed to process sequences up to 512 tokens.

To better understand the dataset, we have prepared a word cloud shown in Figure 4.2b, using the vocabulary of the Fiverr dataset. Visualization helps us to determine that the most frequent phrases are language competencies (*English*, *Espanol*), and skills related to marketing (*social medium*, *digital marketing*) and graphic design (*graphic design*, *Adobe Photoshop*).

In Figures 4.2c, 4.2d the most common skills and languages known to sellers are presented. It is no wonder that the most common skills are typical for graphic designers (Adobe Photoshop), digital marketers (social media marketing) and writers (content writing). In total, the sellers reported having knowledge of 134 different languages. For each language, one of the five levels of proficiency could be chosen: unknown, beginner, conversational, fluent, and native. It is no surprise that the language most frequently known by Fiverr users is English, but there are only around 8,500 users who state that they are native English speakers. More surprising is the large participation of native Urdu speakers (Urdu is the official language of Pakistan).

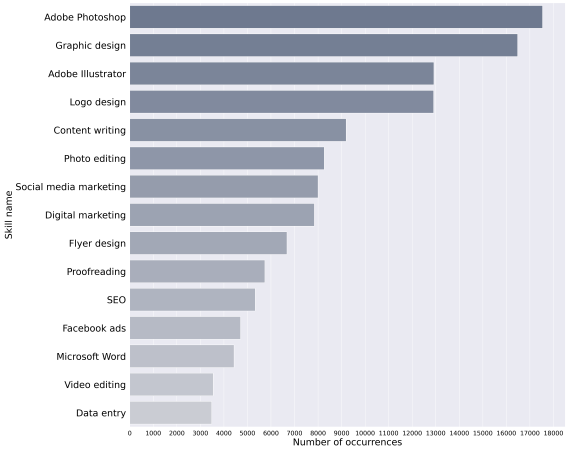
4.1.3 Data augmentation

The input of the DRAV network is a textual description of an applicant for the job. One of the goals of the DRAV network is to distinguish four generative factors hidden in this textual form. The data on the Fiverr website are consistent and provide the segments in fixed order: [description, languages, skills, and education]. As a form of a data augmentation, we add random beginnings and endings to the sections of skills, education, and languages.

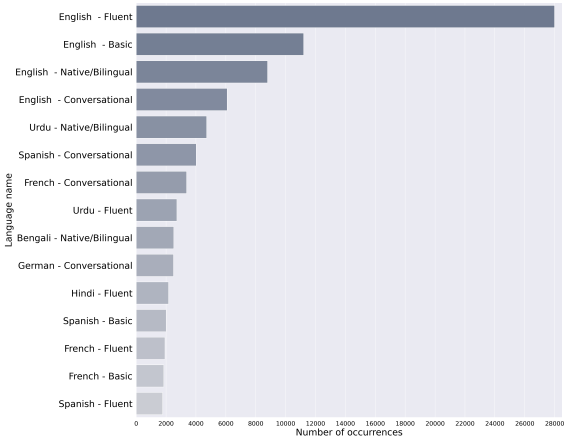


(a) Number of words distribution in each section of the description

(b) The dataset as a word cloud



(c) Most frequent skills in the dataset



(d) Most frequent language skills in the dataset

Figure 4.2: Dataset Visualizations

4.1.4 Training and testing data preparation

As mentioned in Section 4.1.2, 810 non-English data points were removed from the original dataset. Furthermore, we have decided to exclude profiles with missing information on any of the generative factors. As a result of filtering, of the initial 56,083 seller profiles, 41,011 documents were used during the research process. After the data were filtered, data cleaning was applied. We have removed all the Non-ASCII characters and punctuation. Also, we have converted all of the text to lowercase.

The last layer of the DRAV network tries to predict the probability distribution of words from a fixed dictionary. The dictionary was created using lowercase words from the training data. To filter out rare words, and the artefacts caused by misspelling, the words with number of occurrences lower than three were removed from the dictionary. As a result, the language dictionary used in the training consists of 20,522 words. Of the 41,011 profiles left after the filtering process, 1,000 of them were randomly sampled as testing data, leaving 40,011 examples as the training dataset.

In the ADRAV architecture and the disentanglement metrics proposed in Section 4.2.2, we use as targets probability distributions of generative factors. To receive such probability distributions, skills, education, and language skills must be vectorized.

Vectorization of skills and education

The sections on skills and education are diverse in terms of vocabulary. Therefore, it is difficult to create a fixed set of skills and degrees that could be used to encode these factors of variation. To facilitate the problem, we propose a similar approach to that used in [65]. Instead of directly encoding skills or degrees, we encode them using the bag-of-words (BoW) [66] technique. The texts of the skills and education sections are preprocessed by (i) removing the stop words and (ii) excluding the sentiment words proposed in [67]. Obtained probability distributions of the sections, can be expressed formally by the following equation:

$$\tilde{v}_N = \frac{[v_{(N,1)}, v_{(N,2)}, ..., v_{(N,M)}]}{\sum_{i=1}^{M} v_{(N,i)}}. \tag{4.1}$$

where:

- N is the selector of the generative factor (s, e),
- M is the size of BoW,
- $v(N, i)$ is the number of occurrences of a word with an index i in a section.

Language skills section vectorization

Language competencies section is treated differently from skills and education. Although according to [68] there are 7,151 languages in the world, only 23 of them represent more than half of the world population. We can think of the languages section in the sellers' profile as a finite and relatively small subset of names, with one of five levels of proficiency attached to each language name. The matrix $\tilde{V}_l$ representing the language skills section can be formally stated:

$$\tilde{V}_l = \frac{[e^T_{(i_1,1)}, e^T_{(i_2,2)}, \ldots, e^T_{(i_M,M)}]}{\sum_{l=1}^{5} \sum_{k=1}^{M} v_{(l,k)}}, \tag{4.2}$$

where:

- $e_{(i_m,m)}$ is one-hot encoded proficiency in a language m,
- M is the total number of languages in the dataset.

4.2 Experiment plan

During experiments on the DRAV network, we focused on verifying the influence of the proposed methods of controlling the disentanglement introduced in Section 3.3.5 on the ability to create useful data representation. We check the performance of DRAV in the tasks of:

- reconstruction of the input text,
- controlling the subspaces meant to encode generative factors,
- creating disentangled data representation,
- generating new texts with desired properties.

The following sections describe the proposed metrics and other evaluation details.

4.2.1 Ablation study

Main focus during the research on DRAV was to test whether the proposed mechanisms, such as variational attention and methods of controlling disentanglement, help to achieve the defined objectives. We find that evaluating the influence of the attention mechanism on the performance of the DRAV network is particularly interesting. On the one hand, attention can help the DRAV model better reconstruct the input, but on the other hand, attention is using information other than directly encoded in the latent vector z, which may cause the z vector to be insufficiently informative. Therefore, the ablation study includes the effects of disabled variational inference in the attention module and fully disabled the attention module.

We have also evaluated whether the presence of disentanglement control mechanisms helps DRAV model to create disentangled representations.

4.2.2 Metrics

Competence disentanglement control metric

One of the challenges discussed in this thesis is to create a method to control the dimensions of generative factors in data representation, using a mechanism called *competence disentanglement control*. We propose a method to measure how well the generative factors described

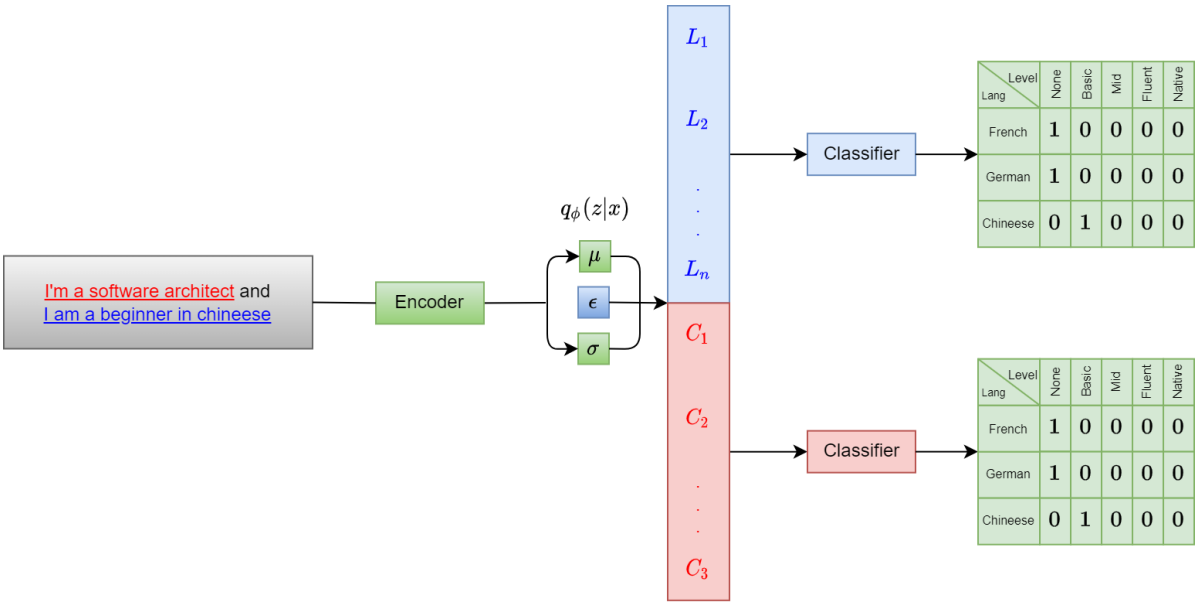


Figure 4.3: The method of measuring the competence disentanglement control in the job candidates' representation

in Section 3.2 (skills, education, language proficiency, and content) are separated in the latent space. The visualization of the method is shown in Figure 4.3.

