Effective Use of Graph Convolution Network and Contextual Sub-Tree for Commodity News Event Extraction

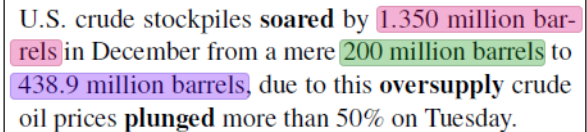**Meisin Lee, Lay-Ki Soon, Eu-Gene Siew**
Monash University
{mei.lee, soon.layki, siew.eu-gene}@monash.edu

Abstract

Event extraction in commodity news is a less researched area as compared to generic event extraction. However, accurate event extraction from commodity news is useful in a broad range of applications such as understanding event chains and learning event-event relations, which can then be used for commodity price prediction. The events found in commodity news exhibit characteristics different from generic events, hence posing a unique challenge in event extraction using existing methods. This paper proposes an effective use of Graph Convolutional Networks (GCN) with a pruned dependency parse tree, termed contextual sub-tree, for better event extraction in commodity news. The event extraction model is trained using feature embeddings from ComBERT, a BERT-based masked language model that was produced through domain-adaptive pre-training on a commodity news corpus. Experimental results show the efficiency of the proposed solution, which outperforms existing methods with F1 scores as high as 0.90. Furthermore, our pre-trained language model outperforms GloVe by 23%, and BERT and RoBERTa by 7% in terms of argument roles classification. For the goal of reproducibility, the code and trained models are made publicly available [1].

1 Introduction

World events such as geo-political and macro-economic-related events have been shown to impact commodity prices in both short-term and long-term (Brandt and Gao, 2019). Generally, events found in commodity news articles can be categorized into geo-political, macro-economic, supply-demand related, and commodity price movements. Commodity news is a valuable source of information to extract and mine for such events. Accurate

Figure 1: An example of a sentence from a piece of commodity news[3], loaded with three events (trigger words are in bold): (1) Crude oil inventory increase, (2) Oversupply and (3) Crude Oil price decrease. Three event arguments that are of the same entity type are highlighted in color.

event extraction is useful for many important downstream tasks, such as understanding event chains and learning event sequence, also known as scripts in (Schank and Abelson, 2013), that can be used for accurate commodity price prediction ultimately.

As defined in ACE (Automatic Content Extraction) Program[2], the event extraction task is made up of two subtasks: (1) event trigger extraction (identifying and classifying event triggers) and (2) event argument extraction (identifying arguments of event triggers and labeling their roles). In this work, we perform event extraction on the Commodity News dataset introduced in (Lee et al., 2021). Figure 1 shows a sample sentence from this dataset.

To illustrate the task of event extraction, consider the example in Figure 1 and the corresponding information tabulated in Table 1. In the event trigger extraction sub-task, the model is trained to identify the trigger word: **soared** and classify the right event type: **movement_up_gain**, while in the event argument extraction sub-task, the model is trained to identify event arguments from a pool of entity mentions within the sentence and then label the argument roles each entity plays in relation to the identified event.

---

[2] https://www.ldc.upenn.edu/collaborations/past-projects/ace

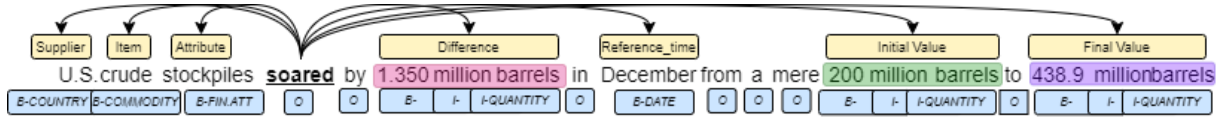[3] This example is used throughout this paper.

Figure 2: The example in Figure 1 is shown here with annotation provided in (Lee et al., 2021); trigger word is underlined and in bold, entity type for each entity mention is shown in blue below its respective word in BIO-tagging format while event argument role for each entity is shown in light yellow above the words; arches link argument to its trigger word.

Event	Entity Mention	Argument Role
Trigger: **soared**	U.S.	Supplier
	crude	Item
	stockpiles	Attribute
	1.350 million barrels	Difference
Event type: **movement_ up_gain**	December	Reference point
	200 million barrels	Initial Value
	438.9 million barrels	Final Value
	more than 50%	NONE

Table 1: Event extraction of "crude oil inventory increase" event (first event in Figure 1).

The types of events found in commodity news are vastly different from generic events in ACE2005[4]. Below is a list of unique characteristics of these events:

1. **Number intensity** - Numbers (e.g., price, difference, percentage of change) and dates (including date of the opening price, dates of closing price) are abundant in commodity news. These numerical data is critical in expressing financial information. Generic information extraction methods may not work well for numeric data, as it is clearly seen in (Saha et al., 2017) where the authors introduced the first Open *numerical* relation extractor specifically to extract Open Information Extraction (IE) tuples that contain numbers or a quantity-unit phrase.

2. **Arguments homogeneity** - Many arguments of the same entity type plays distinct roles in an event. Figure 2 shows that *1.350 million barrels*, *200 million barrels*, *438.0 million barrels* are tagged as *QUANTITY* (see BIO-tagging at the bottom row). However, all three arguments play a different role in relation to the event (see Argument roles at the top row).

3. **Undifferentiated event types without its arguments** - Consider the event triggered by **"soared"** in Figure 1, it is incomplete to identify the event (movement-up-gain) without knowing "what" soared. In commodity

news, possible events are 'price soared', 'supply soared', and 'demand soared'. To disambiguate events and represent them accurately, it is important to extract event arguments accurately as well.

