AN ATTENTION-DRIVEN VARIATIONAL AUTOENCODER FOR UNSUPERVISED DISENTANGLEMENT OF SYNTACTIC ROLES

Anonymous authors
Paper under double-blind review

ABSTRACT

Linking neural representations to linguistic factors is crucial in order to build and analyze interpretable NLP models. Among these factors, syntactic roles (e.g., subjects, verbs,...) and their realizations are essential markers since they constitute an understandable decomposition of the predicative structure. In this paper, we show that it is possible to obtain, without supervision, representations of sentences where the predicative structure is separated via the realizations of core syntactic roles. To that end, and drawing inspiration from machine translation Transformers, we first design an Attention-Driven Variational Autoencoder(ADVAE) that enables analyzing the interaction between latent variables and input tokens through attention. We then develop an evaluation procedure to measure disentanglement with regard to the realizations of syntactic roles both on the encoder and on the decoder of ADVAE. We perform experiments on the raw text of the SNLI dataset showing that disentanglement of syntactic roles can be induced without supervision, that ADVAE performs it better than classical sequence VAEs, and that syntactic roles can be separately modified in sentences by mere intervention on the associated latent variables. Shedding light on this behaviour in neural representations constitutes a first step towards unsupervised controllable content generation.The code for our work is publicly available[1].

1 INTRODUCTION

A disentangled representation of data describes information as a combination of separate *understandable* factors. This separation provides better transparency, but also better transfer performance (Burgess et al., 2017).

When it comes to disentanglement, Variational Autoencoders (VAEs; Kingma & Welling, 2014) were extensively proven effective (Higgins et al., 2017; Chen et al., 2018; Rolinek et al., 2019). and were used throughout several recent works (Chen et al., 2019; Li et al., 2020; John et al., 2020). In NLP, disentanglement has been mostly performed to separate the semantics (or content) in a sentence from characteristics such as style and structure in order to generate paraphrases (Chen et al., 2019; John et al., 2020; Bao et al., 2020; Huang & Chang, 2021; Huang et al., 2021). We show in our work that our VAE-based model can also separate the information in the content itself. In contrast to the aforementioned works, we use neither supervision, nor input syntactic information for this separation. We demonstrate this ability by controlling the lexical realization of core syntactic roles. For example, the subject in a sentence can be encoded separately, and controlled to generate the same sentence with another subject.

The model we introduce is an Attention-Driven VAE (ADVAE), which we train on the SNLI raw text dataset (Schmidt et al., 2020). It draws its inspiration from attention-based machine translation models (Bahdanau et al., 2015; Luong et al., 2015). Such models translate sentences between languages with different underlying structures, and can be inspected to show a coherent alignment between spans from both languages. Our ADVAE uses Transformers (Vaswani et al., 2017), an attention-based architecture, to map sentences from a language to independent latent variables, then

[1]URL to be disclosed upon publication.

map these variables back to the same sentences. Although our model is generic and allows for the study of other attributes, we show that it is able to align syntactic roles with separate latent variables.

Evaluating disentanglement with regard to spans is challenging. After training the model and only for this evaluation, we use linguistic information (from an off-the-shelf dependency parser) to first extract syntactic roles from sentences, and then study their relation to latent variables. To study this relation on the ADVAE decoder, we repeatedly *i)* generate a sentence from a sampled latent vector *ii)* perturb this latent vector at a specific location *iii)* generate a sentence from this new vector and observe the difference. On the encoder side, we study the attention values to see whether each latent variable is focused on a particular syntactic role in input sentences. The latter procedure is only possible through the way our ADVAE uses attention to produce latent variables. To the best of our knowledge, we are the first to introduce this transparency mechanism in a latent variable model.

Our contributions can be summarized as follows:

- We introduce the ADVAE, a model that is designed for *unsupervised* disentanglement of syntactic roles, and that enables analyzing the interaction between latent variables and observations through the values of attention (§2).
- We design an experimental protocol for the challenging assessment of disentanglement over realizations of syntactic roles, based on perturbations on the decoder side and attention on the encoder side (§3).
- Our empirical results show that our architecture disentangles syntactic roles better than standard sequence VAEs, and that it is capable of controlling realizations of syntactic roles separately during generation (§4).

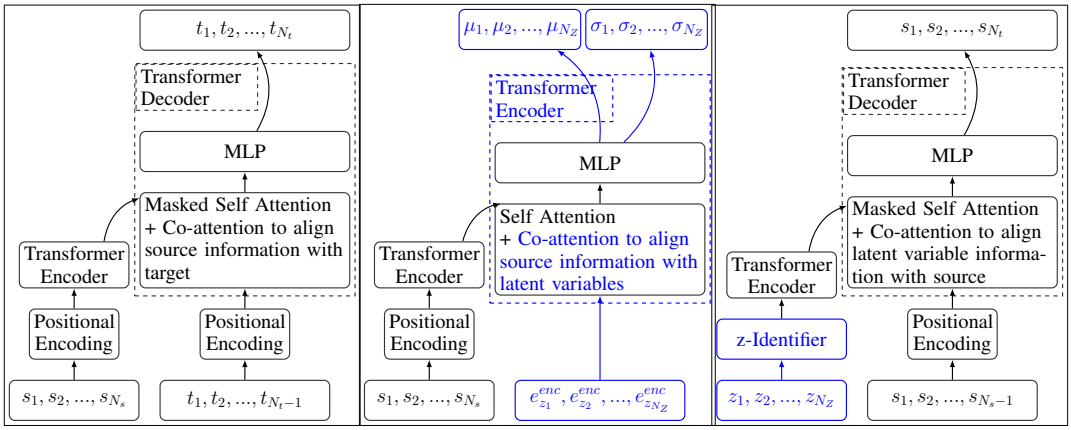


Figure 1: A Transformer-based translation model (a), our encoder q_ϕ (b) and our decoder p_θ (c). In (b) we highlight in blue the difference between our inference model and a source-to-target translation model, while in (c) we highlight the difference with regard to a target-to-source translation model.

2 Model Description

The usual method to obtain sentence representations from Transformer models uses only a Transformer encoder either by taking an average of the token representations, or by using the representation of a special token (*e.g* [SEP] in BERT(Devlin et al., 2019)). Our model, the ADVAE, is novel in that it adapts Machine Translation (MT) Transformers (an encoder and a decoder) to produce sentence representations. In the following sections, we explain the motivation for such a design (§2.1), we present the objective we use (§2.2), and we describe and justify the architectural changes that it requires compared to a MT Transformer (§2.3 and Fig. 1). The parallel between our model and MT Transformers is illustrated in Figure 1.

2.1 THE INTUITION BEHIND OUR MODEL

Consider $s = (s_j)_{1 \leq j \leq N_s}$ and $t = (t_j)_{1 \leq j \leq N_t}$, two series of tokens forming respectively a sentence in a source language and a sentence in a target language. Given s, attention-based translation models are capable of yielding t while also providing information about the alignment between the groups of tokens (of different sizes) in both sentences (Bahdanau et al., 2015; Luong et al., 2015)). This evidence suggests that attention-based architectures are capable of factoring information from groups of words according to a source structure, and redistributing it according to a target structure.

The aim of our design is to use, as a target, a set of N_Z *independent* latent variables that will act as fixed placeholders for the information in sentences. We stress that N_Z is fixed and independent of the input sentence size N_s. Combining Transformers, an attention-based MT model, and the VAE framework for disentanglement, our ADVAE is intended to factor information from independent groups of words into separate latent variables. We hypothesize that, by seeking independent groups of words, our ADVAE will factor the realizations of syntactic roles that appear frequently in a dataset into separate latent variables.

2.2 OPTIMIZATION OBJECTIVE

We train our ADVAE using the β-VAE (Higgins et al., 2017) objective, which is the Evidence Lower-Bound (ELBo) with a controllable weight on its Kullback-Leibler (KL) term:

$$\log p_\theta(s) \geq \mathbb{E}_{(z) \sim q_\phi(z|s)} \left[\log p_\theta(s|z)\right] - \beta D_{\mathrm{KL}}[q_\phi(z|s)||p(z)] \tag{1}$$

In Eq. 1, s is a sample from our dataset, z is our latent variable and the distributions $p_\theta(s) = \int p_\theta(s|z)p(z)dz$ and $q_\phi(z|s)$ are respectively the generation model and the inference model. We use a standard Gaussian distribution as prior $p(z)$ and a diagonal Gaussian distribution as the approximate inference distribution $q_\phi(z|s)$. The weight β is used (as in Chen et al., 2018, Xu et al., 2020, Li et al., 2020) to control disentanglement, but also to find a balance between the expressiveness of latent variables and the generation quality.