To evaluate the DRAV network in the competence disentanglement control task, we take the trained encoder of the DRAV network. The encoder is used to produce latent vectors z from the textual descriptions of the job candidates stored in the test datasets. For each of the three generative factors: *skills*, *education*, and *language proficiency*, we create four linear layers. A multitask layer and three adversarial layers are trained to predict the distribution of *skills*, *education*, and *language skills* vectors receptively using certain parts of the latent vector z. The process of creating vectorizing *skills*, *education*, and *language skills* is described in Section 4.1.

The difference between multitask and adversarial layers is that the multitask layer uses the dimensions of the latent space corresponding to the predicted generative factor. The adversarial layer has access only to the dimensions that were intended to encode another factor of variance. For each of the three factors of variance, we report four values: L_{mul} and L_{adv_1}, L_{adv_2}, L_{adv_3}, The first value – L_{mul} – is the cross-entropy between the target distribution and the output of the multitask layer after applying the softmax function on top of it. L_{mul} determines whether we were able to encode information about the chosen factor of variance in the designated dimensions of the latent space. The values L_{adv_1}, L_{adv_2}, L_{adv_3} are the cross-entropy between the target distributions and the output of adversarial layers with the applied softmax function. The adversarial scores determine how much knowledge about the generative factor leaked into dimensions designed to be occupied by other factors; the higher the reported cross-entropy value, the better. Generally, if the score for the multitask layer is lower than that for the adversarial layer, then we find that the *competence disentanglement control* task is successful. The networks are trained on the test dataset, using the Adam optimizer with the learning rate 0.01 until convergence. We report a mean score and the

standard deviation of 10 iterations.

General disentanglement metric

In Section 2.4.3, a method to quantify the disentanglement of the representation proposed in [47] was described. We have adopted the method for the dataset of the seller profiles. The schematic of the proposed disentanglement assessment process is shown in Figure 4.4.

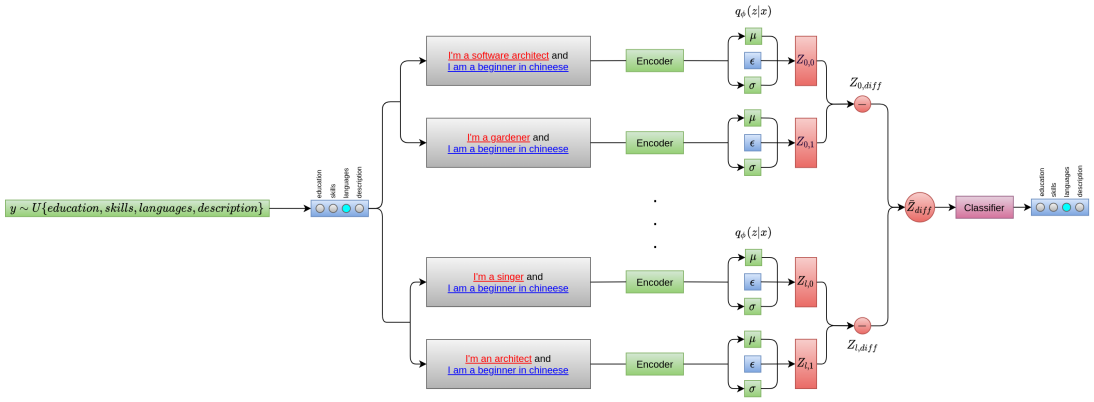


Figure 4.4: The method of measuring the disentanglement of the job candidates' representation

The algorithm for calculating the proposed metric is as follows.

1. Create 16 pairs of data points:

$$b = [(x_{(1,1)}, x_{(2,1)}), (x_{(1,2)}, x_{(2,2)}), \ldots, (x_{(1,16)}, x_{2,16})],$$

   namely a batch. In each batch, we choose one generative factor Gf. In each pair, the value of the generative factor Gf is fixed, while the other factors are selected randomly.

2. Process each data point using the encoder network, receiving pairs of latent vectors

$$b_z = [(z_{1,1}, z_{2,1}), (z_{1,2}, z_{2,2}), \ldots, (z_{1,16}, z_{2,16})],$$

3. Calculate the absolute difference of latent vectors:

$$b_{(z,diff)} = [(|z_{1,1} - z_{2,1}|), (|z_{1,2} - z_{2,2}|), \ldots, (|z_{1,16} - z_{2,16}|)].$$

4. Calculate the mean of the absolute differences received in the previous step $b_{z,mean}$.

5. The linear classifier is trained to predict the fixed generative factor using $b_{z,mean}$.

6. Repeat steps until convergence.

Classifiers are trained on the test dataset, using the Adam optimizer with the learning rate 0.001 until convergence. We report the mean accuracy along the standard deviation of 3 iterations.

4.3 Experiments setup

We have evaluated 12 different versions of the DRAV network:

- ADRAV with variational attention mechanism (VAttn), multitask loss weight $\lambda_{\text{mul}} = 0$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 0$,
- ADRAV with variational attention mechanism (VAttn), multitask loss weight $\lambda_{\text{mul}} = 2$, adversarial entropy loss weight $\lambda_{\text{adv}} = 1$,
- ADRAV with variational attention mechanism, multitask loss weight $\lambda_{\text{mul}} = 200$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 100$,
- ADRAV with deterministic attention mechanism (DAttn), multitask loss weight $\lambda_{\text{mul}} = 0$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 0$,
- ADRAV with deterministic attention mechanism (DAttn), multitask loss weight $\lambda_{\text{mul}} = 2$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 1$,
- ADRAV with deterministic attention mechanism, multitask loss weight $\lambda_{\text{mul}} = 200$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 100$,
- ADRAV without attention mechanism, with multitask loss weight $\lambda_{\text{mul}} = 0$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 0$,
- ADRAV without attention mechanism, with multitask loss weight $\lambda_{\text{mul}} = 2$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 1$,
- ADRAV without attention mechanism, with multitask loss weight $\lambda_{\text{mul}} = 200$ and adversarial entropy loss weight $\lambda_{\text{adv}} = 100$,
- MLDRAV with variational attention mechanism,
- MLDRAV with deterministic attention mechanism,
- MLDRAV without attention mechanism,

During training, we set the weights of the multitask and adversarial networks of ADRAV to the same values:

$$\lambda_{(\text{adv})} = \lambda_{(\text{adv, skills})} = \lambda_{(\text{adv, edu})} = \lambda_{(\text{adv, lang})}. \tag{4.3}$$

$$\lambda_{(\text{mul})} = \lambda_{(\text{mul, skills})} = \lambda_{(\text{mul, edu})} = \lambda_{(\text{mul, lang})}. \tag{4.4}$$

Setting $\lambda_{\text{mul}} = 0$ and $\lambda_{\text{adv}} = 0$ means that all the auxiliary losses used in the disentanglement control mechanism have been disabled.

Other training details are described in the following sections.

Parameter name	Parameter value
Maximum length of char n-gram	6
Minimum length of char n-gram	3
Training epochs	5
Learning rate	0.05
Size of word vectors	100
Size of the context window	5
Model type	skipgram

Table 4.1: Hyperparameters used during the training of the custom fastText model

4.3.1 Training fastText embeddings

For research needs, the custom fastText model was trained on the dataset described in Section 4.1. The hyperparameters used during fastText model training are presented in Table 4.1.

4.3.2 DRAV training details

β annealing schedule

Training the deterministic autoencoder is straightforward. Training variational autoencoders can be more challenging due to *KL vanishing issue* [61]. The first problem related to *KL vanishing issue* is posterior almost identical to the Gaussian prior distribution. The second problem is that due to *KL vanishing*, a decoder ignores the information encoded in the latent space. To facilitate this problem, we use the β annealing schedule proposed by Li et al. [69].

During the training process consisting of 100 epochs, we schedule 10 training cycles:

- $\beta = 0$ for 25% of the cycle,
- β increasing linearly from 0 to $\beta_{\max}$ for 25% of the cycle,
- $\beta = \beta_{\max}$ for 50% of the training cycle.

Other optimization tricks

To accelerate the optimization and help VAE achieve better optima, we use a modified objective function with *free bits* proposed in [70]. To reduce the problem of exploding gradients, we also clip the gradient to the defined maximum value.

Parameters setup

During the research we used the parameters of the DRAV network presented in Table 4.2.

DRAV general config	
latent size z_{dim}	512
skills latent space size $z_{\text{dim},S}$	64
education latent space size $z_{\text{dim},S}$	64
languages latent space size $z_{\text{dim},S}$	64
description latent space size $z_{\text{dim},S}$	320
maximum beta value β_{max}	1
weight of variational attention KL divergence ψ	0.1
embedding technique	fastText
bypassing	disabled
free bits λ	0.125
gradient clip	50
$\lambda_{(kl,s)}$	0.05
$\lambda_{(kl,e)}$	0.05
$\lambda_{(kl,l)}$	0.05
$\lambda_{(kl,c}$	0.05
Encoder	
hidden dimension of the encoder LSTM	512
hidden dimensions of the sequence of linear layers in encoder	$[768, 512]$
dropout rate	0.1
optimizer type	Adam
optimizer learning rate	1.0×10^{-4}
Decoder	
hidden dimension of the decoder LSTM	512
number of layers of the decoder LSTM	2
hidden dimensions of the sequence of linear layers in decoder	[768]
dropout rate	0.1
optimizer type	Adam
optimizer learning rate	5.0×10^{-4}
ADRAV architecture specific	
training batch size	32
number of layers of the encoder LSTM	3
adversarial networks learning rate	1.0×10^{-4}
MLDRAV architecture specific	
training batch size	24
number of layers of the encoder LSTM	2

Table 4.2: Fixed hyperparameters of DRAV network

4.4 Results

In this section, we present the results achieved by the different versions of the DRAV network.

