Variational autoencoders for argumentation representation learning

Kuan Yu, Jan Milde
{kuanyu, jmilde}@uni-potsdam.de
Master's Program in *Cognitive Systems*
University of Potsdam

March 2019

Abstract

We trained two recurrent variational autoencoders for learning argumentation representation with datasets assembled from the internet, one for sentence embedding and another for post embedding. We evaluated the learned representation in supervised tasks with a third dataset containing online debate posts annotated with labels regarding the content of argumentation. The results were a mixture of successes and failures. This paper gives a detailed analysis.

1 Introduction

Argumentation mining is a developing field of natural language processing in pursuit of the automatic identification and extraction of information from argumentative texts. We are particularly interested in the content of argumentation, such as the topic of debate and the argumentative reason. This project explored the unsupervised learning of argumentation representation with variational autoencoders, and applied the learned representation in supervised tasks for evaluation. The results fell short of our expectations, but our explorations were fruitful and educational. The implementation for our experiments is openly available.[1]

We start this paper with a description of the theoretical and technical background (Section 2). Then we explain the general setup of our experiments and detail the training of our unsupervised models (Section 3). Next

[1] https://github.com/argsim/argsim

we present and examine the results of supervised evaluation (Section 4). At last we conclude our paper with a summary and some discussions (Section 5).

2 Background

2.1 Argumentation embedding

There is an abundance of data available for argumentation mining from discussions led online which, if being ordered and structured, could give a well rounded view on a topic. However, data labeling is time consuming and labor intensive. Effective unsupervised methods for argumentation mining are desired. Boltužić and Šnajder (2015) experimented with unsupervised clustering of argumentative sentences based on the similarity of their sentence embeddings, which were derived by adding up all the word embeddings of the sentence.

Following the invention of the word2vec model (Mikolov et al. 2013), word embeddings have risen to popularity and have become a standard when working with text. The reason for the success is that the embedding model learns a meaningful representation space. For example, within this space are linear substructures for gender representation such that the distance between "man" and "woman" would be highly similar to the distance between "king" and "queen". Similarity metrics such as the cosine similarity between word vectors prove to be an effective way to represent their semantic similarity. In general, the closest neighbors are highly related. I.e. conjugations, synonyms, or words of the same topic (Pennington, Socher, and Manning 2014). This motivated researchers to explore methods to embed longer texts, such as sentences or paragraphs. As one of the first steps in this direction, simply adding up weighted word vectors to be sentence embeddings and weighted sentence embeddings to be paragraph embeddings was a working approach (Arora, Liang, and Ma 2016). It could be shown that sentence or paragraph embeddings can also have a meaningful spacial interpretation (Dai, Olah, and Le 2015). This makes research for sentence or paragraph embeddings highly relevant for the field of argumentation similarity. However, the composition of texts from words is a process far more complicated than the linear combination of word vectors. Emergent linguistic properties such as the grammar formalism and the writing style require more sophisticated modeling tools. Hence as of now, the search for the best text embedding method is still in progress.

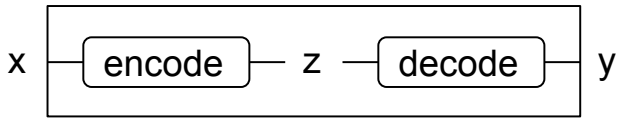

Figure 1: A variational autoencoder ($x = y$).

2.2 Recurrent variational autoencoder

Probabilistic graphical models based on the principle of Boltzmann machines saw the success of data-driven modeling (Ackley, Hinton, and Sejnowski 1985). In this connectionist approach, a generative model is trained on the raw data. Such a generative model typically consists of visible and hidden nodes. The data is input to the visible nodes of the model, and values for the hidden nodes are computed through probabilistic inference. A trained model can be used to generate data from a particular configuration of hidden values, namely the hidden state, and can also be used for inferring the hidden state which explains the structure of a given data point. The hidden states are constrained by the architecture of the model to have simpler structures than the raw data, which makes these models useful for extracting non-linear features from the raw data for linear classification or clustering (Hinton and Salakhutdinov 2006). Exact inference in a generative model is often intractable, for which approximation methods such as variational inference is usually used. An effective training method for deep belief networks triggered the revival of artificial neural networks in statistical machine learning (Hinton, Osindero, and Teh 2006). Modern variants of generative models have maintained their popularity in the form of differentiable generative networks which allow for effective training through backpropagation, such as variational autoencoders (VAEs) (Kingma and Welling 2013).

