IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)

An International Open Access, Peer-reviewed, Refereed Journal

CAUSAL EVENT EXTRACTION

[1]R.C. Dyana Priyatharsini, [2]R. Charan, [3]Rohit Waman Chavan, [4]Rishikesh Kumar, [5]R. Srikanth

[1]Assistant professor, School of Computing, Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai, India- 600073.

[2345]Student, School of Computing, Department of Computer Science and Engineering, Bharath Institute of Higher Education and Research, Chennai, India- 600073.

Abstract: The cause of the manner of causal extraction in herbal language is detection and Drawing motive and impact relationships among events or moves in the text. This purpose is to automatically become aware of useful cause-impact relationships Many applications which include know-how map creation and facts retrieval. Advanced deep studying algorithms together with LSTM and Causal BERT are also wanted. A style of techniques together with semantic coding. Become built a representation of motive and effect relationships between activities is the causal effect of the occasion approach of extraction. Because of the richness of natural language, extracting purpose and effect relationships is a tough task. A complex operation, but it can offer high-quality data the courting among occasions and activities. Supervised algorithms can extract cause-and-impact conclusions because they do not exist Training requires labeled records. But the set of rules does not paintings. They are performed and monitored on labeled records. We analyze the corpus created by using extending SemEval annotations. 2010 Question eight is given within the exam. We extract all reasons and consequences from the texts and keys of herbal languages Proper identification of "C" (purpose), "E" (effect) and tags is a critical a part of our work. "Emb" (Embedded Causation), means the semantic characteristic of causal events.

Keywords: Causal Event Extraction, LSTM, Language

I. INTRODUCTION

The mission of figuring out purpose and impact relationships between occasions in herbal text the linguistic technique is referred to as causal occasion retrieval. Because it allows Identify enormous cause and effect relationships among events to be used for choice making or further evaluation that is a very crucial assignment Information Extraction and Text Mining. Natural language processing methods Including element-of-speech tagging, dependency evaluation and component-covering Tagging is used to pick out gadgets and hyperlinks in textual content. Draw a reason and effect courting. Here are methods you could assist the matters within the text and the conjunctions that meet their purpose. We have exclusive tactics to locating reasons and extracting relationships and effects. Between texts of course. Using learning gadget learning models Annotation is a technique of identifying purpose and impact relationships in datasets. Linkage of products. Another approach is rule-based totally additionally, it seems at reason and effect relationships between users Set policies and examples. Finding and extracting cause and effect relationships of path use examples of strategies between templates in both eventualities Adaptation, organizational reputation, and dating extraction. There is information to be extracted, issues to be solved and choices to be made. Examples of applications for developing motive and effect relationships. Let's use that as an instance to decide the causes of scientific activities for analysis and treatment Diseases or reasons of forecasting marketplace financial tendencies.

II. OBJECTIVE

Determining motive and impact relationships of activities and expressing pairs of related occasions. And the purpose and impact relationship among them. This consists of growing automated techniques that are correct in extracting purpose and impact conclusions Identifying and extracting purpose and impact relationships from large volumes of text.

III. RELATED WORK

The version in Article [1] makes use of an independent bivariate design Transformed with LSTM-CRF (Conditional Random Field) embedding's Identify motive and impact relationships from the text. Bivariate sampling uses LSTM to version. Context and attention to choose up words and expressions. Guaranteed it is used to extract the impact of case coherence and CRF layer relationships. Upgrade The first-rate of phrase symbols additionally uses the interpretation of version concepts. The reason of the version is to separate causality into two According to the killed information.

Paper [2] dataset and version for computerized extraction of motive and impact relationships. Clinical capabilities are offered in the observe as MIMI Cause. MIMI-III 7,794 annotated databases include the dataset. Convolutional Neural Network networks (CNN) and lengthy-brief-time period reminiscence networks (LSTM). In the version, to express relationships between clinical standards and causation. To do It also makes use of a model to jointly expect the type and course of causality Multiple study techniques. The assessment results show that the proposed model outperforms present procedures to show motive and effect relationships from a medical textual content. Proposed transfer method and MIMI Cause dataset this will facilitate medical doctor decision-making and enhance the excellent of affected person care. Results

Article [3] studies proposes a deep model of an automatic neural network extracting purpose and effect relationships from linguistically knowledgeable text. Officer Accurate extraction of cause and impact relationships, syntactic composition and Semantic information in the form of dependency bushes and WorldNet hyperonyms. A model uses a bipartite LSTM network to capture contextual data a softmax classifier is used to expect the lifestyles and nature of purpose and impact relationships. Correlation The proposed model outperforms modern day models. Methods for extracting motive and effect relationships based on evaluation outcomes According to the killed statistics. This concept will have many packages; consisting of textual content mining, statistics extraction and expertise acquisition; It's clean to accept.

Article [4] suggests the proposed technique of automatically extracting the method Drug-associated activities (ADE) from scientific organic tissues. Officer ADE makes use of a unique extraction technique, external facts Resources which include clinical ontologies, dictionaries and semantic networks. It is called the identical object as the relation extraction module is used by the machine initially to know the medicines and associated sicknesses, their Correlation So this device makes use of formal rule-primarily based understanding ADES may be primarily based on information assets and found drug-disorder associations. The set of rules then uses a device learning method to set ADs are divided into several companies in line with severity and chance. The effects of the assessment of the simple facts show the targets to offer higher than expertise based ADE extraction technique Latest era. Technology can help pharmacovigilance Work for better fitness and be affected person

Article [5] discusses tiers of Graph Convolutional Network (GCN) based paper A derivation of motive-effect relationship model is proposed. It uses the GCN version to seize Use syntactic and semantic relationships between phrases in a sentence these representations represent feasible motive and impact relationships. It is believed Causal relationships are refined the use of the second GCN level version. Run away with false positives. Develop accurate cause and effect relationships Extraction, get admission to understanding also includes outside goals Resources in the shape of scientific ontologies. The proposed version performs nicely Modern techniques of extracting motive and effect relationships, as proven via evaluation consequences According to the killed facts. Many packages including pharmacy tracking; Journal of Decision Making and Biomedical Research might also benefit precedent

Article [6] gives a detailed evaluation of the observe of advanced medical technology. Text occasion extraction technique. The trouble of extracting mass and so forth item recognition, stimulus occasion identification, and the argument about extraction is placed first within the article. He then discusses the take a look at various deep mastering frameworks which include Convolutional Neural Networks (CNN), Recurrent Neural Networks (RNN) and their variants are long Short Term Memory (LSTM) networks and Gate Repetition Units (GRU), which they are sure to return returned. The have a look at additionally examines how deep the studying is Models may be related to diverse outside facts sources Lexicons, ontologies and pre-trained language models must make matters better Extraction of results. The document concludes with a summary of several problems. Extraction packages such as organic research, herbal language Managing and studying social networks. Working researchers and clinicians of route, the usage of deep mastering strategies to extract from textual content can carry splendid benefits From Bologna

Article [7] provides an in depth and in-intensity evaluation of inspection technologies. Methods for extracting motive and effect relationships from biomedical textual content. Question Extraction of motive and impact relationships and lots of other responsibilities, incl Knowledge of purpose and impact of things and determination of course Cause and impact courting for the first time at paintings. Hence the thing makes diverse analyses deep getting to know frameworks along with Convolutional Neural Networks (CNN); Recurrent Neural Networks (RNN) and Graph Convolutional Networks (GCN); to explicit motive and effect relationships. The file also talks about integration Deep learning models for developing medical ontologies and dictionaries Extracting reason and impact relationships. The article concludes through offering there are many apps for extracting causes like Maecenas Pharmacovigilance, clinical selection making and biological studies. To Researchers and artists working on deep learning primarily based on reason. A survey is a useful tool for extracting relationships from biomedical textual content.