4.4.1 Reconstruction capabilities

Since the DRAV network is mainly a sequence-to-sequence model, it should be capable of creating reconstructions of the input texts. We have tested how the attention mechanism and auxiliary losses influence the ability of the DRAV network to reconstruct the seller profiles. In Table 4.3 we report the final training reconstruction losses of the evaluated models. In Tables 4.4, 4.5, 4.6, and 4.7, we have shown examples of reconstructed texts. Both profiles used as examples were sampled from the test dataset and the models did not have access to them during the training phase.

We can clearly see that models equipped with deterministic attention mechanisms outperform models with variational attention and without any attention mechanism at all. In ADRAV with small lambda values and MLDRAV the deterministic attention allows the models to reconstruct outputs of excellent quality. The large λ_{mul} and λ_{adv} in the DRAV model cause reconstruction to be not as good as if the λ_{mul} and λ_{adv} were relatively small. The reason for this observation could be the fact that auxiliary losses with huge λ_{mul} and λ_{adv} are an order of magnitude larger than the training negative log-likelihood reconstruction loss. It should be noted that variational attention does not seem to improve the reconstruction results compared to the method with no attention mechanism applied.

One of the goals during the training of the MLDRAV architecture is to reconstruct certain parts of the input text. In Table 4.8 we show the reconstructions of skills, education, language skills, and descriptions retrieved using the MLDRAV architecture. As we can see, MLDRAV models with deterministic models can accurately extract the desired text segments. Variational attention improves the results to a small extent when comparing the MLDRAV architecture without the attention mechanism.

Type	VAttn	DAttn	λ_{mul}	λ_{adv}	Training NLL loss ↓
ADRAV	✓	✗	0	0	0.220
ADRAV	✗	✓	0	0	0.085
ADRAV	✗	✗	0	0	0.219
ADRAV	✓	✗	2	1	0.222
ADRAV	✗	✓	2	1	0.008
ADRAV	✗	✗	2	1	0.217
ADRAV	✓	✗	200	100	0.225
ADRAV	✗	✓	200	100	0.183
ADRAV	✗	✗	200	100	0.216
MLDRAV	✓	✗	–	–	0.223
MLDRAV	✗	✓	–	–	0.099
MLDRAV	✗	✗	–	–	0.289

Table 4.3: Reconstruction training loss of multiple variants of DRAV

4.4.2 Controlling the disentanglement of factors of variation

Using the competence disentanglement control metric described in Section 4.2.2 we verified whether the proposed multitask, adversarial, and LSTM multitask networks are capable of controlling the disentanglement of the generative factor in the created representation. In Tables 4.9, 4.10, 4.11, 4.12 the results of the evaluation are presented. The *target* in the tables is the probability distribution predicted by the linear layer. The *latent source* refers to the intended purpose of a subspace taken as an input of the multitask or adversarial linear layer.

ADRAV models with $\lambda_{\text{mul}} = \lambda_{\text{adv}} = 0.$ have no direct mechanisms to control the separation of generative factors. Therefore, we expect them to achieve similar results for subspaces of equal dimensionality: skills, education, and languages. The content latent subspace space has 320 dimensions, instead of 64 dedicated for the rest of considered factors, for that reason we the results of the evaluation on content subspace can differ. The results shown in Table 4.9 confirm the assumptions. The results of the evaluation achieved on ADRAV models, without the auxiliary losses, are even for the dimensions of equal dimensionality. Using content space as input of the linear layer improves results for language skills but worsens them for skills and education.

Taking into account the results of ADRAV models with $\lambda_{\text{mul}} = 2$, $\lambda_{\text{adv}} = 1$, presented in Table 4.10, we can see that the content space is much less informative for other generative factors, which manifests itself in a significantly higher negative log-likelihood for all generative factors, compared to other source subspaces. We expected multitask networks to achieve the best results by using as input the latent subspace dedicated to the given generative factor. The results show that unfortunately, this goal has not been achieved by the ADRAV models with $\lambda_{\text{mul}} = 2$, $\lambda_{\text{adv}} = 1$.

The results of the ADRAV models with $\lambda_{\text{mul}} = 200$, $\lambda_{\text{adv}} = 100$, presented in Table 4.11, show that increasing the weight of the auxiliary losses makes the content space even less informative for the generative factors of skills, education and languages, compared to other source subspaces. Similarly to the ADRAV models with $\lambda_{\text{mul}} = 2$, $\lambda_{\text{adv}} = 1$, ADRAV models with $\lambda_{\text{mul}} = 200$, $\lambda_{\text{adv}} = 100$ were unable to successfully prevent the leakage of information to subspaces other than the content subspace.

Analyzing the results of the MLDRAV models presented in Table 4.12, we can see that the distribution of the scores reported is similar to that reported using the ADRAV models with $\lambda_{\text{mul}} = \lambda_{\text{adv}} = 0$. The content subspace provides better results for language skills, but worse for skill and education. The scores reported by MLDRAV are slightly worse than those achieved by the compared method. The reason for this observation may be the fact that the MLDRAV architecture does not have direct mechanisms that prevent information leakage between subspaces.

To answer the question of how the model architecture affects the general ability to retrieve the probability distributions of the generative factors, graphs of the losses of multitask networks shown in Figure 4.5 were prepared. Analysis of them can tell us that the best evaluation results, in general, are achieved in the DRAV model without auxiliary competence

Input: satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied ba graphicdesign national college of arts nca pakistan graduated 2018	
Model	**Output**
ADRAV VAttn $\lambda_{\text{mul}} = 0$ $\lambda_{\text{adv}} = 0$	i am am is and i and and and and and and and and i and and i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and to to to to to to to to to to to to english fluent english english fluent english english design fluent design
ADRAV DAttn $\lambda_{\text{mul}} = 0$ $\lambda_{\text{adv}} = 0$	my main is is is is that that that ensure ensure ensure with results that that why not not but such as well and and they they out out out out there that i i i to to success success satisfaction satisfaction that be be by by my services services my skills english fluent fluent fluent fluent fluent fluent fluent skills skills graphic design design design adobe adobe adobe adobe i have studied studied arts arts college arts arts arts nca pakistan graduated
ADRAV no Attn $\lambda_{\text{mul}} = 0$ $\lambda_{\text{adv}} = 0$	hi i am is a and and a and and and and and and and i i i i i i i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and to to to to to to to to to to to to to to to english english fluent fluent fluent fluent fluent fluent fluent fluent fluent design
Input: i am an american native english speaker and freelance copy writer i can create engaging content for your website or blog think of me as a beacon for your content guiding viewers past the sea of endless google results i know english fluent content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing ba english and history university of texas at austin united states graduated 2020	
ADRAV VAttn $\lambda_{\text{mul}} = 0$ $\lambda_{\text{adv}} = 0$	hi i am is and and and and and and and and and and i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and i and and to to and to to to to to to to to to to to english english fluent english english fluent fluent design design fluent design
ADRAV DAttn $\lambda_{\text{mul}} = 0$ $\lambda_{\text{adv}} = 0$	i am an native native speaker and and and writer i i write create engaging content for or or or or sort of a a a for for your that audience the the the of of google know i know english fluent content writing writing writing typing typing microsoft microsoft microsoft microsoft microsoft writing writing writing writing ba ba ba and university university university university arbor united united states graduated
ADRAV no Attn $\lambda_{\text{mul}} = 0$ $\lambda_{\text{adv}} = 0$	hi i am is a and and a and and and and and and and i i i i i i i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and to to to to to to to to to to to to to to to english english fluent fluent fluent fluent fluent fluent fluent fluent fluent design

Table 4.4: Examples of reconstructed texts using ADRAV with $\lambda_{\text{mul}} = 0$ and $\lambda_{\text{adv}} = 0$