In the following sections, our latent variable z will be described as a set of N_Z independent latent variables (*i.e.* $z = (z_i)_{1 \leq i \leq N_Z}$). Notice that the concatenation of N_Z independent diagonal Gaussian variables of dimension D_Z can be viewed as a single diagonal Gaussian variable of dimension $D_Z * N_Z$, and therefore allows for a standard use of the formalism above. We will refer to z as the latent *vector*, and to each z_i as a latent *variable*.

2.3 MODEL ARCHITECTURE

Inference model: This is the inference model q_ϕ (Fig. 1.b) for our latent variables $z = (z_i)_{1 \leq i \leq N_Z}$. It differs in 2 aspects from a MT Transformer.

The first is that it uses as an input a sentence s, and N_Z trainable embedding vectors $(e^{enc}_{z_i})_{1 \leq i \leq N_Z}$ instead of the target tokens t used in translation. The second difference is that its output is not used to select a token from a vocabulary but rather passed to a linear layer (resp. a linear layer followed by a softplus non-linearity) to yield the mean parameters $(\mu_i)_{1 \leq i \leq N_Z}$ (resp. the standard deviation parameters $(\sigma_i)_{1 \leq i \leq N_Z}$) to parameterize the diagonal Gaussian distributions $(q^{(i)}_\phi(z_i|s))_{1 \leq i \leq N_Z}$. The Transformer Decoder is therefore replaced in Fig 1.b by a Transformer Encoder that uses Co-attention (Lu et al., 2019) to factor information from the sentence. The distribution of the whole latent vector is simply the product of Gaussians $q_\phi(z_1, \ldots, z_{N_Z}|s) = \prod_i^{N_Z} q^{(i)}_\phi(z_i|s)$.

Generation model: Our generation model (Fig. 1.c) consists of a decoder $p_\theta(s|z_1, \ldots, z_{N_Z})$ and a prior assuming independent standard Gaussian variables, *i.e.* $p(z_1, \ldots, z_{N_Z}) = \prod_i^{N_Z} p(z_i)$. It identifies latent variables z_i by concatenating them with trainable embeddings $(e^{dec}_{z_i})_{1 \leq i \leq N_Z}$ (*z-Identifier* in Fig. 1.c). From there on, the latent variables are used like source tokens in a MT transformer.

3 EVALUATION PROCEDURE

Before we give the global metrics to quantify disentanglement (§3.4), we present the procedure to extract measurements on the interaction between latent variables and syntactic roles.

In order to get a cleaner picture, we focus on what we call the *core syntactic roles*: subject, direct object, prepositional object, and also verbal root of sentences (to which we will refer as verb). We chose these roles because they map directly to syntactic dependencies, they appear frequently in sentences and because the realizations of these roles are highly informative about the predicative structure and the content of sentences.[2] A dependency parser is used to extract them (§3.1) at evaluation time only (not during training).

For the VAE encoder we use attention to determine the syntactic role followed by each latent variable (§3.3). For the VAE decoder, we repeatedly perturb latent variables, and measure their influence on the realizations of the syntactic roles in generated sentences (§3.2).

3.1 SYNTACTIC ROLE EXTRACTION

We use the Spacy[3] dependency parser (Honnibal & Montani, 2017) trained on Ontonotes5 (Weischedel et al., 2013). For each sentence the realization of *verb* is the root *token* if its POS tag is *VERB*. Realizations of *subj* (subject), *dobj* (direct object), and *pobj* (prepositional object) are *spans* of subtrees whose roots are labelled resp. *nsubj*, *dobj*, and *pobj*.

A realization of a syntactic role in $R = \{verb, subj, dobj, pobj\}$ is empty if no node in the dependency tree satisfies its extraction condition.[4]

3.2 LATENT VARIABLE INFLUENCE ON DECODER

Intuitively, we repeatedly compare the text generated from a sampled latent vector to the text generated using the same *vector* where only one latent *variable* is resampled. Thus we can isolate the effect of each latent $variable$ on output text and gather statistics.

More precisely, we sample T^{dec} latent *vectors* $(z^{(l)})_{1 \leq l \leq T^{dec}} = (z_i^{(l)})_{1 \leq l \leq T^{dec}, 1 \leq i \leq N_Z}$. Then for each z^l, and for each i we create an altered version $\tilde{z}^{(li)} = (\tilde{z}_{i'}^{(li)})_{1 \leq i' \leq N_Z}$ where we resample only the ith latent *variable* (*i.e.* $\forall i' \neq i,\ \tilde{z}_{i'}^{(li)} = z_i^{(l)}$).

Generating the corresponding sentences with $p_\theta(s|z)$ yields a list of original sentences $(s^{(l)})_{1 \leq l \leq T^{dec}}$, and a matrix of sentences displaying the effect of modifying each latent variable $(\tilde{s}^{(li)})_{1 \leq l \leq T^{dec}, 1 \leq i \leq N_Z}$. For each syntactic role $r \in R$, we will denote the realization extracted from a sentence s with $\rho_r(s)$.

To measure the influence of a variable z_j on the realization of a syntactic role r, denoted Γ_{rj}^{dec}, we estimate the probability that a change in this latent variable incurs a change in the span corresponding to the syntactic role. We first discard, for the influence on a role r, sentence pairs $(s^{(l)}, \tilde{s}^{(li)})$ where it appears or disappears, because the presence of a syntactic role is a property of its parent word, which is usually a verb or the dummy root in the case of *core syntactic roles* (*e.g.* if a *dobj* is present, the verb is *transitive*). We leave such changes (diathesis) for future works, since we are interested in realization changes when the predicative structure remains the same. We denote the flag for these cases $\delta_{dec}(r, l, i)$, and renormalize our estimation, after discarding them, with the remaining number of samples T'^{dec}_{ri}. In the following, we use operator $\mathbf{1}\{.\}$, which is equal to 1 when the boolean expression it contains is true and to 0 when it is false, *xor* operator written $\oplus$, and operator $\emptyset(.)$ that is true if its string input is empty, and false otherwise. This process yields a matrix Γ^{dec} of shape $(|R|, N_Z)$ which summarizes interactions in the *decoder* between syntactic roles and latent variables:

[2] Results including all syntactic roles are given in Appendix F.

[3] https://spacy.io/models/en#en_core_web_sm

[4] Examples of syntactic role extractions can be found in Appendix C.

$$\Gamma^{dec}_{ri} = \sum_{l=1}^{T^{dec}} \gamma_{dec}(r,l,i) \; \frac{\delta_{dec}(r,l,i)}{T'^{dec}_{ri}} \tag{2}$$
$$s.t. \;\; T'^{dec}_{ri} = \sum_{l=1}^{T^{dec}} \delta_{dec}(r,l,i), \;\; \gamma_{dec}(r,l,i) = \mathbf{1}\{\rho_r(s^{(l)}) \neq \rho_r(\tilde{s}^{(li)})\},$$
$$\delta_{dec}(r,l,i) = \mathbf{1}\{\emptyset(\rho_r(s^{(l)})) \oplus \emptyset(\rho_r(\tilde{s}^{(li)}))\}$$

3.3 Encoder Influence on Latent Variables

We compute this on a held out set of size T^{enc} of sentences $(s^{(l)}_j)_{1\leq l\leq T^{enc}, 1\leq j\leq N_{s^{(l)}}}$. Each sentence $s^{(l)}$ of size $N_{s^{(l)}}$ generates an attention matrix $(a^{(l)}_{ij})_{1\leq i\leq N_Z, 1\leq j\leq N_{s^{(l)}}}$. Attention values are available in the Transformer Encoder with co-attention computing the inference model.[5]

For the encoder, we consider the influence of a syntactic role on a latent variable to be the probability for the attention values of the latent variable to reach their maximum on the index of a token in that syntactic role realization. For each syntactic role r and sentence $s^{(l)}$, we discard (with a flag $\delta_{enc}(r,l)$) inputs where this syntactic role cannot be found. Γ^{enc}_{ri} denotes the influence of syntactic role r on variable z_i. This process yields matrix Γ^{enc} of shape $(|R|, N_Z)$ which summarizes interactions in the *encoder* between syntactic roles and latent variables:

$$\Gamma^{enc}_{ri} = \sum_{l=1}^{T^{enc}} \gamma_{enc}(r,l,i) \; \frac{\delta_{enc}(r,l)}{T'^{enc}_{r}} \tag{3}$$
$$s.t. \;\; T'^{enc}_{r} = \sum_{l=1}^{T^{enc}} \delta_{enc}(r,l), \;\; \gamma_{enc}(r,l,i) = \mathbf{1}\{\arg\max_j(a^{(l)}_{ij}) \in \rho_r(s^{(l)})\},$$
$$\delta_{enc}(r,l) = \mathbf{1}\{\neg\, \emptyset(\rho_r(s^{(l)}))\}$$

3.4 Disentanglement Metrics

We measure disentanglement taking inspiration from the Mutual Information Gap (MIG) (Chen et al., 2018). MIG consists in measuring the difference between the first and second latent variables with highest mutual information with regard to a target information, and is intended to quantify the extent to which a target information is concentrated in a single variable. This metric assumes knowledge of the underlying distribution of the target information in the dataset, and has been used for cases where this information is categorical. However, there is no straightforward or agreed upon way to set this distribution for text spans, and therefore to calculate MIG in our case. Hence, we propose to use influence metrics defined in §3.2 and §3.3.