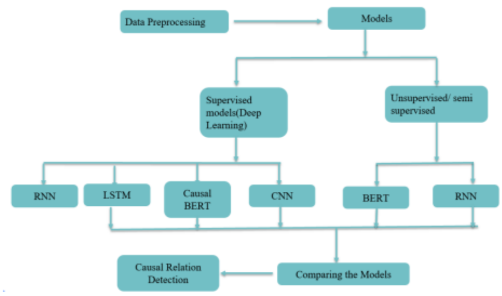
Article [8] introduces the SSA-UO look at, a totally new and unexpected approach. To understand the evaluation of Twitter data. It uses impartial mode Organization diagram (SOM) to organization words in line with frequency as one among a massive variety of tweets. Proximity Pair words as predetermined seed words with a demonstration of sense. Then use that sentence in base words. Officer With an extremely good sense of analysis, he additionally makes use of the ideal new word for design A semantic disambiguation technique based at the distribution of similarity terms. The consequences of assessment of baseline datasets display that SSA-UO Outperforms contemporary unsupervised sensitivity detection algorithms Twitter Analytics. The proposed approach is useful for plenty people Applications which include brand popularity control, patron remarks Analysis and tracking of social networks.

Article [9] placed college students into organizations with special dropout danger The take a look at suggests that classes are primarily based on the density of implicit mastering The art algorithm makes use of area to create a multi-dimensional function Student statistics such as demographics, performance and engagement The answers algorithm then makes use of a density-primarily based clustering approach to discover them Groups of students with positive behaviors that present a danger to each group Depending at the extent to which it is possibly to occur. Proposed the benefit of this method is that it does not require supervision can be used efficiently to kill named statistics and big records. Proposed this method is superior to different approaches currently used to assess dropout threat preaching according to results evaluated by means of student testimony. . The proposed guidelines will assist educational institutions to discover danger groups. They assist college students improve their instructional fulfillment behavior

Article 10 inside the article is a way for robotically extracting causal relationships Links from biomedical texts are counseled. Using throwing strategies Traditional manager algorithms require categorized facts, which may be costly. It's time to attend to it in order that they spontaneously collect purpose and impact relationships; Teachers use texts. The steps within the technique are as follows: Drawing, drawing format and model order. Syntax and Semantics Analytical strategies are used to locate styles that propose cause and effect relationships. These sorts of conversation are then common and thru it is used to create and extract guests. That does the trick Better without the want to label the facts whilst checking out for importance Welcome to the textual content dataset.

IV. SYSTEM ARCHITECTURE

The complete dataset was divided into checking out and education. Information the password is pre-processed from the dataset such as stop word elimination after the pre-processing, we started to create the fashions. Here is our plan 6 examples have been built. Four copies are given in Govt. And in 2 fashions uncontrolled records. We have advanced supervised fashions of RNN, LSTM, Causal BERT and CNN. We advanced BERT and RNN based on unsupervised models. Then we positioned the whole thing together we took the version very accurately and exactly and used this prototype He anticipated the event. In all of the above models, the variety is higher compared to the accuracy with all fashions. So, the use of CNN model, we predicted the result. Finally, if we provide Condemnation with organization predicts the relationship among the 2 all fashions with essence structure are without a doubt explained beneath.



SYSTEM REQUIREMENTS

Hardware Requirements:

- 8 GB RAM or higher
- 512 GB SSD ROM or higher
- Processor: Intel i5 8th gen, AMD Ryzen 5 or higher

Software Requirements

- Python
- Windows 10

MODULES AND THEIR DESCRIPTION

The test data and the train data is used to test and train the Models

```
train_df
```

	label	sentence
0	Component-Whole(e2,e1)	the system as described above has its greatest...
1	Other	the e1start child e1end was carefully wrapped ...
2	Instrument-Agency(e2,e1)	the e1start author e1end of a keygen uses a e2...
3	Other	a misty e1start ridge e1end uprises from the e...
4	Member-Collection(e1,e2)	the e1start student e1end e2start association ...
...	...	...
7995	Other	when the e1start notice e1end is sent by e2sta...
7996	Entity-Origin(e1,e2)	the e1start herbicide e1end is derived from a ...
7997	Entity-Destination(e1,e2)	to test this we placed a kitchen e1start match...
7998	Other	the farmers and city officials in the region h...
7999	Product-Producer(e2,e1)	the e1start surgeon e1end cuts a small e2start...

8000 rows × 2 columns

```
test_df
```

	label	sentence
0	Message-Topic(e1,e2)	the most common e1start audits e1end were abou...
1	Product-Producer(e2,e1)	the e1start company e1end fabricates plastic e...
2	Instrument-Agency(e2,e1)	the school e1start master e1end teaches the le...
3	Entity-Destination(e1,e2)	the suspect dumped the dead e1start body e1end...
4	Cause-Effect(e2,e1)	avian e1start influenza e1end is an infectious...
...	...	...
2712	Instrument-Agency(e2,e1)	after seating all the idols which itself takes...
2713	Product-Producer(e1,e2)	the minister attributed the slow production of...
2714	Component-Whole(e2,e1)	the e1start umbrella e1end e2start frame e2end...
2715	Product-Producer(e1,e2)	manos the hands of fate is a lowbudget horror ...
2716	Entity-Destination(e2,e1)	a few days before the service tom burris had t...