<table>
<tr><td colspan="2">Input: satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied ba graphicdesign national college of arts nca pakistan graduated 2018</td></tr>
<tr><th>Model</th><th>Output</th></tr>
<tr><td>ADRAV VAttn
$\lambda_{mul} = 2$ $\lambda_{adv} = 1$</td><td>hello i a is a and and a and and i and i and and i i and i and i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and i and i to and and to to to to to to to to to to to to to to i english english english english english english english english english english english design</td></tr>
<tr><td>ADRAV DAttn
$\lambda_{mul} = 2$ $\lambda_{adv} = 1$</td><td>satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied ba graphicdesign national college of arts nca pakistan graduated 2018</td></tr>
<tr><td>ADRAV no Attn
$\lambda_{mul} = 2$ $\lambda_{adv} = 1$</td><td>hi am a is i i a a and and and i i i i i i i i i i i i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and</td></tr>
<tr><td colspan="2">Input: i am an american native english speaker and freelance copy writer i can create engaging content for your website or blog think of me as a beacon for your content guiding viewers past the sea of endless google results i know english fluent content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing ba english and history university of texas at austin united states graduated 2020</td></tr>
<tr><td>ADRAV VAttn
$\lambda_{mul} = 2$ $\lambda_{adv} = 1$</td><td>hello i a is a and a a and and and and i and i i i i i i i and i and i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and i and and and and and to and and i to to and to to to to to to to to to to to to to english english english english english english english english english design english design</td></tr>
<tr><td>ADRAV DAttn
$\lambda_{mul} = 2$ $\lambda_{adv} = 1$</td><td>i am an american native english speaker and freelance copy writer i can create engaging content for your website or blog think of me as a p for your content guiding viewers past the sea of endless google results i know english fluent content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing ba english and history university of texas at austin united states graduated 2020</td></tr>
<tr><td>ADRAV no Attn
$\lambda_{mul} = 2$ $\lambda_{adv} = 1$</td><td>hi am a is a i a a and and i i i i i i i i i i i i i i and and i i and and and and and and and and and and and and i i and i i i i and and and and and and and and and and and and and and and and and and and and and and i i i i and i i to to</td></tr>
</table>

Table 4.5: Examples of reconstructed texts using ADRAV with $\lambda_{mul} = 2$ and $\lambda_{adv} = 1$

<table>
<tr><td colspan="2">Input: satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied ba graphicdesign national college of arts nca pakistan graduated 2018</td></tr>
<tr><th>Model</th><th>Output</th></tr>
<tr><td>ADRAV VAttn
$\lambda_{mul} = 200$
$\lambda_{adv} = 100$</td><td>hi am a a i and and and and and and and and and and and and and and and and and and and and and and and and and i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and design i and and and i and design design and and design design design design design design design design design design design design design design design design design</td></tr>
<tr><td>ADRAV DAttn
$\lambda_{mul} = 200$
$\lambda_{adv} = 100$</td><td>i am a to to i to to to to to to to to to to to to to to to to to to to i i i to to to to to to to to to to to to to to to to to to to to english english english english english fluent fluent fluent fluent design design design design design design design design design design design design design design design design education education bsc university university university graduated graduated graduated graduated</td></tr>
<tr><td>ADRAV no Attn
$\lambda_{mul} = 200$
$\lambda_{adv} = 100$</td><td>hi i a is i i a a and and and and and and and and and and and and and and and i i i i i i i i i i i i i i i i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and i i i to to to to to to to to to to to to to english english english english english fluent fluent fluent fluent fluent fluent fluent design</td></tr>
<tr><td colspan="2">Input: i am an american native english speaker and freelance copy writer i can create engaging content for your website or blog think of me as a beacon for your content guiding viewers past the sea of endless google results i know english fluent content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing ba english and history university of texas at austin united states graduated 2020</td></tr>
<tr><td>ADRAV VAttn
$\lambda_{mul} = 200$
$\lambda_{adv} = 100$</td><td>hi am a i i and and and and i and and and i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and design design and and design design and and and</td></tr>
<tr><td>ADRAV DAttn
$\lambda_{mul} = 200$
$\lambda_{adv} = 100$</td><td>i am a a and writer and and and and and and i and and and and and and and and and and and and and and and to to to to to to to english english english english english english english native writing writing writing writing writing writing writing writing writing writing writing writing writing writing writing writing writing writing education education education university university university university university university graduated graduated graduated graduated</td></tr>
<tr><td>ADRAV no Attn
$\lambda_{mul} = 200$
$\lambda_{adv} = 100$</td><td>hi i a is i i a a and and and and and and and and and and and and and and and i i i i i i i i i i i i i i i i i and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and i i i to to to to to to to to to to to to to english english english english english fluent fluent fluent fluent fluent fluent fluent design</td></tr>
</table>

Table 4.6: Examples of reconstructed texts using ADRAV with $\lambda_{mul} = 200$ and $\lambda_{adv} = 100$

<table>
<tr><td colspan="2">Input: satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied ba graphicdesign national college of arts nca pakistan graduated 2018</td></tr>
<tr><th>Model</th><th>Output</th></tr>
<tr><td>MLDRAV VAttn</td><td>is my is is my my my my to to to your your for with i to to me me for and and and i i that that the to to is is a i i i to to to to to to to to to to my my my skills english english english fluent fluent fluent fluent fluent skills skills design design design design design design design design design design design education university graduated graduated graduated graduated graduated graduated</td></tr>
<tr><td>MLDRAV DAttn</td><td>satisfying my client is priority priority work tirelessly to ensure that your experience with me not just alone alone enjoyable as well and what separates me me out out there that i i commitment to your success satisfaction satisfaction that cant be be by services services services language skills fluent fluent fluent fluent fluent fluent fluent fluent my skills design design design adobe adobe adobe adobe adobe design design design design studied studied of of of of pakistan nca graduated graduated</td></tr>
<tr><td>MLDRAV no Attn</td><td>hi i a packaging and a a and and and and and and renderings i i renderings i i i i i i i i and i and and and i and and and i and and i and and and rating and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and i to english english english english i english english english english english english english design</td></tr>
<tr><td colspan="2">Input: i am an american native english speaker and freelance copy writer i can create engaging content for your website or blog think of me as a beacon for your content guiding viewers past the sea of endless google results i know english fluent content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing ba english and history university of texas at austin united states graduated 2020</td></tr>
<tr><td>MLDRAV VAttn</td><td>i i am a and and and content writer i can and and content and your or website copy your i i a a a a and and and and and and and and and and english english english english writing writing writing writing writing writing microsoft microsoft microsoft writing microsoft writing writing writing writing writing writing writing writing university university university university university graduated graduated graduated</td></tr>
<tr><td>MLDRAV DAttn</td><td>i am american native native speaker speaker freelance creator writer i can create engaging content for your or blog think of me as a for for your content narratives the the sea of the google i know english fluent writing writing writing writing writing writing microsoft microsoft microsoft microsoft microsoft microsoft microsoft microsoft microsoft writing writing writing education ba university university united united united united united graduated graduated</td></tr>
<tr><td>MLDRAV no Attn</td><td>hi am am packaging specialist a a and and and and and and and infographic i i and renderings i and i i i i i i and and and i and and and i and and and and and 3k and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and and i i to english english english english to english english english english english design</td></tr>
</table>

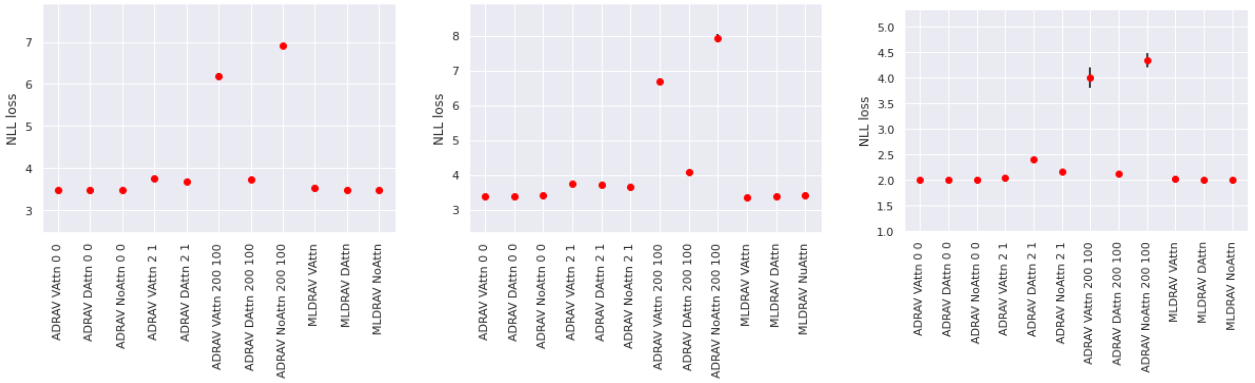
Table 4.7: Examples of reconstructed texts using MLDRAV

Input: satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied ba graphicdesign national college of arts nca pakistan graduated 2018				
Model	**Skills**	**Education**	**Languages**	**Description**
MLDRAV VAttn	my my logo design graphic design adobe adobe web adobe illustrator	i have studied ba civil university university of of sri lanka graduated graduated	my language skills english fluent spanish fluent spanish fluent spanish fluent	is is is is my my my to to to to your your for i to to to me place right and and i i to that to to to to is i i i i to to to your your me me me to to
MLDRAV DAttn	my skills logo design graphic design adobe photoshop adobe illustrator	my education ba graphicdesign national college of arts nca pakistan graduated 2018	my language skills english fluent german fluent italian fluent spanish fluent	satisfying my client my priority i work tirelessly ensure that your experience with is not just successful enjoyable enjoyable well well and except from others out there is i have commitment to your and satisfaction satisfaction that cant be be by services
MLDRAV No Attn	my skills skills belong design design design design	my education education belong university university university graduated graduated graduated graduated	english fluent english fluent english fluent	hi i a is a and a a and packaging and i i i i i i infographic infographic lifestyle lifestyle
Input: i am an american native english speaker and freelance copy writer i can create engaging content for your website or blog think of me as a beacon for your content guiding viewers past the sea of endless google results i know english fluent content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing ba english and history university of texas at austin united states graduated 2020				
MLDRAV VAttn	content writing proofreading proofreading data entry blog editing microsoft word microsoft powerpoint creative writing blog writing	ba english and university university of of united united united states graduated	i know english fluent	i am am a english english writer and and i can can and and for and or or or and and i a a a a a and and and and and and and and and
MLDRAV DAttn	content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing	ba english and history university of at austin united states graduated 2020	i know english fluent	i am american native english speaker and content writer i can engaging content your website or think of me as a for your content narratives over the of of google
MLDRAV No Attn	my skills skills belong design design design	my education education belong university university university graduated graduated graduated graduated	english fluent english fluent english fluent	hi i a is a and a and and packaging label i and i i i i infographic i infographic

Table 4.8: Examples of reconstructed generative factors using MLDRAV

disentanglement control mechanisms, while the huge λ_{mul} and λ_{adv} prevent the linear layer from achieving better results.

