Adversarial Decomposition of Text Representation

Alexey Romanov, Anna Rumshisky, Anna Rogers and David Donahue
Department of Computer Science
University of Massachusetts Lowell
Lowell, MA 01854
{aromanov,arum,arogers}@cs.uml.edu
david_donahue@student.uml.edu

Abstract

In this paper, we present a method for adversarial decomposition of text representation. This method can be used to decompose a representation of an input sentence into several independent vectors, where each vector is responsible for a specific aspect of the input sentence. We evaluate the proposed method on several case studies: the conversion between different social registers, diachronic language change and the decomposition of the sentiment polarity of input sentences. We show that the proposed method is capable of fine-grained controlled change of these aspects of the input sentence. The model uses adversarial-motivational training and includes a special motivational loss, which acts opposite to the discriminator and encourages a better decomposition. Finally, we evaluate the obtained meaning embeddings on a downstream task of paraphrase detection and show that they are significantly better than embeddings of a regular autoencoder.

1 Introduction

Despite the recent successes in using neural models for representation learning for natural language text, learning a meaningful representation of input sentences remains an open research problem. A variety of approaches, from sequence-to-sequence models that followed the work of (Sutskever et al., 2014) to the more recent proposals (Arora et al., 2017; Nangia et al., 2017; Conneau et al., 2017; Logeswaran and Lee, 2018; Subramanian et al., 2018; Cer et al., 2018) share one common drawback. Namely, all of them encode the input sentence into just *one* single vector of a fixed size. One way to bypass the limitations of a single vector representation is to use an attention mechanism (Bahdanau et al., 2014; Vaswani et al., 2017). We propose to approach this problem differently and design a method for adversarial decomposition of the learned input representation into multiple components. Our method encodes the input sentence into *several* vectors, where each vector is responsible for a specific aspect of the sentence.

In terms of learning different separable components of input representation, our work most closely relates to the style transfer work, which has been applied to a variety of different aspects of language, from diachronic language differences (Xu et al., 2012) to authors' personalities (Lipton et al., 2015) and even sentiment (Hu et al., 2017; Fu et al., 2017). The style transfer work effectively relies on the more classical distinction between **meaning** and **form** (de Saussure, 1959), which accounts for the fact that multiple surface realizations are possible for the same meaning. For simplicity, we will use this terminology throughout the rest of the paper. However, following previous work in style transfer, we also use the same methods to learn a separate representation for different aspects of meaning proper (e.g. different sentiment polarity).

Consider the case when we encode an input sentence into a meaning vector and a form vector. We are then able to perform a controllable change of meaning or form by a simple change applied to these vectors. For example, we can encode two sentences written in two different styles, then swap the form vectors while leaving the meaning vectors intact. We can then generate new unique sentences with the original meaning, but written in a different style.

In the present work, we propose a novel model for this type of decomposition based on adversarial-motivational training and design an architecture inspired by the GANs (Goodfellow et al., 2014) and adversarial autoencoders (Makhzani et al., 2015). In addition to the adversarial loss, we use a special motivator, which, in contrast to the discriminator, is used to provide a motivational loss (Albanie et al., 2017) to encourage the model to better decomposition of the meaning and the form, as well as specific aspects of meaning. We make all the code publicly available on GitHub [1] to facilitate further research in this direction.

We evaluate the proposed methods for learning separate aspects of input representation on the following case studies:

1. Learning to separate out a representation of the specific diachronic slice of language. One may express the same meaning using the Early Modern English (e.g. *We think not so.*) and the contemporary English (*We dont think so.*)

2. Learning a representation for a social register (Halliday et al., 1968) – that is, subsets of language appropriate in a given context or characteristic of a certain group of speakers. These include formal and informal language, the language used in different genres (e.g., fiction vs. newspapers vs. academic texts), different dialects, and even literary idiostyles. We experiment with the registers corresponding to the titles of scientific papers vs. newspaper articles.

3. Learning different aspects of meaning, and specifically, a separate representation for sentiment polarity.

2 Related work

As mentioned above, the most relevant previous work comes from the style transfer research, and it can be divided into two groups:

1. Approaches that aim to generate text in a given form. For example, the task may be to produce just any verse as long as it is in the "style" of the target poet.

2. Approaches that aim to induce a change in either the "form" or the "meaning" of an existing utterance. For example, "Good bye, Mr. Anderson." can be transformed to "Fare you well, good Master Anderson" (Xu et al., 2012)).

[1] http://github.com/placeholder

An example of the first group is the work by Potash et al. (2015), who trained several separate networks on verses by different hip-hip artists. An LSTM network successfully generated verses that were stylistically similar to the verses of the target artist (as measured by cosine distance on Tf-Idf vectors). More complicated approaches use language models that are conditioned in some way. For example, Lipton et al. (2015) produced product reviews with a target rating by passing the rating as an additional input at each timestep of an LSTM model. Tang et al. (2016) generated reviews not only with a given rating but also for a specific product. At each timestep a special context vector was provided as input, gated so as to enable the model to decide how much attention to pay to that vector and the current hidden state. Li et al. (2016) used "speaker" vectors as an additional input to a conversational model, improving consistency of dialog responses. Finally, Ficler and Goldberg (2017) performed an extensive evaluation of conditioned language models based on "content" (theme and sentiment) and "style" (professional, personal, length, descriptiveness). Importantly, they showed that it is possible to control both "content" and "style" simultaneously.