2717 rows × 2 columns

LSTM Model Implementation

The first implementation of the LSTM version. It does this by using taking enter and alerts to extract phrases Scoring is pre-figuring out the weightage of words. Intellectual facts from Wikipedia or many other resources of information kept inside the glove box It is described. Therefore, data is indicated and predicted. LSTM is similar to RNN. But in LSTM the weights are remembered however in RNN the gate is forgotten He did not take into account the burden of his earlier phrases.

```
class RelationClassification_Model2(nn.Module):

    def __init__(self, embedding_size, hidden_size, vocab_size, num_layers,class_num,pretrained_embedding,pad_idx):
        super(RelationClassification_Model2, self).__init__()

        self.hidden_size = hidden_size
        self.embedding =  nn.Embedding.from_pretrained(pretrained_embedding,freeze = True,padding_idx = pad_idx)
        self.lstm = nn.LSTM(embedding_size, hidden_size, num_layers,bidirectional = True)
        self.linear = nn.Linear(2*hidden_size, class_num)
        self.softmax = nn.LogSoftmax(dim=1)
        self.dropout = nn.Dropout(0.7)

    def forward(self, X, mask_e1, mask_e2):
        embeddings = self.dropout(self.embedding(X))
        hiddens, _ = self.lstm(embeddings)
        x = hiddens.view((hiddens.shape[0], hiddens.shape[1], 2, self.hidden_size))
        e1, e2 = max_avg_pooling(x,mask_e1,mask_e2)
        concats = torch.cat((e1, e2), dim = 1)

        outputs = self.softmax(self.linear(concats))

        return outputs
```

```
def max_avg_pooling(x,mask_e1,mask_e2):
    e1 = torch.zeros((x.shape[1],x.shape[3])).to(device)
    e2 = torch.zeros((x.shape[1],x.shape[3])).to(device)

    for i in range(x.shape[1]):
        sample = x[:,i,:,:].to(device)
        y = sample[mask_e1[:,i]].to(device)
        avg = torch.zeros((1,x.shape[3])).to(device)
        for j in  range(y.shape[0]):
            avg += (torch.maximum(y[j,0,:],y[j,1,:]) - avg).to(device) / (j + 1)
        e1[i,:] = avg

        y = sample[mask_e2[:,i]].to(device)
        avg = torch.zeros((1,x.shape[3])).to(device)
        for j in  range(y.shape[0]):
            avg += (torch.maximum(y[j,0,:],y[j,1,:]) - avg).to(device) / (j + 1)

        e2[i,:] = avg
```

```
torch.manual_seed(2045)

int_2_cat_class = dict(enumerate(train_df['label'].cat.categories))

# Training and validation Data Generators
training_set = SemEval2010_task8_Dataset2(train_df,partition['train'],vocab)

pad_idx = 1917494

# Parameters
params = {'batch_size': 256,
          'shuffle': True,
          'num_workers':2,
          'collate_fn' : BatchPadCollate2(pad_idx=pad_idx)
          }

training_generator = torch.utils.data.DataLoader(training_set, **params)

validation_set =  SemEval2010_task8_Dataset2(train_df,partition['validation'],vocab)
validation_generator = torch.utils.data.DataLoader(validation_set, **params)

# Hyperparameters
embedding_size = 300
hidden_size = 150
vocab_size = len(vocab_df)
num_layers = 2
class_num = len(int_2_cat_class)
learning_rate = 5e-3
num_epochs = 15

# Model
model3 = RelationClassification_Model2(embedding_size, hidden_size, vocab_size, num_layers,class_num, embedding,pad_idx).to(device)

# Define Loss function and Optimizer
criterion = nn.NLLLoss(ignore_index = pad_idx)
optimizer = optim.Adam(model3.parameters(), lr = learning_rate, weight_decay= 1e-3)
# scheduler = MultiStepLR(optimizer, milestones=[30,80], gamma=0.5)

train_losses3 = []
validation_losses3 = []

train_accs3 = []
val_accs3 = []
```

RNN Model Implementation

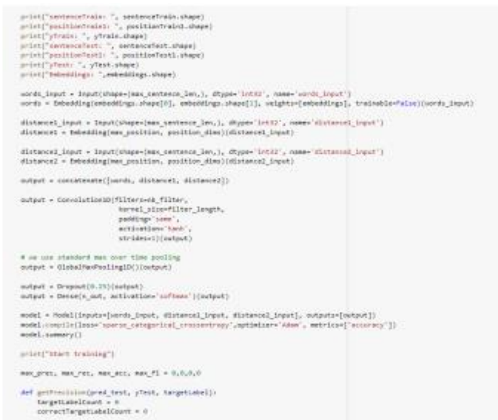
It talks about enforcing an RNN version the usage of methods. Like a glove masking the first pre-processed text enter, replaces the words a vector of numbers. Then, a neural network (RNN) is dedicated to serial records processing, A vector of words belongs to the list. At every time step, the hidden layer of the RNN updates itself the country takes under consideration both the present day enter and the preceding hidden country. Consequently, the model can seize the context and relationships of the words in the sentence. After the RNN layer, greater complicated features can be extracted from the hidden country Add layers such as tight or dropped layers. Finally, we finish Predicts category probabilities in textual content using a dense layer with softmax implementation Office

Causal BERT Model Implementation

Describe the implementation of the BERT causal model, which first makes use of the BERT tokenizer to gain textual content information with sub words as a part of the pre-processing. After this, we run the Causal BERT layer on consecutive signatures. Each signal methods most effective the left context on the left facet of the right input sequence of the causal BERT layer. For this motive, an instance may additionally contain temporal relationships between the phrases of a sentence. After including extra layers which includes dense layer to extract greater complexes from the latent country, we use the remaining dense layer with softmax activation characteristic to output the expected class chances for the text.

CNN Model Implementation

He talks about the implementation of the CNN model. That word the order vector is first pre-processed after which a layer (CNN) is applied Extract nearby features from the enter array. Slipped via the filters given a listing of phrase vectors, the CNN calculates the layer of the detail among the filter out weights and the local phrase window at any time. As this step is repeated to extract a hard and fast of local gadgets from the enter array various filters. The softmax activation characteristic can be output thru the last dense layer Type of reality predicted within the textual content. Bundled and dense extra layers may be brought to extract greater complicated functions from the extract Local features.



BERT Unsupervised Model Implementation

BERT talks about carvings. We have begun by analysing the text, we want to break it into sub-words using the BERT tokenizer. Unsupervised text category. Then he gave a practice model the following subscript is used for tokenized codes and creates an array Contextual embedding for each signal. Then, we follow embedding using the context sign. Methods which includes K, mode or hierarchical partitioning into indices how many bunches? It allows locating groups of words inside the input textual content. What is said in semantics? Finally, we train and expect the result the use of clustering functions as capabilities.

```
        result = {'cycle':cycle, 'unsup_epoch':epoch, 'class_epoch':class_epoch,
              'cm':cm, 'stats':stats, 'macro_f1':stats['macro_f1'], 'model':model_file}
        results.append(result)
        print("Macro P: %2.4f, R: %3.4f, F1: %0.4f" % (stats['macro_precision'], stats['macro_recall'], stats['macro_f1']))

    num_steps = len(train['labels']) // batch_size

    for class_epoch in range(num_class_epochs):
        print("*"*80)
        print("*** CLASS EPOCH %i ***" % class_epoch)
        print("*"*80)
        for class_step in range(num_steps):
            inputs, targets, labels, lens = DH.classification_batch(batch_size, train['sdps'], train['targets'], train['labels'], offset=class_step)
            class_batch = zip(inputs, targets, labels, lens)
            random.shuffle(class_batch)
            class_batch = zip(*class_batch)
            xent = bert.partial_class_fit(*class_batch)
            if class_step % display_mod == 0:
                m,s = divmod(time()-start, 60)
                h,m = divmod(m, 60)
                print("(%i:%i:%i) s %i/%i, e %i avg class xent loss = %0.4f" % (h,m,s, class_step, num_steps, class_epoch, xent))
            if class_step % valid_mod == 0:
                valid_batch = DH.classification_batch(len(valid['labels']), valid['sdps'], valid['targets'], valid['labels'])
                valid_xent = bert.validation_class_loss(*valid_batch)
                m,s = divmod(time()-start, 60)
                h,m = divmod(m, 60)
                print("="*80)
                print("(%i:%i:%i) s %i/%i, e %i validation avg class xent loss = %0.4f" % (h,m,s, class_step, num_steps, class_epoch, valid_xent))
                print("="*80)

        label_set = set(train['labels'])
        preds = bert.predict(valid_batch[0], valid_batch[1], valid_batch[3])
        cm, stats = eh.confusion_matrix(preds, valid['labels'], label_set)
        print("Saving model...")
        model_file = bert.checkpoint()
        result = {'cycle':cycle, 'unsup_epoch':epoch, 'class_epoch':class_epoch,
              'cm':cm, 'stats':stats, 'macro_f1':stats['macro_f1'], 'model':model_file}
        results.append(result)
        print("Macro P: %2.4f, R: %3.4f, F1: %0.4f" % (stats['macro_precision'], stats['macro_recall'], stats['macro_f1']))
    print("Done")
best = sorted(results, key=lambda x:x['macro_f1'], reverse=True)[0]
print("Best result %2.4f, C:%i, UE:%i, CE:%i" %
    (best['macro_f1'], best['cycle'], best['unsup_epoch'], best['class_epoch']))
```

```
with open('SemEval2010_task8_all_data/train_keys.txt', 'w') as f:
    i = 1
    for label in train['labels']:
        f.write("%s\t%s\n" % (i, int2label[label]))
        i += 1
with open('SemEval2010_task8_all_data/valid_keys.txt', 'w') as f:
    i = len(train['labels'])
    for label in valid['labels']:
        f.write("%s\t%s\n" % (i, int2label[label]))
        i += 1
```

```
valid_batch = DM.classification_batch(len(train['labels']), train['sdps'], train['targets'], train['labels'])
preds = bert.predict(valid_batch[0], valid_batch[1], valid_batch[3], return_probs=False)
with open('SemEval2010_task8_all_data/train_preds.txt', 'w') as f:
    i = 1
    for label in preds:
        f.write("%s\t%s\n" % (i, int2label[label]))
        i += 1
```