For Γ^* (either Γ^{dec} or Γ^{enc}) each line corresponds to a syntactic role in the data. The disentanglement metric for role r is the following:

$$\Delta\Gamma^*_r = \Gamma^*_{rm_1} - \Gamma^*_{rm_2} \tag{4}$$
$$s.t. \;\; m_1 = \arg\max_{1\leq i\leq N_Z} \Gamma^*_{ri}, \;\; m_2 = \arg\max_{1\leq i\leq N_Z, j\neq m1} \Gamma^*_{ri}$$

We calculate total disentanglement scores for syntactic roles using Γ^{dec}, Γ^{enc} as follows:

$$\mathbb{D}_{dec} = \sum_{r\in R} \Delta\Gamma^{enc}_r \;, \;\; \mathbb{D}_{enc} = \sum_{r\in R} \Delta\Gamma^{enc}_r \tag{5}$$

In summary the more each syntactic role's information is concentrated in a single variable, the higher the values of $\mathbb{D}_{dec}$ and $\mathbb{D}_{enc}$. However, similar to MIG, these metrics do not say whether variables capturing our concepts of interest are *distinct*.

[5]For simplicity, attention values are averaged over attention heads and transformer layers.

Therefore, we also report the number of distinct variables that capture the most each syntactic role (*i.e* the number of distinct values of m_1 in Eq. 4 when looping over r). This is referred to as $N_{\Gamma^{\text{enc}}}$ for the encoder and $N_{\Gamma^{\text{dec}}}$ for the decoder.

4 EXPERIMENTS

Dataset Previous unsupervised disentanglement works (Higgins et al., 2017; Kim & Mnih, 2018; Li et al., 2020) tend to use relatively homogeneous and low complexity data. The data has *low complexity* if it varies along clear factors which correspond to what the model aims to disentangle. This allows to draw conclusive results on the inductive bias of the unsupervised model at hand. The work of Xu et al. (2020) on unsupervised disentanglement of sentiment is an example where the used data, a sentiment analysis dataset, displays a very clear variation along the factors they aim to disentangle. Similarly, we use a dataset where samples exhibit low variance in terms of syntactic structure, while providing a high diversity of realizations for the syntactic roles composing the sentences, which is an adequate test-bed for unsupervised disentanglement of syntactic roles' realizations. This dataset is the plain text from the SNLI dataset (Bowman et al., 2015) extracted[6] by Schmidt et al. (2020). The SNLI data is a collection of premises (on average 8.92 ± 2.66 tokens long) made for Natural Language Inference. We use 90K samples as a training set, 5K samples as a development set, and 5K samples as a test set.

Setup Our objective is to check whether the architecture of our ADVAE, induces better syntactic role disentanglement when compared to standard Sequence VAEs (Bowman et al., 2016). This comparison is performed using the same β-VAE objectives, and the decoder disentanglement scores as metrics. Training specifics, and hyper-parameter settings are detailed in Appendix D. The latent variables that we vary for the vanilla VAE model during the decoder's evaluation are the mono-dimensional components of its latent vector. It is easier to pack information about the realizations of multiple syntactic roles into D_z dimensions than into a single dimension. Consequently, the single dimensions we study for the vanilla VAE should be at an advantage to separate information into different variables.

Scoring disentanglement on the encoder side will not be possible for the standard VAE as it requires attention values. To establish that our model effectively tracks syntactic roles, we compare it to a baseline that locates each syntactic role through its median position across the dataset. We refer to this Position Baseline as PB.

The scores are given for different values of β (Eq. 1). Raising β lowers the expressiveness of latent variables, but yields better disentanglement (Higgins et al., 2017). Following Xu et al. (2020), we set β to low values to avoid posterior collapse. In our case, we observed that the models do not collapse for $\beta < 0.5$. Therefore, we display results for $\beta \in \{0.3, 0.4\}$. We stop at 0.3 as lower values for β result in poorer generation quality. For our model we report performance for instances with $N_Z = 4$ (*ours-4*) and $N_Z = 8$ (*ours-8*).

Results The global disentanglement metrics are reported in Table 1.[7]

Model	β	$\mathbb{D}_{enc} \uparrow$	$N_{\Gamma^{\text{enc}}} \uparrow$	$\mathbb{D}_{dec} \uparrow$	$N_{\Gamma^{\text{dec}}} \uparrow$
VAE	0.3	-	-	0.60(0.09)	2.40(0.55)
	0.4	-	-	**1.28**(0.24)	1.40(0.55)
PB	-	0.98 (-)	3.00(-)	-	-
ours-4	0.3	1.30(0.09)	3.00(0.00)	0.68(0.22)	2.80(0.45)
	0.4	**1.46**(0.33)	3.00(0.00)	0.81(0.05)	**3.00**(0.00)
ours-8	0.3	1.36(0.13)	**3.40**(0.89)	0.60(0.10)	**3.00**(0.00)
	0.4	1.44(0.79)	**3.40**(0.55)	0.63(0.35)	2.80(0.45)

Table 1: Disentanglement quantitative results for the encoder (enc) and the decoder (dec). N_Γ indicates the number of separated syntactic roles, and $\mathbb{D}$ measures concentration in a single variable. Values are averaged over 5 experiments. Standard deviation is between parentheses. Best performance in each column is in bold.

[6] https://github.com/schmiflo/crf-generation/blob/master/generated-text/train

[7] Fine-grained scores are given in Appendix E.

ID	Original sentence	Resampled subject	Resampled verb	Resampled dobj/pobj
1	people are sitting on the beach	a young man is sitting on the beach	people are playing in the beach	people are sitting on a bench
2	a man and woman are sitting on a couch	a man is sitting on a park bench	a man and woman are running on a grassy field	the man and woman are on a beach
3	a man is playing with his dog	a boy is playing in the snow	a man is selling vegetables	a man is playing the game with his goal .

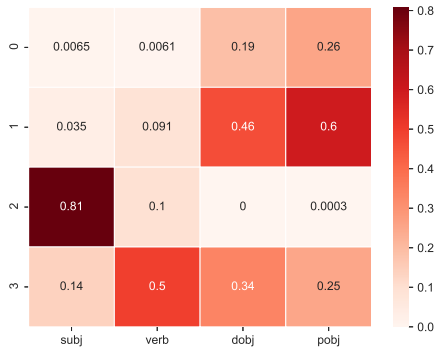
Table 2: Resampling a specific latent variable for a sentence. The ID column is an identifier for the example.

On the decoder side, the Vanilla VAE exhibits disentanglement scores in the range of those reported for our model for $\beta = 0.3$, and higher for $\beta = 0.4$. However, $N_{\Gamma^{\text{dec}}}$ shows that it struggles to factor the realizations of different syntactic roles in different latent variables, and the higher score shown for $\beta = 0.4$ is accompanied with a lower tendency to separate the information from different syntactic roles. In contrast, our model is consistently able to separate 3 out of 4 syntactic roles, and while a higher β raises its $\mathbb{D}_{dec}$, it does not decrease its $N_{\Gamma^{\text{dec}}}$. We can also see that a higher β gives *ours-4* a steady (low variance) improvement in decoder disentanglement score, while for *ours-8* the improvement is slight and comes with a higher variance. As *ours-8* has more latent variables, this may encourage the model to further split the information in each syntactic role between more latent variables.

On the encoding side, our models consistently score above the baseline, showing that our latent variables actively follow the syntactic roles.

In Figures 2 and 3, we display the influence matrices Γ^{enc} and Γ^{dec} for an instance of our ADVAE with $N_Z = 4$ as heatmaps. The vertical axes correspond to the latent variables. As can be seen, our model successfully associates latent variables to verbs and subjects, but chooses not to separate direct objects and prepositional objects into different latent variables. Upon further inspection of the same heatmaps for the VAE baseline, it appears that it most often uses a single latent variable for *verb* and *subj*, and another for *dobj* and *pobj*.