Work from the second group can further be divided into two clusters by the nature of the training data: parallel aligned corpora, or non-aligned datasets. The aligned corpora enable approaching the problem of form shift as a paraphrasing or machine translation problem. Xu et al. (2012) used statistical and dictionary-based systems on a dataset of original plays by Shakespeare and their contemporary translations. Carlson et al. (2017) trained an LSTM network on 33 versions of the Bible. Jhamtani et al. (2017) used a Pointer Network (Vinyals et al., 2015), an architecture that was successfully applied to a wide variety of tasks (Merity et al., 2016; Gulcehre et al., 2016; Potash et al., 2017), to enable direct copying of the input tokens to the output. Note that these works use BLEU (Papineni et al., 2002) as the main, or even the only evaluation measure. This is only possible in cases where a parallel corpus is available.

Recently, new approaches that do not require a parallel corpora were developed in both CV (Zhu et al., 2017) and NLP. (Hu et al., 2017) succeeded in changing tense and sentiment of sentences with a two steps procedure based on a variational autoencoder (VAE) (Kingma and Welling, 2013). After training a VAE, a discriminator and a generator are trained in an alternate manner, where the discriminator tries to correctly classify the target sentence attributes. A special loss component forces the hidden representation of the encoded sentence to not have any information about the target sentence attributes. Mueller et al. (2017) used a VAE to produce a hidden representation of a sentence, and then modify it to match the desired form. Unlike Hu et al. (2017), they do not separate the form and meaning embeddings. Shen et al. (2017) applied a GAN to align the hidden representation of sentences from two corpora and force them to do not have any information about the form via adversarial loss. During the decoding, similarly the work by Lipton et al. (2015), special "style" vectors are passed to the decoder at every timestep to produce a sentence with the desired properties. The model is trained using the Professor-Forcing algorithm (Lamb et al., 2016). Kim et al. (2017) worked directly on hidden space vectors that are constrained with the same adversarial loss instead of outputs of the generator, and use two different generators for two different "styles". Finally, Fu et al. (2017) proposed two models for generating sentences with the target properties using an adversarial loss, similarly to Shen et al. (2017) and Kim et al. (2017).

3 Formulation

Let us formulate the problem of decomposition of text representation on an example of controlled change of linguistic form and conversion of Shakespeare plays in the original Early Modern to contemporary English. Let $\boldsymbol{X}^a$ be a corpus of texts $\boldsymbol{x}_i^a \in \mathcal{X}^a$ in Early Middle English $\mathbf{f}^{\boldsymbol{a}} \in \mathcal{F}$, and $\boldsymbol{X}^b$ be a corpus of texts $\boldsymbol{x}_i^b \in \mathcal{X}^b$ in modern English $\mathbf{f}^{\boldsymbol{b}} \in \mathcal{F}$. We assume that the texts in both $\boldsymbol{X}^a$ and $\boldsymbol{X}^b$ has the same distribution of meaning $\mathbf{m} \in \mathcal{M}$. The form $\mathbf{f}$, however, is different and generated from a mixture of two distributions:

$$\mathbf{f}_i = \alpha_i^a p(\mathbf{f}^a) + \alpha_i^b p(\mathbf{f}^b)$$

where $\mathbf{f}^a$ and $\mathbf{f}^b$ are two different languages (Early Modern and contemporary English). Intuitively, we say that a sample $\boldsymbol{x}_i$ has the form $\mathbf{f}^{\boldsymbol{a}}$ if $\alpha_i^a > \alpha_i^b$, and it has the form $\mathbf{f}^{\boldsymbol{b}}$ if $\alpha_i^b > \alpha_i^a$.

The goal of dissociation meaning and form is to learn two encoders $E_{\mathbf{m}} : \mathcal{X} \to \mathcal{M}$ and $E_{\mathbf{f}} : \mathcal{X} \to \mathcal{F}$ for the meaning and form correspondingly, and the generator $G : \mathcal{M}, \mathcal{F} \to \mathcal{X}$ such that

$$\forall j \in \{a, b\}, \forall k \in \{a, b\} :$$
$$G(E_{\mathbf{m}}(\boldsymbol{x}^k), E_{\mathbf{f}}(\boldsymbol{x}^j)) \to \mathcal{X}^j$$

That is, the form of a generated sample depends exclusively on the provided $\mathbf{f}_j$ and can be the in the same domain for two different $\mathbf{m}_u$ and $\mathbf{m}_v$ from two samples from different domains $\mathcal{X}^a$ and $\mathcal{X}^b$.