RNN Unsupervised Model Implementation

It talks approximately the advent of RNN. They suit the relationships between phrases inside the sentence are captured through the RNN layer. To encrypt we used automated text enter and truncation to go back to the authentic textual content Encoder losses permit education without supervision. Coded A passenger can be ordered into a predetermined quantity of clusters the usage of edges Methods After RNN-based totally auto encoder learns the ideal input sample to transcribe Text this permits finding corporations of texts which have semantic similarity. Finally, as opposed to popular cluster labels, we can assign cluster labels all documents. Where they cohere in articles with a similar distribution of phrases. And the outcome is predictable.

```
num_epochs = 1
batch_size =50
neg_per = 25
num_nearby = 50
nearby_mod = 50
sample_power = .75
DM.scale_vocab_dist(sample_power)
num_steps = DM.num_steps(batch_size)
total_step = 1
save_interval = 30 * 60 # half hour in seconds
save_time = time()

#timing stuff
start = time()
fit_time = 0
nearby_time = 0

for epoch in range(num_epochs):
    offset = 0
    DM.shuffle_data()
    for step , batch in enumerate(DM.batches(batch_size, offset=offset, neg_per=neg_per)):
        if not step: step = offset
        t0 = time()
        loss = drnn.partial_unsup_fit(*batch)
        fit_time = (fit_time * float(total_step) +  time() - t0) / (total_step + 1) # running average
        if step % 10 == 0:
            m,s = divmod(time()-start, 60)
            h,m = divmod(m, 60)
            left = time_left(num_epochs, num_steps, fit_time, nearby_time, start, nearby_mod)
            ml,sl = divmod(left, 60)
            hl,ml = divmod(ml, 60)
            pps = batch_size*(neg_per + 1) / fit_time
            print("(%i:%i:%i) step %i/%i, epoch %i Training Loss = %1.5f :: %0.3f phrases/sec :: (%i:%i:%i) hours left"
                  % (h,m,s, step, num_steps, epoch, loss, pps, hl, ml, sl))
        if (total_step-1) % nearby_mod == 0:
            t0 = time()
            run_validation_test(num_nearby)
            valid_loss = drnn.validation_loss(*DM.validation_batch())
            print("Validation loss: %0.4f" % valid_loss)
            nearby_time = (nearby_time * float(total_step) + time() - t0) / (total_step + 1) # running average

        if (time() - save_time) > save_interval:
            print("Saving model...")
            drnn.checkpoint()
```

```
start = 200
stride = 100
end = start + stride
lowd = TSNE(perplexity=20, n_components=2, init='pca', n_iter=5000)
target_2d = lowd.fit_transform(word_embeddings[start:end])
fig, ax = plt.subplots(figsize=(20,16))
for i, label in enumerate(DM.vocab[start:end]):
    label = "%s" % (label)
    x, y = target_2d[i,:]
    ax.scatter(x, y, color='b')
    ax.annotate(label, xy=(x, y), xytext=(5, 2), textcoords='offset points',
                ha='right', va='bottom')
plt.savefig('image_DRNN.png', dpi=200)
```

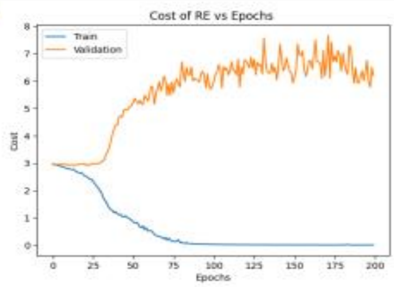
```
zip_train = zip(train['raws'], train['sents'], train['sdps'], train['targets'], train['labels'])
zip_valid = zip(valid['raws'], valid['sents'], valid['sdps'], valid['targets'], valid['labels'])
zip_test = zip(test['raws'], test['sents'], test['sdps'], test['targets'])
```

```
for i, (raw, _, sdp, target, label) in enumerate(zip_train):
    if i > 5:
        break
    print(raw)
    print("%s :: %s" % (DM.sequence_to_sentence(sdp, show_dep=True), DM.sequence_to_sentence(target)))
    print(int2label[label])
    print("="*80)
```

V. RESULT AND DISCUSSION

The graph underneath shows that all facts is divided into elements: schooling and trying out. Dates for both dates are proven right here.

```
plot_cost = 1
# plot the cost
if plot_cost:
    plt.plot(train_losses)
    plt.plot(validation_losses)
    plt.title("Cost of RE vs Epochs ")
    plt.ylabel('Cost')
    plt.xlabel('Epochs')
    plt.legend(labels = ["Train","Validation"])
    plt.show()
```

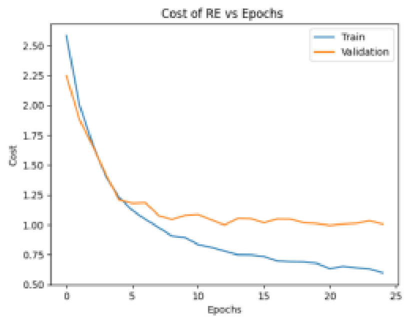


SUPERVISED MODELS

LSTM Model

The graph beneath shows age and statistics loss for fitness and marketing Check up to age 25 There is greater statistics loss at the start up age As age slowly will increase, facts loss decreases. Uses many Over time, the version will become more knowledgeable and greater relevant.

```
plt.plot(train_losses2)
plt.plot(validation_losses2)
plt.title("Cost of RE vs Epochs ")
plt.ylabel('Cost')
plt.xlabel('Epochs')
plt.legend(labels = ["Train","Validation"])
plt.show()
```