One can also notice on Figures 2 and 3, that the encoder matrix is sparser than the decoder matrix (which is consistent with the higher encoder disentanglement scores in Table 1). This is to be expected as the decoder $p_\theta(s|z)$ adapts the realizations of syntactic roles to each other after they are sampled separately from $p(z)$. The reason for this is that the language modeling objective requires some coherence between syntactic roles (conjugating verbs with subjects, changing objects that are semantically inadequate for a verb, etc). This *co-adaptation*, contradicts the independence of our latent variables. It will be further discussed in §4.



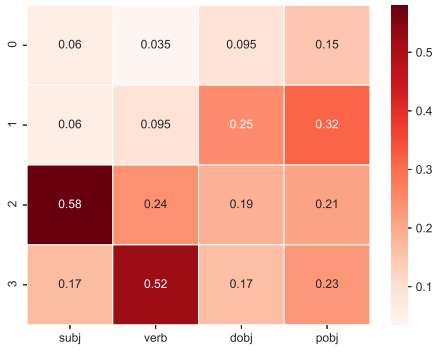


Figure 2: Encoder influence heatmap (Γ^{enc}). Figure 3: Decoder influence heatmap (Γ^{dec}).

Changing the Realizations of Syntactic Roles In this section, we display of few qualitative examples of how the realizations of syntactic roles can be separately changed using an instance of our ADVAE.

ID	Sentence 1	Sentence 2	SSR	Swapped Sentence 1	Swapped Sentence 2
1	a woman is talking on a train	people are sitting on the beach	subj	people are talking on a train	a woman is sitting on the beach
2	a man and woman are sitting on a couch	a woman is talking on a train	verb	a man puts a boy on a park building	a woman is sitting on a train
3	people are sitting on the beach	a woman is talking on a train	verb	people are talking on a beach	a woman is standing on a train
4	a woman is talking on a train	a man is playing with his dog	dobj/ pobj	a man is playing the guitar with a goal	a woman is performing a trick

Table 3: Swapping the value of a specific latent variable between two sentences.The SSR (Swapped Syntactic Role) column indicates the syntactic role that has been swapped.

As a first example, we generate a sentence from a random latent vector, then resample for each syntactic role the corresponding disentangled latent variable to observe the change on the subsequently generated altered sentence. The results of this manipulation are in Table 2[8]. As can be seen, some examples exhibit changes that only affect the target syntactic role (example 1). However, the model often produces co-adaptations that go past the target syntactic role either for semantic soundness (example 2, resampled verb adapts the object), or simply for lack of generalization as a byproduct of the simplicity and narrowness of the SNLI data used for training.

A second example we display is a swap of syntactic role realizations between sentences. A few examples are given in Table 3. Similar to Table 2, the model often yields the expected result. Co-adaptation is best seen here, as taking a syntactic role to a sentence with which it is incompatible results in unexpected changes (example 4).

5 Additional Investigations

Towards Structured Latent Variables As we stated, our ADVAE aims to factor sentences into independent latent variables. However, given the dependency structure of sentences, realizations of syntactic roles are known to be interdependent to some degree in general. Therefore one may think that a structured latent variable model would be better suited to model the realizations of syntactic roles. In fact, such a model could absorb the language modeling co-adaptation between syntactic roles. For instance, instead of sampling an object and a verb from $p(z)$ that are inadequate, then co-adapting them through $p_\theta(s|z)$, a structured $p_\theta(z)$ could produce an *adequate* object for the verb.

Unfortunately, hierarchical latent variable models are notoriously difficult to train (Zhao et al., 2017). An attempt we made at building such a hierarchical model[9] yielded lower disentanglement scores, and often failed to converge to disentanglement. Our independent latent variable model is therefore preferable to the structured one due to these empirical results. More advanced hierarchical latent variable training techniques (such as Progressive Learning and Disentanglement (Li et al., 2020)) may, however, provide better results.

The Behavior of our ADVAE on the Yelp Dataset We investigated the behavior of our ADVAE of on the user generated reviews from the Yelp dataset used in Li et al. (2018) using the same procedure we used for SNLI. The length of sentences from this dataset (8.88 ± 3.64) is similar to the length of sentences from the SNLI dataset. However, Yelp exhibits high syntactic variability (sentences with nominal roots, passive forms, enumerations, etc). Appendix C contains samples from both datasets.

As expected, the variability harms training and evaluation. Although the results show similar trends and lead to similar comparisons between models, they show weak statistical significance[10].

[8]More Examples are available in Appendix G

[9]Full description and results in Appendix A.

[10]These results are in Appendix B.

6 Related Works and Discussion

Our work relates to literature seeking linguistic information in neural networks, and to disentanglement literature.

Linguistics information in neural models Accounting for linguistic information provides better inductive bias in NLP machine learning systems. RNNG (Dyer et al., 2016), ON-LSTM (Shen et al., 2019), SIVAE (Zhang et al., 2019), and ONLSTM-SYD (Du et al., 2020) are examples among others of successful attempts at showing how linguistic information can be captured by neural models. Many ensuing works have also dived into their linguistic capabilities, the types of linguistic annotations that emerge best in them, and syntactic error analyses (Hu et al., 2020; Kodner & Gupta, 2020; Marvin & Linzen, 2020; Kulmizev et al., 2020). BERT (Devlin et al., 2019) has, in turn, been subject to studies showing that the linguistics information it captures are organized among its layers in a way remarkably close to the way a classical NLP pipeline works (Tenney et al., 2020). Furthermore, (Clark et al., 2019), showed that many attention heads in BERT specialize in dependency parsing. We refer the reader to (Rogers et al., 2020) for an extensive review of Bert-related studies. However, most of those studies rely on structural probes to explain representations, probes which are not without issues, as shown by Pimentel et al. (2020). In that regard, the generative capabilities and the attention mechanism of our model offer an intuitive alternative to probing.

Disentanglement in NLP The main line of work in this area revolves around using multitask learning to separate concepts in neural representations (*e.g.* style vs content (John et al., 2020), syntax vs semantics (Chen et al., 2019; Bao et al., 2020)). Alternatively, Huang & Chang (2021) and Huang et al. (2021) use syntactic trees *as inputs* to separate syntax from semantics, and generate paraphrases without a paraphrase corpus. Towards less supervision, Cheng et al. (2020) only uses style information to separate style from content in representations. Literature on *unsupervised* disentanglement in NLP remains sparse. The only example we know of is the work of Xu et al. (2020) which improve the latent space of β-VAEs for unsupervised controllable generation conditioned on categorical generative factors (sentiment and topic).

Our work is novel in that it focuses on syntactic roles. It also adds an entry to the NLP unsupervised disentanglement literature. Additionally, our framework provides an evaluation procedure to score disentanglement with regard to spans. We focus on syntactic roles that are most informative about the content of a sentence, or more precisely, most important to the predicative structure of the sentence. The underlying assumption is that our ADVAE will group information according to the arguments of the main predicate structure in the sentence. We provide empirical observations supporting this assumption in Appendix F by displaying heatmaps such as in Figures 2 and 3, but for all syntactic roles and all PoS Tags.

7 Conclusion

We introduced the ADVAE, a latent variable model that uses attention to map spans to independent latent variables inspired by machine translation. We also introduced a framework to study the unsupervised disentanglement and metrics to quantify this disentanglement, and applied it to control the realization of *core* syntactic roles in short sentences.

ADVAEs have several advantages over standard VAEs for this task. Latent variables in ADVAEs tend to map syntactic roles to separate latent variables more often and more clearly than standard VAEs for this task. Moreover, attention in ADVAEs can be utilized to study the interaction between latent variables and spans. This additional feature helps us measure disentanglement on the encoder side, which gives a cleaner view of disentanglement, unaffected by the co-adaptation of syntactic roles observed on the decoder side.

Despite its limited scope, we are confident that our study constitutes a first step in a promising process towards *unsupervised* explainable modeling and fine grained control over the content of sentences. Although we focused on syntactic roles realizations, this architecture as well as the evaluation method are generic and could be applied to other tasks. The architecture could be used at the document level (*e.g.* disentangling discourse relations) or at the word level (*e.g.* disentangling

morphological affixations), while the evaluation procedure could be applied to other types of spans such as constituents or predicate arguments.

REFERENCES

Dzmitry Bahdanau, Kyung Hyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *3rd International Conference on Learning Representations, ICLR 2015 - Conference Track Proceedings*, pp. 1–15, 2015.

Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xinyu Dai, and Jiajun Chen. Generating sentences from disentangled syntactic and semantic spaces. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 6008–6019, 2020. doi: 10.18653/v1/p19-1602.

Samuel R. Bowman, Gabor Angeli, Christopher Potts, and Christopher D. Manning. A large annotated corpus for learning natural language inference. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pp. 632–642, Lisbon, Portugal, September 2015. Association for Computational Linguistics. doi: 10.18653/v1/D15-1075. URL https://www.aclweb.org/anthology/D15-1075.