Note that, in contrast to the previously proposals, the form $\mathbf{f}$ is not a categorical variable but a continuous vector. This enables fine-grained controllable change of form: the original form $\mathbf{f}_i$ is changed to reflect the form of the specific target sentence $\mathbf{f}_j$ with its own unique α^a and α^b while preserving the original meaning $\mathbf{m}_i$.

An important caveat concerns the core assumption of the similar meaning distribution in the two corpora, which is also made in all other works reviewed in Section 2. It limits the possible use of this approach to cases where the distributions are in fact similar (i.e. parallel or at least comparable corpora are available). It does not apply to many cases that could be analyzed in terms of meaning and form. For example, books for children and scholarly papers are both registers, they have their own form (i.e. specific subsets of linguistic means and structure conventions) – but there is little overlap in the content. This would make it hard even for a professional writer to turn a research paper into a fairy tale.

4 Method description

Inspired by Makhzani et al. (2015), Kim et al. (2017), and Albanie et al. (2017), we propose ADNet, a new model for adversarial decomposition of text representation (Figure 1).

Our solution is based on a widely used sequence-to-sequence framework (Sutskever et al., 2014) and consists of four main parts. The encoder E encodes the inputs sequence $\boldsymbol{x}$ into two latent vectors $\mathbf{m}$ and $\mathbf{f}$ which capture the meaning and the form of the sentence correspondingly. The generator G then takes these two vectors as the input and produces a reconstruction of the original input sequence $\hat{\boldsymbol{x}}$.

The encoder and generator by themselves will likely not achieve the dissociation of the meaning and form. We encourage this behavior in a

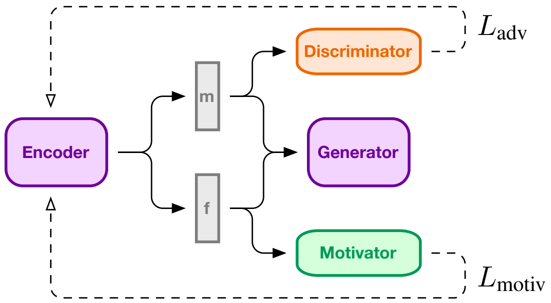


Figure 1: Overview of ADNet. *Encoder* encodes the inputs sentences into two latent vectors $\mathbf{m}$ and $\mathbf{f}$. The *Generator* takes them as the input and produces the output sentence. During the training, the *Discriminator* is used for an adversarial loss that forces $\mathbf{m}$ to do not carry any information about the form, and the *Motivator* is used for a motivational loss that encourages $\mathbf{f}$ to carry the needed information about the form.

way similar to Generative Adversarial Networks (GANs) (Goodfellow et al., 2014), which had an overwhelming success the past few years and have been proven to be a good way of enforcing a specific distribution and characteristics on the output of a model.

Inspired by the work of Albanie et al. (2017) and the principle of "carrot and stick" (Safire, 1995), in contrast to the majority of work that promotes pure adversarial approach (Goodfellow et al., 2014; Shen et al., 2017; Fu et al., 2017; Zhu et al., 2017), we propose two additional components, the discriminator D and the motivator M to force and motivate the model to learn the dissociation of the meaning and the form. Similarly to a regular GAN model, the adversarial discriminator D tries to classify the form $\mathbf{f}$ based on the latent meaning vector $\mathbf{m}$, and the encoder E is penalized to make this task as hard as possible.

Opposed to such vicious behaviour, the motivator M tries to classify the form based on the latent form vector $\mathbf{f}$, as it should be done, and encourages the encoder E to make this task as simple as possible. We could apply the adversarial approach here as well and force the distribution of the form vectors to fit a mixture of Gaussians (in this particular case, a mixture of two Guassians) with another discriminator, as it is done by Makhzani et al. (2015), but we opted for the "dualistic" path of two complimentary forces.

4.1 Encoder-Decoder

Both the encoder E and the generator G are modeled with a neural network. Gated Recurrent Unit (GRU) (Chung et al., 2014) is used for E to encode the input sentence $\boldsymbol{x}$ into a hidden vector $\boldsymbol{h} = \text{GRU}(\boldsymbol{x})$.

The vector $\boldsymbol{h}$ is then passed through two different fully connected layers to produce the latent vectors of the form and the meaning of the input sentence:

$$\mathbf{m} = \tanh(\boldsymbol{W_m}\boldsymbol{h} + \boldsymbol{b}_m)$$
$$\mathbf{f} = \tanh(\boldsymbol{W_f}\boldsymbol{h} + \boldsymbol{b}_f)$$

We use $\boldsymbol{\theta}_E$ to denote the parameters of the encoder E: $\boldsymbol{W_m}$, $\boldsymbol{b}_m$, $\boldsymbol{W_f}$, $\boldsymbol{b}_f$, and the parameters of the GRU unit.