A VAE consists of an encoder and a decoder (Figure 1). The encoder is a neural network which receives a data point as input and outputs a latent vector z, from which the decoder network reconstructs the original data point. Conceptually, the decoder is a directed graphical model from the latent space to the data space whereas the encoder performs variational inference in the backward direction. The components of the latent vector z are assumed to be independent and identically distributed random variables, typically following a standard isotropic multivariate normal distribution. This assumption is enforced during training through backpropagation using the reparameter-

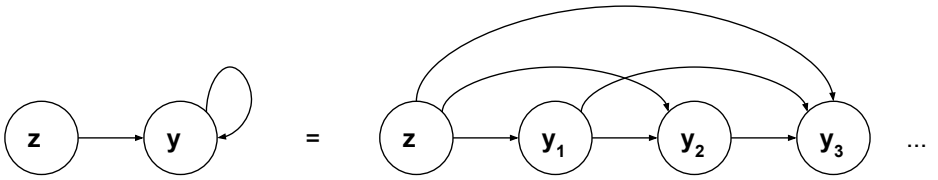

Figure 2: An autoregressive decoder models the probability of a sequence y given a latent state z, with the sequence factorized causally by position: $p(y \mid z) = p(y_1 \mid z)\ p(y_2 \mid y_1, z)\ p(y_3 \mid y_2, y_1, z)\ \ldots$

ization trick where random noises are added to the latent variables. The training objective is to raise the variational lower bound by minimizing the Kullback–Leibler divergence (KL) between the distributions of the latent variables and their assumed prior while maximizing the likelihood of the training data or equivalently minimizing the reconstruction error.

$$\begin{aligned}\mathcal{L}(\theta; x) &= \mathbb{E}_{q_\theta(z|x)}[\log p_\theta(x|z)] - \mathtt{KL}(q_\theta(z|x) \,\|p(z)) \\ &\leq \log p(x)\end{aligned}$$

The prior imposed on z by the KL term in the loss function enforces information to be encoded in a disentangled manner. The encoder acts as an embedding function from the data space to the latent space, where each dimension represents a distinctly interpretable feature of the data. For this reason, VAEs serve as an interesting bridge between the connectionist approach and the symbolist approach to data modeling, which is evident in their capability of separating the style and the content of images (Kingma, Mohamed, et al. 2014).

In natural language processing, VAEs have been successfully applied for sentence representation learning (Bowman et al. 2015), discourse-level diversity learning (Zhao, Zhao, and Eskenazi 2017), topic model learning (Srivastava and Sutton 2017), semantic space learning (Jang, Seo, and Kang 2018), and target-level sentiment analysis (Xu and Tan 2018).

In those applications, recurrent neural networks (RNNs) are a typical choice. The encoder RNN transforms an input sequence to a single state vector, while the decoder autoregressively generates an output sequence, with each step conditioning on the encoded state and previously generated partial sequence (Figure 2). During generation, an output is sampled from the decoder predictions at each step, and fed back together with the recurrent state for the next step. Since the search space grows exponentially for such a

structure prediction problem, beam search is commonly used to approximate the optimal solution (Freitag and Al-Onaizan 2017). A cheaper alternative is greedy decoding, where the sampling is simply performed by picking the prediction with the highest probability. For training an autoregressive model, the teacher-forcing method is typically used (Williams and Zipser 1989). Instead of running the feedback loop, the true sequence is given as input with a begin-of-sentence symbol padded at the front to predict the same sequence with an end-of-sentence symbol padded at the end.

