Disentangling the Latent Space of (Variational) Autoencoders for NLP

Gino Brunner, Yuyi Wang, Roger Wattenhofer, and Michael Weigelt

ETH Zurich, Switzerland
brunnegi@ethz.ch

Abstract. We train multi-task (variational) autoencoders on linguistic tasks and analyze the learned hidden sentence representations. The representations change significantly when translation and part-of-speech decoders are added. The more decoders are attached, the better the models cluster sentences according to their syntactic similarity, as the representation space becomes less entangled. We compare standard unconstrained autoencoders to variational autoencoders and find significant differences. We achieve better disentanglement with the standard autoencoder, which goes against recent work on variational autoencoders in the visual domain.

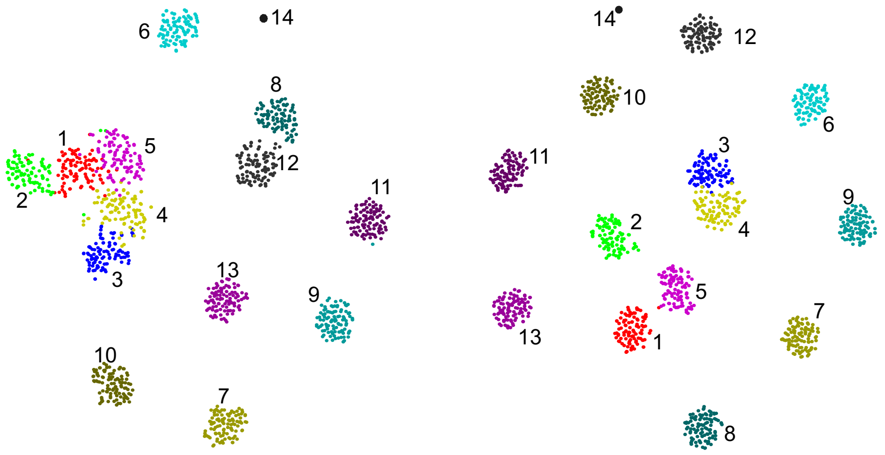
Keywords: NLP, variational,autoencoder, disentanglement, representation learning, syntax

1 Introduction

Learning good representations lies at the core of Deep Learning [1]. We would like algorithms to automatically extract the most salient features instead of having to rely on expert knowledge to manually design complex preprocessing pipelines. If a model can learn good features, it will likely perform well in an array of (downstream) tasks. Another important aspect is that of transfer learning, where a model is trained on multiple tasks that mutually benefit from each other, leading to better performance in each task. A model that can learn good representations is likely to perform better in a transfer learning setting. For more background on what makes good representations, we refer the interested reader to [1]. Higgins et al. [3] have shown that a simple modification to the standard Variational autoencoder (VAE) objective enables disentanglement of independent linear data generating factors for an artificial dataset of simple 2D shapes. Other works have achieved similar results for the visual domain. However, the progress for discrete sequences, such as natural language, has been much slower. Prior work shows the efficacy of Variational autoencoders, which are generally good at learning representations, for complex tasks such as sentiment transfer ([9]). It has also been shown that multi-task learning is beneficial in the context of NLP [6–8]. In this paper, we focus on analyzing the learned representations and investigate the disentanglement capabilities of (variational) autoencoders in a Multitask setting for Natural Language Processing (NLP). A commonly used definition of disentanglement (e.g., [3]) is that small changes in one dimension of the hidden representation should result in small changes in only one data generating factor, where data generating factors could be the size of an object. In the context of language however, it is considerably more difficult to come up with such linear data generating factors, and thus, there are not yet any general definitions of disentanglement. In this paper we therefore only look at one specific factor of language: syntax. We investigate the ability of autoencoder based language models to learn disentangled representations of syntax. We define a representation to be disentangled if the hidden representations of sentences with different syntactic structures can be clustered with little to no overlap. To this end we train several multi-task autoencoder models, where each decoder performs a distinctive linguistic task. We compare the sentence representations our models have learned and explore how representations of different sentences relate to each other.

2 Models

Our models are based on the autoencoder (AE) and variational auteoncoder (VAE) [4] frameworks. In both cases, an encoder transforms the data into a lower dimensional representation, from which a decoder tries to replicate the input. We use Long short-term memory (LSTM) neural networks for all encoders and decoders. The VAE formulation additionally encourages the latent variables to be distributed according to a prior (usually an isotropic Gaussian with unit variance). This is achieved by adding a second term to the AE loss function that minimizes the KL-divergence between the chosen prior and the true posterior. When weighted appropriately, this constraint acts as a regularizer on the number of latent dimensions that are used by the model, which in turn promotes disentanglement of the latent dimensions ([2, 3]). Higgins et al. formally introduce this weight in the VAE loss function as β, and thus call their VAE variant β-VAE. Apart from the standard replicating decoder (REP(R)), we attach additional decoders to perform different tasks. The multi-task models in this paper use a subset of the following three decoders in addition to the REP decoder. The German and French (GER(G)/FR(F)) decoders translate the input sentence to German and French respectively. The part-of-speech (POS(P)) decoder learns to tag words in the input sequence with part-of-speech tags, such as *verb*, *noun*, *adjective*. To train our models on the three tasks replication, translation and POS, we use the aligned multilingual transcripts of the European Parliament sessions ([5]). The subset of this dataset we use contains over 1.7 million sentences, 1.5 million of which were used as the training set. The remaining 0.2 million sentences form the test set. Our models are trained on character-sequences.