With these challenges in mind, this paper proposes a solution applying Graph Convolutional Networks (GCN) over contextual sub-tree for the task of event extraction. The contextual sub-tree is a dependency parse tree uniquely pruned that provides not just dependency path information but also off-path information. The off-path information adds more context to the existing dependency path between two nodes; hence it is termed contextual sub-tree.

The contributions of this work are as follows: (1) We show that a domain-adaptive pre-trained Language Model, ComBERT, can yield promising performance over generic pre-trained language model when fine-tuned on event extraction tasks, and (2) we propose an effective usage of a graph neural network, in the form of a GCN with contextual sub-tree that outperforms other existing approaches. The usefulness is particularly apparent in the subtask of event argument classification.

2 Commodity News Dataset

The dataset introduced in (Lee et al., 2021) is a collection of annotated commodity news articles where its annotation is based on standards introduced by canonical programs such as ACE and TAC-KBP[5]. Figure 2 captures in graphical form, the annotatations found in the dataset: (1) Entity Mentions - both named and nominal (entity type shown in blue under the sentence in BIO-tagging format); (2) Event Trigger Words (word underlined and in bold); and (3) Argument roles (labels are in light yellow above the sentence) with arches linking each argument to the event it belongs. This dataset consists of 21 entity types, 18 event types, and 19 argument role types (each event type has

---

[4]ACE2005 is a multilingual Training Corpus developed under the ACE program

[5]https://tac.nist.gov/2015/KBP

its own set of argument roles. Event types are defined based on Ravenpack's[6] event taxonomy, out of which 18 event types are chosen. In terms of size, this dataset contains 8,850 entity mentions and 3,949 events. Table 2 shows the event type distribution. More information about the dataset, including argument roles, entity types, and example event types are found in Appendix A.

Event type	Type ratio	# sentence
1. Cause-movement-down-loss	13.35%	524
2. Cause-movement-up-gain	2.23%	88
3. Civil-unrest	2.53%	100
4. Crisis	0.76%	30
5. Embargo	3.75%	148
6. Geopolitical-tension	1.70%	67
7. Grow-strong	6.03%	238
8. Movement-down-loss	22.69%	896
9. Movement-flat	1.52%	60
10. Movement-up-gain	22.13%	874
11. Negative-sentiment	4.79%	189
12. Oversupply	2.63%	104
13. Position-high	3.82%	151
14. Position-low	3.11%	123
15. Prohibiting	1.06%	42
16. Shortage	1.04%	41
17. Slow-weak	5.47%	216
18. Trade-tensions	1.39%	55
Total		3949

Table 2: Event type distribution and sentence level counts

3 Related Work

Event Extraction in Finance/Economics. Generic event extraction-related work is covered in detail in the survey paper (Xiang and Wang, 2019) and in (Hogenboom et al., 2016). Here we focus specifically on event extraction within the domain of finance and economics. Most of the event extraction tasks in this domain focuses on extracting company-related events. Here is a summary of related methods in recent literature:

1. Rule-based approach: authors in (Malik et al., 2011) introduced statistical classifiers aided by rules, while authors in (Hogenboom et al., 2013) used rule-sets and domain ontology knowledge-bases together with other semantically-enabled components;
2. Usage of external resources: Semantic Frame was in (Xie et al., 2013) and Wikipedia for weak supervision in (Ein-Dor et al., 2019);

3. Open IE (Open Information Extraction): authors in (Ding et al., 2014) extracted events from news headline via Open IE; while in (Saha et al., 2017), authors introduced an extension of Open IE to extract numerical arguments in each Open IE tuple.
4. Deep learning approach: (Yang et al., 2018) proposed a deep learning approach to extract financial events from Chinese text.

Although company financial events and commodity news fall under the same domain, and both may involve numerical data as event arguments, existing methods for company financial event extractions are rather limited. For example, the solution in (Saha et al., 2017) caters for extracting only one numerical argument for each Open IE tuple. In comparison, the task of extracting company financial information in (Yang et al., 2018) is the closest match to this work in terms of extracting numerical data as event arguments. However, it is a solution for Chinese text and focuses on document-level event extraction.

Graph Convolutional Networks. The usage of Graph Convolutional Networks (GCN) coupled with syntactic information from dependency parse tree has been used for event extraction in (Nguyen and Grishman, 2018) and in (Liu et al., 2018). In (Nguyen and Grishman, 2018), the authors proposed using GCN over syntactic dependency graphs of sentences to produce non-consecutive k-grams as an effective mechanism to link words to their informative content directly for event detection. Authors in (Liu et al., 2018) on the other hand, used attention-based GCN to model graph information to extract multiple event triggers and arguments jointly. Their proposed solution, Joint Multiple Events Extraction (JMEE) framework, focuses on modeling the association between events to enhance the accuracy of event extraction. Both these solutions use the shortest dependency path.

Apart from event extraction, GCN has been used successfully for relation extraction in (Zhang et al., 2018). Instead of obtaining tokens strictly from the shortest dependency path, authors in (Zhang et al., 2018) made modifications to produce pruned a sub-dependency tree to include off-path information as well such as negation cue words. Among the related work listed here, the one that is closest to our work in terms of the task (event extraction) and scope (sentence level) is JMEE by (Liu et al., 2018).

---

[6]RavenPack is an analytics provider for financial services. Among their products are finance and economic data. More information can be found on their page: https://www.ravenpack.com/

4 Proposed Solution

In reference to the list of unique characteristics of events found in commodity news (listed in Section 1), this paper proposes a solution using Graph Convolutional Networks (GCN) with contextual sub-tree for effective event extraction in commodity news.