2.3 Sentence piece

When treating texts as sequences generated from a set of symbols, it is common and intuitive to choose words as the symbolic units. This however has many drawbacks. The vocabulary size of any corpus is huge, often in an order of magnitude around or above 10^6, resulting in a large number of parameters in the input and output layers of the model. On top of that, due to their power law distribution, the majority of words in the vocabulary of a language remains unseen in the available training data, and most words in the observed vocabulary occur infrequently. Furthermore, morphology is an important part of language, especially for agglutinative languages, which can only be modeled at the subword level. Modeling at the character-level can be effective (Kalchbrenner et al. 2016). However, the sequences become much longer, making computation prohibitively expensive. An often adopted compromise is to model a fixed set of frequent words while segmenting the rest into word pieces using byte-pair encoding (BPE) (Sennrich, Haddow, and Birch 2015). An equivalent approach which does not require tokenization is a sentence-piece model (Kudo 2018).

Sentence pieces are character ngrams, which are whole words at the maximum level and single characters at the minimum level. Segmentation can be performed using BPE or a unigram language model (with a unigram being a sentence piece). A sentence-piece model is trained on the data directly based on the frequencies. A unigram sentence-piece model can be used by picking the optimal segmentation, or by sampling from the space of all possible segmentations according to the unigram probabilities.

As an example, the phrase "argumentation mining" can be segmented in the following ways, ranked by the sampling likelihood. (Here _ denotes word boundary and space denotes segmentation.)

- `_argument ation _mining`
- `_argument at ion _mining`

Topic	Posts	Sentences (annotated)	Reason labels
Abortion	463	876	13
Gay Rights	560	865	9
Obama	446	651	16
Marijuana	432	846	10
Total	1901	3238	48

Table 1: Evaluation data.

- `_argument ation _ min ing`
- `_argument ation _mini ng`
- `_argument a tion _mining`
- ...

While the sampled segmentations do not always reflect the morphology, they are often informative and provide potentially unlimited variations from limited data. Training a model with sentence-piece sampling is an effective regularization technique and was shown by Kudo (2018) to improve the quality of machine translation.

3 Experiments

3.1 Datasets and preprocessing

Three datasets were involved in our experiments, one for evaluation and two for training.

We trained separate sentence-piece model on each set of training data. The vocabulary size was set to 8192 with a total character coverage of 99.95%. When sentence-piece sampling was applied, the smoothing parameter was set to $\alpha = 0.5$ with an unlimited sampling size.

Evaluation data

For evaluating the quality of the learned text representation, we used the stance and reason dataset assembled by Hasan and Ng (2014). The dataset consisted of online debate posts on four topics: *Abortion*, *Gay Rights*, *Obama*, and *Marijuana*. Each post was annotated with a binary stance in terms of pro and con and contained an arbitrary number of argumentative sentences

Topic	Reason	Sentence
Marijuana	c-health	Smoking a useless plant isn't "recreational".
Marijuana	p-medicine	I'm saying that during jazz's first creative explosion, weed was a major factor in the community. And yes, lowered stress is an excellent way to make better music.
GayRights	c-born	love has no gender so why should marriage.
GayRights	c-religion	I believe that homosexuallity is morally wrong and I don't want it in my country.

Table 2: Examples of questionable labeling.

annotated with labels for the reason of argumentation (Table 1). Accompanying the dataset was the partition information of five-fold cross-validation for stance and reason classification. This dataset was used in order to have comparable results to Boltužić and Šnajder (2015), but there are several problems within the dataset. Additional to having duplicate sentences with differed labels and occasional different stances within one post, the main issue is that often questionable choices are made regarding the labeling (Table 2).

For embedding these sentences and posts, we learned separate VAE models using different training data.

Training data for sentence modeling

For sentence representation learning, we used the *IBM Debater Claim Sentences Search* dataset (Levy et al. 2018). The dataset contained over 1.49 million sentences covering 150 topics, which were automatically retrieved from Wikipedia. We reserved a random subset of 4096 instances for validation, and used the rest for training. To avoid memory resource error, we only used instances with with no more than 256 sentence-piece segments.

