Unsupervised Learning of Discourse Structures using a Tree Autoencoder

Patrick Huber, Giuseppe Carenini
Department of Computer Science, University of British Columbia
Vancouver, BC, Canada
{huberpat, carenini}@cs.ubc.ca

Abstract

Discourse information, as postulated by popular discourse theories, such as RST and PDTB, has been shown to improve an increasing number of downstream NLP tasks, showing positive effects and synergies of discourse with important real-world applications. While methods for incorporating discourse become more and more sophisticated, the growing need for robust and general discourse structures has not been sufficiently met by current discourse parsers, usually trained on small scale datasets in a strictly limited number of domains. This makes the prediction for arbitrary tasks noisy and unreliable. The overall resulting lack of high-quality, high-quantity discourse trees poses a severe limitation to further progress. In order the alleviate this shortcoming, we propose a new strategy to generate tree structures in a task-agnostic, unsupervised fashion by extending a latent tree induction framework with an auto-encoding objective. The proposed approach can be applied to any tree-structured objective, such as syntactic parsing, discourse parsing and others. However, due to the especially difficult annotation process to generate discourse trees, we initially develop a method to generate larger and more diverse discourse treebanks. In this paper we are inferring general tree structures of natural text in multiple domains, showing promising results on a diverse set of tasks.

Introduction

Discourse Parsing is a key Natural Language Processing (NLP) task for processing multi-sentential text. Most research in the area focuses on one of the two main discourse theories – RST (Mann and Thompson 1988) or PDTB (Prasad et al. 2008). The latter thereby postulates shallow discourse structures, combining adjacent sentences and mainly focuses on explicit and implicit discourse connectives. The RST discourse theory, on the other hand, proposes discourse trees over complete documents in a constituency-style manner, with tree leaves as so called Elementary Discourse Units (or EDUs), representing span-like sentence fragments. Internal tree-nodes encode discourse relations between sub-trees as a tuple of {Nuclearity, Relation}, where the nuclearity defines the sub-tree salience in the local context, and the relation further specifies the type of relationship between the binary child nodes (e.g. Elaboration)[1]. While both discourse theories are of great value to the field of NLP, and have stimulated much progress in discourse parsing, there are major drawbacks when data is annotated according to these theories:

(1) Since both theories rely on annotation-guidelines rather than data-driven algorithms, the human factor plays a substantial role in generating treebanks, posing a difficult task on linguistic experts. In this work, we are eliminating the human component from the annotation process by employing a data-driven approach to generate discourse trees directly from natural language, capturing commonly occurring phenomena in an unsupervised manner.

(2) The annotation process following human-generated guidelines, especially following the RST discourse theory, is expensive and tedious, as the annotation itself requires linguistic expertise and a full understanding of the complete document. This limits available RST-style discourse corpora in both, size and number of domains where gold-standard datasets exist. Using an automated, data-driven approach as described in this paper allows us to crucially expand the size and domain-coverage of datasets annotated with RST-style discourse structures.

With the rapidly growing need for robust and general discourse structures for many downstream tasks and real-world applications (e.g. Gerani et al. (2014); Nejat, Carenini, and Ng (2017); Ji and Smith (2017); Xiao, Huber, and Carenini (2020); Huber and Carenini (2020a)), the current lack of high-quality, high-quantity discourse treebanks poses a severe shortcoming.

Fortunately, more data-driven alternatives to infer discourse structures have been previously proposed. For example, our recently published MEGA-DT discourse treebank (Huber and Carenini 2020b) with automatically inferred discourse structures and nuclearity attributes from large-scale *sentiment* datasets already reached state-of-the-art (SOTA) performance on the inter-domain discourse parsing task. Similarly, Liu and Lapata (2018) infer latent discourse trees from the *text classification* task, and Liu, Titov, and Lapata (2019) employ the downstream task of *summarization* using a transformer model to generate discourse trees. Outside the area of discourse parsing, syntactic trees have previously

[1] We only generate plain discourse structures in this work, not considering nuclearity and relation labels.

been inferred according to several strategies, e.g. Socher et al. (2011); Yogatama et al. (2016); Choi, Yoo, and Lee (2018); Maillard, Clark, and Yogatama (2019).

In general, the approaches mentioned above have shown to capture valuable structural information. Some models outperform baselines trained on human-annotated datasets (see Huber and Carenini (2020b)), others have proven to enhance diverse downstream tasks (Liu and Lapata 2018; Liu, Titov, and Lapata 2019; Choi, Yoo, and Lee 2018). However, despite these initial successes, one critical limitation that all aforementioned models share is the task-specificity, possibly only capturing downstream-task related information. This potentially compromises the generality of the resulting trees, as for instance shown for the model using *text classification* data (Liu and Lapata 2018) in Ferracane et al. (2019). In order to alleviate this limitation of task-specificity, we propose a new strategy to generate tree structures in a task-agnostic, unsupervised fashion by extending the latent tree induction framework proposed by Choi, Yoo, and Lee (2018) with an auto-encoding objective. Our system thereby extracts important knowledge from natural text by optimizing both the underlying tree structures and the distributed representations. We believe that the resulting discourse structures effectively aggregate related and commonly appearing patterns in the data by merging coherent text spans into intermediate subtree encodings, similar to the intuition presented in Drozdov et al. (2019). However, in contrast to the approach by Drozdov et al. (2019), our model makes discrete structural decisions, rather than joining possible subtrees using a soft attention mechanism. We believe that our discrete tree structures allow the model to more efficiently achieve the autoencoder objective in reconstructing the inputs, directly learning how written language can be aggregated in the wild (comparable to previous work in language modelling (Jozefowicz et al. 2016)). In general, the proposed approach can be applied to any tree-structured objective, such as syntactic parsing, discourse parsing and further problems outside of NLP, like tree-planning (Guo et al. 2014) and decision-tree generation (Irsoy and Alpaydin 2016). Yet, due to the especially difficult annotation process to generate discourse trees, we initially develop a method to generate much larger and more diverse discourse treebanks.