Syntactic dependency graphs represent sentences as directed trees with head-modifier dependency arcs between related words. Each word in such graphs is surrounded by its direct syntactic governor and dependent words (the neighbors), over which convolution operations can be performed on the most relevant words and avoid the modeling of unrelated words. Rather than using the full dependency parse tree, we propose to use a uniquely pruned dependency tree that is made up of the shortest path between two nodes (in our case - the **trigger candidate** and **entity mention**) and additional **off-path** nodes. The off-path nodes are included to provide additional contextual information. The resulting tree is termed **contextual sub-tree**.

4.1 Contextual Sub-tree

The dependency tree for the example sentence is shown in Figure 3 with one of the many candidate contextual sub-trees highlighted. Inspired by (Zhang et al., 2018), we prune the dependency tree to obtain the sub-tree rooted at the Least Common Ancestor (LCA) between the trigger candidate and the entity mention candidate while also contains off-path nodes. These off-path nodes provide additional and crucial contexts that enable better results in argument role classification. Off-path information is made up of tokens that are up to distance $DIST$ away from the dependency path. Algorithm 1 shows the steps of how to build the contextual sub-tree. As shown in (Zhang et al., 2018), $DIST = 1$ achieves the best balance between including contextual information and keeping irrelevant ones out of the resulting sub-tree as much as possible.

The usage of contextual sub-tree is targeted at **argument role classification**, the subtask within event extraction that classifies the argument role each entity plays in an event. Entity candidates are classified into one of the 19 argument roles. Figure 4 (Left) shows the sub-tree with the LCA path between event trigger and argument, while Figure 4 (Right) shows a slightly different sub-tree

---

Algorithm 1: Build sub-parse tree from dependency head indexes

Result: sub-tree structure

convert head indexes to tree object;

if *DIST < 0* **then**
| build the whole tree;
else
| find all ancestor nodes of trigger;
| find all ancestor nodes of entity;
| find lowest common ancestor;
| generate PathNodes (common nodes between trigger & entity);
| insert more nodes based on DIST away from PathNodes;
end

---

that not only contains the LCA path but also "off-path" information.

Below are three examples of contextual sub-trees (in words, without tree structure) between an event trigger and an entity mention. Off-path information are in *italics*. In these examples, the off-path words are prepositional words that help with the correct classification of the entity mention's argument role even though all the entities are of the same type, namely "quantity":

1. **soared** *by* <u>1.350 million barrels</u>
   argument role - difference
2. **soared** *from* <u>a mere 200 million barrels</u>:
   argument role - initial value
3. **soared** *to* <u>438.9 million barrels</u>
   argument role - final value

4.2 Domain-Adaptive Pre-training: ComBERT

Pre-trained language models such as BERT have been shown to produce SOTA results in many NLP tasks, including event extraction in the ACE2005 corpus (Yang et al., 2019). Instead of using BERT, we have decided to further pre-train BERT on a commodity news corpus, adapting the model to the finance and economics domains. The resulting model is referred to here as ComBERT. In the commodity news corpus, there are some commodity-specific polysemous words that can be better represented with further pre-training with in-domain data. Apart from the famous 'bank' example where the word could mean (1) financial institution or (2) terrain that is part of the river, there are some commodity-specific polysemous words in the com-
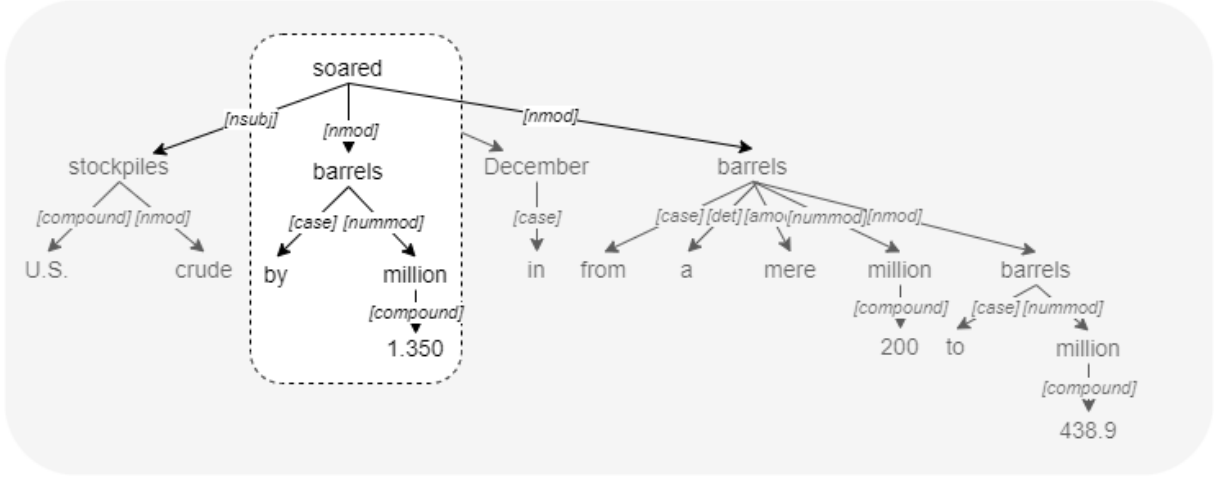
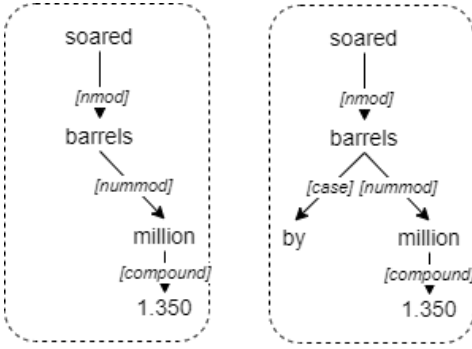Figure 3: The dependency parse tree of the example sentence.