We performed no text normalization on sentences. Comparing with the training data the evaluation data was irregular and ill-formed, with many instances of run-on sentences, spelling errors, and casing inconsistencies. We were aware that this disparity could be problematic, but could not conceive a practical solution. In the meanwhile, we were curious whether a model trained on clean data could handle noisy evaluation data.

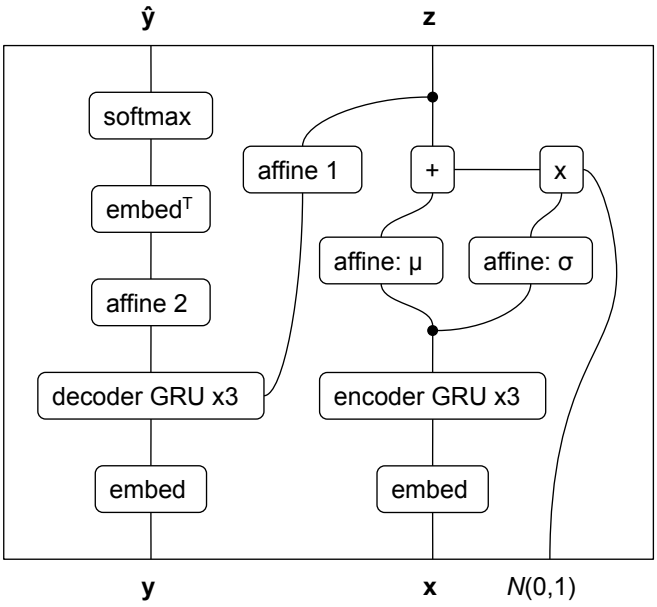

Figure 3: The architecture.

Training data for post modeling

For post representation learning, we used the *Internet Argument Corpus* containing over 390 thousand online posts (Walker et al. 2012). The characteristics of this dataset is much closer to the evaluation data. Due to the limited size of this dataset, we did not perform a split, and simply used the evaluation data for validation.

The main problem with training on whole posts was the long sequence lengths. We limited the number of sentence pieces to 512 segments, and truncated longer posts after sentence boundary segmentation. We also normalized whitespaces and lower-cased all characters.

3.2 Architecture and training

Our model architecture was a recurrent VAE as described in Section 2.2 and shown in Figure 3. The encoder consisted of a linear layer for sentence-piece embedding lookup, followed by three layers of stacked bidirectional RNNs with gated recurrent units (GRU) (Cho et al. 2014). The final recurrent state was connected to 2 affine layers in parallel for computing the mean and log variance of the latent variables. The latent vector was then computed

from the mean and variance with random noises during training. (Outside of training, the latent vector was simply the mean.) The decoder likewise consisted of a linear layer for sentence-piece embedding lookup, followed by three layers of unidirectional GRU RNNs receiving the latent vector after an affine layer as the initial recurrent states. The recurrent output was connected to an affine layer followed by a linear layer for predicting the logit probabilities over sentence pieces. The reconstruction error was softmax cross-entropy.

We applied input-output embedding sharing following Press and Wolf (2016). Specifically, the two linear input layers for the encoder and the decoder shared the same embedding matrix, and the linear output layer in the decoder used the transposed matrix. We found that input-output embedding sharing was only beneficial when we scaled the weight matrix by $\sqrt{d}$ for the input layers with d being the model dimension. All weights in the model were initialized according to Glorot and Bengio (2010).

The training procedure was stochastic gradient descent with Adam optimization (Kingma and Ba 2014). We monitored the reconstruction accuracy on the validation data for hyperparameter tuning. The optimal model dimension was 512 with the latent dimension doubled due to the bidirectional encoding. The learning rate was scheduled $lr/\left(1 + dr \times \sqrt{s}\right))$ by the training step s, where $lr = 0.001$ was the initial learning rate and $dr = 0.01$ the decay rate.