(a) R(EP) model with $CE = 51$. Some sentence prototype representation clusters are very close together or overlapping.

(b) RGP model with $CE = 0$. No sentence representation clusters are overlapping, and only type 3 and 4 are close together.

Fig. 1: Syntax clusters for the autoencoder models visualized with t-SNE.

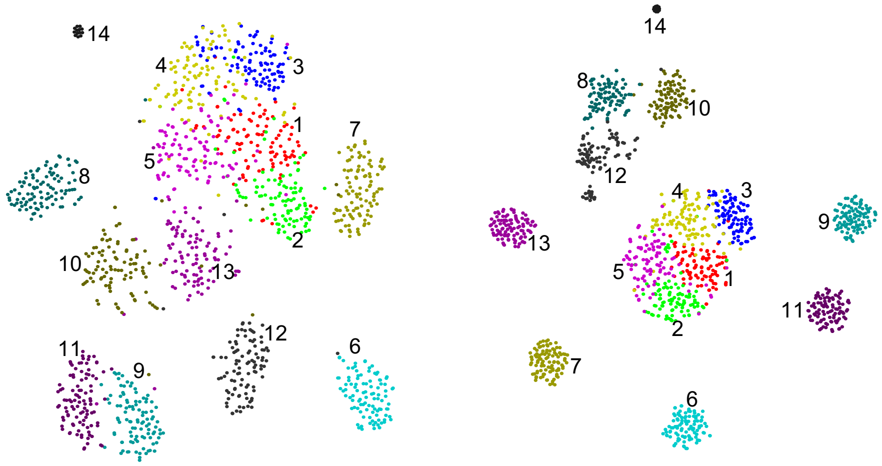
3 Results

3.1 Syntax clustering

To compare the learned representations of different models, we examine how well they cluster syntactically similar sentences in latent space. We define 14 sentence prototypes (see Table 1) with different syntactic structures. N, V, A and D are placeholders for nouns, verbs, adjectives and adverbs. Each sentence prototype is randomly populated by common English words 100 times. These sentences are then fed through the encoders to obtain their representation vectors, which are then clustered by K-means with $K = 14$. For each resulting cluster, we count how many sentences of each prototype it contains. The cluster is then labeled

Table 1: Sentence prototypes.

1: The N is A.	2: The N Vs.	3: The N has a N.	4: The N Vs a N.
5: The N Vs a N.	6: No N ever Vs.	7: Are Ns A?	
8: The Ns of N D V the A N, but some Ns still V their N.			
9: In the N of a A N, the N will V the N of Ving the N.			
10: Ns V the A N of Ns Ving on the N.			
11: In the N of N, Ns would rather V without N than V any A Ns.			
12: N Vs in order to V on a N.		13:A Ns often V like Ns.	
14: *whitespace*			



(a) R(EP) model with $CE = 197$. Several sentence types are highly overlapping or close together. The individual clusters have a large diameter.

(b) RGP model with $CE = 25$. The latent space is much less entangled and the clusters have smaller diameters, but several sentence types are still overlapping or very close together.

Fig. 2: Syntax clusters for the β-VAE models visualized with t-SNE.

with the majority prototype. The per-cluster error is defined as the number of sentences in the cluster that are not of the majority type. The sum of errors of all 14 clusters is the *clustering error* (CE), which is our quality metric for this experiment. Since K-means clustering is nondeterministic, we run the algorithm 100 times. Table 2 shows the best-of-100 clustering errors.

Table 2: AE vs β-VAE (β=0.001)

Model	R	RF	RGF	RG	RP	RGP
AE	51	26	24	22	8	0
VAE	197	87	26	98	58	25

For the standard autoencoder, adding more tasks clearly helps reduce the clustering error. The POS tagging decoder brings the highest benefit. This makes sense, since most sentence prototypes have a unique POS tag sequence, and thus separating the sentence prototypes in latent space will make the POS tagging decoder's job easier. Attaching either a German or French translation decoders also help reduce the clustering error. Figure 1 shows the sentence representation of two different AE models, visualized using t-SNE. The RGP model is significantly better at disentangling syntax.