Related Work

Within the last decade, general autoencoder frameworks have been frequently used to compress data, such as in Srivastava, Mansimov, and Salakhudinov (2015). More recently, sequential autoencoders have been applied in the area of NLP (Li, Luong, and Jurafsky 2015), with many popular approaches, such as sequence-to-sequence learning models (Sutskever, Vinyals, and Le 2014) having strong ties to sequential autoencoders. Based on the promising results of the sequential autoencoder, researchers started to compress and reconstruct more general structures in tree-style models, such as Chen, Liu, and Song (2018) showing that with available gold-standard trees, the programming-language translation task (e.g. from CoffeeScript to JavaScript) can be learned with a tree-to-tree style neural autoencoder network. Furthermore, variational autoencoders have been shown effective for the difficult task of grammar induction (Kusner, Paige, and Hernández-Lobato 2017).

While both previously mentioned applications for tree-style autoencoder models require readily available tree structures to guide the aggregation process, another line of work by Socher et al. (2011) overcomes this requirement by using the reconstruction error of an autoencoder applied to every two adjacent text spans as an indicator for syntactic correctness within a sentence. In their model, Socher et al. (2011) combine the tree-inference objective with the autoencoder topology, training an unsupervised tree-structured model, which is subsequently fine-tuned on a small-scale supervised dataset. While their model is clearly comparable to our approach, there are three major differences: (1) They make sequential, local decisions on the aggregation of spans to generate a tree structure, rather than optimizing the complete process holistically. (2) Their model uses an unsupervised objective in the initial step but requires supervision in later stages and (3) The model has been only applied to syntactic parsing. In contrast, we apply our model to discourse parsing, which arguably introduces further difficulties, as we will discuss later.

Recently, Choi, Yoo, and Lee (2018) showed a promising approach to infer tree structures in a holistic and parallelizable manner, generating task-depended trees solely relying on sentiment-related information. In their model, they make use of the Gumbel-Softmax (Jang, Gu, and Poole 2016) (also used in similar ways in Corro and Titov (2018, 2019)), allowing the neural network to make discrete decisions while still being able to use standard approaches like back-propagation to optimize the model. By combining a similar objective to Socher et al. (2011) and Chen, Liu, and Song (2018), we utilize the discrete decision-process in Choi, Yoo, and Lee (2018), positioning our work at the intersection of these two lines of research. The general task of tree inference has been mostly explored on sentence-level. For instance in Choi, Yoo, and Lee (2018) and Socher et al. (2011) as described above, or by applying a reinforcement approach (Yogatama et al. 2016) or CKY methodology (Maillard, Clark, and Yogatama 2019) to syntactic parsing. Our work employs a novel, fully differentiable approach to a similar problem in the area of discourse parsing.

In discourse parsing itself, there have been multiple attempts to overcome the aforementioned limitations of small-scale human annotated datasets. However, all previous models (such as Liu and Lapata (2018); Huber and Carenini (2019); Liu, Titov, and Lapata (2019); Huber and Carenini (2020b)) use downstream tasks to infer discourse structures. While this is a valid strategy, shown to achieve SOTA results on the inter-domain discourse parsing task (Huber and Carenini 2020b), as well as performance gains on downstream tasks (e.g., Liu and Lapata (2018); Liu, Titov, and Lapata (2019)), those discourse structures are likely task-depended and need to be either combined across multiple downstream tasks or can only be applied in similar domains. Further work has been trying to infer RST-style discourse structures in a linguistically supervised manner (Nishida and Nakayama 2020), showing good performance when heavily exploiting syntactic markers in combination with general

