The Thirty-First International Florida
Artificial Intelligence Research Society Conference (FLAIRS-31)

Tree Structured Multimedia Signal Modeling

Weicheng Ma
Department of Computer Science
Boston University
Boston, MA 02135
wm724@nyu.edu

Kai Cao, Xiang Li
Chief AI Office
Cambia Health Solutions
Seattle, WA 98101
{kai.cao, xiang.li}@cambiahealth.com

Peter Chin
Department of Computer Science
Boston University
Boston, MA 02135
spchin@cs.bu.edu

Abstract

Current solutions to multimedia modeling tasks feature sequential models and static tree-structured models. Sequential models, especially models based on Bidirectional LSTM (BLSTM) and Multilayer LSTM networks, have been widely applied on video, sound, music and text corpora. Despite their success in achieving state-of-the-art results on several multimedia processing tasks, sequential models always fail to emphasize short-term dependency relations, which are crucial in most sequential multimedia data. Tree-structured models are able to overcome this defect. The static tree-structured LSTM presented by Tai et al. (Tai, Socher, and Manning 2015) forcingly breaks down the dependencies between elements in each semantic group and those outside the group, while preserves chain-dependencies among semantic groups and among nodes in the same group. Though the tree-LSTM network is able to better represent the dependency structure of multimedia data, it requires the dependency relations of the input data to be known before it is fed into the network. This is hard to achieve since for most types of multimedia data there exists no parsers which can detect the dependency structure of every input sequence accurately enough. In order to preserve dependency information while eliminating the necessity of a perfect parser, in this paper we present a novel neural network architecture which 1) is self-expandable and 2) maintains the layered dependency structure of incoming multimedia data. We call our new neural network architecture Seq2Tree network. A Seq2Tree model is applicable on classification, prediction and generation tasks with task-specific adjustments of the model. We prove by experiments that our Seq2Tree model performs well in all the three types of tasks.

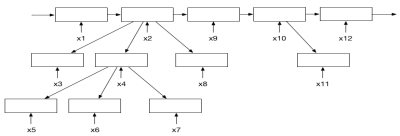

Figure 1: A tree-structured model with three layers. Hidden states of lower-level nodes are inherited from parent nodes.

Introduction

Multimedia signal modeling is the basis of multimedia signal processing tasks across a wide range of research fields such as Computer Vision, Natural Language Processing, and Sound Signal Processing. However on sequential-data-related research remain not fully developed due to flexibility in structure of sequential data. In this paper, we aim at tackling the problems static neural network solutions have in modeling sequential signals with a dynamically self-adjustable neural network architecture.

In sequential data each unit contributes to the prediction of all its following units, so traditional sequential models such as Hidden Markov Model (HMM) and Recurrent Neural Networks (RNN) are the first choice when processing this kind of data. With the ability of weakening gradient vanishment and explosion problems, Long Short Term Memory (LSTM) network has been very popular in sequential data processing tasks. Seq2seq network (Sutskever, Vinyals, and Le 2014), one of the most famous applications of LSTM network for example, has thus attracted much attention in machine translation research (Luong et al. 2015). Its use has also been extended to multiple other tasks such as speech to text conversion (Zhang, Chan, and Jaitly 2017). Moreover, HMM achieves high performance in music style classification task, especially when differentiating composer characteristics (Buzzanca 2002; Chai and Vercoe 2001).

However, simple sequential models over single data points can sometimes misrepresent complex multimedia signals. Thus, some variants of the sequential models are introduced. Bidirectional LSTM (BLSTM) network (Schuster and Paliwal 1997) for example, combines two LSTM networks each accepting the input forwards and backwards. Because of the backward LSTM, BLSTM is able to foresee possible boundaries of unit patterns in the future. The current state-of-the-art results in speech and noise separation task, as is reported in the 2nd CHiME challenge (Vincent et al. 2013), is achieved by a BLSTM-based system (Erdogan et al. 2015). Multilayer LSTM is another popular variant of the original LSTM network. This neural network architecture allows different features to sit on different layers of the network, so as to divide a sequence into patterns based on learnt boundary characteristics. As an example, a three-layer Seq2seq model achieves over 95% accuracy in the evaluation of constituent parsers (Vinyals et al. 2015).