The generator G is also modelled with a GRU unit. The generator takes as input the meaning vector $\mathbf{m}$ and the form vector $\mathbf{f}$, concatenates them, and passes trough a fully-connected layer to obtain a hidden vector $\boldsymbol{z}$ that represents both meaning and form of the original input sentence:

$$\boldsymbol{z} = \tanh(\boldsymbol{W}_z[\mathbf{m}; \mathbf{f}] + \boldsymbol{b}_m)$$

After that, we use a GRU unit to generate the output sentence as a probability distribution over the vocabulary tokens:

$$p(\hat{\boldsymbol{x}}) = \prod_{t=1}^{T} p(\hat{\boldsymbol{x}}_t | \boldsymbol{z}, \hat{\boldsymbol{x}}_1, \ldots, \hat{\boldsymbol{x}}_{t-1})$$

We use $\boldsymbol{\theta}_G$ to denote the parameters of the generator G: $\boldsymbol{W}_z$, $\boldsymbol{b}_m$, and the parameters of the used GRU. The encoder and generator are trained using the standard reconstruction loss:

$$\begin{aligned}\mathcal{L}_{\text{rec}}(\boldsymbol{\theta}_E, \boldsymbol{\theta}_G) = &\mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{X}^a}[-\log p(\hat{\boldsymbol{x}}|\boldsymbol{x})] + \\ &\mathbb{E}_{\boldsymbol{x} \sim \boldsymbol{X}^b}[-\log p(\hat{\boldsymbol{x}}|\boldsymbol{x})]\end{aligned}$$

4.2 Discriminator

The representation of the meaning $\mathbf{m}$ produced by the encoder E should not contain any information about the form $\mathbf{f}$. We achieve this by using an adversarial approach. First, we train a discriminator D, consisting of several fully connected layers with $\texttt{ELU}$ activation function (Clevert et al., 2015) between them, to predict the form $\mathbf{f}$ of a sentence by its meaning vector: $\hat{\boldsymbol{f}}_D = D(\mathbf{m})$, where $\hat{\boldsymbol{f}}$ is the score (logit) reflecting the probability of the sentence $\boldsymbol{x}$ to belong to one of the form domains.

Motivated by the Wasserstein GAN (Arjovsky et al., 2017), we use the following loss function instead of the standard cross-entropy:

$$\mathcal{L}_D(\boldsymbol{\theta}_D) = \mathbb{E}_{\boldsymbol{x}\sim \boldsymbol{X}^a}[D(E_{\mathbf{m}}(\boldsymbol{x}))] - \mathbb{E}_{\boldsymbol{x}\sim \boldsymbol{X}^b}[D(E_{\mathbf{m}}(\boldsymbol{x}))]$$

Thus, a successful discriminator will produce negative scores $\hat{\boldsymbol{f}}$ for sentences from $\boldsymbol{X}^a$ and positive scores for sentences from $\boldsymbol{X}^b$. This discriminator is then used in an adversarial manner to provide a learning signal for the encoder and force dissociation of the meaning and form by maximizing $\mathcal{L}_D$: $\mathcal{L}_{\text{adv}}(\boldsymbol{\theta}_E) = -\lambda_{\text{adv}}\mathcal{L}_D$, where λ_{adv} is a hyperparameter reflecting the strength of the adversarial loss. Note that this loss applies to the parameters of the encoder.

4.3 Motivator

Our experiments showed that it is enough to have just the discriminator D and the adversarial loss $\mathcal{L}_{\text{adv}}$ to force the model to dissociate the form and the meaning. However, in order to achieve a better dissociation, we propose to use a motivator M (Albanie et al., 2017) and the corresponding motivational loss. Conceptually, this is the opposite of the adversarial loss, hence the name. As the discriminator D, the motivator M learns to classify the form $\mathbf{f}$ of the input sentence. However, its input is not not the meaning vector but the form vector: $\hat{\boldsymbol{f}}_M = M(\mathbf{f})$.

The motivator has the same architecture as the discriminator, and the same loss function. While the adversarial loss forces the encoder E to produce a meaning vector $\mathbf{m}$ with no information about the form $\mathbf{f}$, the motivational loss encourages E to encode this information in the form vector by minimizing $\mathcal{L}_M$: $\mathcal{L}_{\text{motiv}}(\boldsymbol{\theta}_E) = \lambda_{\text{motiv}}\mathcal{L}_M$.

4.4 Training procedure

The overall training procedure follows the methods for training GANs (Goodfellow et al., 2014; Arjovsky et al., 2017) and consists of two stages: training the discriminator D and the motivator M, and training the encoder E and the generator G.

In contrast to Arjovsky et al. (2017), we do not train the D and M more than the E and the G. In our experiments we found that simple training in two stages is enough to achieve dissociation of the meaning and the form. Encoder and generator are trained with the following loss function that combines reconstruction loss with the losses from the discriminator and the motivator:

$$\mathcal{L}_{\text{total}}(\theta_E, \theta_G) = \mathcal{L}_{\text{rec}} + \mathcal{L}_{\text{adv}} + \mathcal{L}_{\text{motiv}}$$

4.5 Comparison with previous work

In contrast to the proposals of Xu et al. (2012), Carlson et al. (2017), Jhamtani et al. (2017), our solution does not require a parallel corpus. Furthermore, unlike the model by Shen et al. (2017), our model works directly on representation of sentences in the hidden space.