Samuel R Bowman, Luke Vilnis, Oriol Vinyals, Andrew M Dai, Rafal Jozefowicz, and Samy Bengio. Generating sentences from a continuous space. *CoNLL 2016 - 20th SIGNLL Conference on Computational Natural Language Learning, Proceedings*, pp. 10–21, 2016. doi: 10.18653/v1/k16-1002. URL http://arxiv.org/abs/1511.06349.

Christopher P Burgess, Irina Higgins, Arka Pal, Loic Matthey, Nick Watters, Guillaume Desjardins, and Alexander Lerchner. Understanding disentangling in-VAE. In *2017 NeurIPS Workshop on Learning Disentangled Representations*, 2017. URL http://arxiv.org/abs/1804.03599.

Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. A multi-task approach for disentangling syntax and semantics in sentence representations. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:2453–2464, 2019. doi: 10.18653/v1/n19-1254. URL http://arxiv.org/abs/1904.01173.

Tian Qi Chen, Xuechen Li, Roger Grosse, and David Duvenaud. Isolating sources of disentanglement in variational autoencoders. In *6th International Conference on Learning Representations, ICLR 2018 - Workshop Track Proceedings*, 2018.

Pengyu Cheng, Martin Renqiang Min, Dinghan Shen, Christopher Malon, Yizhe Zhang, Yitong Li, and Lawrence Carin. Improving Disentangled Text Representation Learning with Information-Theoretic Guidance. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 7530–7541, 2020. doi: 10.18653/v1/2020.acl-main.673.

Kevin Clark, Urvashi Khandelwal, Omer Levy, and Christopher D Manning. What Does BERT Look at? An Analysis of BERT's Attention. In *BlackBoxNLP@ACL*, 2019. doi: 10.18653/v1/w19-4828. URL http://arxiv.org/abs/1906.04341.

Marie-Catherine De Marneffe and Christopher D Manning. Stanford typed dependencies manual. Technical report, Technical report, Stanford University, 2008.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1:4171–4186, 2019. ISSN 0140-525X. doi: arXiv:1811.03600v2. URL http://arxiv.org/abs/1810.04805.

Wenyu Du, Zhouhan Lin, Yikang Shen, Timothy J O'Donnell, Yoshua Bengio, and Yue Zhang. Exploiting Syntactic Structure for Better Language Modeling: A Syntactic Distance Approach. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 6611–6628, 2020. doi: 10.18653/v1/2020.acl-main.591. URL https://www.aclweb.org/anthology/2020.acl-main.591/.

Chris Dyer, Adhiguna Kuncoro, Miguel Ballesteros, and Noah A Smith. Recurrent neural network grammars. *2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference*, pp. 199–209, 2016. doi: 10.18653/v1/n16-1024.

Irina Higgins, Loic Matthey, Arka Pal, Christopher Burgess, Xavier Glorot, Matthew Botvinick, Shakir Mohamed, and Alexander Lerchner. B-VAE: Learning basic visual concepts with a constrained variational framework. *5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings*, pp. 1–22, 2017.

Matthew Honnibal and Ines Montani. spaCy 2: Natural language understanding with Bloom embeddings, convolutional neural networks and incremental parsing. To appear, 2017.

Jennifer Hu, Jon Gauthier, Peng Qian, Ethan Wilcox, and Roger P Levy. A Systematic assessment of syntactic generalization in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1725–1744, 2020. doi: 10.18653/v1/2020.acl-main.158. URL https://www.aclweb.org/anthology/2020.acl-main.158.

James Y. Huang, Kuan-Hao Huang, and Kai-Wei Chang. Disentangling semantics and syntax in sentence embeddings with pre-trained language models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 1372–1379, Online, June 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.naacl-main.108. URL https://aclanthology.org/2021.naacl-main.108.

Kuan-Hao Huang and Kai-Wei Chang. Generating syntactically controlled paraphrases without using annotated parallel pairs. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1022–1033, Online, April 2021. Association for Computational Linguistics. URL https://aclanthology.org/2021.eacl-main.88.

Vineet John, Lili Mou, Hareesh Bahuleyan, and Olga Vechtomova. Disentangled representation learning for non-parallel text style transfer. In *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 424–434, 2020. ISBN 9781950737482. doi: 10.18653/v1/p19-1041.

Hyunjik Kim and Andriy Mnih. Disentangling by factorising. *35th International Conference on Machine Learning, ICML 2018*, 6:4153–4171, 2018.

Diederik P Kingma and Jimmy Ba. Adam: A Method for Stochastic Optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*, 2015. ISBN 9781450300728. doi: 10.1145/1830483.1830503. URL http://arxiv.org/abs/1412.6980.

Diederik P. Kingma and Max Welling. Auto-encoding variational bayes. In Yoshua Bengio and Yann LeCun (eds.), *2nd International Conference on Learning Representations, ICLR 2014, Banff, AB, Canada, April 14-16, 2014, Conference Track Proceedings*, 2014. URL http://arxiv.org/abs/1312.6114.

Jordan Kodner and Nitish Gupta. Overestimation of syntactic representation in neural language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 1757–1762, 2020. doi: 10.18653/v1/2020.acl-main.160. URL https://www.aclweb.org/anthology/2020.acl-main.160.

Artur Kulmizev, Vinit Ravishankar, Mostafa Abdou, and Joakim Nivre. Do neural language models show preferences for syntactic formalisms? In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4077–4091, 2020. doi: 10.18653/v1/2020.acl-main.375. URL https://www.aclweb.org/anthology/2020.acl-main.375%0A.

Juncen Li, Robin Jia, He He, and Percy Liang. Delete, retrieve, generate: a simple approach to sentiment and style transfer. In *Proceedings of the 2018 Conference of the North American Chapter of*

the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers), pp. 1865–1874, New Orleans, Louisiana, June 2018. Association for Computational Linguistics. doi: 10.18653/v1/N18-1169. URL https://www.aclweb.org/anthology/N18-1169.

Zhiyuan Li, Jaideep Vitthal Murkute, Prashnna Kumar Gyawali, and Linwei Wang. Progressive Learning and Disentanglement of Hierarchical Representations. *arXiv*, 2 2020. ISSN 23318422. URL https://openreview.net/forum?id=SJxpsxrYPShttp://arxiv.org/abs/2002.10549.

Jiasen Lu, Dhruv Batra, Devi Parikh, and Stefan Lee. Vilbert: Pretraining task-agnostic visiolinguistic representations for vision-and-language tasks. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett (eds.), *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper/2019/file/c74d97b01eae257e44aa9d5bade97baf-Paper.pdf.

Minh Thang Luong, Hieu Pham, and Christopher D Manning. Effective approaches to attention-based neural machine translation. *Conference Proceedings - EMNLP 2015: Conference on Empirical Methods in Natural Language Processing*, pp. 1412–1421, 2015. doi: 10.18653/v1/d15-1166.

Rebecca Marvin and Tal Linzen. Targeted syntactic evaluation of language models. *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, EMNLP 2018*, pp. 1192–1202, 2020. doi: 10.18653/v1/d18-1151.

Tiago Pimentel, Josef Valvoda, Rowan Hall Maudslay, Ran Zmigrod, Adina Williams, and Ryan Cotterell. Information-theoretic probing for linguistic structure. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 4609–4622, Online, July 2020. Association for Computational Linguistics. doi: 10.18653/v1/2020.acl-main.420. URL https://aclanthology.org/2020.acl-main.420.

Anna Rogers, Olga Kovaleva, and Anna Rumshisky. A Primer in BERTology: What we know about how BERT works. *arXiv*, 2020.

Michal Rolinek, Dominik Zietlow, and Georg Martius. Variational autoencoders pursue pca directions (by accident). *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2019-June:12398–12407, 2019. ISSN 10636919. doi: 10.1109/CVPR.2019.01269.

Florian Schmidt, Stephan Mandt, and Thomas Hofmann. Autoregressive text generation beyond feedback loops. In *EMNLP-IJCNLP 2019 - 2019 Conference on Empirical Methods in Natural Language Processing and 9th International Joint Conference on Natural Language Processing, Proceedings of the Conference*, number 2003, pp. 3400–3406, 2020. ISBN 9781950737901. doi: 10.18653/v1/d19-1338.

Yikang Shen, Shawn Tan, Alessandro Sordoni, and Aaron Courville. Ordered neurons: Integrating tree structures into recurrent neural networks. In *7th International Conference on Learning Representations, ICLR 2019*, pp. 1–14, 2019.

Milan Straka. UDPipe 2.0 prototype at CoNLL 2018 UD shared task. In *Proceedings of the CoNLL 2018 Shared Task: Multilingual Parsing from Raw Text to Universal Dependencies*, pp. 197–207, Brussels, Belgium, October 2018. Association for Computational Linguistics. doi: 10.18653/v1/K18-2020. URL https://www.aclweb.org/anthology/K18-2020.