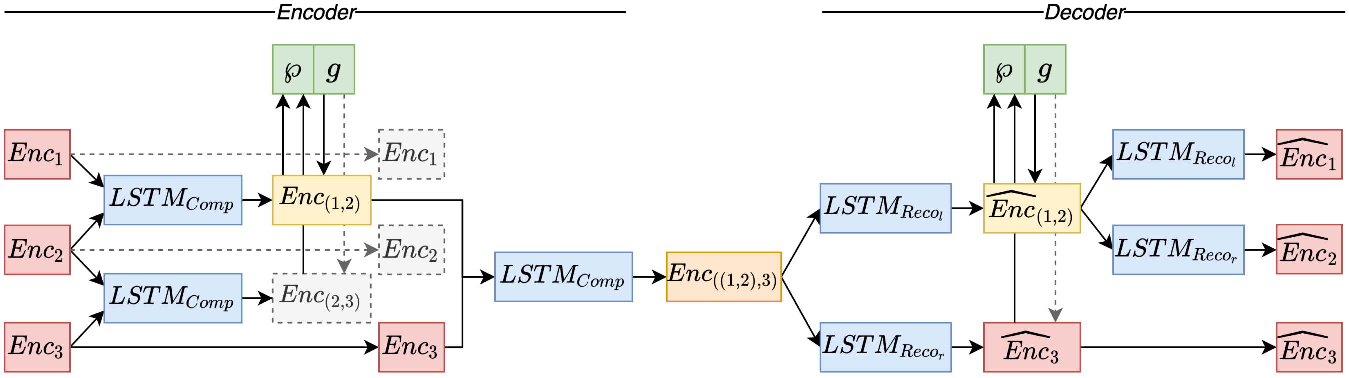

Figure 1: T-AE (Tree-AutoEncoder) topology for unsupervised tree inference. Inputs and outputs are dense encodings, Comp = Compression, Reco = Reconstruction, $\widehat{Enc}_x$ represents reconstruction of spans. $\wp$ represents the pointer-network, $g \sim G(0,1)$ denotes the Gumbel-softmax (in the forward-pass with an additional straight-through computation, not shown here). Grey /Dashed components represent actions outside the computational path chosen. red = model inputs/outputs, blue = TreeLSTM cells, green = discrete structure selector as in Choi, Yoo, and Lee (2018), yellow = hidden subtree encodings, orange = hidden state of the complete input.

linguistic priors. Yet, the approach appears to be very specific to the data at hand – News articles from the Wall Street Journal – raising questions in regards to overfitting.

In this work, we explore a purely unsupervised approach: instead of relying on domain specific syntactic features, we infer general discourse trees (structure only) by exploiting inherently available information from natural data (not requiring any supervision), making our model similar to approaches in language modelling (Jozefowicz et al. 2016). More specifically, our proposal extends the previously proposed Gumbel-TreeLSTM method (Choi, Yoo, and Lee 2018) by substituting the original downstream-task related objective with an autoencoder-style reconstruction.

Unsupervised Tree Autoencoder

We now outline our general tree autoencoder model. The description is thereby purposely general, as the model is independent of a specific application and we believe can be utilized in manifold scenarios.

Generally speaking, our proposed model induces tree structures through compression and reconstruction of raw inputs in a tree autoencoder style architecture. The model is similar in spirit to the commonly used sequence-to-sequence (seq2seq) architecture (Sutskever, Vinyals, and Le 2014), which has also been interpreted as a sequential autoencoder (Li, Luong, and Jurafsky 2015). However, our approach generalizes on the seq2seq model, which is essentially a special (left-branching) case of a tree-structured autoencoder. While the sequential structure of a document is naturally given by the order of words, EDUs, and sentences, moving towards more general tree representations adds the additional difficulty to infer valid tree structures alongside the hidden states. To generate these discrete tree structures during training, in conjunction with the hidden states of the neural network, we make use of the Gumbel-softmax decision framework, allowing us to discretely generate tree-aggregations alongside intermediate sub-tree encodings (Gumbel 1948; Maddison, Tarlow, and Minka 2014; Jang, Gu, and Poole 2016). As presented in Figure 1, the structure of our novel *T-AE* (**T**ree-**A**uto**E**ncoder) model comprises of an encoder, compressing the input into a fixed-size hidden vector and a subsequent decoder component, reconstructing the inputs in an autoencoder-style fashion.

Encoder Component

The computational steps performed in our encoder are akin to the approach described in Choi, Yoo, and Lee (2018), computing a single document encoding through a tree-style aggregation procedure. Our approach generates a hidden state $Enc_{l,r} = [c_p, h_p] = LSTM_{Compress}(l, r)$ for every two adjacent input embeddings $l = [c_l, h_l]$ (left) and $r = [c_r, h_r]$ (right) using a binary TreeLSTM cell as proposed by Tai, Socher, and Manning (2015)[2].

$$\begin{bmatrix} i \\ f_l \\ f_r \\ o \\ u \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \cdot (W \begin{bmatrix} h_l \\ h_r \end{bmatrix} + b)$$
$$c_p = f_l \cdot c_l + f_r \cdot c_r + i \cdot u$$
$$h_p = o \cdot tanh(c_p) \qquad (1)$$

With $W \in \mathbb{R}^{5|h_p| \times 2|h_p|}$ and $b \in \mathbb{R}^{2|h_p|}$. Based on the $(n-1)$ sub-tree candidates $Enc_{l,r}$ with $0 \leq l < (n-1)$ and $r = l + 1$ of the given inputs I ($|I| = n$), an unnormalized attention computation (or pointer network) $\wp = Pointer(\cdot, \cdot)$ (Vinyals, Fortunato, and Jaitly 2015) is used

[2] Equation 1 is modified from Choi, Yoo, and Lee (2018) and Tai, Socher, and Manning (2015).

to predict which two adjacent units should be merged. Randomly uniform Gumbel noise, obtained from the Gumbel distribution $G(0,1)$, effectively sampling $g \sim G(0,1)$ as $g_i = -log(-log(u_i))$ and $u_i = Uniform(0,1)$ is added to the un-normalized scores. Subsequently, the scores are normalized across aggregation candidates according to the temperature coefficient τ to obtain $p(l,r)$ (see equation 2).

$$p(l,r) = \frac{exp[(\wp(l,r)+g)/\tau]}{\sum_{k=0}^{n-1} exp[(\wp(I_k, I_{k+1})+g)/\tau]} \tag{2}$$

In the forward pass, the straight-through (ST) Gumbel-distribution is used to enforce a discrete selection p_{st}, as commonly done using the Gumbel-softmax trick (see equation 3 (Jang, Gu, and Poole 2016; Choi, Yoo, and Lee 2018; Corro and Titov 2018, 2019)).

$$p_{st}(l,r) = \begin{cases} 1, & \text{if } \underset{k=0,\ldots,n-2}{\arg\max}\ p(I_k, I_{k+1}) = l \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

Given this one-hot encoding for a set of aggregation candidates, the most appropriate aggregation, as predicted by the pointer component and pertubed with the Gumbel-softmax, is executed. All other inputs with $p_{st} = 0$ are directly forwarded to the next step and the respective TreeLSTM computations are discarded (grey/dashed boxes in Figure 1). In the example shown in Figure 1, Enc$_1$ and Enc$_2$ are aggregated, while Enc$_3$ is directly forwarded to the next step without any aggregation computation.

