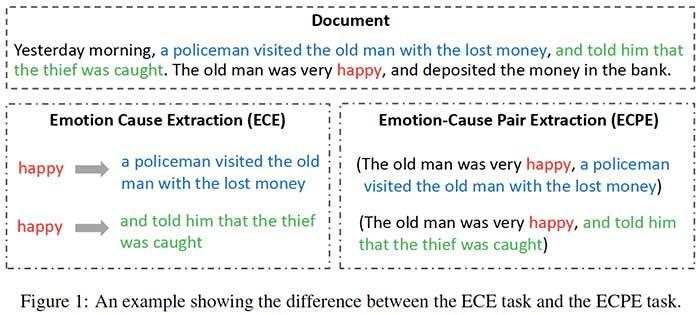
### **1.** [**EMOTION-CAUSE PAIR EXTRACTION: A NEW TASK TO EMOTION ANALYSIS IN TEXTS**](https://arxiv.org/abs/1906.01267)**, BY RUI XIA AND ZIXIANG DING**

#### **ORIGINAL ABSTRACT**

Emotion cause extraction (ECE), the task aimed at extracting the potential causes behind certain emotions in text, has gained much attention in recent years due to its wide applications. However, it suffers from two shortcomings: 1) the emotion must be annotated before cause extraction in ECE, which greatly limits its applications in real-world scenarios; 2) the way to first annotate emotion and then extract the cause ignores the fact that they are mutually indicative. In this work, we propose a new task: emotion-cause pair extraction (ECPE), which aims to extract the potential pairs of emotions and corresponding causes in a document. We propose a 2-step approach to address this new ECPE task, which first performs individual emotion extraction and cause extraction via multi-task learning, and then conduct emotion-cause pairing and filtering. The experimental results on a benchmark emotion cause corpus prove the feasibility of the ECPE task as well as the effectiveness of our approach.

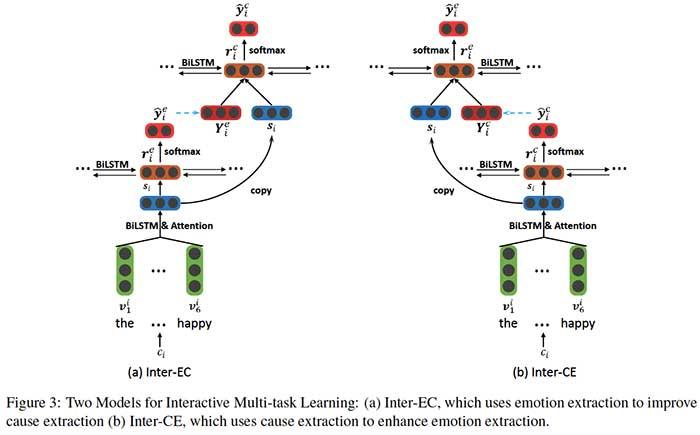
#### **OUR SUMMARY**

Emotion cause extraction (ECE) is an approach used in natural language processing to identify statements containing the causes behind vocabulary expressing emotion. However, ECE requires emotions to first be annotated and ignores mutual relationships between causes and emotional effects. The researchers sought to solve this problem by simultaneously identifying pairs of emotions and causes in a task they call emotion-cause pair extraction (ECPE). ECPE uses a two-step approach: the first step uses two multi-task learning networks to identify emotion and cause clauses, while the second step pairs all causes and emotions, and uses a trained filter to eliminate pairings that do not contain a causal relationship. The resulting ECPE task is able to identify emotion-cause pairs at an accuracy on par with existing ECE methods but without requiring emotion annotation.



#### **WHAT’S THE CORE IDEA OF THIS PAPER?**

* The paper introduces a new emotion-cause pair extraction (ECPE) task to overcome the limitations of the traditional ECE task, where emotion annotation is required prior to cause extraction and mutual indicativeness of emotion and cause is not taken into account.
* The introduced approach consists of two steps:
  + In the first step, the two individual tasks of emotion extraction and cause extraction are performed via two kinds of multi-task learning networks:
    - Inter-EC that uses emotion extraction to improve cause extraction;
    - Inter-CE that leverages cause extraction to enhance emotion extraction.
  + In the second step, the model combines all elements of the two sets into pairs by applying a Cartesian product. Then, a logistic regression model is trained to eliminate pairs that do not contain a causal relationship.