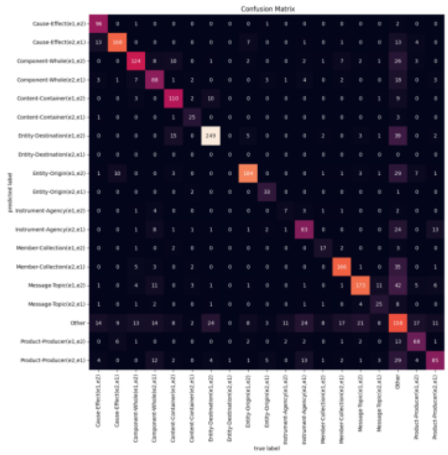
In this example the diploma of accuracy is calculated as zero.6430807820742049. It suggests that about sixty four.31% of the predicted causal hyperlinks are accurate. Obsessive Mera Dilai Edoi Osenka Ravana zero.6894858053578062 68.95% of genuine causal links have been effectively detected. Accuracy indicator for this the score is zero.6915715863084284, which represents approximately 69.60% of all instances. The data set contained correct predictions made through the version. F1 score for this rating 0.6574352187234178, which suggests the general overall performance of the version adjusting for both accuracy and remarks.

```
[ ] precision_metric3,recall_metric3,accuracy_metric3 ,f1_metric3

(0.6430807820742049,
 0.6894858053578062,
 0.6915715863084284,
 0.6574352187234178)
```

Accuracy assesses the accuracy of predictions, taking the whole thing into consideration Categories. The accuracy become determined to be zero.68, i.e. 68% of the cases given the facts predicted the model accurately. Macro Average: Macro average offers same weight to each category within the calculation. Average precision, do not forget and F1 score throughout all training. Careful, remember and F1-. They have macro scores of 0.66, zero. Sixty three and 0.Sixty four respectively. Section: Recall and F1 score for all when calculating mean statement Classes get the common imply (variety of occurrences) for every Various precision, bear in mind and F1 score averaged zero.68, 0.68 and 0.Sixty eight respectively.

```
import seaborn as sns
print(classification_report(y_test2, y_predictions2))

labels = list(int_2_cat_class.values())
f, ax = plt.subplots(figsize=(16, 12))
cnf_matrix = confusion_matrix(y_test2, y_predictions2)
ax = sns.heatmap(cnf_matrix.T, square=True, annot=True, fmt='d', cbar=False, xticklabels=labels , yticklabels=labels)
plt.xlabel('true label')
plt.ylabel('predicted label')
plt.title(' Confusion Matrix')
plt.show()
```

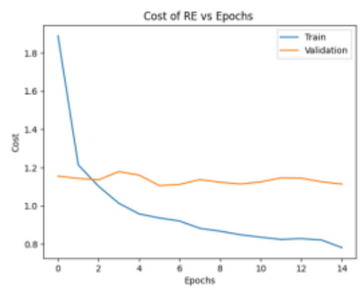
```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:1344: UndefinedMetricWarning: Precision and F-
  _warn_prf(average, modifier, msg_start, len(result))
              precision    recall  f1-score   support

           0       0.96      0.72      0.82       134
           1       0.81      0.87      0.84       194
           2       0.66      0.77      0.71       162
           3       0.66      0.59      0.62       150
           4       0.81      0.72      0.76       153
           5       0.83      0.64      0.72        39
           6       0.79      0.86      0.82       291
           7       0.00      0.00      0.00         1
           8       0.76      0.87      0.81       211
           9       0.92      0.70      0.80        47
          10       0.37      0.32      0.34        22
          11       0.61      0.62      0.61       134
          12       0.68      0.53      0.60        32
          13       0.78      0.83      0.80       201
          14       0.67      0.82      0.74       210
          15       0.57      0.49      0.53        51
          16       0.43      0.35      0.38       454
          17       0.69      0.63      0.66       108
          18       0.51      0.69      0.59       123

    accuracy                           0.68      2717
   macro avg       0.66      0.63      0.64      2717
weighted avg       0.68      0.68      0.68      2717
```



RNN Model

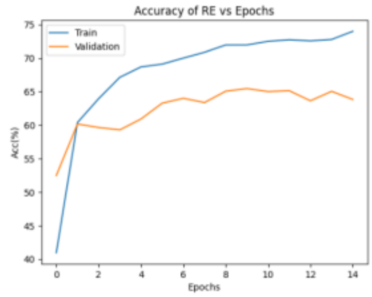
Data loss for education and trying out epochs up to 14 epochs is proven in Fig. The diagram under. Data loss is excessive inside the early epoch; however, whilst as the epoch step by step will increase, records loss decreases. These are centuries of assist the version is surprisingly practical and accurate.

```
plt.plot(train_losses4)
plt.plot(validation_losses4)
plt.title("Cost of RE vs Epochs ")
plt.ylabel('Cost')
plt.xlabel('Epochs')
plt.legend(labels = ["Train","Validation"])
plt.show()
```



The graph below suggests the truthful period and their accuracy set up check and validation records. Accuracy also will increase with age. Escalation is a totally found out example. 14 We came to be destroyed.

```
plt.plot(train_acc_list)
plt.plot(test_acc_list)
plt.title("Accuracy of RE vs Epochs")
plt.ylabel('Acc(%)')
plt.xlabel('Epochs')
plt.legend(labels = ["Train","Validation"])
plt.show()
```



Precision: In this situation, the value of Precision_metric3 is zero.5975, which means approx. 59.79% of expected advantageous cases virtually occur. With a value of Recall_metric3 0.6177, the RNN model correctly detected 61.Seventy seven% real positives. Captain. An accuracy_metric3 value of 0.6445 shows common accuracy the performance of RNN version is set 64.Forty five%. F1 score is balanced score Model overall performance, which is calculated as a harmonious common of keep in mind and precision. It is calculated as 2* (Precision * Recall)/ (Precision + Recall). F1_metric3 value a cost of 0.6009 shows that the version plays equally nicely in terms of don't forget and precision.

```
[ ] precision_metric3,recall_metric3,accuracy_metric3 ,f1_metric3

(0.5975492099575979,
 0.6176925594152751,
 0.6444608023555392,
 0.6008836051694597)
```

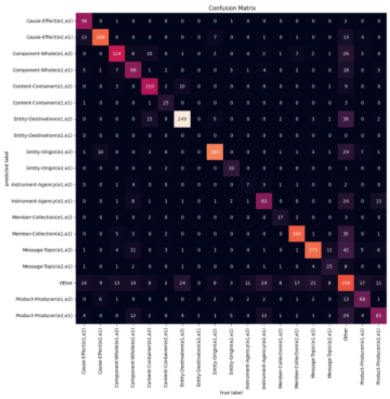
```
import seaborn as sns
print(classification_report(y_test2, y_predictions2))

labels = list(int_2_cat_class.values())
f, ax = plt.subplots(figsize=(16, 12))
cnf_matrix = confusion_matrix(y_test2, y_predictions2)
ax = sns.heatmap(cnf_matrix.T, square=True, annot=True, fmt='d', cbar=False, xticklabels=labels , yticklabels=labels)
plt.xlabel('true label')
plt.ylabel('predicted label')
plt.title(' Confusion Matrix')
plt.show()
```

Accuracy reflects the general capacity of the version to make correct predictions in any respect degrees. It measures the share of efficaciously anticipated cases Samples. In this case the RNN model achieves 68% accuracy on the dataset. Zero.68 typical accuracy. Measure the weights of both lessons similarly while calculating the common overall performance throughout all training. He makes an example in trendy, for all training, it is exactly clean from the common macros, I don't forget; and F1 rating, every around zero.66.