As was shown, the proposed techniques have difficulties in separating subspaces of skills, education, and languages. However, ADRAV networks can prevent the leakage of information on any of these factors to the content space. The reason for this observation can be the fact that the content subspace during training is used by adversarial networks, but no multitask network is assigned to it. The network then just has to prevent the subspace to code any valuable information, not bothering minimizing the multitask loss.



(a) Losses of skills multitask layers (b) Losses of education multitask layers (c) Losses of languages multitask layers

Figure 4.5: Multitask network losses

Increasing the multitask loss weights and adversarial entropy loss weights causes the model to achieve worse evaluation results for each of the generative factors.

4.4.3 General disentanglement evaluation

Using the method described in Section 4.2.2 we have tested how well the latent space received by the DRAV network is disentangled. The results are shown in Table 4.13. We can clearly see that the ADRAV architecture outperforms DRAV without auxiliary losses and the MLDRAV architecture. Furthermore, we have proven that increasing the multitask and adversarial entropy loss weights can increase the level of general disentanglement.

4.4.4 Competence transfer

As shown in Tables 4.4, 4.5, 4.6, 4.7, not all proposed network configurations provide sufficient performance to consider them in the transfer of competence task. The only networks with good quality reconstruction are those with deterministic attention. Furthermore, it is not necessary to test networks without auxiliary losses controlling the disentanglement of the latent space ($\lambda_{\text{mul}} = 0$ and $\lambda_{\text{adv}} = 0$). In Table 4.14 we show the results of the competence transfer task using three versions of the DRAV network using the deterministic attention mechanism. Unfortunately, the results show that DRAV models with deterministic attention do not use much information from the latent vector z, which causes the reconstruction to be invariant of the operations carried out in the latent space.

Type	VAttn	DATTN	λ_{mul}	λ_{adv}	Target	Latent Source	Score ↓
ADRAV	✓	✗	0	0	Skills	*Skills*	3.482±0.010
						Education	3.474±0.015
						Languages	3.476±0.015
						Content	3.639±0.013
					Education	Skills	3.408±0.011
						Education	3.401±0.008
						Languages	3.405±0.008
						Content	4.237±0.026
					Languages	Skills	2.012±0.011
						Education	1.996±0.015
						Languages	2.003±0.013
						Content	1.510±0.031
ADRAV	✗	✓	0	0	Skills	*Skills*	3.476±0.008
						Education	3.477±0.009
						Languages	3.466±0.007
						Content	3.563±0.023
					Education	Skills	3.403±0.007
						Education	3.395±0.005
						Languages	3.400±0.009
						Content	4.247±0.036
					Languages	Skills	2.000±0.020
						Education	2.000±0.014
						Languages	1.998±0.013
						Content	1.421±0.010
ADRAV	✗	✗	0	0	Skills	*Skills*	3.474±0.004
						Education	3.481±0.012
						Languages	3.480±0.013
						Content	3.689±0.019
					Education	Skills	3.403±0.010
						Education	3.410±0.006
						Languages	3.410±0.007
						Content	4.279±0.015
					Languages	Skills	2.009±0.017
						Education	2.004±0.016
						Languages	2.006±0.010
						Content	1.579±0.036

Table 4.9: Competence disentanglement control metric of ADRAV with $\lambda_{\text{mul}} = 0$ and $\lambda_{\text{adv}} = 0$

Type	VAttn	DATTN	λ_{mul}	λ_{adv}	Target	Latent Source	Score ↓
ADRAV	✓	✗	2	1	Skills	*Skills*	3.754±0.019
						Education	3.735±0.018
						Languages	3.555±0.010
						Content	4.677±0.050
					Education	Skills	3.761±0.019
						Education	3.755±0.010
						Languages	3.530±0.008
						Content	5.389±0.053
					Languages	Skills	2.137±0.024
						Education	2.139±0.021
						Languages	2.056±0.012
						Content	2.399±0.092
ADRAV	✗	✓	2	1	Skills	*Skills*	3.685±0.011
						Education	3.707±0.014
						Languages	4.171±0.019
						Content	4.882±0.042
					Education	Skills	3.687±0.016
						Education	3.713±0.014
						Languages	4.337±0.011
						Content	5.416±0.042
					Languages	Skills	2.129±0.014
						Education	2.138±0.018
						Languages	2.417±0.037
						Content	2.448±0.117
ADRAV	✗	✗	2	1	Skills	*Skills*	3.680±0.011
						Education	3.665±0.016
						Languages	3.783±0.016
						Content	4.681±0.000
					Education	Skills	3.688±0.011
						Education	3.671±0.018
						Languages	3.810±0.014
						Content	5.073±0.043
					Languages	Skills	2.117±0.025
						Education	2.117±0.021
						Languages	2.161±0.033
						Content	2.482±0.153

Table 4.10: Competence disentanglement control metric of ADRAV with $\lambda_{\text{mul}} = 2$ and $\lambda_{\text{adv}} = 1$

Type	VAttn	DATTN	λ_{mul}	λ_{adv}	Target	Latent Source	Score ↓
ADRAV	✓	✗	200	100	Skills	*Skills*	6.183±0.074
						Education	5.689±0.058
						Languages	6.724±0.086
						Content	36.231±0.853
					Education	Skills	7.524±0.043
						Education	6.692±0.097
						Languages	8.512±0.127
						Content	52.423±1.518
					Languages	Skills	3.496±0.137
						Education	3.208±0.122
						Languages	4.010±0.199
						Content	11.098±0.588
ADRAV	✗	✓	200	100	Skills	*Skills*	3.725±0.014
						Education	3.946±0.026
						Languages	3.705±0.000
						Content	5.037±0.075
					Education	Skills	3.757±0.021
						Education	4.087±0.013
						Languages	3.709±0.023
						Content	5.601±0.085
					Languages	Skills	2.158±0.029
						Education	2.227±0.027
						Languages	2.120±0.021
						Content	2.917±0.095
ADRAV	✗	✗	200	100	Skills	*Skills*	6.915±0.000
						Education	6.344±0.063
						Languages	7.445±0.134
						Content	40.587±0.611
					Education	Skills	9.006±0.188
						Education	7.931±0.115
						Languages	10.096±0.163
						Content	59.64±0.915
					Languages	Skills	3.787±0.098
						Education	3.535±0.103
						Languages	4.350±0.148
						Content	12.629±0.735

Table 4.11: Competence disentanglement control metric of ADRAV with $\lambda_{\text{mul}} = 200$ and $\lambda_{\text{adv}} = 100$

Type	VAttn	DATTN	λ_{mul}	λ_{adv}	Target	Latent Source	Score ↓
MLDRAV	✓	✗	–	–	Skills	*Skills*	3.522±0.009
						Education	3.431±0.009
						Languages	3.462±0.010
						Content	3.807±0.019
					Education	Skills	3.426±0.007
						Education	3.347±0.008
						Languages	3.374±0.008
						Content	4.187±0.017
					Languages	Skills	2.067±0.015
						Education	1.984±0.012
						Languages	2.023±0.008
						Content	1.915±0.046
MLDRAV	✗	✓	–	–	Skills	*Skills*	3.472±0.011
						Education	3.477±0.010
						Languages	3.465±0.015
						Content	3.929±0.026
					Education	Skills	3.395±0.005
						Education	3.393±0.008
						Languages	3.374±0.005
						Content	4.395±0.020
					Languages	Skills	2.011±0.015
						Education	2.004±0.016
						Languages	2.003±0.013
						Content	1.710±0.028
MLDRAV	✗	✗	–	–	Skills	*Skills*	3.492±0.009
						Education	3.489±0.014
						Languages	3.485±0.015
						Content	3.812±0.050
					Education	Skills	3.402±0.009
						Education	3.409±0.010
						Languages	3.406±0.013
						Content	4.350±0.013
					Languages	Skills	2.009±0.014
						Education	2.014±0.022
						Languages	2.010±0.015
						Content	1.658±0.038

Table 4.12: Competence disentanglement control metric of MLDRAV

Type	VAttn	DAttn	λ_{mul}	λ_{adv}	Accuracy ↑
ADRAV	✓	✗	0	0	0.237 ± 0.002
ADRAV	✗	✓	0	0	0.239 ± 0.001
ADRAV	✗	✗	0	0	0.239 ± 0.000
ADRAV	✓	✗	2	1	0.253 ± 0.006
ADRAV	✗	✓	2	1	0.268 ± 0.006
ADRAV	✗	✗	2	1	0.251 ± 0.013
ADRAV	✓	✗	200	100	0.307 ± 0.003
ADRAV	✗	✓	200	100	0.298 ± 0.010
ADRAV	✗	✗	200	100	0.300 ± 0.012
MLDRAV	✓	✗	–	–	0.264 ± 0.007
MLDRAV	✗	✓	–	–	0.240 ± 0.001
MLDRAV	✗	✗	–	–	0.240 ± 0.001

Table 4.13: General disentanglement metric for DRAV

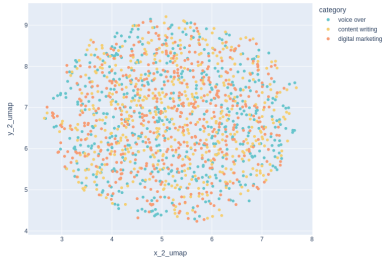
Although the proposed method of mixing different parts of the latent space does not provide satisfying results, in this work we have already presented an easier and more reliable approach that can be used to the task of transferring competences between applicants. Instead of manipulating the latent space, we can use trained multitask LSTM networks from the MLDRAV architecture, which can extract different sections of the job applicant's description with high accuracy (see Table 4.8). Rather than replacing fragments of the latent space and using the decoder to produce the output using the encoded information, we can generate each section separately and concatenate the output. The results of the approach using multitask LSTM networks of the MLDRAV model with deterministic attention are shown in Table 4.15.