Figure 4: Left: Sub-tree with shortest path, Right: Contextual sub-tree with off-path information

modity news corpus that can be better represented with further pre-training with in-domain data, for example:

- **stocks**: (1) inventory and (2) shares
- **tank**: (1) storage vessel (noun), (2) market / price drop (verb)

The commodity news corpus is made up of about 20k news articles extracted from https://www.investing.com/commodities/crude-oil-news[7], with publishing dates ranging from 2013 to 2019. We initialized ComBERT with `bert-base-cased`, one of the pre-trained BERT models provided by (Devlin et al., 2019a) and have the same model settings of transformer and pre-training hyperparameters as BERT. We have decided to use cased vocabulary `bert-base-cased` instead of uncased because the event extraction task involves extracting event arguments that are made up of named

---

[7]the same source as the annotated dataset used here in event extraction.

entities and nominal entities. Examples of named entities are countries, organizations, and specific commodities-related terms such as WTI, ICE, NYMEX, Brent and etc. Having a case-sensitive model yielded slightly better performance for the downstream event extraction task.

The proposed solution is built on leveraging the power of a domain-adaptive pre-training model, ComBERT, to further fine-tune for event extraction with a smaller dataset and with fewer training steps.

4.3 Event Extraction

Both of the sub-tasks within event extraction: (1) event trigger extraction and (2) event argument extraction are conceived as sentence-level multi-label classification tasks. The overall architecture is shown in Figure 5.

4.4 Data Proprocessing

Input. The annotation files made public in (Lee et al., 2021) were first converted from Brat Annotation standoff format (*.ann* files) along with their corresponding news articles (*.txt* files) to *json* format. Each sentence in the dataset was parsed using Stanford CoreNLP toolkit, including sentence splitting, tokenization, POS-tagging, NER-tagging, and dependency parsing to generate dependency parse trees. For input to the model, we adopt the "multi-channel" strategy (shown in "1" in Figure 5) by concatenating three components listed below. Let $W = w_1, w_2, ...w_n$ be a sentence of length n where w_i is the *i*-th token:

1. The word embedding vector of w_i: this is the feature representation from a word embedding of ComBERT. It is made up of WordPiece
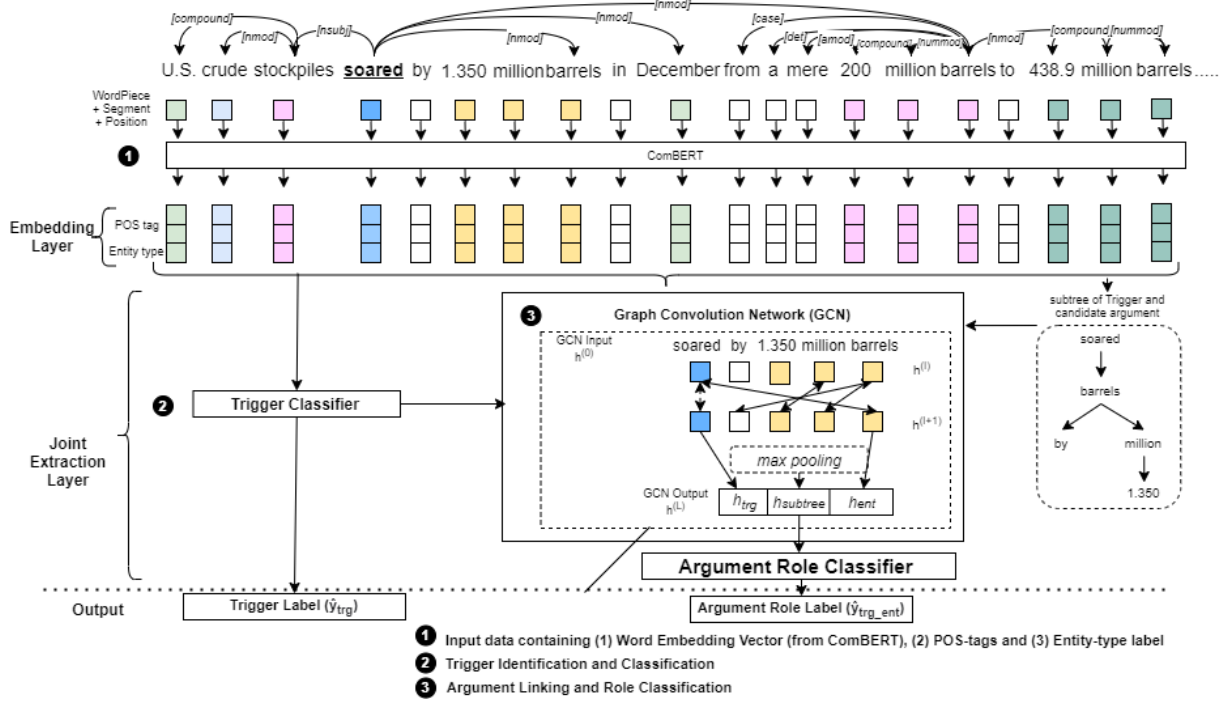
Figure 5: Architecture of the framework

tokenization (Wu et al., 2016) with [CLS] and [SEP][8] are placed at the start and end of the sentence.