Ian Tenney, Dipanjan Das, and Ellie Pavlick. BERT rediscovers the classical NLP pipeline. *ACL 2019 - 57th Annual Meeting of the Association for Computational Linguistics, Proceedings of the Conference*, pp. 4593–4601, 2020. doi: 10.18653/v1/p19-1452. URL http://arxiv.org/abs/1905.05950.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention Is All You Need. In *NeurIPS*, number Nips, 2017. ISBN 1469-8714. doi: 10.1017/S0952523813000308. URL http://arxiv.org/abs/1706.03762.

Ralph Weischedel, Martha Palmer, Mitchell Marcus, Eduard Hovy, Sameer Pradhan, Lance Ramshaw, Nianwen Xue, Ann Taylor, Jeff Kaufman, Michelle Franchini, Mohammed El-Bachouti, Robert Belvin, and Ann Houston. OntoNotes Release 5.0, 2013. URL https://hdl.handle.net/11272.1/AB2/MKJJ2R.

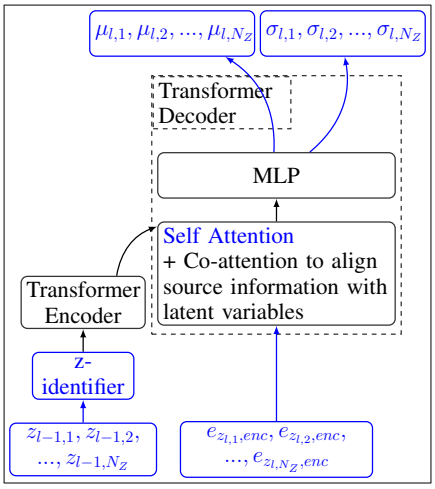
Peng Xu, Jackie Chi Kit Cheung, and Yanshuai Cao. On Variational Learning of Controllable Representations for Text without Supervision. *The 37th International Conference on Machine Learning (ICML 2020)*, 2020. URL http://arxiv.org/abs/1905.11975.

Xinyuan Zhang, Yi Yang, Siyang Yuan, Dinghan Shen, and Lawrence Carin. Syntax-Infused Variational Autoencoder for Text Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pp. 2069–2078, Stroudsburg, PA, USA, 6 2019. Association for Computational Linguistics. ISBN 9781950737482. doi: 10.18653/v1/P19-1199. URL http://arxiv.org/abs/1906.02181https://www.aclweb.org/anthology/P19-1199.

Shengjia Zhao, Jiaming Song, and Stefano Ermon. Learning hierarchical features from deep generative models. *34th International Conference on Machine Learning, ICML 2017*, 8:6195–6204, 2017.

A A HIERARCHICAL VERSION OF OUR ADVAE

For this experiment, rather than using an independent prior $p(z)$, we use a structured prior $p_\theta(z) = p(z_0)\prod_{l=1}^{L} p_\theta(z_l|z_{l-1})$ where $p(z_0)$ is a standard Gaussian, and all subsequent $L-1$ hierarchy levels are parameterized by learned conditional diagonal Gaussians. The model used for each $p_\theta(z_l|z_{l-1})$ is shown in Figure 4 below:



(b)

Figure 4: The conditional inference module linking each of the hierarchy levels in our prior with the next level $p_\theta(z_l|z_{l-1})$. This module treats latent variables from previous layers as they are treated in our original decoder, and generates parameters for latent variables in subsequent hierarchy levels as it is done in our encoder.

We display the results for $L = 2$ and $L = 3$ in Table 4. For both models, we set N_Z to 4.

Depth	β	$\mathbb{D}_{enc}$	$N_{\Gamma^{enc}}$	$\mathbb{D}_{dec}$	$N_{\Gamma^{dec}}$
$L = 2$	0.3	0.79(0.36)	3.60(0.55)	0.51(0.22)	2.60(0.55)
	0.4	0.42(0.23)	2.80(0.45)	0.12(0.20)	2.20(0.45)
$L = 3$	0.3	0.90(0.25)	3.14(0.69)	0.52(0.20)	2.43(0.53)
	0.4	0.32(0.38)	2.75(0.50)	0.25(0.42)	2.25(0.50)

Table 4: Disentanglement results for structured latent variable models on SNLI.

The results show lower mean disentanglement scores, and high standard deviations compared to the standard version of our ADVAE. By inspecting individual training instances of this hierarchical model, we found that some instances achieve disentanglement with close scores to those of the standard ADVAE, while others completely fail (which results in the high variances observed in Table 4). These results show that learning structured latent variables requires non-trivial changes to our model.

B EXPERIMENTING WITH THE YELP DATASET

Model	β	$\mathbb{D}_{enc}$	$N_{\Gamma^{enc}}$	$\mathbb{D}_{dec}$	$N_{\Gamma^{dec}}$
VAE	0.3	-	-	0.44(0.09)	2.20(0.45)
	0.4	-	-	1.21(0.06)	2.25(0.50)
PB	-	0.33(-)	2.00(-)	-	-
ours-4	0.3	0.48(0.07)	2.00(0.00)	0.18(0.02)	2.50(0.58)
	0.4	0.54(0.04)	3.00(0.00)	0.23(0.03)	2.40(0.55)
ours-8	0.3	0.44(0.04)	3.80(0.45)	0.17(0.04)	2.80(0.84)
	0.4	0.57(0.26)	3.40(0.55)	0.15(0.10)	2.40(0.89)

Table 5: Disentanglement results for the Yelp dataset

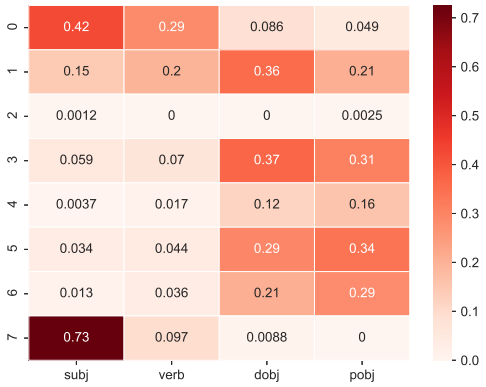


Figure 5: Encoder influence heatmap for Yelp(Γ^{enc}).

Figure 6: Decoder influence heatmap for Yelp(Γ^{dec}).

Similar to the experiments in the main body of the paper, we display the disentanglement scores in Table 5, and the influence metrics of one of the instances of our model as heatmaps in Figures 5 and 6.

Over the entire table, the results are weaker than what we obtained for SNLI. Given the difference between SNLI and Yelp (displayed in Appendix C) there are two clear reasons for this decrease. The first is that Yelp is a dataset where it is harder to locate the syntactic roles. This is illustrated by the fact that the PB baseline obtains a much lower score. The second is that our syntactic role extraction heuristics are tailored for regular sentences with verbal roots, which subjects the evaluation metrics on Yelp to a considerable amount of noise. Nevertheless, the comparisons between a VAE, an ADVAE and PB retain the same conclusions, but with lower margins and some overlapping standard deviations.

Through manual inspection of examples, we observed that the various structural characteristics (enumerations, sentences with nominal roots, presence of coordinating conjunctions, etc) were captured by different variables. This indicates that future iterations of our model need to provide ways to separate structural information from content related information.

C Example Sentences from Yelp and SNLI and their Corresponding Syntactic Extractions

Table 6 shows some samples from SNLI and Yelp reviews. Samples from Yelp Reviews exhibit a clearly higher structural diversity. On the other hand, most of SNLI samples are highly similar in structure.

Our syntactic role extraction heuristics were tailored for sentences with verbal roots. As a results, it can be seen that they struggle with sentences with nominal roots as well as other forms of irregular utterances present in Yelp. For SNLI, our extractions mostly yield the expected results, allowing for a reliable global assessment of our models.

D Training Details and Hyper-Parameter Settings

Our ADVAE's Hyper-parameters Our model has been set to be large enough to reach a low reconstruction error during the initial reconstruction phase of the training. We use 2-layer Transformers with 4 attention heads and a hidden size of 192. Contrary to Vanilla VAEs, our model seems to perform better with high values of N_Z. Therefore, we set our latent vector to a size of 768, and divide it to 96-dimensional variables for our $N_Z = 8$ model and to 192-dimensional latent variables for our $N_Z = 4$ model. No automated hyper-parameter selection has been done afterwards.

Vanilla VAE Hyper-parameters As is usually done for this baseline (Xu et al., 2020), we set both the encoder and the decoder to be 2-layer LSTMs.