We recursively generate $n-1$ tree-candidates using the TreeLSTM cell in conjunction with the pointer-component and the Gumbel-softmax to build a discrete tree in bottom-up fashion, along with sub-tree hidden states[3]. Once the tree is aggregated, a single hidden-state represents the complete input. Given this dense hidden-state (orange in Fig. 1), Choi, Yoo, and Lee (2018) add a multi-layer-perceptron (MLP) to predict the sentence-level sentiment on the Stanford Sentiment Treebank (SST) (Socher et al. 2013). As a result, the obtained tree structures are mostly task-dependent, as shown in Williams, Drozdov, and Bowman (2018). With the goal to generate task-independent structures, we replace the task-dependant MLP layer with our autoencoder objective to reconstruct the original inputs.

Decoder Component

Besides similar pointer network and Gumbel softmax components as used in the encoder, the decoder component is implemented as an inverse TreeLSTM containing two independent LSTM cells, recursively splitting hidden states into two separate encodings to reconstruct the left and right child-node states (c_l, h_l and c_r, h_r) for a given parent node (c_p, h_p), as shown in equation 4).

$$\begin{bmatrix} i_l \\ f_l \\ o_l \\ u_l \\ i_r \\ f_r \\ o_r \\ u_r \end{bmatrix} = \begin{bmatrix} \sigma \\ \sigma \\ \sigma \\ tanh \\ \sigma \\ \sigma \\ \sigma \\ tanh \end{bmatrix} \cdot (Wh_p + b)$$
$$\begin{aligned} c_l &= f_l \cdot c_p + i_l \cdot u_l \\ c_r &= f_r \cdot c_p + i_r \cdot u_r \\ h_l &= o_l \cdot tanh(c_l) \\ h_r &= o_r \cdot tanh(c_r) \end{aligned} \tag{4}$$

With $W \in \mathbb{R}^{8|h_p| \times |h_p|}$ and $b \in \mathbb{R}^{|h_p|}$. Guided by the predicted tree structure of the ST Gumbel-softmax, as shown in Figure 1 and equations 2 and 3, the structural decision process in the reconstruction phase selects the highest scoring node to be further subdivided into a local sub-tree. This reconstruction approach, generating two child-node encodings given the parent encoding $Enc_p \rightarrow [Enc_l, Enc_r]$ is recursively applied top-down until the original number of inputs $|I| = n$ is reached. Finally, the reconstructed dense encodings $[\widehat{Enc}_1, ..., \widehat{Enc}_n]$ are evaluated against the model input encodings, following the autoencoder objective.

Discourse Tree Generation

The T-AE approach described above has been kept deliberately general. In this section, we outline the application-specific extensions required in order to deal with the inputs, assumptions, and granularity of the discourse parsing task. First, for the task of discourse parsing, the model inputs I are clause-like EDUs, representing sentence fragments containing multiple words. While the input- and output-encodings for word-level autoencoders are naturally represented as the respective one-hot vectors of words in the vocabulary, this approach is not directly applicable for discourse parsing. Hence, we encode the EDUs as dense representations and execute the autoencoder objective directly on these embeddings (Press and Wolf 2016). Second, as discourse parsing considers complete documents, frequently containing a large number of sentences with oftentimes diverse content, we apply a commonly used approach in this area by separating within-sentence and between-sentence sub-trees (Joty, Carenini, and Ng 2015). In this setup, we apply the model described above for each sentence individually, trying to infer general patterns on sentence-level and subsequently using the learned sentence encodings (orange in Figure 1) as the starting point of the document-level T-AE. Having two separate models on sentence- and document-level further aligns with previous work in discourse parsing, postulating different sets of features relevant on different levels of the tree-generation process (Joty, Carenini, and Ng 2015; Wang, Li, and Wang 2017).

[3] Please note that the computation of the hidden states in the TreeLSTM cell and the tree structure prediction using the pointer-network with Gumbel pertubation are non-overlapping, allowing for independent optimization of either component.

Evaluation

Tasks

To fully evaluate the performance of our T-AE method, we conduct experiments on three distinct tasks, focusing on the two learning goals of our model: **(1)** Evaluating if the model is able to infer valuable and general discourse-structures and **(2)** Assessing the ability of the model to learn task-independent hidden states, capturing important relationships between instances. The three tasks are:

Alignment with existing RST-style Discourse Structures: Proposing an unsupervised approach to generate (discourse) tree structures allows us, in principle, to generate trees in any domain with sufficient raw text for training. However, due to the expensive and tedious annotation of gold-standard discourse trees, only very few datasets in some narrow domains are augmented with full RST-style trees, required to evaluate our generated structures. Despite their limited coverage, we believe that comparing the discourse-structures produced by our newly proposed model on the alignment with human-annotated discourse structures can be insightful.