Taking under consideration the squared distribution of the statistics set, the weighted suggest cost is determined Average overall performance indicators for all instructions. In those instances, for the duration of instructions Unbalanced, he offers greater representational dimensions. A weighted average value the accuracy, completeness and F1 index are near 0.Sixty eight, which shows the index of RNN version. General productivity, taking into consideration the distribution of instructions.

```
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:13
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:13
  _warn_prf(average, modifier, msg_start, len(result))
/usr/local/lib/python3.10/dist-packages/sklearn/metrics/_classification.py:13
  _warn_prf(average, modifier, msg_start, len(result))
```

	precision	recall	f1-score	support
0	0.96	0.72	0.82	134
1	0.81	0.87	0.84	194
2	0.66	0.77	0.71	162
3	0.66	0.59	0.62	150
4	0.81	0.72	0.76	153
5	0.83	0.64	0.72	39
6	0.79	0.86	0.82	291
7	0.00	0.00	0.00	1
8	0.76	0.87	0.81	211
9	0.92	0.70	0.80	47
10	0.37	0.32	0.34	22
11	0.61	0.62	0.61	134
12	0.68	0.53	0.60	32
13	0.78	0.83	0.80	201
14	0.67	0.82	0.74	210
15	0.57	0.49	0.53	51
16	0.43	0.35	0.38	454
17	0.69	0.63	0.66	108
18	0.51	0.69	0.59	123
accuracy			0.68	2717
macro avg	0.66	0.63	0.64	2717
weighted avg	0.68	0.68	0.68	2717



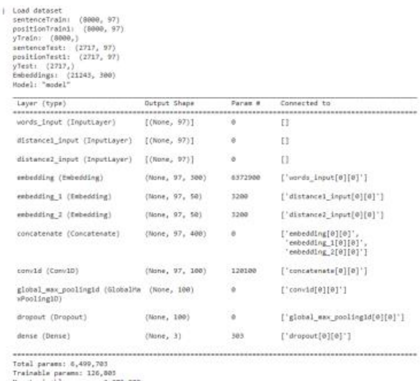
BERT causal version

The accuracy of the version based at the experimental facts is eighty four.14%, as shown by the precision zero.8414. Recall the accuracy and efficiency of the opposition as proven within the F1 score. 0.8414. The fee of the loss characteristic is determined in the evaluation manner Indicates loss. It calculates the distinction between expected effects Signs additionally appear. In this example the loss is 1.0142. These assessment results provide a comprehensive evaluation model. Performance on the test information set. The model seems to predict nicely High precision and effectively label the project causal fabric extraction according to F1 The loss price shows how properly the test version can reduce the difference Actual labels and expected results in training

```
[ ] args.do_eval = True
    if args.do_eval:
        trainer.load_model()
        res, test_loss, preds, labels  = trainer.evaluate("test")

    Evaluating:   0%|          | 0/85 [00:00<?, ?it/s]***** Running evaluation on %s dataset ***** test
      Num examples = %d 2717
      Batch size = %d 32
    Evaluating: 100%|██████████| 85/85 [00:30<00:00,  2.74it/s]***** Eval results *****
      acc = 0.8414
      f1 = 0.8414
      loss = 1.0142
```

CNN Model

In the enter layer report input, distance1_input and distance2_input. They are sentences, first degree and second degree respectively. Embed_1 for the first stage and embed_2 for the second level. Place is the achievement of three orders. An integrated layer is used to connect those\ Linked embedding's are subjected to a one-dimensional layer transformation. (Change 1d). Dimensionality is decreased the use of a 1D Global Max layer. (global_max_pooling1d). A dropout layer is used to keep away from redundancy. The output runs thru the dense layer to represent the three classes. With three gadgets. A general of 126,803 parameters are trainable. 499 703 in VI version.

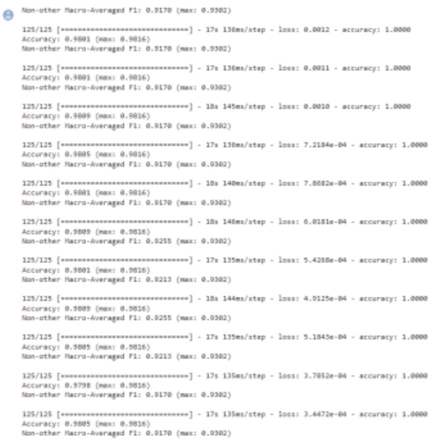


```
Load dataset
sentenceTrain:  (8000, 97)
positionTrain1:  (8000, 97)
yTrain:  (8000,)
sentenceTest:  (2717, 97)
positionTest1:  (2717, 97)
yTest:  (2717,)
Embeddings:  (21243, 300)
Model: "model"
```

Layer (type)	Output Shape	Param #	Connected to
words_input (InputLayer)	[(None, 97)]	0	[]
distance1_input (InputLayer)	[(None, 97)]	0	[]
distance2_input (InputLayer)	[(None, 97)]	0	[]
embedding (Embedding)	(None, 97, 300)	6372900	['words_input[0][0]']
embedding_1 (Embedding)	(None, 97, 50)	3200	['distance1_input[0][0]']
embedding_2 (Embedding)	(None, 97, 50)	3200	['distance2_input[0][0]']
concatenate (Concatenate)	(None, 97, 400)	0	['embedding[0][0]', 'embedding_1[0][0]', 'embedding_2[0][0]']
conv1d (Conv1D)	(None, 97, 100)	120100	['concatenate[0][0]']
global_max_pooling1d (GlobalMaxPooling1D)	(None, 100)	0	['conv1d[0][0]']
dropout (Dropout)	(None, 100)	0	['global_max_pooling1d[0][0]']
dense (Dense)	(None, 3)	303	['dropout[0][0]']

```
Total params: 6,499,703
Trainable params: 126,803
```

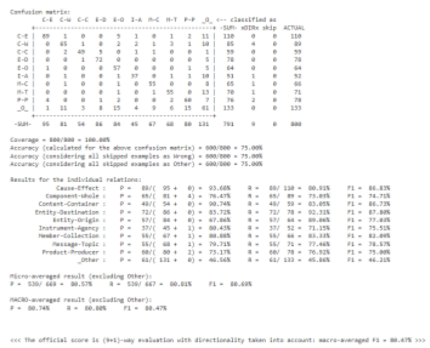
The model has 126,803 out of 6,499,703 trainable parameters. That is, the version is tested the use of check data after schooling using schooling facts. Accuracy and other macro-averaged F1 rankings are typed after each exercise. In addition to the epoch, the outcomes of most accuracy and F1 ratings are proven. Version Shown by using High, it appears to do a very good process of drawing cause and effect relationships Accuracy and F1 ratings. However, let them take into account that according to the movement and context some aspects of architectural fashions, training techniques and assessment metrics can trade.