Nevertheless, both BLSTM and Multilayer LSTM are additive combinations of multiple original LSTM networks which easily lose geometric information of the units. This undermines the performance of systems based on BLSTM and Multilayer LSTM. With careful examination of multiple multimedia data samples we found that most multimedia signals share two characteristics: 1) inner-group dependencies are stronger for every meaningful unit group and 2) meaningful groups form a chain-structured dependency path. These characteristics of temporally successive multimedia data lead us to a natural selection of a tree-structured representation which 1) expands along one direction and 2) branches only when a pattern starts while 3) ends a branch and continues expanding on higher level when it reaches the end of current pattern.

This special tree structure satisfies our needs of modeling multimedia signals in terms of meaningful segments but not single units by locating the units of the same semantic group in the same subtree. We call these meaningful unit groups segments. Currently no existing model works for bounding segments with flexible length. To build up such tree structure from sequential input, we highlight the ability of our self-expandable tree model to find boundaries of segments by branching at proper positions. We call this novel neural network architecture Seq2Tree network.

For generosity, we fit our tree model to three different types of tasks, namely classification, prediction and generation tasks with necessary modifications to the network. We designed experiments to prove the correctness of the our tree-structured models built by Seq2Tree network and its advancement over the traditional LSTM-based models.

Seq2Tree networks were introduced by (Ma et al. 2018). The structure has also been used in AI tasks such as signal processing (Ma et al. 2017).

Multimedia Signal Modeling

Among all types of multimedia signals we concentrate on temporally successive signals. To be more specific, in this paper we focus on text, video, music and sound signals. We mainly study mainly three types of tasks, namely classification, prediction and generation tasks.

Classification Model

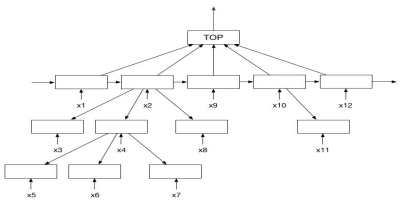

Figure 2: A tree-structured classification model. Information from every top-level node is summarized into the top node. The hidden state of the top node can be fed into a classifier.

As is shown in Figure 2, in the classification model there is one root node above the entire tree structure. The root node incorporates the hidden state of all top-level nodes in the original tree structure.

For the classification model we build a softmax classification model by adding a softmax layer on top of the root node subject to the error function:

$$p(y|h_{top}) = softmax(U^{(c)}h_{top} + b^{(c)}),$$
$$y_{top} = argmax_y p(y|h_{top})$$

where $U^{(c)}$ is the classification matrix, $b^{(c)}$ is the bias and h_{top} is the hidden state at the top of the tree.

The cost function we choose here is the cross-entropy loss of the predicted label y:

$$J(\theta) = -\frac{1}{C}\sum_{i=1}^{C} p(y|h_{top})logp(y|h_{top})$$

where C is the number of classes.

In our experiments we test our classification model on a text classification task. The task requires classification of Arxiv paper abstracts into their categories. Our data on this task is collected from Arxiv.

Generative Model

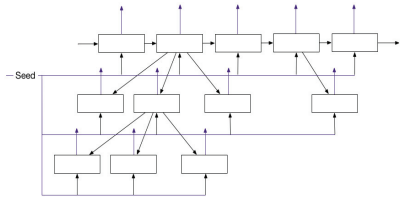

Figure 3: The generative version of a tree-structured model. Renders one output at every time step. All inputs come from the seed.

In our generative model we yield one output at every node in the tree with one fixed input and an inherited hidden state. Parent nodes are updated every time a signal is generated at their children nodes so the order of nodes affects the generation result.

Since in our model parent nodes are always temporally preceding units of their children nodes, the generation is done post-orderly but the final results are stacked according to the original temporal order.