Ability to Predict Important Downstream Tasks: Besides evaluating the overlap with existing, human-annotated discourse trees, we investigate into the generality of the T-AE model by evaluating the performance when applied to an important downstream task in NLP – sentiment analysis. We therefore use the document-level hidden state of our model (orange in Figure 1), trained on the unsupervised autoencoder objective, and add a single feed-forward neural network layer on top, reducing the hidden state of our model to the number of sentiment classes required for the sentiment prediction task. Training this linear combination on top of the model's document-level encoding gives further insight into the information contained in the hidden state and its alignment with the downstream task of sentiment analysis.

General Representational Consistency: In this third task, we further explore the information captured by the document-level hidden state by qualitatively comparing the dense encoding of a random (short) sample document with its most similar/most different documents, giving intuition about the relatedness of similarly encoded documents.

Datasets

The RST-DT Treebank published by Carlson, Marcu, and Okurowski (2003) is the most popular RST treebank. It contains 385 documents of the Wall Street Journal (WSJ) corpus, split into 344 documents in the training-set and 39 documents in the test-portion. In order to obtain a development set, we subdivide the training-portion into 308 documents for training and 36 documents for a length-stratified development set. Each document in the RST-DT treebank is annotated with a complete discourse tree according to the RST discourse theory and segmented into EDUs by human annotators. N-ary subtrees are converted into a sequence of right-branching constituents.

The Yelp'13 Dataset by Tang, Qin, and Liu (2015) is a review dataset published as part of the 2013 Yelp Dataset Challenge. The corpus contains predominantly restaurant reviews alongside a 5-point star rating. Frequently used in previous work, the dataset has been pre-segmented into EDUs by Angelidis and Lapata (2018), using the discourse-segmenter proposed in Feng and Hirst (2012). The complete dataset contains 335,018 documents in an 80-10-10 data-split, resulting in 268,014 training documents and 33,502 documents each in the development and test sets.

Baselines

We compare our new model against a task-dependent set of baselines. For the **Alignment with RST-style Discourse Structures**, we evaluate three sets of related approaches: For supervised models, we compare against a diverse set of previously proposed, fully supervised discourse parsers, trained and evaluated on the RST-DT dataset. These include the CODRA model by Joty, Carenini, and Ng (2015), the Two-Stage approach by Wang, Li, and Wang (2017) and the neural topology by Guz, Huber, and Carenini (2020). We further compare against our two distantly supervised models recently proposed in Huber and Carenini (2019, 2020b), using sentiment analysis to inform the generation of discourse structures with distant supervision. Our last set of baselines for this task contains linguistically supervised approaches. We compare our model against fully left- and right-branching trees, as well as hierarchically left- and right-branching tree structures (separated on sentence-level), encoding basic rhetorical strategies. Left-branching trees generally reflect a common sequential strategy while right-branching tree structures oftentimes accurately represent documents where the main objective is initially expressed and then further evaluated throughout the document (e.g. news)[4]. For these reasons, we consider the left- and right-branching tree structures as linguistically supervised approaches. We further show the recently proposed model by Nishida and Nakayama (2020) in our evaluation. In their best model setting, Nishida and Nakayama (2020) also heavily exploit basic rhetorical strategies of natural language by aggregating a document into right-branching trees on sentence- and paragraph-level and joining paragraphs using left-branching constituents. Starting from this linguistically inspired tree (already achieving remarkable performance on the well-structured news documents), they apply a Viterbi EM algorithm to achieve further improvements. Despite the promising results on RST-DT, we believe that such high performance is mostly due to the well-structured nature of news documents and not generally applicable to other domains – the main objective of our presented approach.

Building on the intuition given in Huber and Carenini (2019, 2020a,b), we further evaluate our model regarding the **Ability to Predict Important Downstream Tasks**. More precisely, we evaluate the sentiment prediction performance of the document-level hidden-state of our model against the HAN model proposed by Yang et al. (2016), the LSTM-GRNN approach by Tang, Qin, and Liu (2015) as well as a document encoding build from average random word encodings and the majority class baseline.

[4] Right-branching trees are further artificially favoured in discourse parsing, since most parsing models convert n-ary sub-trees into a sequence of right-branching constituents.

Model	Structure
Human (2017)	88.30
Supervised	
CODRA(2015)	83.84
Two-Stage(2017)	86.00
Neural-SR(2020)	**86.47**
Distantly Supervised	
Two-Stage$_{\text{Yelp13-DT}}$(2019)	76.41
Two-Stage$_{\text{MEGA-DT}}$(2020b)	**77.82**
Linguistically Supervised	
Left Branching	53.73
Right Branching	54.64
Hier. Left Branching	70.58
Hier. Right Branching	74.37
ViterbiEM(2020)	**84.30**
Unsupervised	
Ours$_{\text{RST-DT}}$	69.68
Ours$_{\text{Yelp'13}}$	**71.32**

Table 1: Results of the average micro-precision measure, evaluated on the RST-DT corpus. Subscripts identify training sets. Best model in each subset is bold.

Model	Accuracy
HAN(2016)	**66.20**
LSTM-GRNN(2015)	65.10
Ours$_{\text{Yelp'13}}$	42.69
Ours$_{\text{RST-DT}}$	40.41
Random Encoding	37.30
Majority Class	35.63

Table 2: Five-class sentiment accuracy scores trained and tested on the Yelp'13 dataset, subscripts in model-names indicate dataset for unsupervised training. Best model is bold.