Most importantly, in contrast to the proposals by Mueller et al. (2017), Hu et al. (2017), Kim et al. (2017), Fu et al. (2017), our model produces a representation for both meaning and form and does not treat the form as a categorical (in the vast majority of works, binary) variable. Although the form was represented as dense vectors in previous works, it is still just a binary feature, as they use a single pre-defined vector for each form, with all sentences of the same form assigned the same form vector. In contrast, our work treats form as a truly continuous variable, where each sentence has its own, unique, form vector. To the best of our knowledge, this is the first model that considers linguistic form in the task of text generation as a continuous variable.

5 Experimental setup

5.1 Evaluation

Similarly to the evaluation of style transfer in CV (Isola et al., 2017), evaluation of this task is difficult. We follow the approach of Isola et al. (2017); Shen et al. (2017) and recently proposed by Fu et al. (2017) methods of evaluation of "transfer strength" and "content preservation". The authors showed the proposed automatic metrics to a large degree correlate with human judgment and can serve as a proxy. Below we give an overview of these metrics.

Transfer Strength. The goal of this metric is to capture whether the form has been changed successfully. To do that, a classifier C is trained on the two corpora, $\boldsymbol{X}^a$ and $\boldsymbol{X}^b$ to recognize the linguistic "form" typical of each of them. After that, a sentence the form/meaning of which was changed is passed to the classifier. The overall accuracy reflects the degree of success of changing the form/meaning. This approach is widely used in CV (Isola et al., 2017), and was applied in NLP as well (Shen et al., 2017).

In our experiments we used a GRU unit followed by four fully-connected layers with ELU activation functions between them as the classifier.

Content preservation Note that transfer strength by itself does not capture the overall quality of a changed sentence. A extremely overfitted model that produces the same, the most characteristic sentence of one corpus all the time would have a high score according to this metric. Thus, we need to measure how much of the meaning was preserved while changing the form. To do that, Fu et al. (2017) proposed to use a cosine similarity based metric using pretrained word embeddings. First, a sentence embedding is computed by concatenation of max, mean, and average pooling over the timesteps:

$$\begin{aligned}\boldsymbol{v} = [\,&\max(\boldsymbol{v}_1, \dots, \boldsymbol{v}_T); \min(\boldsymbol{v}_1, \dots, \boldsymbol{v}_T);\\ &\mathrm{mean}(\boldsymbol{v}_1, \dots, \boldsymbol{v}_T)]\end{aligned}$$

Next, the cosine similarity score s_i between the embedding $\boldsymbol{v}_i^s$ of the original source sentence and the target sentence with the changed form $\boldsymbol{v}_i^t$ is computed, and the scores across the dataset are averaged to obtain the total score:

$$\mathrm{s} = \frac{1}{2}\left[\frac{1}{|\boldsymbol{X}^a|}\sum_{i=1}^{|\boldsymbol{X}^a|}\mathrm{s}_i + \frac{1}{|\boldsymbol{X}^b|}\sum_{i=1}^{|\boldsymbol{X}^b|}\mathrm{s}_i\right]$$

5.1.1 Continuous form

Note that the metrics described above treat the form as a categorical (in most cases, even binary) variable. This was not a problem in previous work since the change of form could be done by just inverting the form vector. Our work, in contrast, treats the form as a continuous variable, and, therefore, we cannot just use the proposed metrics directly. To enable a fair comparison we propose the following procedure.

For each sentence s_s^a in the test set from the corpus $\boldsymbol{X}^a$ we sample ten random sentence from the corpus $\boldsymbol{X}^b$. We then generate ten corresponding sentences with the form $\boldsymbol{f}_j$ and the meaning $\boldsymbol{m}_s$ and apply the classifier C to estimate the transfer strength. If the change of the form was successful in strictly more than half of the cases (six of more), we count this a successful change of form for this sentence s_s^a. The scores of content preservation are simply averaged across the ten generated sentences. This process enables a fair comparison with the previous works that treat form as a binary variable.

5.2 Datasets

We performed an extensive evaluation of the proposed method on several dataset that reflect different changes of meaning, form, or specific aspects of meaning, such as sentiment polarity.

Changing form: register This experiment is conducted with a dataset of titles of scientific papers and news articles published by Fu et al. (2017). This dataset (referred to as "Headlines") contains titles of scientific articles crawled from online digital libraries, such as "ACM Digital Library" and "arXiv". The titles of the news articles are taken from the "News Aggregator Data Set" from UCI Machine Learning Repository (Dheeru and Karra Taniskidou, 2017)

Changing form: language diachrony Diachronic language change is explored with the dataset composed by Xu et al. (2012). It includes the texts of 17 plays by William Shakespeare in the original Early Modern English, and their translations into contemporary English. We randomly permuted all sentences from all plays and sampled the training, validation, and test sets. Note that this is the smallest dataset in our experiments.