Source	Sentence	subj	verb	dobj	pobj
Yelp	i was originally told it would take _num_ mins .	it	told	_ num _ mins	
Yelp	slow , over priced , i 'll go elsewhere next time .	i	go		
Yelp	we will not be back	we			
Yelp	terrible .				
Yelp	at this point they were open and would be for another hour .	they			this point
SNLI	people are outside playing baseball .	people		baseball	
SNLI	two dogs pull on opposite ends of a rope .	two dogs	pull	opposite ends of a rope	a rope
SNLI	a lady lays at a beach .	a lady	lays		a beach
SNLI	people are running through the streets while people watch .	people	running		the streets
SNLI	someone prepares food into bowls	someone	prepares	food	bowls

Table 6: Example syntactic role extractions from both SNLI and Yelp

Model	β	$\mathbb{D}_{dec}$	$N_{\Gamma^{\text{dec}}}$	$\Delta\Gamma_{dec,verb}$	$\Delta\Gamma_{dec,subj}$	$\Delta\Gamma_{dec,dobj}$	$\Delta\Gamma_{dec,pobj}$
ADVAE-4	0.3	0.68(0.22)	2.80(0.45)	0.19(0.04)	0.35(0.18)	0.06(0.03)	0.07(0.03)
	0.4	0.81(0.05)	3.00(0.00)	0.21(0.04)	0.47(0.03)	0.06(0.02)	0.07(0.02)
ADVAE-8	0.3	0.60(0.10)	3.00(0.00)	0.17(0.04)	0.31(0.08)	0.05(0.04)	0.07(0.04)
	0.4	0.63(0.35)	2.80(0.45)	0.17(0.10)	0.32(0.18)	0.05(0.04)	0.08(0.05)
VAE	0.3	0.60(0.09)	2.40(0.55)	0.24(0.06)	0.03(0.04)	0.03(0.02)	0.31(0.03)
	0.4	1.28(0.24)	1.40(0.55)	0.45(0.12)	0.23(0.02)	0.02(0.02)	0.57(0.11)

Table 7: Complete decoder disentanglement scores for SNLI

We run this model for hidden LSTM sizes in [256, 512], and latent vector sizes in [16, 32]. The results for the model scoring the highest $\mathbb{D}_{dec}$ are then reported. Even though selection has been done according to $\mathbb{D}_{dec}$, we checked the remaining instances of our baselines and they also yielded low $N_{\Gamma^{\text{dec}}}$ values.

Training Phases All our models are trained using ADAM(Kingma & Ba, 2015) with a batch size of 128 and a learning rate of 2e-4 for 20 epochs. The dropout is set to 0.3. To avoid posterior collapse, we train all our models for 3000 steps with $\beta = 0$ (reconstruction phase), then we linearly increase β to its final value for the subsequent 3000 steps. Following Bowman et al. (2016), we also use word-dropout. We set its probability to 0.1.

Evaluation For the evaluation, T^{dec} is set to 2000 and T^{enc} is equal to the size of the test set. To get the decoder metrics, the sentences are generated using greedy sampling (sampling the maximum-probability word at each step).

E Disentanglement Scores for each Syntactic Role

The full disentanglement scores are reported in Table 7 for the decoder, and in Table 8 for the encoder.

Model	β	$\mathbb{D}_{enc}$	$N_{\Gamma^{\text{enc}}}$	$\Delta\Gamma_{enc,verb}$	$\Delta\Gamma_{enc,subj}$	$\Delta\Gamma_{enc,dobj}$	$\Delta\Gamma_{enc,pobj}$
ADVAE-4	0.3	1.30(0.09)	3.00(0.00)	0.28(0.05)	0.65(0.02)	0.08(0.03)	0.29(0.03)
	0.4	1.46(0.33)	3.00(0.00)	0.38(0.12)	0.64(0.10)	0.14(0.04)	0.30(0.10)
ADVAE-8	0.3	1.36(0.13)	3.40(0.89)	0.44(0.12)	0.60(0.18)	0.21(0.08)	0.11(0.06)
	0.4	1.44(0.79)	3.40(0.55)	0.42(0.23)	0.61(0.34)	0.17(0.10)	0.23(0.16)
Average Position	-	0.98 (-)	3.00(-)	0.12(-)	0.70(-)	0.12(-)	0.04(-)

Table 8: Complete encoder disentanglement scores for SNLI

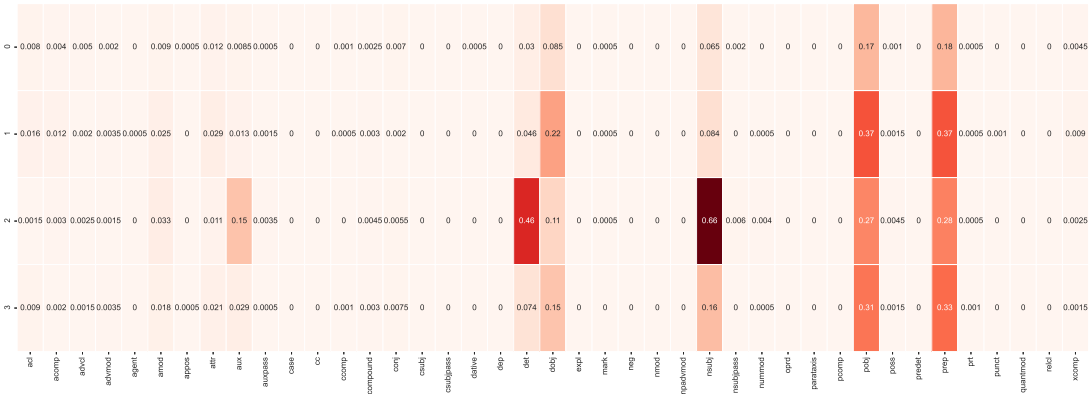


Figure 7: Decoder influence heatmap for all SD syntactic roles.

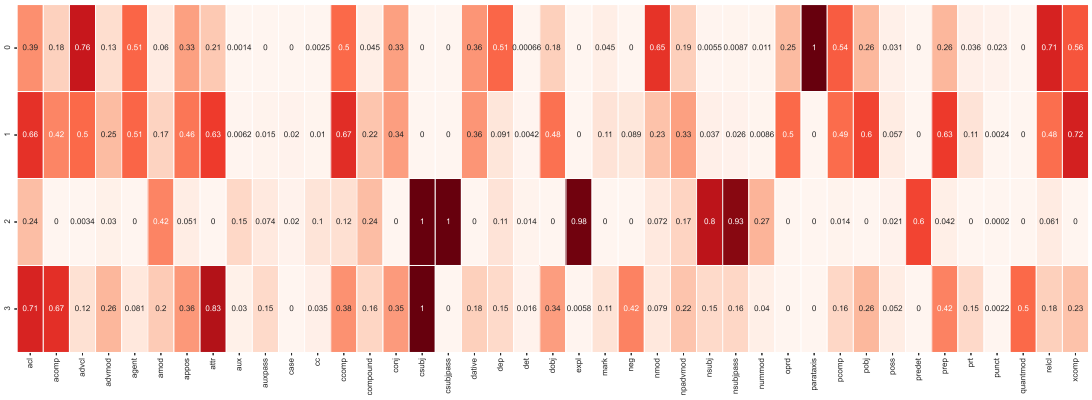


Figure 8: Encoder influence heatmap for all SD syntactic roles.

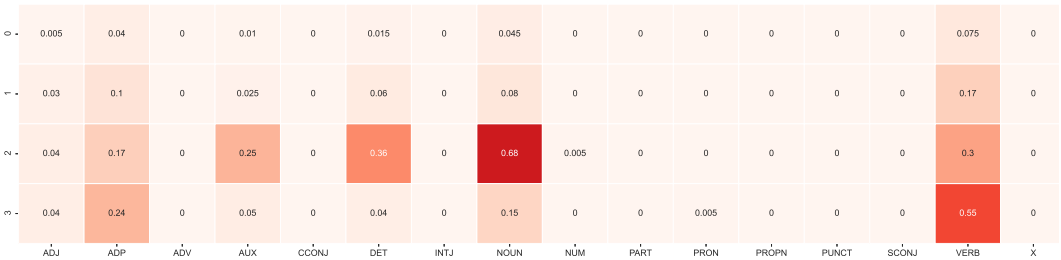


Figure 9: Decoder influence heatmap for all PoS Tags.

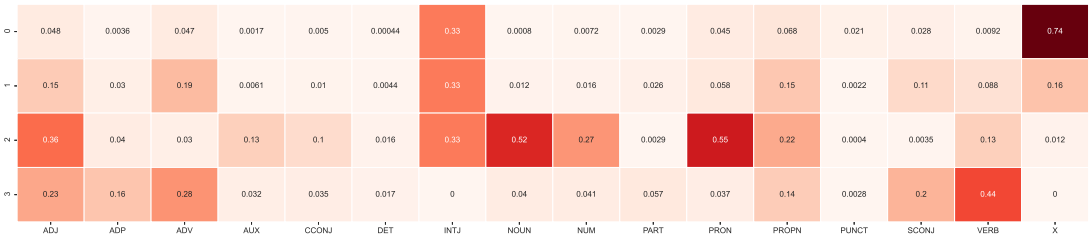


Figure 10: Encoder influence heatmap for all PoS Tags.