4.4.5 Latent space visualization

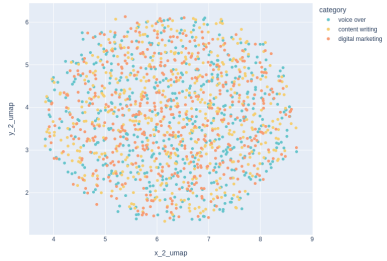
One of the commonly used techniques for evaluating data representation is visualizing the latent space using dimensionality reduction techniques such as t-SNE [71] or UMAP [72]. We have selected 500 users of each of the three groups of users who specialize in voice over, content writing, and digital marketing. We assume that the job applicants assigned to each of these groups have a similar set of skills and that the skills necessary to do voiceover, content writing, or digital marketing differ from each other. In Figure 4.6 we present visualizations of the latent subspace of skills for each of the methods examined. We can see that ADRAV methods that use attention mechanisms with large loss weight values $\lambda_{\text{mul}} = 200$, $\lambda_{\text{mul}} = 100$ can create disentangled data representations. Unfortunately, the mentioned groups are not easily separable using any of the proposed visualizations.

Model	Source factor	Target with source's factor
Target: satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills **english fluent german fluent italian fluent spanish fluent** my skills **logo design graphic design adobe photoshop wordpress adobe illustrator** i have studied **ba graphicdesign national college of arts nca pakistan graduated 2018**		
ADRAV DAttn $\lambda_{\mathbf{mul}} = 2$ $\lambda_{\mathbf{adv}} = 1$	**skills:** content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing	satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills **logo design graphic design adobe photoshop wordpress adobe illustrator** i have studied ba graphicdesign national college of arts nca pakistan graduated 2018
	education: ba english and history university of texas at austin united states graduated 2020	satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied **ba graphicdesign national college of arts nca pakistan graduated 2018**
	languages: english fluent	satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills **english fluent german fluent italian fluent spanish fluent** my skills logo design graphic design adobe photoshop wordpress adobe illustrator i have studied ba graphicdesign national college of arts nca pakistan graduated 2018
ADRAV DAttn $\lambda_{\mathbf{mul}} =$ 200 $\lambda_{\mathbf{adv}} =$ 100	**skills:** content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing	i am a to to i to to to to to to to to to to to to to to to to to to to i i i to to to to to to to to to to to to to to to to to to to to english english english english english fluent fluent fluent fluent design design design design design design design design design design design design design design design design education education bsc university university university graduated graduated graduated graduated
	education: ba english and history university of texas at austin united states graduated 2020	i am a to to i to to to to to to to to to to to to to to to to to to to i i i to to to to to to to to to to to to to to to to to to to to english english english english english fluent fluent fluent fluent design design design design design design design design design design design design design design design design education education bsc university university university graduated graduated graduated graduated
	languages: english fluent	i am a to to i to to to to to to to to to to to to to to to to to to to i i i to to to to to to to to to to to to to to to to to to to to english english english english english fluent fluent fluent fluent design design design design design design design design design design design design design design design design education education bsc university university university graduated graduated graduated graduated
MLDRAV DAttn	**skills:** content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing	satisfying my client is my priority work tirelessly to ensure that your experience with me not just alone that enjoyable well well and what separates me me out out there that i have commitment to your success satisfaction that that cant be be by services services services language skills fluent fluent fluent fluent fluent fluent fluent skills skills skills design design design design adobe adobe adobe adobe adobe design studied studied design of of of of of pakistan pakistan graduated graduated
	education: ba english and history university of texas at austin united states graduated 2020	satisfying my client is priority priority work work to ensure that your experience with me not just alone alone enjoyable as well and what separates me me out out there that i have commitment to your success satisfaction that that be be be by services services services my skills fluent fluent fluent fluent fluent fluent fluent fluent skills skills design design design adobe design adobe adobe adobe design adobe studied studied studied of arts studied of of pakistan pakistan graduated pakistan
	languages: english fluent	satisfying my client is priority priority work tirelessly to ensure that your experience with me not just successful that enjoyable well well and what separates me me out out there is that have commitment to your success and satisfaction that that be be by services services services language skills fluent fluent fluent fluent fluent fluent fluent fluent skills skills design design design adobe adobe adobe adobe adobe adobe design design studied design of arts arts of of pakistan graduated graduated graduated

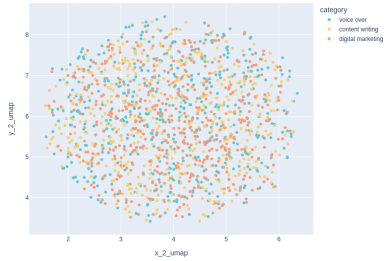
Table 4.14: The results of competence transfer task using manipulation of the latent space



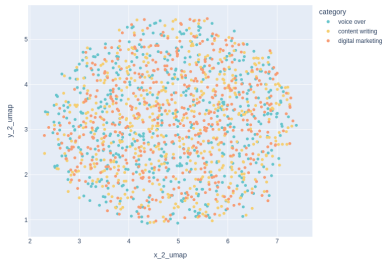
(a) Skills latent space generated by ADRAV with variational attention and $\lambda_{\text{mul}} = 0$, $\lambda_{\text{mul}} = 0$



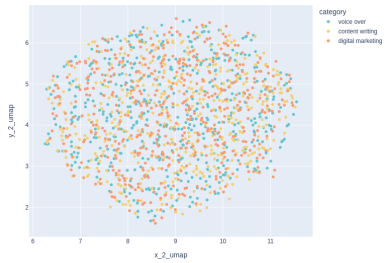
(b) Skills latent space generated by ADRAV with deterministic attention and $\lambda_{\text{mul}} = 0$, $\lambda_{\text{mul}} = 0$



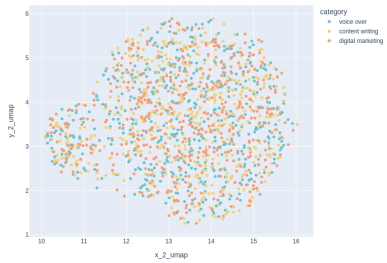
(c) Skills latent space generated by ADRAV with no attention and $\lambda_{\text{mul}} = 0$, $\lambda_{\text{mul}} = 0$



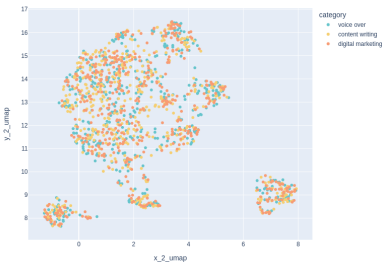
(d) Skills latent space generated by ADRAV with variational attention and $\lambda_{\text{mul}} = 2$, $\lambda_{\text{mul}} = 1$



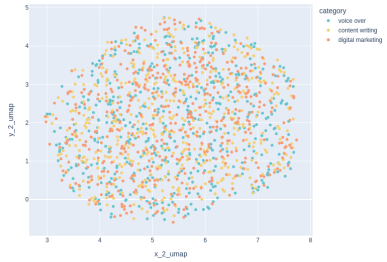
(e) Skills latent space generated by ADRAV with deterministic attention and $\lambda_{\text{mul}} = 2$, $\lambda_{\text{mul}} = 1$



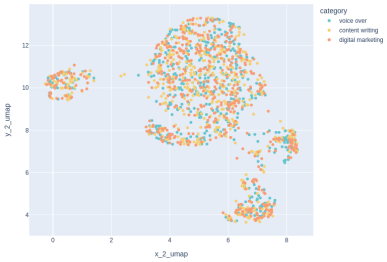
(f) Skills latent space generated by ADRAV with no attention and $\lambda_{\text{mul}} = 2$, $\lambda_{\text{mul}} = 1$



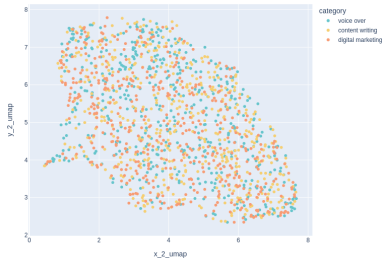
(g) Skills latent space generated by ADRAV with variational attention and $\lambda_{\text{mul}} = 200$, $\lambda_{\text{mul}} = 100$