2. The Part-of-Speech-tagging (POS-tagging) label embedding vector of w_i: This is generated by looking up the POS-tagging label embedding.

3. The entity type label embedding vector of w_i: Similar to the POS-tagging label embedding vector of w_i, entity mentions in a sentence were annotated using BIO annotation schema, and the entity type labels were transformed to real-valued vectors by looking up the entity label embedding.

4.4.1 Trigger Candidate and Argument Joint-Extraction

The experiments in this paper were conducted based on the joint-extraction approach where event trigger and arguments extraction are trained together.

Event Trigger Extraction. The event trigger extraction is setup as a token classification task, similar to that of Named Entity Recognition. The trigger classifier (shown as "2" in Figure 5) is a simple multi-layer perceptron (MLP) with a single

hidden layer where each token within the input sentence is classified into one of 19 classes (18 event types and 'NONE' for non-event). All predicted event triggers are added to a list, $\widehat{y}_{trg}$ where $\widehat{y}_{trg} = t_1, t_2, ...t_j$.

Event argument extraction. For the second subtask, the event argument extraction task is setup as a sequence classification task. Candidate arguments are selected from the pool of entity mentions within the sentence. Each candidate argument will be paired with a candidate trigger for argument role classification. The classifier will classify each trigger-entity pair into one of 20 classes (19 argument roles and 'NONE' for entities with no links to the candidate trigger).

In the experiments, gold-standard entity mention E were used, $E = e_1, e_2, ...e_k$ where k is the number of the entity mentions in a sentence. With the list of predicted candidate triggers $\widehat{y}_{trg}$ with j number of triggers, and the list of golden entity mentions E with k number of entity mentions, we pair each candidate trigger with an entity, resulting in $j \times k$ number of pairs. For the pair $t_x e_y$, the task is to classify the argument roles entity e_y plays in event t_x.

---

[8][CLS], [SEP], [MASK] are special tokens of BERT. For experiments involving RoBERTa, Byte-Pair Encoding (BPE) tokenization and its special tokens are used.

4.4.2 Graph Convolutional Networks over Dependency Tree

A sentence's syntactic parse tree can be seen as a directed graph. Let $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$ be the dependency parse tree for the sentence w with $\mathcal{V}$ and $\mathcal{E}$ as the sets of nodes and edges of $\mathcal{G}$ respectively. $\mathcal{V}$ contains n nodes corresponding to the n tokens $w_1, w_2, ..., w_n$ in w. Each edge $(v_i, v_j) \in \mathcal{E}$ is directed from the head word w_i to the dependent word w_j) with the Universal Dependency (UD) relation tags. Given a sentence's dependency parse tree with with n nodes, we convert each tree into its corresponding $n \times n$ adjacency matrix A with the following modifications:

1. Treating the dependency graph as undirected, i.e $\forall i, j, A_{i,j} = A_{j,i}$, where $A_{i,j} = A_{j,i} = 1$ if there is a dependency edge between tokens i and j;
2. Adding self-loops to the each node in the graph, following (Kipf and Welling, 2017): $\widetilde{A} = A + I$ with I being the $n \times n$ identity matrix

Stacking a GCN layer L times gives us a L-layer GCN where L is a hyperparameter of the model. During graph convolution at each layer l, each node gathers and summarizes information from its connected nodes ($\widetilde{A}_{i,j} = 1$) in the graph. We set $h^{(0)}$ as the input word vectors for an L-level GCN network and $h^{(L)}$ as the output word representations. The graph convolution operation of a single node, node i at level l of the GCN is as follows:

$$h_i^{(l)} = \sigma(\sum_{j=1}^{n} \widetilde{A}_{ij} W^{(l)} h_j^{l-1}/d_i + b^{(l)}) \quad (1)$$

where $h_i^{(l-1)}$ is the input vector, $h_i^{(l)}$ denotes the collective hidden representations, $W^{(l)}$ is the weight matrix, $b^{(l)}$ is a bias term, σ is the sigmoid activation function and $d_i = \sum_{j=1}^{n} \widetilde{A}_{ij}$, is the number of arches in the resulting graph.

4.4.3 Argument Role Classification with GCN

This section describes the operations shown as "3" in Figure 5.

Encoding Trigger-Entity Pair. Given a trigger-entity pair $t_x e_y$, we prune the dependency tree to obtain the contextual sub-tree between trigger t_x and entity e_y based on Algorithm 1. The nodes of

this sub-tree form the input word vectors $h^{(0)}$ to the L-layer GCN network.

The subtree representation after L times of graph convolution is obtained as follows:

$$h_{subtree} = f(h^{(L)}) = f(GCN(h^{(0)})) \quad (2)$$

where $h^{(L)}$ is the output word representations produced by the L-layer GCN network and f is a max-pooling function that maps the input to the subtree vector, $h_{subtree}$. Besides the subtree representation, we also obtained a representation h_{trg} for trigger and h_{ent} for entity:

$$h_{trg} = f(h_t^{(L)}), h_{ent} = f(h_e^{(L)}) \quad (3)$$

Besides max-pooling, we have also experimented with average-pooling and sum-pooling to obtain the final vector for all three vectors (substree, trigger and entity). All three vectors are then concatenated into a vector which is then propagated through a fully-connected layer to classify the argument role:

$$\widehat{y}_{trg\_ent} = g(W_a[h_{subtree}; h_{trg}; h_{ent}] + b_a) \quad (4)$$

where g is the *softmax* operation to obtain a probability distribution over argument roles. $\widehat{y}_{trg\_ent}$ is the final output of the role the entity *ent* plays in the event triggered by the trigger candidate *trg*.