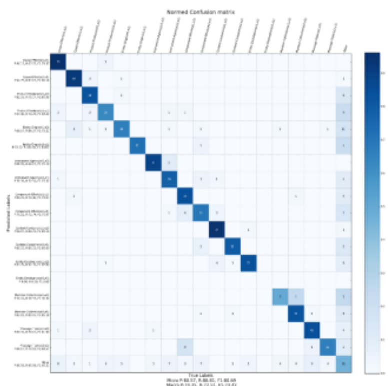


UNSUPERVISED MODELS

Unsupervised Bert

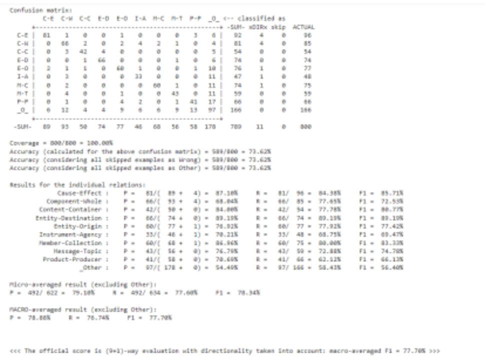
In this version, we calculated the remember accuracy and familiarity of all f1 statements. In the information set. The accuracy of the cause-effect courting is 93.Sixty eight%, i.e. of all reason-effect instances, 93.68% were efficiently expected. Component-integer courting, do not forget seventy three.03%, 73.03% i.e. The cases that Component-Toti should have had are accurate anticipated. The F1 rating for the content material dating is 86.73%, which shows there is often a robust balance among don't forget and precision. Except for the "_Other" elegance, the implementation of the complete model is taken into consideration. Taking under consideration the calculation of the small mean effect. Calculates precision, take into account, and F1 primarily based on the variety of fake positives, false positives, and false positives. Denies all relationships. The average F1 is small at 80.Sixty nine%. Precision, keep in mind, and F1 rankings had been averaged for every setting, except the "_Other" elegance is the result of the chat within the macro. No matter what Relationships arise greater regularly, which gives them equal weight. Average F1 motive force this case is 80.47%. Official evaluation based totally on record F1 macro imply rating is 80.Forty seven% and specific (9+1) path Directional estimation is taken into consideration.



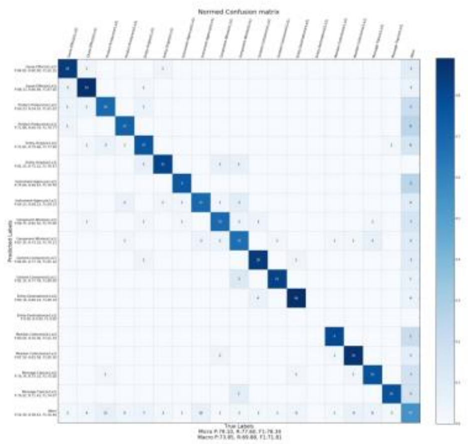


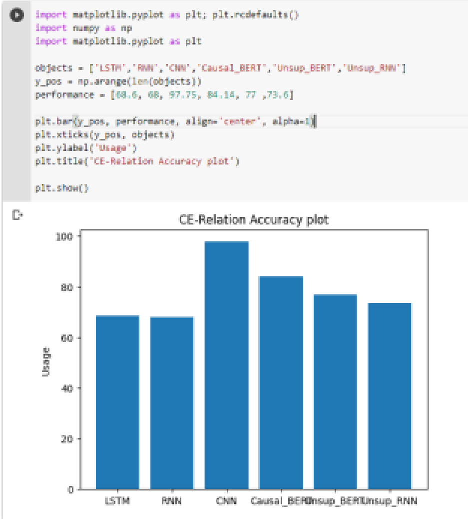
Rnn Unsupervised

Accuracy is calculated by dividing the wide variety of outcomes by means of (800). The range of efficiently diagnosed cases (589), is given accurately this precision is maintained regardless of how the 73.62% missing cases are dealt with. It's true. Precision, do not forget, and F1 ratings are once more provided for each statement Results of each record. These metrics examine how properly the version is acting. Relationship with every different. The typical sample performance is "_Other" special. Class is considered whilst calculating average performance for signs True effective, false high-quality and fake negative. Average accuracy in this immediate seventy nine.10%, Recall seventy seven.60%, F1 score seventy eight.34%. Precision, do not forget, and F1 ratings had been averaged for every setting, except the "_Other" magnificence is the result of the chat inside the macro. No rely what Attitudes often arise, giving equal weight. Careful, recall and F1-. In this case, the macro averages are 78.88%, 76.Seventy four% and 77.70% respectively. The official result is based on a unique (9+1)-thing score that takes direction, and macro F1 averaged seventy seven.70%. This is Formal evaluation is the evaluation of activities related to tactics Information assessment.



```
Confusion matrix:
        C-E  C-W  C-C  E-D  E-O  I-A  M-C  M-T  P-P  _O_ <-- classified as
      +--------------------------------------------------+ -SUM- xDIRx skip  ACTUAL
  C-E |  81    1    0    0    1    0    0    0    3    6 |   92     4     0     96
  C-W |   0   66    2    0    2    4    2    1    0    4 |   81     4     0     85
  C-C |   0    3   42    4    0    0    0    0    0    5 |   54     0     0     54
  E-D |   0    0    1   66    0    0    0    1    0    6 |   74     0     0     74
  E-O |   2    1    1    0   60    1    0    0    1   10 |   76     1     0     77
  I-A |   0    3    0    0    0   33    0    0    0   11 |   47     1     0     48
  M-C |   0    2    0    0    0    0   60    1    0   11 |   74     1     0     75
  M-T |   0    4    0    0    1    0    0   43    0   11 |   59     0     0     59
  P-P |   0    1    0    0    4    2    0    1   41   17 |   66     0     0     66
  _O_ |   6   12    4    4    9    6    6    9   13   97 |  166     0     0    166
      +--------------------------------------------------+
 -SUM-   89   93   50   74   77   46   68   56   58  178    789    11     0    800

Coverage = 800/800 = 100.00%
Accuracy (calculated for the above confusion matrix) = 589/800 = 73.62%
Accuracy (considering all skipped examples as Wrong) = 589/800 = 73.62%
Accuracy (considering all skipped examples as Other) = 589/800 = 73.62%

Results for the individual relations:
          Cause-Effect :    P =   81/(  89 +   4) =  87.10%     R =   81/  96 =  84.38%     F1 =  85.71%
       Component-Whole :    P =   66/(  93 +   4) =  68.04%     R =   66/  85 =  77.65%     F1 =  72.53%
     Content-Container :    P =   42/(  50 +   0) =  84.00%     R =   42/  54 =  77.78%     F1 =  80.77%
    Entity-Destination :    P =   66/(  74 +   0) =  89.19%     R =   66/  74 =  89.19%     F1 =  89.19%
         Entity-Origin :    P =   60/(  77 +   1) =  76.92%     R =   60/  77 =  77.92%     F1 =  77.42%
     Instrument-Agency :    P =   33/(  46 +   1) =  70.21%     R =   33/  48 =  68.75%     F1 =  69.47%
     Member-Collection :    P =   60/(  68 +   1) =  86.96%     R =   60/  75 =  80.00%     F1 =  83.33%
         Message-Topic :    P =   43/(  56 +   0) =  76.79%     R =   43/  59 =  72.88%     F1 =  74.78%
      Product-Producer :    P =   41/(  58 +   0) =  70.69%     R =   41/  66 =  62.12%     F1 =  66.13%
                _Other :    P =   97/( 178 +   0) =  54.49%     R =   97/ 166 =  58.43%     F1 =  56.40%

Micro-averaged result (excluding Other):
P =  492/ 622 =  79.10%     R =  492/ 634 =  77.60%     F1 =  78.34%

MACRO-averaged result (excluding Other):
P =  78.88%     R =  76.74%     F1 =  77.70%

<<< The official score is (9+1)-way evaluation with directionality taken into account: macro-averaged F1 = 77.70% >>>
```





```
import matplotlib.pyplot as plt; plt.rcdefaults()
import numpy as np
import matplotlib.pyplot as plt

objects = ['LSTM','RNN','CNN','Causal_BERT','Unsup_BERT','Unsup_RNN']
y_pos = np.arange(len(objects))
performance = [68.6, 68, 97.75, 84.14, 77 ,73.6]

plt.bar(y_pos, performance, align='center', alpha=1)
plt.xticks(y_pos, objects)
plt.ylabel('Usage')
plt.title('CE-Relation Accuracy plot')

plt.show()
```