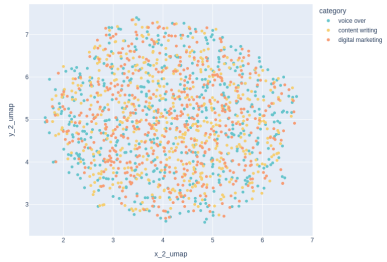
(h) Skills latent space generated by ADRAV with deterministic attention and $\lambda_{\text{mul}} = 200$, $\lambda_{\text{mul}} = 100$



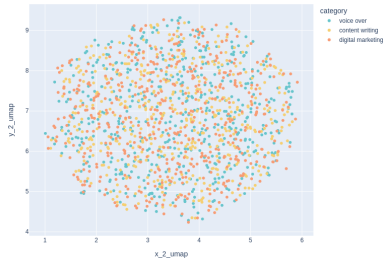
(i) Skills latent space generated by ADRAV with no attention and $\lambda_{\text{mul}} = 200$, $\lambda_{\text{mul}} = 100$



(j) Skills latent space generated by MLDRAV with variational attention



(k) Skills latent space generated by MLDRAV with deterministic attention



(l) Skills latent space generated by MLDRAV with no attention

Figure 4.6: Skills latent space visualizations

Target: satisfying my client is my priority i work tirelessly to ensure that your experience with me is not just successful but enjoyable as well and what separates me from others out there is that i have commitment to your success and satisfaction that cant be matched by other services my language skills **english fluent german fluent italian fluent spanish fluent** my skills **logo design graphic design adobe photoshop wordpress adobe illustrator** i have studied **ba graphicdesign national college of arts nca pakistan graduated 2018**		
Model	**Source factor**	**Target with source's factor**
MLDRAV DAttn	**skills:** content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing	satisfying my client my priority i work tirelessly to that your experience with is not just successful enjoyable enjoyable well and and separates from others out there is i have commitment to your and and satisfaction that that be be by services my language skills english fluent german fluent italian fluent spanish fluent **content writing proofreading typing data entry copy editing microsoft word microsoft powerpoint creative writing blog writing** my education ba graphicdesign national college of arts nca pakistan graduated 2018
	education: ba english and history university of texas at austin united states graduated 2020	satisfying my client my priority i work tirelessly to that your experience me is not just successful enjoyable enjoyable well and and separates from others out there is i have commitment to your and satisfaction satisfaction that cant be be by services my language skills english fluent german fluent italian fluent spanish fluent my skills logo design graphic design adobe photoshop wordpress adobe illustrator **ba english and history university of at austin united states graduated 2020**
	languages: english fluent	satisfying my client my priority i work tirelessly to that your experience me is not just successful enjoyable enjoyable well and and separates from others out there is i have commitment to your and satisfaction satisfaction that cant be be other services i know **english fluent english** skills logo design graphic design adobe photoshop wordpress adobe my education ba graphicdesign national college of arts nca pakistan graduated 2018

Table 4.15: The results of competence transfer task using multitask LSTM networks

4.5 Summary

In the experiments conducted, we have shown that different variations of the DRAV network can perform well in different tasks. ADRAV with no auxiliary losses, produces the latent space, which achieves the best results in predicting the probability distributions of generative factors. However, it does not separate the content subspace from subspaces designed to encode different generative factors. This separation is done relatively well by ADRAV architecture with large values of multitask loss weight and adversarial entropy loss weight. However, it achieves poor quality results in the task of predicting the probability distributions of generative factors. The trade-off can be decreasing the loss weights.

The deterministic attention mechanism greatly improves the performance of DRAV network in tasks of reconstructing the input. Unfortunately, it does not use much information encoded in latent vector, making the model invariant of different latent space manipulations, as the one used in the competence transfer technique. To work around this issue, the MLDRAV with the attention mechanism can be used. Due to the good accuracy of retrieving different sections of job profiles, competences can be transformed simply by concatenating their texts.

We have also shown that ADRAV performs better than the MLDRAV architecture in terms of disentangling generative factors in a representation of job applicants.

5 Discussion and conclusion

We have achieved all the goals defined in the introduction in Section 1.2. In Chapter 2 we have conducted an extensive review of the literature on deep learning, representation learning techniques, variational autoencoders, and current trends in labor market intelligence. In Chapter 3 we have proposed the *Disentangled Representation of Applicants with VAE* (DRAV) model designed to create a disentangled representation of job applicants in two variants: *Adversarial Disentangled Representation of Applicants with VAE* (ADRAV) and *Multi-LSTM Disentangled Representation of Applicants with VAE* (MLDRAV). In Section 4.2.2 we have proposed two custom metrics to measure the quality of the representations: (i) the competence disentanglement control metric and (ii) the general disentanglement metric. In Section 4.4 we have evaluated 12 different combinations of DRAV parameters.

In conclusion, we have shown that the ADRAV architecture is capable of disengaging generative factors in the latent space. The level of measured disentanglement is not high and there is still room for improvement.

5.1 Future work

We believe that future improvements can be made to the DRAV architecture and its training process. Future experiments may include the following.

- evaluating the influence of the different DRAV hyperparameters such as hidden network dimensions on the achieved results,
- creating hybrid architecture consisting of disentanglement control mechanisms from both ADRAV and MLDRAV architectures,
- introducing more complex auxiliary losses capable of better disentangling the latent space,
- training DRAV network large amount diverse data from different sources, such as CV or other job marketplaces,

Bibliography

[1] A. Ryzhkov and D. Zrnić, "Assessment of rainfall measurement that uses specific differential phase," *Journal of Applied Meteorology and Climatology*, vol. 35, no. 11, pp. 2080 – 2090, 1996.

[2] Y. Bengio, A. Courville, and P. Vincent, "Representation learning: A review and new perspectives," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 8, pp. 1798–1828, 2013.

[3] P. Domingos, "A few useful things to know about machine learning," *Communications of the ACM*, vol. 55, pp. 78–87, 10 2012.

[4] I. Millington and J. Funge, eds., *Artificial Intelligence for Games (Second Edition.* Boston: Morgan Kaufmann, second edition ed., 2009.

[5] G. E. Moore *et al.*, "Cramming more components onto integrated circuits," 1965.

[6] G. Brunner, O. Richter, and R. Wattenhofer, "Deep learning for natural language processing (nlp) using variational autoencoders (vae)," 2018.

[7] F. Rosenblatt, "The perceptron: a probabilistic model for information storage and organization in the brain.," *Psychological review*, vol. 65, no. 6, p. 386, 1958.

[8] B. C. Csáji *et al.*, "Approximation with artificial neural networks," *Faculty of Sciences, Etvs Lornd University, Hungary*, vol. 24, no. 48, p. 7, 2001.

[9] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning internal representations by error propagation," in *Parallel Distributed Processing: Explorations in the Microstructure of Cognition, Volume 1: Foundations* (D. E. Rumelhart and J. L. Mcclelland, eds.), pp. 318–362, Cambridge, MA: MIT Press, 1986.

[10] K. Fukushima and S. Miyake, "Neocognitron: A self-organizing neural network model for a mechanism of visual pattern recognition," in *Competition and cooperation in neural nets*, pp. 267–285, Springer, 1982.

[11] S. Hochreiter, "The vanishing gradient problem during learning recurrent neural nets and problem solutions," *International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems*, vol. 6, pp. 107–116, 04 1998.

[12] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, pp. 1735–80, 12 1997.

[13] K. Cho, B. van Merrienboer, D. Bahdanau, and Y. Bengio, "On the properties of neural machine translation: Encoder-decoder approaches," 2014.

[14] P. M. Nadkarni, L. Ohno-Machado, and W. W. Chapman, "Natural language processing: an introduction," *Journal of the American Medical Informatics Association*, vol. 18, pp. 544–551, 09 2011.

[15] R. Feldman, "Techniques and applications for sentiment analysis," *Commun. ACM*, vol. 56, p. 82–89, apr 2013.

[16] J. Kocoń, P. Miłkowski, and M. Zaśko-Zielińska, "Multi-level sentiment analysis of PolEmo 2.0: Extended corpus of multi-domain consumer reviews," in *Proceedings of the 23rd Conference on Computational Natural Language Learning (CoNLL)*, (Hong Kong, China), pp. 980–991, Association for Computational Linguistics, Nov. 2019.

[17] S. Poria, N. Majumder, R. Mihalcea, and E. Hovy, "Emotion recognition in conversation: Research challenges, datasets, and recent advances," *IEEE Access*, vol. 7, pp. 100943–100953, 2019.

[18] S. Shaheen, W. El-Hajj, H. Hajj, and S. Elbassuoni, "Emotion recognition from text based on automatically generated rules," in *2014 IEEE International Conference on Data Mining Workshop*, pp. 383–392, 2014.

[19] M. Crawford, T. M. Khoshgoftaar, J. D. Prusa, A. N. Richter, and H. Al Najada, "Survey of review spam detection using machine learning techniques," *Journal of Big Data*, vol. 2, no. 1, pp. 1–24, 2015.

[20] B. Markines, C. Cattuto, and F. Menczer, "Social spam detection," in *Proceedings of the 5th International Workshop on Adversarial Information Retrieval on the Web*, AIRWeb '09, (New York, NY, USA), p. 41–48, Association for Computing Machinery, 2009.

[21] J. L. Elman, "Finding structure in time," *Cognitive Science*, vol. 14, no. 2, pp. 179–211, 1990.

[22] G. E. Hinton, J. L. McClelland, and D. E. Rumelhart, *Distributed Representations*, p. 77–109. Cambridge, MA, USA: MIT Press, 1986.