We encountered the common problem when using VAEs for language modeling tasks where the KL loss vanished right at the beginning of training. This happens when the model simply ignores the encoder and learns the decoder alone to be a language model. To avoid this problem, we applied KL term weighting and decoder word dropout following Bowman et al. (2015). The KL weight was scheduled $\tanh(dr^2 s)$ which started from 0 and increased to 1 after $4e5$ steps. The word dropout rate was scheduled $1/\left(1 + \exp(dr^2 s)\right)$ which started from 50% and reduced to 0% after $8e5$ steps.

The optimized setting was used for the training of all our models. We used mini-batches of 100 instances. For sentence modeling, we trained the model without sampling for around $2e5$ steps. The validation accuracy was over 95%. The sentence model did not benefit from sentence-piece sampling due to the large size of the dataset. We trained a model with sampling for twice as long which still under-performed. For post modeling, the model trained without sampling exploded after $9e5$ steps. The exact cause was uncertain. The post model trained with sampling did not converge after almost $4e6$ steps. The validation accuracy was still under 60%. However, we terminated training since it took 52.5 days and the rate of improvement

	Reason			Stance		
	Baseline	J3	VAE	J3	VAE	VAE-sampled
Abortion	32.7	**39.5**	34.4	66.3	56.75	56.53
Gay Rights	23.3	31.4	**34.8**	65.7	56.10	55.92
Obama	19.5	**25.1**	20.6	69.0	58.83	58.73
Marijuana	28.7	35.1	**36.0**	64.0	58.77	57.65

Table 3: Classification F-scores.

became increasingly slow.

3.3 Embedding sentences and posts

To embed a sentence or a post using a trained VAE model, we fed the sequence as input to the encoder and retrieved the latent representation. We used the model trained without sentence-piece sampling for sentence embedding, and the model trained with sampling for post embedding.

Since the model trained with sampling could receive the same text with different segmentations and produce different representations, we tried a novel approach. For each text, 128 segmentations were randomly sampled and the averaged output was taken as the latent representation. We hypothesized that this averaging method could produce a more robust representation by factoring away minor variations in the input forms through sampling.

In summary, we produced a 1024 dimensional embedding for each sentence and post. For each post, we additionally obtained a sampled embedding. These embedding vectors were then used for classification and clustering analysis.

4 Results

4.1 Classification results

We trained logistic classifiers with L2 regularization to compare against the results reported by Hasan and Ng (2014) using the same five-fold cross-validation split.

Sentence embedding for reason classification

Table 3 shows the reason classification results, compared against two models Baseline and J3 from Hasan and Ng (2014). The Baseline model was a logistic classifier based on ngram, dependency, frame-semantic, quotation, and positional features. The J3 model used joint density estimation (stance & reason) with reasons predicted for the preceding post, which was the best model reported by Hasan and Ng (2014). Our logistic classifier (cost = 0.001) using VAE embedding achieved better results on two topics, but performed worse on the other two.

Post embedding for stance classification

The stance classification results are also included in Table 3. Our logistic classifier (cost = 0.1) using VAE embedding performed significantly worse than J3, and did not benefit from the sampled representation. The post VAE model was under-trained. Comparing with the sentence VAE, the main difference was that the posts are much longer than the sentences, and the recurrent VAE could not reconstruct the sequences well due to the latent bottleneck.

Post embedding for topic classification

We also used the post embedding for topic classification on the five-fold cross-validation split for the stance dataset, although (Hasan and Ng 2014) did not report results for topic classification. Our logistic classifier (cost = 0.01) yielded an F-score of 84.6, using either sampled or unsampled representation.

4.2 Clustering analysis

We tried to follow the approach of Boltužić and Šnajder (2015) where the sentences were clustered using Hierarchical Agglomerative Clustering (HAC) and the cluster labels were matched against the reason labels to compute the Adjusted Rand Score (ARS) and the V-measure (V-MSR).