In our experiments we apply Seq2Tree network, which we will introduce in the next section, as the generator and Long Short Term Memory (LSTM) network as discriminator to build a Generative Adversarial Network (GAN) (Goodfellow et al. 2014). The generation goal is music signal given a seed vector representing the music style of Bach and several real music pieces by Bach. In training the model we use the original GAN loss function:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N} logD(r_i) + \frac{1}{M}\sum_{i=1}^{M} logD(G(s_i))$$

where N and M are the sizes of real music signals in the training set and the generated musics, respectively. $D(r_i)$ stands for the discriminator output for the i-th real music, while $D(G(s_i))$ is the discriminator output given the generated music with the seed s_i.

Predictive Model

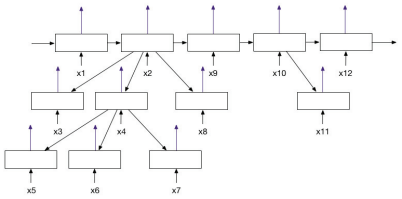

Figure 4: A predictive model with tree structure. Requires one output at every node. The inputs are independent at different time steps.

The predictive model is slightly different from the generative model in the sense that each node gets different input. Data processing order and output reformation are both similar to what the generative model does.

In the experiments we evaluate our predictive model on a speech feature extraction task. The goal of this task is to extract key features from audio files. Our solution to this task is by applying an autoencoder on the original sound signal directly. The two ends of the autoencoder are both implemented using Seq2Tree network. For comparison we also implemented an LSTM-LSTM autoencoder on the same task. Our data for the speech feature extraction task comes from the ComParE challenge.

The autoencoder is optimized with the absolute difference loss between the decoded signal and the original speech signal:

$$J(\theta) = -\frac{1}{N}\sum_{i=1}^{N} \|sig_{gen} - sig_{orig}\|$$

where N is the size of training set, and sig_{gen} and sig_{orig} are the generated sound signal and the original sound signal, respectively.

Seq2Tree Network

Theoretically our tree-structured model fits multimedia signal modeling well because of its ability to emphasize local relatedness while keeping track of global chain dependency. However none of the existing neural network architectures is able to deal with dynamic branching of previously unknown depth. This motivated us to design a new neural network architecture which could discover tree-structured dependency relations from sequential data.

To satisfy the needs of modeling multimedia data with non-fixed tree structure, the network architecture has to be able to dynamically decide the depth each new state is on. We reduce the problem of node layering to that of choosing the preceding state from all the former nodes in time sequence.

To maintain the tree-structured dependency, each new node introduces new information that influences all the ancestor nodes of the new unit. Similar to the definition in tree-structured LSTM network, we pass this update information through the forget gates of the ancestor nodes. The transition functions of our neural network architecture then becomes as follows:

$$\begin{aligned}
d_t &= \sigma(W^{(d)}x_t + U^{(d)}h_{parents} + b^{(d)}),\\
h_{parent} &= d_t h_{parents},\\
c_{parent} &= d_t c_{parents},\\
i_t &= \sigma(W^{(i)}x_t + U^{(i)}h_{parent} + b^{(i)}),\\
f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_{parent} + b^{(f)}),\\
o_t &= \sigma(W^{(o)}x_t + U^{(o)}h_{parent} + b^{(o)}),\\
u_t &= tanh(W^{(u)}x_t + U^{(u)}h_{parent} + b^{(u)}),\\
c_t &= i_t \odot u_t + f_t \odot c_{parent},\\
h_t &= u_t \odot tanh(c_t),\\
\Delta f_t &= \sigma(W^{(f)}x_t + U^{(f)}h_t + b^{(f)}),\\
\Delta c_t &= \Delta f_t \odot c_t,\\
l_t &= \sigma(W^{(l)}h_t + U^{(l)}h_{parents} + b^{(l)}),\\
c_{parents} &= c_{parents} + l_t \odot \Delta c_t,\\
h_{parents} &= o_{parents} \odot tanh(c_{parents}).
\end{aligned} \tag{1}$$

where d is the direction gate of Seq2Tree network, l gate calculates the influence distribution of h_t over $h_{parents}$ and c_{paents}. i, f, o gates, c cell and the hidden state h follow the same definition as is in the original LSTM network.

Experiments and Discussion

Arxiv Text Classification

For the text classification task we collected over 3 million abstracts from 18 classes of papers from Arxiv with a 80%/20% split for training/test sets.