#### **WHAT’S THE KEY ACHIEVEMENT?**

* ECPE is able to achieve F1 scores of 0.83 for emotion extraction, 0.65 for cause extraction, and 0.61 for emotion-cause pairing.
* On the ECE benchmark dataset, ECPE performs on par with existing ECE methods that require emotion annotation before causal clauses can be identified.

#### **WHAT DOES THE AI COMMUNITY THINK?**

* The paper received an Outstanding Paper award at ACL 2019.

#### **WHAT ARE FUTURE RESEARCH AREAS?**

* Altering the ECPE approach from a two-step to a one-step process that directly extracts emotion-cause pairs in an end-to-end fashion.

#### **WHAT ARE POSSIBLE BUSINESS APPLICATIONS?**

* Sentiment analysis for marketing campaigns.
* Opinion monitoring from social media.

#### **WHERE CAN YOU GET IMPLEMENTATION CODE?**

* The code used in this study is available on [GitHub](https://github.com/NUSTM/ECPE).

### **2. improving Lives of Indebted Farmers Using Deep Learning: Predicting Agricultural Produce Prices Using Convolutional Neural Networks**

### 

Hangzhi Guo,1 Alexander Woodruff,2 Amulya Yadav2 1Wenzhou Kean University, Wenzhou, China 325060 2Pennsylvania State University, University Park, PA 16802 guoha@kean.edu, arw5550@psu.edu, amulya@psu.edu

Abstract

Farmer suicides have become an urgent social problem which governments around the world are trying hard to solve. Most farmers are driven to suicide due to an inability to sell their produce at desired profit levels, which is caused by the widespread uncertainty/fluctuation in produce prices resulting from varying market conditions. To prevent farmer suicides, this paper takes the first step towards resolving the issue of produce price uncertainty by presenting PECAD, a deep learning algorithm for accurate prediction of future produce prices based on past pricing and volume patterns. While previous work presents machine learning algorithms for prediction of produce prices, they suffer from two limitations: (i) they do not explicitly consider the spatio-temporal dependence of future prices on past data; and as a result, (ii) they rely on classical ML prediction models which often perform poorly when applied to spatio-temporal datasets. PECAD addresses these limitations via three major contributions: (i) we gather real-world daily price and (produced) volume data of different crops over a period of 11 years from an official Indian government administered website; (ii) we pre-process this raw dataset via state-of-the-art imputation techniques to account for missing data entries; and (iii) PECAD proposes a novel wide and deep neural network architecture which consists of two separate convolutional neural network models (trained for pricing and volume data respectively). Our simulation results show that PECAD outperforms existing stateof-the-art baseline methods by achieving significantly lesser root mean squared error (RMSE) - PECAD achieves ∼25% lesser coefficient of variance than state-of-the-art baselines. Our work is done in collaboration with a non-profit agency that works on preventing farmer suicides in the Indian state of Jharkhand, and PECAD is currently being reviewed by them for potential deployment.

### **3.** [**ALBERT: A LITE BERT FOR SELF-SUPERVISED LEARNING OF LANGUAGE REPRESENTATIONS**](https://arxiv.org/abs/1909.11942v1)**, BY ZHENZHONG LAN, MINGDA CHEN, SEBASTIAN GOODMAN, KEVIN GIMPEL, PIYUSH SHARMA, RADU SORICUT**

#### **ORIGINAL ABSTRACT**

Increasing model size when pretraining natural language representations often results in improved performance on downstream tasks. However, at some point further model increases become harder due to GPU/TPU memory limitations, longer training times, and unexpected model degradation. To address these problems, we present two parameter-reduction techniques to lower memory consumption and increase the training speed of BERT. Comprehensive empirical evidence shows that our proposed methods lead to models that scale much better compared to the original BERT. We also use a self-supervised loss that focuses on modeling inter-sentence coherence, and show it consistently helps downstream tasks with multi-sentence inputs. As a result, our best model establishes new state-of-the-art results on the GLUE, RACE, and SQuAD benchmarks while having fewer parameters compared to BERT-large.