4.4.4 Loss Function for Joint-extraction

Event trigger and arguments extraction are jointly trained by minimizing the cross-entropy loss function.

$$\mathcal{L}_{joint} = \mathcal{L}_{trg} + \beta(\mathcal{L}_{arg}) \quad (5)$$

where β is the weightage placed on the loss of argument extraction task. In the experiment we use the value of 2, training the model with double the weightage on argument extraction.

5 Experiment Setup

Parameter settings. The data is split into 70% for training and 30% for testing. For all the experiments, the word embedding is of size 768 dimensions (same as `bert-base-cased`) while 50 dimensions for the other two embeddings - POS-tag embedding and entity-type embedding. For the GCN module, we use a two-layer GCN ($L = 2$) with a batch size of 4. The model is trained using the Cross-entropy loss function and Adam optimizer.

Method	Trigger Identification (%)			Trigger Classification (%)			Argument Identification (%)			Argument Role (%)		
	P	R	F1	P	R	F1	P	R	F1	P	R	F1
Sequence representation (A)	0.81	0.82	0.72	0.75	0.71	0.70	0.71	0.70	0.7	0.60	0.62	0.62
JMEE (B)	0.91	0.92	0.92	0.89	0.91	0.90	0.79	0.79	0.78	0.72	0.76	0.75
GCN with full tree (C)	0.92	0.97	0.95	0.92	0.94	0.92	0.80	0.81	0.81	0.74	0.75	0.73
GCN with LCA sub-tree (D)	0.92	0.96	0.94	0.93	0.95	0.94	0.89	0.90	0.89	0.87	0.86	0.85
GCN with contextual sub-tree (E)	0.93	0.98	0.95	0.91	0.97	0.93	**0.92**	**0.91**	**0.92**	**0.90**	**0.89**	**0.90**

Table 3: Comparing results (Precision, Recall, and F1 scores) across various methods with gold-standard entity mentions.

Models Settings. The architecture and setup for models listed in Table 3 are as follows:
(1) **Model A** - The embedding of trigger and candidate argument (from ComBERT) are concatenated and fed into a Bi-LSTM, which is then fed into a classifier with one fully connected (FC) layer.
(2) **Model B** - Jointly Multiple Events Extraction via Attention-based Graph Information Aggregation (JMEE) as presented in (Liu et al., 2018)[9].
(3) **Model C** - GCN with Full Tree uses the full dependency tree, $h_{fulltree}$. The same convolution operations are done on $h_{fulltree}$ in the place of $h_{subtree}$.
(4) **Model D** - GCN with LCA sub-tree with shortest dependency path between trigger candidate and entity candidate.
(5) **Model E** - GCN with contextual sub-tree, this setup is described in the Proposed Solution section.

6 Results and Analysis

6.1 Trigger Classification

As shown in Table 3, Trigger Identification and Trigger Classification achieve rather high F1 scores in all experiments regardless of whether full dependency tree or sub-tree, sequential or syntactic representation approach.

6.2 Argument Role Classification

From the results shown in Table 3, it can be concluded that syntactic representation (**Model C, D, E)** of a sentence yields better event extraction results. The results of Model B and C are not as good as using sub-tree because the full dependency tree contains unnecessary and noisy information that is not helpful in argument role classification.

---

[9]This was developed for the ACE2005 dataset.

As for (Liu et al., 2018) (Model B), it did not produce the best results because it was designed for capturing the association between multiple events within a sentence via the attention mechanism. The events in the commodity news dataset do not exhibit the same strong association as the events in ACE2005 dataset. Model D uses the LCA sub-tree that has only the "bare minimum" information, while Model E contains additional crucial context information that has proved to be useful in argument role classification.

Table 4 presents the breakdown of Argument Classification by Argument Types to fully provide evidence to the effectiveness of the proposed solution on the corpus, which exhibits the characteristic of arguments homogeneity. It is shown clearly that arguments of the same entity type, for example, *Final_value*, *Initial_value* and *Difference* can be better differentiated and classified using a contextual sub-tree that contains the shortest path between an event trigger and its event argument as well as crucial off-path information. Symbols (♣, ◇, ♠) in Table 4 indicate the grouping of arguments by entity type.

As for the characteristic of having multiple events in a sentence, the proposed solution is able to detect and classify the events as well as link arguments to their rightful event, as shown in both Table 3 and Table 4.

6.3 Comparing Word Embedding and Pre-trained Language Models

Model E in Table 5 was further experimented using GloVe (Pennington et al., 2014) and other pre-trained language models namely BERT (Devlin et al., 2019b) and RoBERTa (Liu et al., 2019). These were compared against ComBERT.