Model	Precision (%)	UAR (%)
LSTM	16.64	N/A
Seq2Tree	22.32	42.60

Table 1: Arxiv text classification results.

In Table 1 we list the overall precision and the Unweighted Average Recall (UAR) of the text classification task on the test set. Though the precision scores of the Seq2Tree model and the LSTM model are close, the LSTM model produces no valid UAR score because it is not able to discriminate texts from different classes. This proves that our tree-structured model could better model text data than the sequential LSTM model.

In Figure 5 we show one example parsed by our Seq2Tree model. In the classification model we included only phrase-level attention which, according to our analysis, is not able to advise the network in learning sentence-level features. The wrongly-ordered words, on the other hand, should be caused

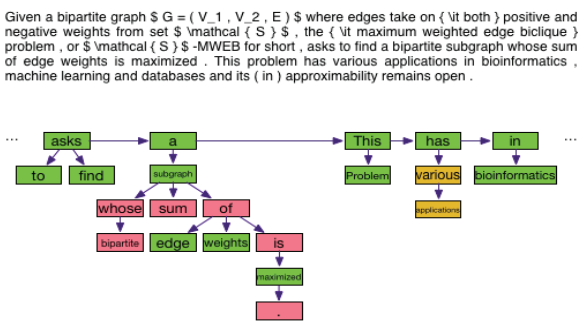

Figure 5: An example of the tree-structured model built by Seq2Tree network. Green nodes are correctly layered, while the red ones show incorrectly or inversely located nodes. The yellow nodes are misplaced, but the dependency relations are maintained.

by the lack of strong enough boundaries in parent node selection when inserting new nodes.

Music Generation

In the music generation task we evaluate the music generated by GAN with Seq2Tree network and LSTM network as generator, respectively. The initial point of the generation is randomized and the seed vector is extracted from the last hidden layer of a music composer classifier which performs 96.86% accuracy in our 10-class composer-based music classification task. The two GAN's share the same LSTM-based discriminator network. The real data we use in this experiment comes from 50 Bach-composed musics and we limited the lengths of these musics to be 500.

Model	Precision (%)
LSTM-GAN	54
Seq2Tree-GAN	72

Table 2: Evaluation results of generated music signals using a pre-trained composer-based music classification model.

We tested a composer-based music signal classifier on the generated musics. The generation result is tagged as good if the prediction result is Bach. As is mentioned earlier in this paper, our classifier claims a 96.86% overall accuracy on the music composer classification task. The classification results are listed in Table 2. Though the experiment is not a formal metric to evaluate music generation models, the better results our Seq2Tree-GAN got reveals the higher ability of the Seq2Tree model to represent music signal flows.

Speech Feature Extraction

Our data for the speech feature extraction experiment comes from the Addressee data in the ComParE challenge. As preprocessing we do MFCC over the audio files and pad them all to length 100. We examine the performance of our autoencoders by training an audio classifier with the output of the autoencoders.

Model	Precision (%)	UAR (%)
Original	14.16	N/A
Seq2Seq Autoencoder	18.44	N/A
Seq2Tree Autoencoder	22.32	42.60

Table 3: Sound signal classification results.

In this experiment we are using an autoencoder as a filter of features before classifying audio signals. In Table 3 we list the precisions and UAR's for all three cases, which are sound signal only, Seq2Tree autoencoder and Seq2Seq autoencoder, respectively. It is clear that without feature selection the classifier does not have any ability to classify those instances in the challenge dataset (all instances tagged with the same label). With the help of both autoencoders the classifier is able to produce valid UAR scores, while the performance of the classifier appears to be better when working with the Seq2Tree autoencoder. This is to say that the tree-structured model generated by Seq2Tree network better simulates the actual way the sound signals are constructed by phoneme units.