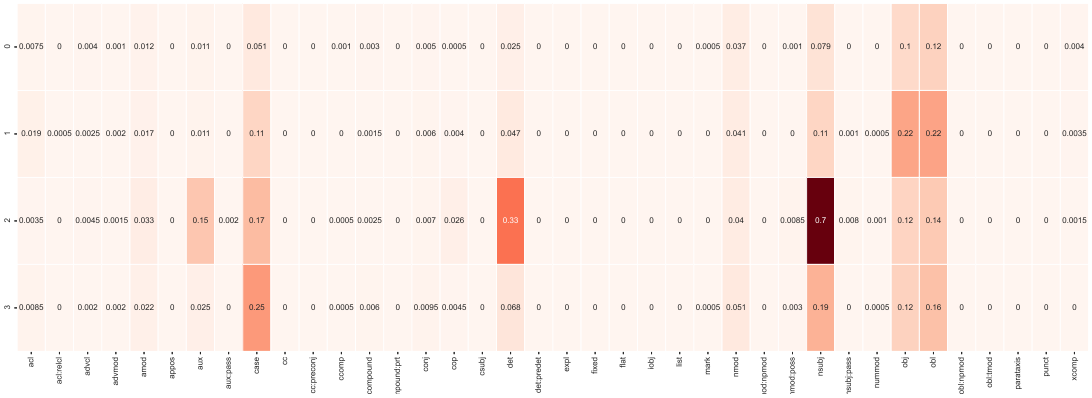


Figure 11: Decoder influence heatmap for all UD syntactic Roles.

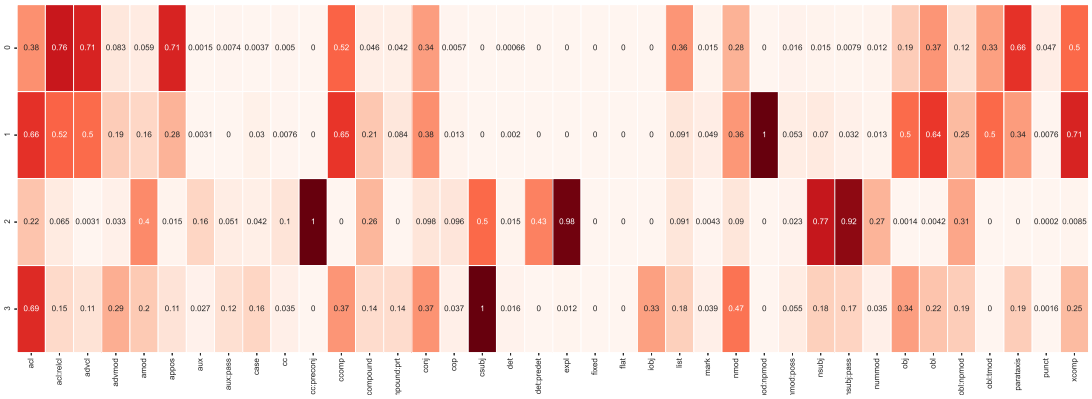


Figure 12: Encoder influence heatmap for all UD syntactic Roles.

F Disentanglement Heatmaps Over the Entire Range of Syntatic Roles and PoS Tags

We report decoder and encoder heatmaps for all the syntactic roles following the Stanford Dependencies (SD; De Marneffe & Manning, 2008) annotation scheme of Ontonotes, which was used to train our Spacy2 parser, in Figures 7 and 8. For the sake of extensiveness and to make sure we did not draw results from some parser biases, we also report the same heatmaps but using UDPipe 2.0 (Straka, 2018), which uses UD type annotations[11], in Figures 11 and 12. Finally, we also report heatmaps for interaction with PoS Tags extracted with Spacy2 in Figures 9 and 10. As was done in the main body of the paper, the span corresponding to each syntactic role (in both annotation

[11] A widely adopted annotation scheme derived from Stanford Dependencies.

schemes) was taken to be the series of words included in its corresponding subtree. In contrast, the span corresponding to each PoS tags was just taken to be the tagged word. Results from UD parsing extraction lead to the same conclusions as from our initial SD results.

The instance of our ADVAE for which we display the above heatmaps is the same one for which we display the heatmaps in Figures 2 and 3 in the main body of the paper. As shown in those Figures, it mostly uses variable 3 for verbs, variables 2 for subjects, and variable 1 for objects. The remaining variable (0) also seems to capture some interaction with objects. As conjectured in our discussion, the heatmaps show that our ADVAE tends to group syntactic roles into latent variables in a way that aligns with the predicative structure of sentences. In fact, variable 2 displays the highest influence on the PoS tag VERB as well as its surroundings as a predicate argument such as adverbs and adverbial phrases. Similarly, latent variable 2 displays a high influence on subjects (nominal or clausal), numeral modifiers, adjectival modifiers, and auxiliaries (for conjugation). Moreover, Variable 1 highly influences the direct and prepositional objects, which we study in the main body of the paper, but also diverse clausal modifiers and obliques which often play similar roles to direct and prepositional objects in a predicate structure.

G Additional Examples of Resampled realizations for each syntactic role

Table 9 contains a wide array of examples where the latent variable corresponding to each syntactic role is resampled.

Original sentence	Resampled subject	Resampled verb	Resampled dobj/pobj
the woman is riding a large brown dog	two men are riding in a large city	the woman is wet	the woman is riding on the bus
the police are running in a strategy	a man is looking at a date	the police are at an arid	the police are running in a wooded area
a man is holding a ball	a man is holding a ball	a man is , and a woman are talking on a road	a man is sitting on a cellphone outside
everyone is watching the game	some individuals are watching tv	everyone is a man	everyone is watching the game in the air
there is a man in the air	a man is sitting in the air	there is no women wearing swim trunks	there is a man in a red shirt
a group of friends are standing on a beach	an elderly father and child are standing on the beach	a group of people are standing on a beach	a group of friends are looking at the beach
the women are in a store	a man is playing a game	two women are on a break	two women are sitting on a bench
a man is playing a game	a little girl is playing with a ball	a man is clean	a man is sitting on a lake to an old country
a man is playing a game	some dogs are playing in the pool	a man is preparing to chase himself	a man is playing a game
the memorial woman is happy	a dog is happy	the memorial workers are in a room	the memorial is happy
a man is wearing a green jacket and a ship	a boy sitting in a green device	a man is dancing for the camera	the man is wearing a hat
a man is playing a game	a man is playing a game	two men are tripod	a man is playing with a guitar
a man is wearing a brown sweater and green shirt	a karate dog is swimming in a chair	a man is bought a brown cat in an airplane	a man is wearing a dress and talks to the woman
the woman is about to visitors	three people are working at a babies	the woman is wearing a sewer	the woman is about to sell a tree
a man is sitting in the snowy field	a man is sitting in the snowy field	a man is wearing electronics	a man is sitting on a park bench
two people are playing in the snow	the motorcycle is a woman on the floor	two people play soccer in the snow	two people are playing in a concert
a man is standing next to another man	a boy is standing next to another man	a man is standing	a man is standing next to a man
a man is on his bike	a man is on his bike	a dog is showing water	a man is on his bike
a man is sitting in front of a tree , taking a picture	a man is sitting in front of a tree	a man is holding a red shirt and climbing a tree	a man is sitting on a suburban own
a man is sitting with a dog	the children are sitting with the dog	a man is playing with a dog	a man is sitting with an umbrella
the man is holding a ball	a boy is playing with a ball	the man is on a bike	the man is waiting for a counts to jump for the first base
a man is holding a game	five people buying a skateboard from easter	a man is on a bicycle	a man is very large
two men are playing in a field	the kids play in the snow	two men are playing a game	two men are playing in a field
a man is wearing a hat	a woman is wearing a hat	the man is they oil	a man is wearing a black bathing suit near buildings
a man is playing a guitar	the man is wearing a blue shirt	a man is sitting on a bench	a man wearing a hat is playing a guitar
a woman is playing a game	a man is playing a game	a woman is playing a game	a woman is playing a game
a man is on the truck	the people are on the truck	a man is holding a truck	a man is on the grass
a man is playing with the cut	a small boy is playing on the cut	a man is warming up the cut	a man is playing a game
a group of people are at a park	the man is wearing a blue shirt	a group of people are at a park	a group of people are at a park

Table 9: More examples where we resample a specific latent variable for a sentence.