Changing specific aspect of meaning: review tonality Change of this aspect of meaning is tested with two sentiment analysis datasets: a collection of Yelp restaurant reviews (Shen et al., 2017) and a collection of product reviews on Amazon by Fu et al. (2017). Note that these are sentence-level datasets, and their construction assumes that all sentences in a review share the same sentiment. To enforce this property, Shen et al. (2017) filtered out reviews that contain more than 10 sentences, and sentences that are more than 15 words long.

6 Results and discussion

Probably, the most recent and similar to our work is the model proposed by Fu et al. (2017), in particular the "style-embedding" model. We implemented this model to provide a baseline for comparison.

The classifier used in the transfer strength metric achieves very high accuracy (0.991, 0.802, 0.971, and 0.807 for the Headlines, Shakespeare,

Dataset	Model	Transfer Strength	Content Preservation
Shakespeare	Fu et al. (2017)	0.381	0.963
	ADNet	0.590	0.838
	ADNet + Motivator	0.587	0.850
Headlines	Fu et al. (2017)	0.623	0.881
	ADNet	0.999	0.783
	ADNet + Motivator	0.997	0.787
Yelp	Fu et al. (2017)	0.166	0.989
	ADNet	0.906	0.815
	ADNet + Motivator	0.935	0.813
Amazon	Fu et al. (2017)	0.252	0.988
	ADNet	0.603	0.865
	ADNet + Motivator	0.729	0.852

Table 1: The results of the evaluation on different datasets.

Yelp, and Amazon datasets correspondingly). These results concur with the results of Shen et al. (2017) and Fu et al. (2017), and show that the two source corpora are significantly different. They also illustrate the complexity of the problem that discriminator and motivator face.

The results of our experiments, evaluated as described in subsection 5.1, are shown in Table 1. It is clear that the proposed method achieves significantly better transfer strength then the previously proposed model[2]. It also has a lower content preservation score, which means that it repeats fewer exact words from the source sentence. Note that a low transfer strength and very high (0̃.99) content preservation score means that the model was not able to successfully learn to transfer the form and the target sentence is almost identical to the source sentence. The Shakespeare dataset is the hardest for the model in terms of transfer strength, probably because it is the smallest dataset, but the proposed method performs consistently well in transfer of both form and meaning and, in contrast to the baseline, it achieves transfer strength over 50% on all datasets.

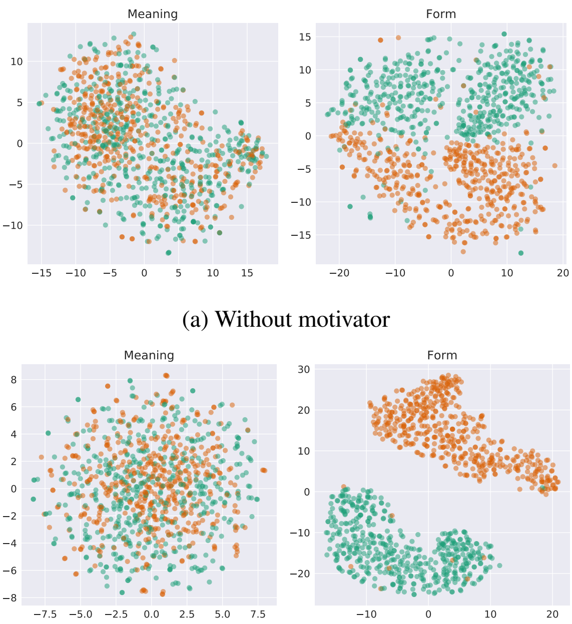
6.1 Impact of the motivational training

To investigate the impact of the motivator, we visualized form and meaning embeddings of 1000 random samples from the Headlines dataset using t-SNE algorithm (Van Der Maaten, 2014) with the Multicore-TSNE library (Ulyanov, 2016). The result is presented in Figure 2.

There are three important observations. First, there is no clear separation in the meaning embeddings, which means that any accurate form transfer is due to the form embeddings, and the dissociation of form and meaning was successful.

Second, even without the motivator the model is able to produce the form embeddings that are clustered into two group. Recall from section 4 that without the motivational loss there are no forces that influence the form embeddings, but nevertheless the model learns to separate them.

However, the separation effect is much more pronounced in the presence of motivator. This explains why the motivator consistently improved transfer strength of ADNet, as shown in Table 1.



(b) With motivator

Figure 2: t-SNE visualization of the form and meaning embeddings of 1000 random sentences. Green point represent sentences form news headlines, and red points represent titles of scientific articles.

6.2 Qualitative evaluation

Table 3 and Table 2 show several examples of the successful form/meaning transfer achieved by ADNet. Table 3 presentes the results of an experiment that to some extent replicates the approach taken by the authors who treat linguistic form as a

[2] Note that the size of the "style" embeddings, as well as the hidden size of the RNN, used in the model of Fu et al. (2017), significantly impacts the results. In our experiments we saw the same behaviour as the authors: the bigger is the style embeddings, the higher is transfer strength and the lower is content preservation. We chose thus the form embeddings of size 64, as a median size that balances these two metrics, and the same size of the RNN as in the rest of our models.