Conclusion

In the paper we present a novel way of modeling multimedia signals using a tree-structured model. In multimedia signals, meanings are usually expressed by segments of units. This makes the model theoretically more advanced than traditional sequential models especially on temporally successive data, because it helps separate local inner-segment dependencies from the chain-structured global dependency paths. We designed experiments to prove the actual applicability of the tree-structured models in multimedia processing tasks. To build such models we introduce a new neural network architecture that works specifically on segment-based data. We call this network architecture Seq2Tree network. From all the three sets of experimental results we can see that tree-structured modeling of data helps improve the performance of classification, generation and prediction tasks. This implies the correctness of the tree-structured model and encourages us to examine deeper into the nature of temporally successive multimedia signals. In our future work we are going to build a Seq2Tree based dependency parser. Dependency parsers have been utilized in quite a few NLP tasks such as Relation Extraction and Event Extraction systems. For example, (Cao, Li, and Grishman 2015) introduces dependency regularizations on dependeny parsers. (Cao, Li, and Grishman 2016) (Cao 2016) include syntactic relations with dependency regularizations in event detection systems. Deep neural networks have also applied in semantic relations such as Abstract Meaning Representation parsers. The Seq2Tree structure can also be applied in AMR parsing because the AMR semantic structure is also a tree. AMR parser is widely explored with different NLP tasks such as event detection (Li et al. 2015) and natural language generation (Flanigan, Dyer, and Smith 2016).

References

Buzzanca, G. 2002. A supervised learning approach to musical style recognition. In *Music and Artificial Intelligence. Additional Proceedings of the Second International Conference, ICMAI*, volume 2002, 167.

Cao, K.; Li, X.; and Grishman, R. 2015. Improving event detection with dependency regularization. In *Proceedings of RANLP*.

Cao, K.; Li, X.; and Grishman, R. 2016. Leveraging dependency regularization for event extraction. In *Proceedings of FLAIRS*.

Cao, K. 2016. *IMPROVING EVENT EXTRACTION : CASTING A WIDER NET*. Ph.D. Dissertation, New York University.

Chai, W., and Vercoe, B. 2001. Folk music classification using hidden markov models. In *Proceedings of International Conference on Artificial Intelligence*, volume 6. sn.

Erdogan, H.; Hershey, J. R.; Watanabe, S.; and Le Roux, J. 2015. Phase-sensitive and recognition-boosted speech separation using deep recurrent neural networks. In *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*, 708–712. IEEE.

Flanigan, J.; Dyer, C.; and Smith, Noah A.and Carbonell, J. 2016. Generation from abstract meaning representation using tree transducers. In *HLT-NAACL*.

Goodfellow, I.; Pouget-Abadie, J.; Mirza, M.; Xu, B.; Warde-Farley, D.; Ozair, S.; Courville, A.; and Bengio, Y. 2014. Generative adversarial nets. In *Advances in neural information processing systems*, 2672–2680.

Li, X.; Nguyen, T. H.; Cao, K.; and Grishman, R. 2015. Improving event detection with abstract meaning representation. In *Proceedings of ACL-IJCNLP Workshop on Computing News Storylines (CNewS 2015)*.

Luong, M.-T.; Le, Q. V.; Sutskever, I.; Vinyals, O.; and Kaiser, L. 2015. Multi-task sequence to sequence learning. *arXiv preprint arXiv:1511.06114*.

Ma, W.; Cao, K.; Ni, Z.; Ni, X.; and Chin, S. 2017. Sound signal processing based on seq2tree network. In *Proceedings of Interspeech workshop on Vocal Interactivity in-and-between Humans, Animals and Robots*.

Ma, W.; Cao, K.; Ni, Z.; Chin, P.; and Li, X. 2018. Seq2tree: A tree-structured extension of lstm network. In *Proceedings of LREC*.

Schuster, M., and Paliwal, K. K. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing* 45(11):2673–2681.

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*.

Vincent, E.; Barker, J.; Watanabe, S.; Le Roux, J.; Nesta, F.; and Matassoni, M. 2013. The second 'chime' speech separation and recognition challenge: Datasets, tasks and baselines. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, 126–130. IEEE.

Vinyals, O.; Kaiser, Ł.; Koo, T.; Petrov, S.; Sutskever, I.; and Hinton, G. 2015. Grammar as a foreign language. In *Advances in Neural Information Processing Systems*, 2773–2781.

Zhang, Y.; Chan, W.; and Jaitly, N. 2017. Very deep convolutional networks for end-to-end speech recognition. In *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*, 4845–4849. IEEE.