Hyper-Parameters Settings

We select our hyper-parameters based on the development-set performance of the respective datasets. Despite the fact that we are training two unsupervised models (on RST-DT and Yelp'13) we use a single set of hyper-parameters, to be more general. We train all models using the Adam optimizer (Kingma and Ba 2014) with the standard learning rate of 0.001. As mentioned before, we are directly training on dense representations of input EDU embeddings, comparing them to the reconstructed representations of EDUs. This setup makes the Kullback-Leibler Divergence (KLD) or the Mean-Squared-Error (MSE) the natural choice for the loss function. In this work we employ MSE due to its superior performance observed on the development-set. Each EDU in the input document is represented as the average GloVe word-embedding (Pennington, Socher, and Manning 2014) as in Choi, Yoo, and Lee (2018). The loss is computed on the softmax of the respective inputs and outputs. We train our model on mini-batches of size 20, due to computational restrictions[5] and apply regularization in form of 20% dropout on the input embeddings, the document-level hidden state and the output embeddings (Choi, Yoo, and Lee 2018). We clip gradients to a max norm of 2.0 to avoid exploding gradients. Documents are limited to 150 EDUs per document and a maximum of 50 words per EDU, similar to Huber and Carenini (2019). We restrict the vocabulary size to the most frequent $50,000$ words with an additional minimal frequency requirement of 10. We train the sentence- and document-level model for 40 epochs and select the best performing generation on the development set. The hidden dimension of our LSTM modules as well as the pointer component is set to 64, due to computational restrictions. To avoid our model to interfere with the input GloVe embeddings, we freeze the word representations. To promote consistency between the encoding and decoding, we tie the decoder tree-decisions to the encoder predictions, enabling a more consistent tree-embedding in the compression and reconstruction phase. Furthermore, to disentangle the optimization of structures and hidden states, we apply a phased approach, alternating the training of the two components in a conditional back-propagation loop with a single objective in each pass over the data (see footnote 3). This way, the hidden states are recalculated based on the last epoch's structure prediction and vice-versa. To be able to explore diverse tree candidates in early epochs and further improve them during later epochs, we start with the diversity factor $\tau = 5$ and linearly reduce the parameter to $\tau = 1$ (see Choi, Yoo, and Lee (2018)) over 3 structure-learning epochs.

Experiments

In this section we evaluate our novel T-AE model on the three tasks described above. Table 1 shows the results on the first task, evaluating our model on RST-style discourse structures from the RST-DT treebank. The first sub-table shows three top-performing, completely supervised models, reaching a structure-prediction performance of 86.47% using the neural approach by Guz, Huber, and Carenini (2020). In comparison, the second sub-table contains our distantly supervised models, achieving a performance of 77.82% (Huber and Carenini 2020b). The third sub-table presents the linguistically supervised models, showing a clear advantage of the right-branching models over left-branching approaches, in line with our intuition given above. Furthermore, considering sentence boundaries and generating hierarchical baselines significantly improves the performance, reaching 74.37% with the hierarchical right branching baseline and 70.58% on the left-branching structures. The linguistically supervised Viterbi EM approach by Nishida and Nakayama (2020) reaches a performance of 84.30% with their multi-level hierarchical approach. Our newly proposed, truly unsupervised and purely data-driven approach is shown in the fourth sub-table. In comparison to the aforementioned linguistically supervised models, this set of results makes no assumptions on the underlying data except the sentence/document split. When trained on the raw-text of the small-scale RST-DT dataset, our T-AE approach reaches a performance of 69.68%, slightly below the linguistically supervised hierarchical left-branching model. Even though the

[5] Trained on a Nvidia GTX 1080 Ti GPU with 11GB of memory.

Document	Prices were cheap, however food was served well after others who came in and they literally put brown gravy on the Mexican food, staff ignored simple requests. Only reason for 1 star was due to price.
Similar-1	This establishment has a good 10$ lunch special with plenty of varity in the bento they offer, service is usually good polite and efficient the only thing that makes me crazy is the crappy usually too loud caned pop music they play.
Similar-2	The good: Awesome complimentary breakfasts, warm gooey chocolate chip cookie at check in, nice pool and and hot tub at the center, fairly large room with 2 TVs (flat screen) and a huge comfortable bed with down pillows. The bad: Not a bad fitness room but could be larger, it doesn't have the feel look of a fancy hotel at first. More like a Motel (but the rooms are nice and the restaurant too). The ugly: No free internet
Similar-3	Just the facts: Great options for healthier eating, unique non-meat sandwich options at lunch (portabello, grilled zucchini, black bean, etc.), decent coffee, cute atmosphere and fun s&p shakers at the table, kind of pricey. I want to go back to try breakfast.
Different-1	Forgot to mention the prices are great & just had the baklava yummm to die for delicious!
Different-2	Bit pricey, but it's always been our favorite place to go for treats.
Different-3	Decent place, but the drinks are too expensive unless its a buy 1 get 1 night.

Table 3: Representationally similar/different document-encodings based on the cosine similarity. For more examples of the representational similarity and additional tree structure comparisons, please check out the arXiv version of our paper.

unsupervised training corpus is within the same domain as the test dataset, the very limited amount of data seems insufficient for the unsupervised model. Training our model on the nearly three orders of magnitude larger Yelp'13 dataset, we reach a performance of 71.32% evaluating the tree structures on RST-DT. This result shows that a larger training dataset, even though containing out-of-domain documents (reviews vs. news), can improve the performance over the within-domain model trained on a small-scale dataset and the hierarchical left-branching model.