We trained a L1-regularized logistic classifier to predict the reason label from the sentence embedding. The regularization would force the classifier to assign non-zero weights to only the dimensions encoding the most relevant information for differentiating the reasons. We then removed the irrelevant dimensions from the embedding space and clustered the sentences. However, the clusters did not match the reason labels. The ARS and V-MSR were

Cluster	Sampling	ARS	V-MSR
Topic	no	0.056	0.118
	yes	0.065	0.130
Reason	no	0.010	0.587
	yes	0.009	0.587
Abortion	no	0.014	0.551
	yes	0.017	0.559
Gay Rights	no	0.009	0.378
	yes	0.011	0.374
Obama	no	0.012	0.560
	yes	0.013	0.555
Marijuana	no	0.026	0.534
	yes	0.037	0.535

Table 4: Clustering results (ARS and V-MSR) regarding topic, reason, and topic-wise reason. Post embedding was used with and without sentence-piece sampling.

not significantly different from zero. To investigate further, we turned to analyzing the post embedding.

Hierarchical Agglomerative Clustering

We used HAC to cluster the posts and evaluated the clusters against these labels: the topic, the reason, and the reason within each topic. The number of clusters was set according to the number of labels. Table 4 shows the results. The ARS was consistently low but the scores for the topic label were higher than the rest. The V-MSR for the topic was low, but significant. This was consistent with the good topic classification score we found. Figure 4 visualizes the post embedding space colored by topic after a dimensionality reduction to 2D using t-SNE (Maaten and Hinton 2008).

For all the reason clustering, the V-MSR score was fairly high. However, this was merely due to the number of clusters being high as a result of how we assigned the reason labels to posts. The reason was originally labeled for sentences, so we took the set of all reasons labeled present in a post as the reason label for that post. The combination considerably raised the number

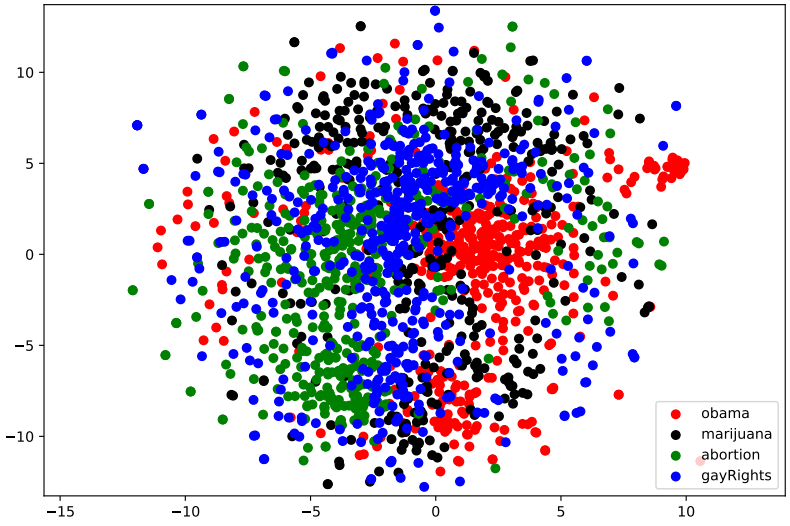

Figure 4: Post embedding reduced to 2D using t-SNE.

of labels and consequently the number of clusters. For example, the number of reason labels for the Obama topic changed form 16 to 101. As a result, there were many very small clusters and even singleton clusters. V-MSR was not suited for evaluating such a clustering result.

As the number of reason labels was high and the analysis was further complicated by the fact that the labeling is often quite questionable, we decided to focus on the stance and topic.

Affinity Propagation Clustering

To further explore the posting embedding, we used Affinity Propagation Clustering (APC). This clustering method considers all points to be potential exemplars and iteratively groups instances by passing messages between the datapoints until a optimal ordering is achieved (Frey and Dueck 2007). Therefore number of clusters is not fixed, but depends on the underlying structure of the data.