### **4.** [**CTRL: A CONDITIONAL TRANSFORMER LANGUAGE MODEL FOR CONTROLLABLE GENERATION**](https://einstein.ai/presentations/ctrl.pdf)**, BY NITISH SHIRISH KESKAR, BRYAN MCCANN, LAV R. VARSHNEY, CAIMING XIONG, RICHARD SOCHER**

#### **ORIGINAL ABSTRACT**

Large-scale language models show promising text generation capabilities, but users cannot easily control particular aspects of the generated text. We release CTRL, a 1.6 billion-parameter conditional transformer language model, trained to condition on control codes that govern style, content, and task-specific behavior. Control codes were derived from structure that naturally co-occurs with raw text, preserving the advantages of unsupervised learning while providing more explicit control over text generation. These codes also allow CTRL to predict which parts of the training data are most likely given a sequence. This provides a potential method for analyzing large amounts of data via model-based source attribution. We have released multiple full-sized, pretrained versions of CTRL at <https://www.github.com/salesforce/ctrl>.

#### **OUR SUMMARY**

Language models used for text generation are very powerful, but they are often “black boxes”, so users do not have much control over the output. To address this problem, the Salesforce research team has introduced the **Conditional Transformer Language (CTRL)** model that conditions on a set of *control codes.* With these codes, the users can control domain, style, topics, dates, entities, relationships between entities, plot points, and task-related behavior. Moreover, all control codes can be traced back to a specific subset of the training data, allowing CTRL to predict the subset of the training data most likely leveraged for a particular sequence. This relationship between CTRL and its training data provides new possibilities for analyzing the correlations learned from each domain.

#### **WHAT’S THE CORE IDEA OF THIS PAPER?**

* Text generation tools are very powerful, but they do not give users much control over the content, style or genre of the generated text.
* The Salesforce research team has released CTRL, a 1.6 billion-parameter conditional transformer language model, that gives users more control over the generated content:
  + CTRL exposes keywords called control codes which allow users to specify a domain, style, topics, dates, entities, relationships between entities, plot points, and task-related behavior.
  + CTRL is trained on control codes derived from the structure that naturally co-occurs with the raw text. In particular, CTRL leverages the fact that training data is usually associated with a URL that contains information relevant to the text it represents.

#### **WHAT’S THE KEY ACHIEVEMENT?**

* Introducing and open-sourcing a language model that:
  + enables more controllable text generation;
  + provides new opportunities for analyzing large amounts of text via model-based source attribution;
  + can be used to detect artificially generated text.

#### **WHAT DOES THE AI COMMUNITY THINK?**

* The community [appreciates](https://www.reddit.com/r/MachineLearning/comments/d2uii7/r_ctrl_a_conditional_transformer_language_mode/) that the researchers offered such a clear discussion on the ethical considerations behind releases of large language models and that they included it in a separate section of the paper rather than relegating it to a blog post.

#### **WHAT ARE FUTURE RESEARCH AREAS?**

* Introducing a greater variety of control codes to allow finer-grained control.
* Extending to other areas of NLP including abstractive summarization and commonsense reasoning.
* Analyzing the relationships between training data and language models.
* Exploring the possibilities to make the interface between humans and language models more explicit and intuitive.

#### **WHAT ARE POSSIBLE BUSINESS APPLICATIONS?**

* Improved and tailored text generation for question-answering systems and other human-computer interaction applications.
* Identifying artificially generated text, to detect malicious uses such as automatically generated essays or fake reviews.

#### **WHERE CAN YOU GET IMPLEMENTATION CODE?**

* The authors have released multiple full-sized, pretrained versions of CTRL on [GitHub](https://www.github.com/salesforce/ctrl).