To evaluate the ability of our model to capture valid information to represent input documents, we assess the document-level hidden state's ability to capture useful information for the downstream task of sentiment analysis. The results of this experiment are provided in Table 2, showing the accuracy of our models when compared against commonly used approaches. The best system (the HAN model) reaches an accuracy of 66.2%, while the random baseline reaches 37.30% and the simple majority class baseline achieves 35.63%. Our models based on the T-AE hidden states obtain accuracy scores in-between those results, reaching 40.41% and 42.69% when trained on RST-DT and the much larger Yelp'13 respectively. While this performance is still far from the results of completely supervised models, the improvements over the simple baselines suggest the usefulness of our learned document-level encodings.

In our third and last experiment, we aim to further evaluate the quality of the document encodings in a qualitative manner. We therefore compare the hidden-state of a random document from the Yelp'13 test-set against all datapoints in the test-portion and show the three most similar/most different documents according to the cosine similarity measure in Table 3. It can be observed that closely related documents have a similar argumentative structure as the core-document (top row in Table 3), initially describing a positive aspect and subsequently evaluating on negative components. The most different documents tend to have an inverse structure.

Conclusion and Future Work

In this paper, we proposed a truly unsupervised and purely data-driven tree-style autoencoder to compress and reconstruct textual data. We show the potential of our T-AE approach on the task of discourse parsing, which severely suffers from training-data sparsity, due to the tedious and expensive annotation process. Our unsupervised model outperforms one of the commonly used, linguistically supervised approaches, without making any assumptions on the underlying data, except the sentence/document split. The superior performance compared to the hierarchical left branching baseline plausibly indicates that our unsupervised structures could be valuable when combined with supervised or distantly supervised models to further improve their joint performance. Furthermore, the superior performance of the large out-of-domain model trained on the Yelp'13 dataset over the small-scale within-domain model trained on the raw text of the RST-DT dataset shows the synergies between these corpora as well as strong potential for even larger datasets to enhance the performance of the approach.

In the future, we intend to extend this work in several ways: First, we want to explore the application of generative models, employing a variational autoencoder. Second, we plan to study further tasks besides predicting discourse, such as syntactic parsing, as well as additional synergistic downstream tasks (e.g. summarization, text classification). To improve our model on important downstream tasks (such as sentiment analysis), we want to explore a pre-training/fine-tuning approach, similar to contextualized language models, such as BERT. Combining our novel approach with distantly-supervised and supervised models is another future direction we want to explore. Lastly, we plan to evaluate additional model adaptions, such as two independent models on sentence- and document-level, incorporating a BERT EDU encoder and an end-to-end model with soft-constraints on sentence-level.

Acknowledgments

We thank the anonymous reviewers, Linzi Xing and the UBC-NLP group for their comments and suggestions. This research was supported by the Language & Speech Innovation Lab of Cloud BU, Huawei Technologies Co., Ltd. We further acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC). Nous remercions le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG) de son soutien.

References

Angelidis, S.; and Lapata, M. 2018. Multiple Instance Learning Networks for Fine-Grained Sentiment Analysis. *Transactions of the Association for Computational Linguistics* 17–31.

Carlson, L.; Marcu, D.; and Okurowski, M. E. 2003. Building a discourse-tagged corpus in the framework of rhetorical structure theory. In *Current and new directions in discourse and dialogue*, 85–112.

Chen, X.; Liu, C.; and Song, D. 2018. Tree-to-tree neural networks for program translation. In *Advances in neural information processing systems*, 2547–2557.

Choi, J.; Yoo, K. M.; and Lee, S.-g. 2018. Learning to compose task-specific tree structures. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Corro, C.; and Titov, I. 2018. Differentiable perturb-and-parse: Semi-supervised parsing with a structured variational autoencoder. *arXiv preprint arXiv:1807.09875* .

Corro, C.; and Titov, I. 2019. Learning latent trees with stochastic perturbations and differentiable dynamic programming. *arXiv preprint arXiv:1906.09992* .

Drozdov, A.; Verga, P.; Yadav, M.; Iyyer, M.; and McCallum, A. 2019. Unsupervised Latent Tree Induction with Deep Inside-Outside Recursive Auto-Encoders. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 1129–1141.

Feng, V. W.; and Hirst, G. 2012. Text-level discourse parsing with rich linguistic features. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 60–68.

Ferracane, E.; Durrett, G.; Li, J. J.; and Erk, K. 2019. Evaluating discourse in structured text representations. *arXiv preprint arXiv:1906.01472* .

Gerani, S.; Mehdad, Y.; Carenini, G.; Ng, R. T.; and Nejat, B. 2014. Abstractive summarization of product reviews using discourse structure. In *Proceedings of the 2014 conference on empirical methods in natural language processing (EMNLP)*, 1602–1613.

Gumbel, E. J. 1948. *Statistical theory of extreme values and some practical applications: a series of lectures*. US Government Printing Office.

Guo, X.; Singh, S.; Lee, H.; Lewis, R. L.; and Wang, X. 2014. Deep learning for real-time Atari game play using offline Monte-Carlo tree search planning. In *Advances in neural information processing systems*, 3338–3346.

Guz, G.; Huber, P.; and Carenini, G. 2020. Unleashing the Power of Neural Discourse Parsers - A Context and Structure Aware Approach Using Large Scale Pretraining. In *Proceedings of the 28th International Conference on Computational Linguistics*, 3794–3805. International Committee on Computational Linguistics.

Huber, P.; and Carenini, G. 2019. Predicting Discourse Structure using Distant Supervision from Sentiment. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2306–2316.