APC picked a considerably higher number of clusters compared to what we used for HAC (Table 5). We tried two distance measures, euclidean and cosine. The results were very different. When using cosine, the number

Dist	Sample	Clusters	Topic	Instances	Cl>50%	%
euc.	no	49	Abortion	463	136	29.4
			Gay Rights	560	179	31.9
			Obama	446	210	47.0
			Marijuana	432	104	24.1
	yes	63	Abortion	463	107	23.1
			Gay Rights	560	165	29.5
			Obama	446	210	47.1
			Marijuana	432	109	25.2
cos.	no	198	Abortion	463	418	90.3
			Gay Rights	560	425	75.9
			Obama	446	362	81.1
			Marijuana	432	386	89.4
	yes	199	Abortion	463	428	**92.4**
			Gay Rights	560	450	**80.4**
			Obama	446	362	81.1
			Marijuana	432	391	**90.5**

Table 5: APC results for two distance measures (euclidean and cosine) and two embedding methods (with and without sentence-piece sampling). The number of clusters and instances are listed. Additionally, the number and percentage of instances that are in clusters where over 50% of the instances are of the same topic are shown.

of clusters was 4–5 times higher. To explore how the clusters represented different topics, we analyzed for each topic how many of its instances were in clusters where they were the majority (>50%). Here, cosine provided much better results. This is understandable considering that the euclidean metric often does not work well in high dimensional spaces. The results also suggest that the embedding produced with sentence-piece sampling yielded better clusters.

5 Conclusion

VAEs are simple generative models for learning argumentation representation requiring only raw texts for training. The encoder could be used to embed texts for further discriminative modeling. Our sentence model performed relatively well on reason classification despite the disparity between the cleanliness of the training data and the messiness of the evaluation data. However, the same embedded representation did not work well when used for clustering.

Training a VAE on long sequences such as whole posts was difficult. Our under-trained post model did not produce adequate representation for stance classification. Further analysis via clustering showed that the learned latent space was locally arranged to some extent according to the topic of discussion. However, globally, the latent space was not simply separable by the topic or the reason of argumentation.

We attempted at charting the latent space by manually introspecting the clusters, which turned out to be extremely difficult. The clusters were not formed according to easily recognizable patterns. We did discover that some clusters contained posts in the same thread due to literal quoting of previous posts in a chain of discussion. Interestingly, within almost all the clusters, the shortest posts were the closest to the centroid. We tried generating posts from the centroids using the decoder with greedy decoding, but the generated texts were mostly lengthy ramblings with no coherent content.

Nevertheless we are optimistic about the research potential along this approach. The training method could be adjusted to incorporate the desired argumentation labeling. Take our sentence VAE for example, the reconstruction accuracy exceeded 95%, but for the purpose of argumentation mining, we are not as interested in the exact wording or the syntactic structure as we are in the semantic and pragmatic aspects. The VAE model could be trained with paraphrasing datasets to regularize the latent space for the desired abstraction. Apart from the usage of embedding, a well-trained VAE

has unparalleled prospects for argumentation generation due to the disentanglement of features in the latent representation.

References

Ackley, David H, Geoffrey E Hinton, and Terrence J Sejnowski (1985). "A learning algorithm for Boltzmann machines". In: *Cognitive science* 9.1, pp. 147–169.

Williams, Ronald J and David Zipser (1989). "A learning algorithm for continually running fully recurrent neural networks". In: *Neural computation* 1.2, pp. 270–280.

Hinton, Geoffrey E, Simon Osindero, and Yee-Whye Teh (2006). "A fast learning algorithm for deep belief nets". In: *Neural computation* 18.7, pp. 1527–1554.

Hinton, Geoffrey E and Ruslan R Salakhutdinov (2006). "Reducing the dimensionality of data with neural networks". In: *science* 313.5786, pp. 504–507.

Frey, Brendan J and Delbert Dueck (2007). "Clustering by passing messages between data points". In: *science* 315.5814, pp. 972–976.