[23] T. Mikolov, K. Chen, G. Corrado, and J. Dean, "Efficient estimation of word representations in vector space," 2013.

[24] P. Bojanowski, E. Grave, A. Joulin, and T. Mikolov, "Enriching word vectors with subword information," 2016.

[25] J. Pennington, R. Socher, and C. D. Manning, "Glove: Global vectors for word representation," in *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543, 2014.

[26] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," 2017.

[27] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," 2018.

[28] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," 2019.

[29] V. Sanh, L. Debut, J. Chaumond, and T. Wolf, "Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter," 2019.

[30] K. Do and T. Tran, "Theory and evaluation metrics for learning disentangled representations," in *International Conference on Learning Representations*, 2020.

[31] A. Wang, A. Singh, J. Michael, F. Hill, O. Levy, and S. R. Bowman, "Glue: A multi-task benchmark and analysis platform for natural language understanding," 2018.

[32] P. Rybak, R. Mroczkowski, J. Tracz, and I. Gawlik, "Klej: Comprehensive benchmark for polish language understanding," 2020.

[33] C. E. Shannon, "A mathematical theory of communication," *The Bell system technical journal*, vol. 27, no. 3, pp. 379–423, 1948.

[34] S. Kullback and R. A. Leibler, "On information and sufficiency," *The annals of mathematical statistics*, vol. 22, no. 1, pp. 79–86, 1951.

[35] D. P. Kingma and M. Welling, "Auto-Encoding Variational Bayes," in *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014.

[36] I. J. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," 2014.

[37] L. Dinh, D. Krueger, and Y. Bengio, "Nice: Non-linear independent components estimation," *arXiv preprint arXiv:1410.8516*, 2014.

[38] D. Bank, N. Koenigstein, and R. Giryes, "Autoencoders," 2020.

[39] E. Plaut, "From principal subspaces to principal components with linear autoencoders," 2018.

[40] C. Zhou and R. C. Paffenroth, "Anomaly detection with robust deep autoencoders," in *Proceedings of the 23rd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '17, (New York, NY, USA), p. 665–674, Association for Computing Machinery, 2017.

[41] L. Gondara, "Medical image denoising using convolutional denoising autoencoders," in *2016 IEEE 16th International Conference on Data Mining Workshops (ICDMW)*, pp. 241–246, 2016.

[42] L. Theis, W. Shi, A. Cunningham, and F. Huszár, "Lossy image compression with compressive autoencoders," 2017.

[43] H. E. Robbins, "A stochastic approximation method," *Annals of Mathematical Statistics*, vol. 22, pp. 400–407, 2007.

[44] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," 2018.

[45] L. A. Gatys, A. S. Ecker, and M. Bethge, "A neural algorithm of artistic style," 2015.

[46] X. Chen, Y. Duan, R. Houthooft, J. Schulman, I. Sutskever, and P. Abbeel, "Infogan: Interpretable representation learning by information maximizing generative adversarial nets," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, (Red Hook, NY, USA), p. 2180–2188, Curran Associates Inc., 2016.

[47] I. Higgins, L. Matthey, A. Pal, C. Burgess, X. Glorot, M. Botvinick, S. Mohamed, and A. Lerchner, "beta-vae: Learning basic visual concepts with a constrained variational framework," 2016.

[48] C. P. Burgess, I. Higgins, A. Pal, L. Matthey, N. Watters, G. Desjardins, and A. Lerchner, "Understanding disentangling in beta-vae," *arXiv preprint arXiv:1804.03599*, 2018.

[49] S. Zhao, J. Song, and S. Ermon, "Infovae: Information maximizing variational autoencoders," 2017.

[50] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 424–434, 8 2018.

[51] D. Bahdanau, K. Cho, and Y. Bengio, "Neural machine translation by jointly learning to align and translate," *CoRR*, vol. abs/1409.0473, 2015.

[52] T. Luong, H. Pham, and C. D. Manning, "Effective approaches to attention-based neural machine translation," in *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, (Lisbon, Portugal), pp. 1412–1421, Association for Computational Linguistics, Sept. 2015.

[53] H. Bahuleyan, L. Mou, O. Vechtomova, and P. Poupart, "Variational attention for sequence-to-sequence models," in *Proceedings of the 27th International Conference on Computational Linguistics (COLING)*, 2018.

[54] L. Van-Duyet, V. M. Quan, and D. Q. An, "Skill2vec: Machine learning approach for determining the relevant skills from job description," 7 2017.

[55] A. Giabelli, L. Malandri, F. Mercorio, M. Mezzanzanica, and A. Seveso, "Skills2job: A recommender system that encodes job offer embeddings on graph databases," *Applied Soft Computing*, vol. 101, 3 2021.

[56] J. L. Holland, *Making vocational choices: A theory of vocational personalities and work environments, 3rd ed.* Psychological Assessment Resources, 1997.

[57] C. Zhu, H. Zhu, F. Xie, P. Ding, H. Xiong, C. Ma, and P. Li, "Person-job fit: Adapting the right talent for the right job with," *Joint Representation Learning. ACM Trans. Manage. Inf. Syst*, vol. 9, 2018.

[58] C. Qin, H. Zhu, T. Xu, C. Zhu, L. Jiang, E. Chen, and H. Xiong, "Enhancing person-job fit for talent recruitment: An ability-aware neural network approach," *41st International ACM SIGIR Conference on Research and Development in Information Retrieval, SIGIR 2018*, pp. 25–34, 12 2018.

[59] E. Lacic, M. Reiter-Haas, D. Kowald, M. R. Dareddy, J. Cho, and E. Lex, "Using autoencoders for session-based job recommendations," *User Modeling and User-Adapted Interaction*, vol. 30, pp. 617–658, 2020.

[60] P. Vincent, H. Larochelle, Y. Bengio, and P.-A. Manzagol, "Extracting and composing robust features with denoising autoencoders," pp. 1096–1103, Association for Computing Machinery, 2008.

[61] S. R. Bowman, L. Vilnis, O. Vinyals, A. Dai, R. Jozefowicz, and S. Bengio, "Generating sentences from a continuous space," in *Proceedings of The 20th SIGNLL Conference on Computational Natural Language Learning*, (Berlin, Germany), pp. 10–21, Association for Computational Linguistics, Aug. 2016.

[62] M. Wierzba, M. Riegel, J. Kocoń, P. Miłkowski, A. Janz, K. Klessa, K. Juszczyk, B. Konat, D. Grimling, M. Piasecki, and A. Marchewka, "Emotion norms for 6000 polish word meanings with a direct mapping to the polish wordnet," *Behavior Research Methods*, 2021.

[63] J. Kocoń, A. Figas, M. Gruza, D. Puchalska, T. Kajdanowicz, and P. Kazienko, "Offensive, aggressive, and hate speech analysis: From data-centric to human-centered approach," *Information Processing & Management*, vol. 58, p. 102643, 2021.

[64] K. Cao and S. Clark, "Latent variable dialogue models and their diversity," in *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics*, 2017.

[65] V. John, L. Mou, H. Bahuleyan, and O. Vechtomova, "Disentangled representation learning for non-parallel text style transfer," 2018.

[66] Z. S. Harris, "Distributional structure," *WORD*, vol. 10, pp. 146–162, 8 1954. doi: 10.1080/00437956.1954.11659520.

[67] M. Hu and B. Liu, *Mining and summarizing customer reviews.* 8 2004.

[68] D. M. Eberhard, G. F. Simons, and C. D. Fennig, *Ethnologue: Languages of the World.* SIL International., twenty-fifth edition ed., 2022.

[69] H. Fu, C. Li, X. Liu, J. Gao, A. Celikyilmaz, and L. Carin, "Cyclical annealing schedule: A simple approach to mitigating kl vanishing," *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, vol. 1, pp. 240–250, 3 2019.

[70] D. P. Kingma, T. Salimans, R. Jozefowicz, X. Chen, I. Sutskever, and M. Welling, "Improved variational inference with inverse autoregressive flow," in *Proceedings of the 30th International Conference on Neural Information Processing Systems*, NIPS'16, (Red Hook, NY, USA), p. 4743–4751, Curran Associates Inc., 2016.

[71] L. Van der Maaten and G. Hinton, "Visualizing data using t-sne.," *Journal of machine learning research*, vol. 9, no. 11, 2008.

[72] L. McInnes, J. Healy, N. Saul, and L. Grossberger, "Umap: Uniform manifold approximation and projection," *The Journal of Open Source Software*, vol. 3, no. 29, p. 861, 2018.

[73] F. Chollet, *Deep Learning with Python, Second Edition*. MANNING, 2021.

[74] L. McInnes, J. Healy, and J. Melville, "Umap: Uniform manifold approximation and projection for dimension reduction," 2 2018.

[75] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," 2014.

List of Figures

List of Tables

A Appendix

A.1 Languages distribution

The distribution of text languages in the Fiverr dataset is shown in Table A.1.

Language	Profiles
English	55,273
Norwegian	252
Spanish	172
Italian	110
German	82
French	76
unknown	18
Portuguese	14
Dutch	13
Danish	11
Turkish	8
Russian	7
Catalan	6
Indonesian	6
Tagalog	4
Welsh	4
Vietnamese	4
Afrikaans	3
Hebrew	3
Somali	3
Romanian	2
Slovak	2
Finnish	2
Estonian	2
Polish	2
Arabic	1
Lithuanian	1
Slovenian	1
Swahili	1

Table A.1: The distribution of text languages in the Fiverr dataset