Argument Roles	Entity Type	Argument Role Classification F1 Score				
		Model A	Model B	Model C	Model D	Model E
NONE	-	0.84	0.84	0.90	0.91	**0.94**
Attribute	Financial Attribute	0.40	0.65	0.79	0.75	**0.83**
Item	Economic Item	0.64	0.85	**0.88**	0.85	**0.88**
Final_value ♣	Money / Production unit / Price unit / Percentage / Quantity	0.43	0.39	0.71	0.75	**0.79**
Initial_value ♣	Money / Production unit / Price unit / Percentage / Quantity	0.56	0.56	0.73	0.69	**0.77**
Difference ♣	Money / Production unit / Price unit / Percentage / Quantity	0.58	0.69	0.84	**0.89**	**0.89**
Reference_point ♢	Date	0.54	0.69	**0.80**	0.71	**0.80**
Initial_reference_point ♢	Date	0.40	0.63	0.63	0.60	**0.66**
Contract_date ♢	Date	0.52	0.54	0.70	0.66	**0.80**
Duration	Duration	0.55	0.55	0.75	0.82	**0.84**
Type	Location	0.52	0.59	0.70	0.68	**0.76**
Imposer ♠	Country / State or province	0.71	0.69	**0.81**	0.79	**0.81**
Imposee ♠	Country / State or province	0.50	0.49	0.60	**0.68**	**0.68**
Place ♠	Country / State or province	0.58	0.69	**0.74**	0.60	**0.74**
Supplier_consumer ♠	Country / State or provience / Nationality / Group	0.49	0.71	0.73	0.73	**0.79**
Impacted_countries ♠	Country	0.42	0.69	0.72	0.70	**0.76**
Participating_countries ♠	Country	0.65	0.75	0.78	0.83	**0.89**
Forecaster	Organization / Group	0.62	0.75	0.78	0.80	**0.82**
Forecast	Forecast_Target	0.61	0.61	0.83	0.67	**0.91**
Situation	Phenomenon / Other acitivites	0.57	0.69	**0.73**	0.67	0.66

Table 4: F1-scores for each argument type.

Method	Trigger Classification (%)			Argument Role Class. (%)		
	P	R	F1	P	R	F1
GloVe	0.67	0.70	0.68	0.65	0.66	0.67
BERT	**0.93**	0.93	0.93	0.85	0.83	0.83
RoBERTa	0.89	0.96	0.93	0.86	0.85	0.83
ComBERT	0.91	**0.97**	**0.93**	**0.90**	**0.89**	**0.90**

Table 5: Comparing Word Embedding and Pre-trained Language Models for Model E

From the results in Table 5, it is shown that ComBERT produced the best result, further proving that a contextualized token representation helps boost the performance of event extraction. ComBERT is used in all models listed in Table 4.

7 Conclusion

This paper presents a graph-based deep learning framework to extract commodity-related events. This framework is tailored for the purpose of commodity news event extraction based on the new commodity news dataset introduced by (Lee et al.,

2021). Our method addresses specific challenges exhibited by the characteristics of this dataset, in particular: (1) sentences containing lots of numerical information such as price, percentage of change, and dates, (2) entities of similar type playing distinctly different argument roles, and (3) the need for arguments extraction to disambiguate the identified events. The proposed solution uses a Graph Convolutional Network with a contextual sub-tree to extract events effectively. Experimental results demonstrate that the proposed solution outperforms existing solutions with higher F1 scores, particularly in argument role classification. With accurate event extraction from commodity news, the extracted information can be used for other downstream tasks such as learning event chains and event-event relations that can be further exploited for commodity price prediction.

References

Michael W Brandt and Lin Gao. 2019. Macro fundamentals or geopolitical events? a textual analysis of news events for crude oil. *Journal of Empirical Finance*, 51:64–94.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019a. BERT: Pre-training of deep bidirectional transformers for language understanding. 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)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019b. BERT: pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Xiao Ding, Yue Zhang, Ting Liu, and Junwen Duan. 2014. Using structured events to predict stock price movement: An empirical investigation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1415–1425.

Liat Ein-Dor, Ariel Gera, Orith Toledo-Ronen, Alon Halfon, Benjamin Sznajder, Lena Dankin, Yonatan Bilu, Yoav Katz, and Noam Slonim. 2019. Financial event extraction using Wikipedia-based weak supervision. In *Proceedings of the Second Workshop on Economics and Natural Language Processing*, pages 10–15, Hong Kong. Association for Computational Linguistics.

Alexander Hogenboom, Frederik Hogenboom, Flavius Frasincar, Kim Schouten, and Otto Van Der Meer. 2013. Semantics-based information extraction for detecting economic events. *Multimedia Tools and Applications*, 64(1):27–52.

Frederik Hogenboom, Flavius Frasincar, Uzay Kaymak, Franciska De Jong, and Emiel Caron. 2016. A survey of event extraction methods from text for decision support systems. *Decision Support Systems*, 85:12–22.

Thomas N. Kipf and Max Welling. 2017. Semi-supervised classification with graph convolutional networks. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

Meisin Lee, Lay-Ki Soon, Eu-Gene Siew, and Ly Fie Sugianto. 2021. An annotated commodity news corpus for event extraction.

Xiao Liu, Zhunchen Luo, and Heyan Huang. 2018. Jointly multiple events extraction via attention-based graph information aggregation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1247–1256, Brussels, Belgium. Association for Computational Linguistics.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Hassan H Malik, Vikas S Bhardwaj, and Huascar Fiorletta. 2011. Accurate information extraction for quantitative financial events. In *Proceedings of the 20th ACM international conference on information and knowledge management*, pages 2497–2500.

Thien Huu Nguyen and Ralph Grishman. 2018. Graph convolutional networks with argument-aware pooling for event detection. In *AAAI*, volume 18, pages 5900–5907.

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

Swarnadeep Saha, Harinder Pal, et al. 2017. Bootstrapping for numerical open ie. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 317–323.

Roger C Schank and Robert P Abelson. 2013. *Scripts, plans, goals, and understanding: An inquiry into human knowledge structures*. Psychology Press.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *CoRR*, abs/1609.08144.

Wei Xiang and Bang Wang. 2019. A survey of event extraction from text. *IEEE Access*, 7:173111–173137.

Boyi Xie, Rebecca J. Passonneau, Leon Wu, and Germán G. Creamer. 2013. Semantic frames to predict stock price movement. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 873–883, Sofia, Bulgaria. Association for Computational Linguistics.