This restaurant is horrible. Thank you soo much .	→	I love so much. This place is horrible.
Otherwise the customer service was terrible. Also, their spray tans are amazing!	→	Also the customer spray service are amazing! However, their service was horrible

Table 2: Flipping the meaning and the form embeddings of two sentence from different sentiment classes.

Aye, sir. (EME)	→	Yes, sir. (CE)
Fare thee well, my lord (EME)	→	Fare you well, my lord (CE)
This guy will tell us everything. (CE)	→	This man will tell us everything. (EME)
I've done no more to caesar than you will do to me. (CE)	→	I have done no more to caesar than, you shall do to me. (EME)

Table 3: Decoding of the source sentence from Early Modern English (EME) into contemporary English (CE), and vice versa.

binary variable (Shen et al., 2017; Fu et al., 2017). The sentences the original Shakespeare plays were averaged to get the "typical" Early Modern English form vector. This averaged vector was used to decode a sentence from the modern English translation back into the original. The same was done in the opposite direction.

Table 2 illustrates the possibilities of ADNet on fine-grained transfer with regards to sentiment polarity. Two sentences with different polarity from the Yelp dataset were encoded into form and meaning embeddings, and then we decoded the first sentence with the meaning embeddings of the second, and vice verse. As can be seen from Table 2, the model correctly captures the meaning of sentences and decodes them using the form of the source sentences. Note how the model preserves specific words and the structure of the source sentence. This is not possible in the previously proposed models, as they treat form as just a binary variable.

Note that the transfer does not always result in coherent or grammatical sentences, highlighting the importance on further work on scalable qualitative evaluation methods for text generation.

6.3 Performance of meaning embeddings on downstream tasks

We conducted some experiments to test the assumption that the derived meaning embeddings should improve performance on downstream tasks that require understanding of the meaning of the sentences regardless of their form. We evaluated embeddings produced by the ADNet, trained in the Headlines dataset, on a task of paraphrase detection. We used the SentEval toolkit (Conneau et al., 2017) and the Microsoft Research Paraphrase Corpus (Dolan et al., 2004). The F1 scores on this task for different models are presented in Table 4. Note that all models, except InferSent, are unsupervised. The InferSent model was trained on a big SNLI dataset, consisting of more than 500,000 manually annotated pairs. ADNet achieves the the highest score among the unsupervised systems and outperforms the regular sequence-to-sequence autoencoder with a large gap.

BoW	Seq2Seq	InferSent	Fu et al. (2017)	ADNet
80.82	74.68	**83.17**	78.88	81.38

Table 4: F1 scores on the task of paraphrase detection using the SentEval toolkit (Conneau et al., 2017)

7 Conclusion

In this paper, we presented ADNet, a new model that performs adversarial decomposition of text representation. In contrast to previous work, it does not require a parallel training corpus and works directly on hidden representations of sentences. Most importantly, is does not treat the form as a binary variable (as done in most previously proposed models), enabling a fine-grained change of the form of sentences or specific aspects of meaning. We evaluate ADNet on three tasks: the shift of language register, diachronic language change, and reversing sentiment polarity. Our solution achieves superior results, and t-SNE visualizations of the learned meaning and style embeddings illustrate that the proposed motivational loss leads to significantly better separation of the form embeddings.

References

Samuel Albanie, Sébastien Ehrhardt, and João F Henriques. 2017. Stopping gan violence: Generative unadversarial networks. *arXiv preprint arXiv:1703.02528*.

Martin Arjovsky, Soumith Chintala, and Léon Bottou. 2017. Wasserstein gan. *arXiv preprint arXiv:1701.07875*.

Sanjeev Arora, Yingyu Liang, and Tengyu Ma. 2017. A simple but tough-to-beat baseline for sentence embeddings. In *International Conference on Learning Representation*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Keith Carlson, Allen Riddell, and Daniel Rockmore. 2017. Zero-shot style transfer in text using recurrent neural networks. *arXiv preprint arXiv:1711.04731*.

Daniel Cer, Yinfei Yang, Sheng-yi Kong, Nan Hua, Nicole Limtiaco, Rhomni St John, Noah Constant, Mario Guajardo-Cespedes, Steve Yuan, Chris Tar, et al. 2018. Universal sentence encoder. *arXiv preprint arXiv:1803.11175*.

Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

Djork-Arné Clevert, Thomas Unterthiner, and Sepp Hochreiter. 2015. Fast and accurate deep network learning by exponential linear units (elus). *arXiv preprint arXiv:1511.07289*.

Alexis Conneau, Douwe Kiela, Holger Schwenk, Loïc Barrault, and Antoine Bordes. 2017. Supervised learning of universal sentence representations from natural language inference data. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 670–680.

Dua Dheeru and Efi Karra Taniskidou. 2017. UCI machine learning repository.

Bill Dolan, Chris Quirk, and Chris Brockett. 2004. Unsupervised construction of large paraphrase corpora: Exploiting massively parallel news sources. In *Proceedings of the 20th international conference on Computational Linguistics*, page 350. Association for Computational Linguistics.