Maaten, Laurens van der and Geoffrey Hinton (2008). "Visualizing data using t-SNE". In: *Journal of machine learning research* 9.Nov, pp. 2579–2605.

Glorot, Xavier and Yoshua Bengio (2010). "Understanding the difficulty of training deep feedforward neural networks". In: *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pp. 249–256.

Walker, Marilyn A et al. (2012). "A Corpus for Research on Deliberation and Debate." In: *LREC*. Istanbul, pp. 812–817.

Kingma, Diederik P and Max Welling (2013). "Auto-encoding variational bayes". In: *arXiv preprint arXiv:1312.6114*.

Mikolov, Tomas et al. (2013). "Efficient estimation of word representations in vector space". In: *arXiv preprint arXiv:1301.3781*.

Cho, Kyunghyun et al. (2014). "Learning phrase representations using RNN encoder-decoder for statistical machine translation". In: *arXiv preprint arXiv:1406.1078*.

Hasan, Kazi Saidul and Vincent Ng (2014). "Why are you taking this stance? identifying and classifying reasons in ideological debates". In: *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 751–762.

Kingma, Diederik P and Jimmy Ba (2014). "Adam: A method for stochastic optimization". In: *arXiv preprint arXiv:1412.6980*.

Kingma, Durk P, Shakir Mohamed, et al. (2014). "Semi-supervised learning with deep generative models". In: *Advances in neural information processing systems*, pp. 3581–3589.

Pennington, Jeffrey, Richard Socher, and Christopher Manning (2014). "Glove: Global vectors for word representation". In: *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, pp. 1532–1543.

Boltužić, Filip and Jan Šnajder (2015). "Identifying prominent arguments in online debates using semantic textual similarity". In: *Proceedings of the 2nd Workshop on Argumentation Mining*, pp. 110–115.

Bowman, Samuel R et al. (2015). "Generating sentences from a continuous space". In: *arXiv preprint arXiv:1511.06349*.

Dai, Andrew M, Christopher Olah, and Quoc V Le (2015). "Document embedding with paragraph vectors". In: *arXiv preprint arXiv:1507.07998*.

Sennrich, Rico, Barry Haddow, and Alexandra Birch (2015). "Neural machine translation of rare words with subword units". In: *arXiv preprint arXiv:1508.07909*.

Arora, Sanjeev, Yingyu Liang, and Tengyu Ma (2016). "A simple but tough-to-beat baseline for sentence embeddings". In:

Kalchbrenner, Nal et al. (2016). "Neural machine translation in linear time". In: *arXiv preprint arXiv:1610.10099*.

Press, Ofir and Lior Wolf (2016). "Using the output embedding to improve language models". In: *arXiv preprint arXiv:1608.05859*.

Freitag, Markus and Yaser Al-Onaizan (2017). "Beam search strategies for neural machine translation". In: *arXiv preprint arXiv:1702.01806*.

Srivastava, Akash and Charles Sutton (2017). "Autoencoding variational inference for topic models". In: *arXiv preprint arXiv:1703.01488*.

Zhao, Tiancheng, Ran Zhao, and Maxine Eskenazi (2017). "Learning discourse-level diversity for neural dialog models using conditional variational autoencoders". In: *arXiv preprint arXiv:1703.10960*.

Jang, Myeongjun, Seungwan Seo, and Pilsung Kang (2018). "Recurrent Neural Network-Based Semantic Variational Autoencoder for Sequence-to-Sequence Learning". In: *arXiv preprint arXiv:1802.03238*.

Kudo, Taku (2018). "Subword Regularization: Improving Neural Network Translation Models with Multiple Subword Candidates". In: *arXiv preprint arXiv:1804.10959*.

Levy, Ran et al. (2018). "Towards an argumentative content search engine using weak supervision". In: *Proceedings of the 27th International Conference on Computational Linguistics*, pp. 2066–2081.

Xu, Weidi and Ying Tan (2018). "Semi-supervised Target-level Sentiment Analysis via Variational Autoencoder". In: *arXiv preprint arXiv:1810.10437*.