Hang Yang, Yubo Chen, Kang Liu, Yang Xiao, and Jun Zhao. 2018. DCFEE: A document-level Chinese financial event extraction system based on automatically labeled training data. In *Proceedings of ACL 2018, System Demonstrations*, pages 50–55, Melbourne, Australia. Association for Computational Linguistics.

Sen Yang, Dawei Feng, Linbo Qiao, Zhigang Kan, and Dongsheng Li. 2019. Exploring pre-trained language models for event extraction and generation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics*, pages 5284–5294.

Yuhao Zhang, Peng Qi, and Christopher D. Manning. 2018. Graph convolution over pruned dependency trees improves relation extraction. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 2205–2215, Brussels, Belgium. Association for Computational Linguistics.

A Commodity News Dataset

A.1 Event Types

There are 18 event types that can be grouped into four main categories: (1) macro-economic, (2) geo-political, (3) supply-demand-related, and (4) commodity price movement events. In-depth details about the event types such as the categories they belong to, description, and as well as more example sentences are found in (Lee et al., 2021).

Event type	Examples
1. Cause-movement-down-loss	The pandemic has **zapped** demand to a level never seen before...
2. Cause-movement-up-gain	IEA tried to **boost** global oil demand by introducing.....
3. Civil-unrest	.....a fragile recovery in Libyan supply outweighed **fighting** in Iraq ......
4. Crisis	.....Ukraine declared an end to an oil **crisis** that has .....
5. Embargo	.....prepared to impose " strong and swift " economic **sanctions** on Venezuela...
6. Geopolitical-tension	....despite geopolitical **war** in Iraq , Libya and Ukraine.
7. Grow-strong	.....as **strong** U.S. employment data.....
8. Movement-down-loss	....further **decreases** in U.S. crude production.....
9. Movement-flat	U.S. crude is expected to **hold** around $105 per barrel.
10. Movement-up-gain	It expects consumption to **trend upward** by 1.05 million bpd.
11. Negative-sentiment	....due to **concern** about softening demand growth and awash in crude.
12. Oversupply	....the market is still working off the **gluts** built up.....
13. Position-high	Oil price remained close to four-year **highs**....
14. Position-low	Oil slipped more than 20% to its **lowest level** in two years on 1980s...
15. Prohibiting	U.S. has **prohibit** the sale of oil from Venezuela.
16. Shortage	......and there is no **shortfall** in supply , the minister added.
17. Slow-weak	U.S. employment data **contracts** with the euro zone....
18. Trade-tensions	... escalating global **trade wars**, especially between the US and China.

Table 6: Event types with examples.

A.2 Event Schema

Each event has its own set of event arguments. Below is the complete list of arguments for the event **movement_up_gain**. The argument text of this table is populated using the example sentence presented in Figure 1. For the complete list of event schemas refer to (Lee et al., 2021).

Argument Role	Entity Type	Argument Text
Type	Nationality, Location	
Supplier_consumer	Organization, Country, State_or_province, Group, Location	U.S.
Reference_point_time	Date	Tuesday
Initial_reference_point	Date	
Final_value	Percentage, Number, Money, Price_unit, Production_unit, Quantity	438.9 million barrels
Initial_value	Percentage, Number, Money, Price_unit, Production_unit, Quantity	200 million barrels
Item	Commodity, Economic_item	crude
Attribute	Financial_attribute	stockpiles
Difference	Percentage, Number, Money, Production_unit, Quantity	1.350 million barrels
Forecast	Forecast_target	
Duration	Duration	
Forecaster	Organization	

Table 7: Event Schema for the event **movement_up_gain**.

A.3 Argument Roles and Entity Types

Table 8 is a repeat of Table 4 showing the association of various entity types to each argument roles. The table below provides examples of entity mention for each argument roles.

Argument Roles	Entity Type	Examples
1. NONE	-	-
2. Attribute	Financial Attribute	*supply, demand, output, production, price, import, export*
3. Item	Economic Item	*economy, economic growth, market, economic outlook, employment data, currency, commodity-oil*
4. Final_value	Money / Production unit / Price unit / Percentage / Quantity	*$60, USD 50, 170,000 bpd, 400,000 barrels per day, 1 million barrels, $40 per barrel, USD58 per barrel, 0.5%*
5. Initial_value	Money / Production unit / Price unit / Percentage / Quantity	`-same as above-`
6. Difference	Money / Production unit / Price unit / Percentage / Quantity	`-same as above-`
7. Reference_point	Date	*1998, Wednesday, Jan. 30, the final quarter of 1991, the end of this year*
8. Initial_reference_point	Date	`-same as above-`
9. Contract_date	Date	`-same as above-`
10. Duration	Duration	*two years, three-week, 5-1/2-year, multiyear, another six months*
11. Type	Location	*global, world, domestic*
12. Imposer	Country / State or province	*China, Iran, Iraq, Washington, Moscow, Cushing, North America, Europe*
13. Imposee	Country / State or province	`-same as above-`
14. Place	Country / State or province	`-same as above-`
15. Supplier_consumer	Country / State or provience / Nationality / Group	`-same as above-` + *OPEC, non-OPEC countries, American, Russian*
16. Impacted_countries	Country	*China, U.S. Russia, Iran, Iraq*
17. Participating_countries	Country	`-same as above-`
18. Forecaster	Organization / Group	*OPEC, European Union, U.S. Energy Information Administration*
19. Forecast	Forecast_Target	*forecast, target, estimate, projection, bets*
20. Situation	Phenomenon / Other activities	`free text`

Table 8: List of Event Argument Roles and their corresponding entity types.