Huber, P.; and Carenini, G. 2020a. From Sentiment Annotations to Sentiment Prediction through Discourse Augmentation. In *Proceedings of the 28th International Conference on Computational Linguistics*, 185–197.

Huber, P.; and Carenini, G. 2020b. MEGA RST Discourse Treebanks with Structure and Nuclearity from Scalable Distant Sentiment Supervision. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 7442–7457.

Irsoy, O.; and Alpaydin, E. 2016. Autoencoder trees. In *Asian Conference on Machine Learning*, 378–390.

Jang, E.; Gu, S.; and Poole, B. 2016. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144* .

Ji, Y.; and Smith, N. A. 2017. Neural Discourse Structure for Text Categorization. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 996–1005.

Joty, S.; Carenini, G.; and Ng, R. T. 2015. CODRA: A Novel Discriminative Framework for Rhetorical Analysis. *Computational Linguistics* .

Jozefowicz, R.; Vinyals, O.; Schuster, M.; Shazeer, N.; and Wu, Y. 2016. Exploring the limits of language modeling. *arXiv preprint arXiv:1602.02410* .

Kingma, D. P.; and Ba, J. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* .

Kusner, M. J.; Paige, B.; and Hernández-Lobato, J. M. 2017. Grammar variational autoencoder. *arXiv preprint arXiv:1703.01925* .

Li, J.; Luong, M.-T.; and Jurafsky, D. 2015. A hierarchical neural autoencoder for paragraphs and documents. *arXiv preprint arXiv:1506.01057* .

Liu, Y.; and Lapata, M. 2018. Learning Structured Text Representations. *Transactions of the Association for Computational Linguistics* 63–75.

Liu, Y.; Titov, I.; and Lapata, M. 2019. Single document summarization as tree induction. In *Proceedings of the 2019*

Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), 1745–1755.

Maddison, C. J.; Tarlow, D.; and Minka, T. 2014. A* sampling. In *Advances in Neural Information Processing Systems*, 3086–3094.

Maillard, J.; Clark, S.; and Yogatama, D. 2019. Jointly learning sentence embeddings and syntax with unsupervised treelstms. *Natural Language Engineering* 433–449.

Mann, W. C.; and Thompson, S. A. 1988. Rhetorical structure theory: Toward a functional theory of text organization. *Text-Interdisciplinary Journal for the Study of Discourse* 243–281.

Morey, M.; Muller, P.; and Asher, N. 2017. How much progress have we made on RST discourse parsing? A replication study of recent results on the RST-DT. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1319–1324.

Nejat, B.; Carenini, G.; and Ng, R. 2017. Exploring Joint Neural Model for Sentence Level Discourse Parsing and Sentiment Analysis. In *Proceedings of the 18th Annual SIGdial Meeting on Discourse and Dialogue*, 289–298.

Nishida, N.; and Nakayama, H. 2020. Unsupervised Discourse Constituency Parsing Using Viterbi EM. *Transactions of the Association for Computational Linguistics* 215–230.

Pennington, J.; Socher, R.; and Manning, C. D. 2014. GloVe: Global Vectors for Word Representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, 1532–1543.

Prasad, R.; Dinesh, N.; Lee, A.; Miltsakaki, E.; Robaldo, L.; Joshi, A.; and Webber, B. 2008. The Penn Discourse TreeBank 2.0. .

Press, O.; and Wolf, L. 2016. Using the output embedding to improve language models. *arXiv preprint arXiv:1608.05859* .

Socher, R.; Pennington, J.; Huang, E. H.; Ng, A. Y.; and Manning, C. D. 2011. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 conference on empirical methods in natural language processing*, 151–161.

Socher, R.; Perelygin, A.; Wu, J.; Chuang, J.; Manning, C. D.; Ng, A. Y.; and Potts, C. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, 1631–1642.

Srivastava, N.; Mansimov, E.; and Salakhudinov, R. 2015. Unsupervised learning of video representations using lstms. In *International conference on machine learning*, 843–852.

Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, 3104–3112.

Tai, K. S.; Socher, R.; and Manning, C. D. 2015. Improved semantic representations from tree-structured long short-term memory networks. *arXiv preprint arXiv:1503.00075* .

Tang, D.; Qin, B.; and Liu, T. 2015. Document modeling with gated recurrent neural network for sentiment classification. In *Proceedings of the 2015 conference on empirical methods in natural language processing*, 1422–1432.

Vinyals, O.; Fortunato, M.; and Jaitly, N. 2015. Pointer networks. In *Advances in neural information processing systems*, 2692–2700.

Wang, Y.; Li, S.; and Wang, H. 2017. A two-stage parsing method for text-level discourse analysis. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, 184–188.

Williams, A.; Drozdov, A.; and Bowman, S. R. 2018. Do latent tree learning models identify meaningful structure in sentences? *Transactions of the Association for Computational Linguistics* 253–267.

Xiao, W.; Huber, P.; and Carenini, G. 2020. Do We Really Need That Many Parameters In Transformer For Extractive Summarization? Discourse Can Help ! In *Proceedings of the First Workshop on Computational Approaches to Discourse*, 124–134.

Yang, Z.; Yang, D.; Dyer, C.; He, X.; Smola, A.; and Hovy, E. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 1480–1489.

Yogatama, D.; Blunsom, P.; Dyer, C.; Grefenstette, E.; and Ling, W. 2016. Learning to compose words into sentences with reinforcement learning. *arXiv preprint arXiv:1611.09100* .