For the β-VAE, the results are much less consistent, and generally worse than for the standard AE, even though β-VAE was shown to disentangle factors of variation in latent space for visual tasks. Especially the high CE for the RP model is surprising. In the experiments performed by [3], $\beta = 4$ yielded the highest degree of disentanglement. Unfortunately, increasing the weight of the KL-term in the VAE loss has a negative effect on reconstruction performance. We were not able to train models with $\beta > 0.1$, which is consistent with existing VAE implementations for sequence tasks [1]. We trained multiple β-VAEs with $\beta \in [0, 0.0001, 0.001, 0.01, 0.1]$ and found that $\beta = 0.001$ generally performs best in terms of reconstruction performance and clustering error. Figure 1 shows the sentence representation of two different VAE models, visualized using t-SNE. The latent space is clearly more entangled than for the AE based RGP model.

3.2 Interpolation and Representation Space Algebra

To further evaluate the properties of our models we traverse the latent space by interpolating between samples from the dataset. We find that the models with the best clustering errors produce smoother interpolations with fewer non-words and more consistent syntactic structure. We also investigate the learned linear relationships between latent sentence representations. To do this we compute a new sentence representation $\hat{s}$ by combining the latent representations of three sentences as $\hat{s} = s_1 - s_2 + s_3$. Intuitively, s_3 should be modified with the difference-vector of s_1 and s_2. For example: *Cats are good pets*(s_1) and *Dogs are good pets*(s_2) should have canceled out the part about good pets and roughly point from *Dogs* to *Cats* ($s_1 - s_2$). Adding this difference-vector to any sentence that contains *Dogs* should then result in a sentence where *Dogs* is replaced with *Cats*. We find that the models with low clustering error perform significantly better at this task, as shown in Table 3.

Table 3: Examples of representation vector algebra for two autoencoder models. The RGP model produces the correct result for the first two examples. Both models manage to replace *small* with *large* in the third example, but also wrongly change most other parts of the sentence.

	s_1		s_2		s_3		s_1 - s_2 + s_3
RG	I am one.	-	I am two.	+	You are two.	=	You ready no.
RGP	I am one.	-	I am two.	+	You are two.	=	**You are one.**
RG	A word in a phrase.	-	A tree in a phrase.	+	A tree is green.	=	A word is purevy?
RGP	A word in a phrase.	-	A tree in a phrase.	+	A tree is green.	=	**A word is green.**
RP	A large number of people want to work.	-	A small number of people want to work.	+	A small sentence is enough.	=	A large senselfeir in or evacce.
RGP	A large number of people want to work.	-	A small number of people want to work.	+	A small sentence is enough.	=	A large sector for challenge.

[1] e.g., https://github.com/tensorflow/magenta/tree/master/magenta/models/music_vae

4 Conclusion

We trained several multi-task autoencoders on linguistic tasks and analyzed the learned sentence representations based on a new clustering based metric using a toy dataset. Adding linguistic tasks helps the models disentangle syntax in latent space. We further found significant differences between standard and variational autoencoders. We built a toy dataset based on sentence prototypes and introduced the clustering error metric to evaluate the disentanglement of the learned representations. In the future we plan to formulate more rigorous definitions of good (e.g., disentangled) representations in the context of natural language, and evaluate models with different degrees of disentanglement on downstream tasks to see if there is any benefit. We will also revisit the use of recurrent neural networks. As recent trends show, CNNs or pure attention based models might be better suited to model sequences. CNNs are known to be powerful feature extractors, which might be one reason for the success of unsupervised representation learning methods for vision.

Bibliography

[1] Bengio Y, Courville AC, Vincent P (2013) Representation learning: A review and new perspectives. IEEE Trans Pattern Anal Mach Intell 35(8):1798–1828, DOI 10.1109/TPAMI.2013.50, URL https://doi.org/10.1109/TPAMI.2013.50

[2] Burgess CP, Higgins I, Pal A, Matthey L, Watters N, Desjardins G, Lerchner A (2018) Understanding disentangling in β-vae. arXiv preprint arXiv:180403599

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

[4] Kingma DP, Welling M (2013) Auto-encoding variational bayes. CoRR abs/1312.6114, URL http://arxiv.org/abs/1312.6114, 1312.6114

[5] Koehn P (2005) Europarl: A Parallel Corpus for Statistical Machine Translation

[6] Liu X, Gao J, He X, Deng L, Duh K, Wang Y (2015) Representation learning using multi-task deep neural networks for semantic classification and information retrieval. In: NAACL HLT 2015, The 2015 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp 912–921

[7] Luong M, Le QV, Sutskever I, Vinyals O, Kaiser L (2015) Multi-task sequence to sequence learning. CoRR abs/1511.06114

[8] Niehues J, Cho E (2017) Exploiting linguistic resources for neural machine translation using multi-task learning. In: Proceedings of the Second Conference on Machine Translation, WMT 2017, pp 80–89

[9] Shen T, Lei T, Barzilay R, Jaakkola T (2017) Style transfer from non-parallel text by cross-alignment. In: Advances in Neural Information Processing Systems, pp 6833–6844