VI. CONCLUSION

Finally, we notice that the purpose of the take a look at is to decide cause and effect relationships among occasions in the text. A crucial mission in natural language processing is referred to as causal characteristic extraction. In many ways in regions inclusive of health, finance and social media, purpose and impact relationships are essential Illuminates the fundamental dynamics and mechanisms of complex structures.

Several techniques have been advanced to extract cause and impact relationships from all facets A rule-based totally method to supervised and unsupervised machine studying models. LSTM, RNN and BERT are examples of deep gaining knowledge of based totally architectures It confirmed a promising ability to seize activities and the complicated interactions among them Basic causal mechanisms.

Although development has been made on this vicinity, many issues remain to be resolved. Solutions, as an example, cope with noise and textual content blurring and make working with fashions easier. Understanding and processing massive amounts of statistics in actual-time programs. It won't take off further research and improvement in the area of natural language processing to meet them problems.

Overall, causal inference is a developing region of research with remarkable promise. It improves our understanding of complicated systems and opens up new opportunities various fields.

FUTURE SCOPE

Future research desires in causal extraction include improving version accuracy and scalability, preventing textual noise and ambiguity, producing large and numerous datasets, incorporating outside expertise, and permitting real-time applications. Understand simple reason and impact mechanisms and the relationships between them Better, greater interpretable fashions may be created.

VII. REFERENCES

1. Z. Li, Q. Li, X. Zou, and J. Ren, "Causality Extraction Based on Self-Attentive BiLSTM-CRF with Transferred Embeddings," in Proceedings of the 2020 5th International Conference on Computational Intelligence and Applications (ICCIA 2020), Nov 2020, pp. 26-31.

2. V. Khetan, M. I. H. Rizvi, J. Huber, P. Bartusiak, B. Sacaleanu, and A. Fano, "MIMICause: Representation and automatic extraction of causal relation types from clinical notes," in Proceedings of the 2022 IEEE International Conference on Healthcare Informatics (ICHI 2022), May 2022, pp. 327-336.

3. T. Dasgupta, R. Saha, L. Dey, and A. Naskar, "Automatic Extraction of Causal Relations from Text using Linguistically Informed Deep Neural Networks," in Proceedings of the 2018 IEEE International Conference on Big Data (Big Data 2018), Seattle, WA, USA, Jul. 2018, pp. 1781-1789.

4. N. Kang, B. Singh, C. Bui, Z. Afzal, E. M. van Mulligen, and J. A. Kors, "Knowledgebased extraction of adverse drug events from biomedical text," in Proceedings of the 2014 IEEE International Conference on Bioinformatics and Biomedicine (BIBM 2014), Belfast, UK, Mar. 2014, pp. 54-59.

5. G. Zhu, Z. Sun, S. Zhang, S. Wei, and K. Li, "Causality Extraction Model Based on Two-stage GCN," in Proceedings of the 2022 International Joint Conference on Neural Networks (IJCNN 2022), April 2022, pp. 1-8.

6. Q. Li, J. Li, J. Sheng, S. Cui, J. Wu, Y. Hei, H. Peng, S. Guo, L. Wang, A. Beheshti, and P. S. Yu, "A Survey on Deep Learning Event Extraction: Approaches and Applications," in Proceedings of the 2022 IEEE International Conference on Data Mining (ICDM 2022), Dec. 2022, pp. 1-10.

7. A. Akkasi and M. Moens, "Causal relationship extraction from biomedical text using deep neural models: A comprehensive survey," Journal of Biomedical Informatics, vol. 117, June 2021, p. 103729.

8. R. Ortega, A. Fonseca, Y. Gutierrez, and A. Montoyo, "SSA-UO: Unsupervised Twitter Sentiment Analysis," in Proceedings of the 4th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis (WASSA 2013), June 2013, pp. 122-127.

9. M.A. Valles Coral, L. Salazar-Ramírez, R. Injante, E.A. Hernandez-Torres, J. JuárezDíaz, J.R. Navarro Cabrera, L. Pinedo, and P. Vidaurre-Rojas, "Density-Based Unsupervised Learning Algorithm to Categorize College Students into Dropout Risk Levels," in Proceedings of the IEEE International Conference on Computational Intelligence and Virtual Environments for Measurement Systems and Applications (CIVEMSA), 2022, pp. 1-6.

10. Sharma, R., Palshikar, G., & Pawar, S. (2018). An Unsupervised Approach for CauseEffect Relation Extraction from Biomedical Text. In 2018 IEEE International Conference on Data Mining Workshops (ICDMW) (pp. 1118-1125). IEEE.

11. Yang, J., Han, S.C. & Poon, J. A survey on extraction of causal relations from natural language text. Knowl Inf Syst 64, 1161–1186 (2022).

12. Tirthankar Dasgupta, Rupsa Saha, Lipika Dey, and Abir Naskar. 2018. Automatic Extraction of Causal Relations from Text using Linguistically Informed Deep Neural Networks. In Proceedings of the 19th Annual SIGdial Meeting on Discourse and Dialogue, pages 306–316, Melbourne, Australia. Association for Computational Linguistics.

13. Q. Li et al., "A Survey on Deep Learning Event Extraction: Approaches and Applications," in IEEE Transactions on Neural Networks and Learning Systems, 2022, doi: 10.1109/TNNLS.2022.3213168.

14. Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. 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.

15. Edoardo Maria Ponti and Anna Korhonen. 2017. Event-Related Features in Feedforward Neural Networks Contribute to Identifying Causal Relations in Discourse. In Proceedings of the 2nd Workshop on Linking Models of Lexical, Sentential and Discourse-level Semantics, pages 25–30, Valencia, Spain. Association for Computational Linguistics.

16. Khoo, C.S., Kornfilt, J., Oddy, R.N., Myaeng, S.H.: Automatic extraction of cause-effect information from newspaper text without knowledge-based inferencing. Lit. Linguist. Comput. 13(4), 177–186 (1998)

17. Sorgente A, Vettigli G, Mele F (2013) Automatic extraction of cause-effect relations in natural language text. DART@ AI* IA 1109:37–48

18. Kang N, Singh B, Bui QC, Afzal Z, van Mulligen EM, Kors J (2014) Knowledge-based extraction of adverse drug events from biomedical text. BMC Bioinform 15:64.