Jessica Ficler and Yoav Goldberg. 2017. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, pages 94–104.

Zhenxin Fu, Xiaoye Tan, Nanyun Peng, Dongyan Zhao, and Rui Yan. 2017. Style transfer in text: Exploration and evaluation. *arXiv preprint arXiv:1711.06861*.

Ian Goodfellow, Jean Pouget-Abadie, Mehdi Mirza, Bing Xu, David Warde-Farley, Sherjil Ozair, Aaron Courville, and Yoshua Bengio. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, pages 2672–2680.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 140–149.

M. A. K. Halliday, A. McIntosh, and P. Stevens. 1968. *The Linguistic Sciences and Language Teaching*. Longmans, Green and Co., London.

Zhiting Hu, Zichao Yang, Xiaodan Liang, Ruslan Salakhutdinov, and Eric P Xing. 2017. Toward controlled generation of text. In *International Conference on Machine Learning*, pages 1587–1596.

Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. 2017. Image-to-image translation with conditional adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1125–1134.

Harsh Jhamtani, Varun Gangal, Eduard Hovy, and Eric Nyberg. 2017. Shakespearizing modern language using copy-enriched sequence-to-sequence models. *arXiv preprint arXiv:1707.01161*.

Yoon Kim, Kelly Zhang, Alexander M Rush, Yann LeCun, et al. 2017. Adversarially regularized autoencoders for generating discrete structures. *arXiv preprint arXiv:1706.04223*.

Diederik P Kingma and Max Welling. 2013. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*.

Alex M Lamb, Anirudh Goyal ALIAS PARTH GOYAL, Ying Zhang, Saizheng Zhang, Aaron C Courville, and Yoshua Bengio. 2016. Professor forcing: A new algorithm for training recurrent networks. In *Advances In Neural Information Processing Systems*, pages 4601–4609.

Jiwei Li, Michel Galley, Chris Brockett, Georgios Spithourakis, Jianfeng Gao, and Bill Dolan. 2016. A persona-based neural conversation model. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, volume 1, pages 994–1003.

Zachary C Lipton, Sharad Vikram, and Julian McAuley. 2015. Generative concatenative nets jointly learn to write and classify reviews. *arXiv preprint arXiv:1511.03683*.

Lajanugen Logeswaran and Honglak Lee. 2018. An efficient framework for learning sentence representations. *arXiv preprint arXiv:1803.02893*.

Alireza Makhzani, Jonathon Shlens, Navdeep Jaitly, Ian Goodfellow, and Brendan Frey. 2015. Adversarial autoencoders. *arXiv preprint arXiv:1511.05644*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Jonas Mueller, David Gifford, and Tommi Jaakkola. 2017. Sequence to better sequence: continuous revision of combinatorial structures. In *International Conference on Machine Learning*, pages 2536–2544.

Nikita Nangia, Adina Williams, Angeliki Lazaridou, and Samuel R Bowman. 2017. The repeval 2017 shared task: Multi-genre natural language inference with sentence representations. *arXiv preprint arXiv:1707.08172*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2015. Ghostwriter: using an lstm for automatic rap lyric generation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 1919–1924.

Peter Potash, Alexey Romanov, and Anna Rumshisky. 2017. Here's my point: Joint pointer architecture for argument mining. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1364–1373.

William Safire. 1995. On Language – Gotcha! Gang Strikes Again.

Ferdinand de Saussure. 1959. *Course in General Linguistics*. New York : Philosophical Library.

Tianxiao Shen, Tao Lei, Regina Barzilay, and Tommi Jaakkola. 2017. Style transfer from non-parallel text by cross-alignment. In *Advances in Neural Information Processing Systems*, pages 6833–6844.

Sandeep Subramanian, Adam Trischler, Yoshua Bengio, and Christopher J Pal. 2018. Learning general purpose distributed sentence representations via large scale multi-task learning. *arXiv preprint arXiv:1804.00079*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Jian Tang, Yifan Yang, Sam Carton, Ming Zhang, and Qiaozhu Mei. 2016. Context-aware natural language generation with recurrent neural networks. *arXiv preprint arXiv:1611.09900*.

Dmitry Ulyanov. 2016. Multicore-tsne. https://github.com/DmitryUlyanov/Multicore-TSNE.

Laurens Van Der Maaten. 2014. Accelerating t-sne using tree-based algorithms. *Journal of machine learning research*, 15(1):3221–3245.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Oriol Vinyals, Meire Fortunato, and Navdeep Jaitly. 2015. Pointer networks. In *Advances in Neural Information Processing Systems*, pages 2692–2700.

Wei Xu, Alan Ritter, Bill Dolan, Ralph Grishman, and Colin Cherry. 2012. Paraphrasing for style. *Proceedings of COLING 2012*, pages 2899–2914.

Jun-Yan Zhu, Taesung Park, Phillip Isola, and Alexei A Efros. 2017. Unpaired image-to-image translation using cycle-consistent adversarial networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2223–2232.