### **5.** [**ORDERED NEURONS: INTEGRATING TREE STRUCTURES INTO RECURRENT NEURAL NETWORKS**](https://arxiv.org/abs/1810.09536)**, BY YIKANG SHEN, SHAWN TAN, ALESSANDRO SORDONI, AARON COURVILLE**

#### **ORIGINAL ABSTRACT**

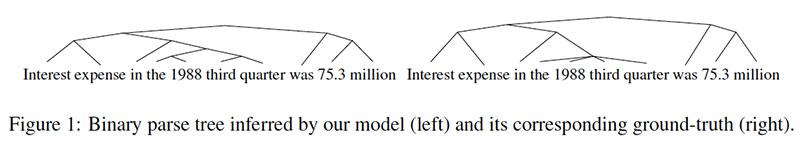
Natural language is hierarchically structured: smaller units (e.g., phrases) are nested within larger units (e.g., clauses). When a larger constituent ends, all of the smaller constituents that are nested within it must also be closed. While the standard LSTM architecture allows different neurons to track information at different time scales, it does not have an explicit bias towards modeling a hierarchy of constituents. This paper proposes to add such an inductive bias by ordering the neurons; a vector of master input and forget gates ensures that when a given neuron is updated, all the neurons that follow it in the ordering are also updated. Our novel recurrent architecture, ordered neurons LSTM (ON-LSTM), achieves good performance on four different tasks: language modeling, unsupervised parsing, targeted syntactic evaluation, and logical inference.

#### **OUR SUMMARY**

The joint group of researchers from the Université de Montréal and Microsoft studies the problem of integrating tree structures into recurrent neural networks (RNNs). Natural language is hierarchically structured, with smaller units (e.g. phrases) nested within larger units (e.g. clauses), but this hierarchy is not reflected in a standard RNN architecture. In this paper, the authors propose to address this problem by ordering neurons. In particular, they use a new activation function, the *cumulative softmax()*, to produce a vector of master input and forget gates: when a given neuron is updated (or erased), all of the neurons that follow it in the ordering are also updated (or erased). The experiments demonstrate that this new architecture, called ON-LSTM, **ordered neurons long short-term memory** network, performs well on a variety of NLP tasks.

#### **WHAT’S THE CORE IDEA OF THIS PAPER?**

* Even though some evidence exists that LSTMs can potentially encode the tree structure implicitly, the researchers believe that better results can be obtained by equipping the model with an inductive bias towards learning such latent tree structures.
* To this end, they introduce a new inductive bias for RNNs, namely **ordered neurons**, that forces neurons to represent information at different time scales:
  + High-ranking neurons will store long-term information to be kept for many steps, while low-ranking neurons will store short-term information that only last one or a few time steps.
  + The differentiation between high-ranking and low-ranking neurons is learned in a data-driven fashion by controlling the update frequency of single neurons.
  + In particular, a new activation function, the *cumulative softmax()*, or *cumax()*, ensures that some neurons are updated more (or less) frequently the others: to erase (or update) high-ranking neurons, the model should first erase (or update) all lower-ranking neurons.



#### **WHAT’S THE KEY ACHIEVEMENT?**

* The proposed model performs better than standard LSTMs on such tasks as:
  + language modeling;
  + targeted syntactic evaluation (on long-term dependency cases);
  + logical inference (especially on longer sequences).
* The ON-LSTM model also performs well on an *unsupervised constituency parsing* task:
  + it outperforms previous models in terms of generalization and robustness toward longer sentences;
  + it gives strong results for phrase detection, including adjective, prepositional, and noun phrases;
  + the model induces the latent structure of natural language in coherence with human annotations.

#### **WHAT DOES THE AI COMMUNITY THINK?**

* The paper received the Best Paper award at ICLR 2019, the key conference in machine learning.

#### **WHAT ARE POSSIBLE BUSINESS APPLICATIONS?**

* The proposed model can benefit many downstream NLP tasks, including question answering, named entity recognition, co-reference resolution, and others.

#### **WHERE CAN YOU GET IMPLEMENTATION CODE?**

* The code used for the word-level language model and unsupervised parsing experiments in this paper is available on [GitHub](https://github.com/yikangshen/Ordered-Neurons).