St. Vincent Pallotti College of Engineering & Technology
Department of Information Technology
Academic Session: 2020-2021

Word Phrase Alignment

Name of the students: Gaurav Bhisikar, Aditya Dale, Sagar Malik
Name of the Guide: Prof. Pooja Walke & Indutry Mentor: Dr. Soma Paul and Dr. Vineet Chaitanya
Alumni Mentor: Ms. CBS Manasvini

ABSTRACT : The main objective of this project is to develop software tools and a scheme for building word or phrase aligned corpus of English and Hindi optimally using existing open resources. Current NMT models are able to translate, but current NMT models are not able to generate correct alignment of translated Hindi sentence. So this project focuses on identifying the relation between words of both English and Hindi sentences and based on this relation to generating a Hindi Translated sentence that is correctly aligned.

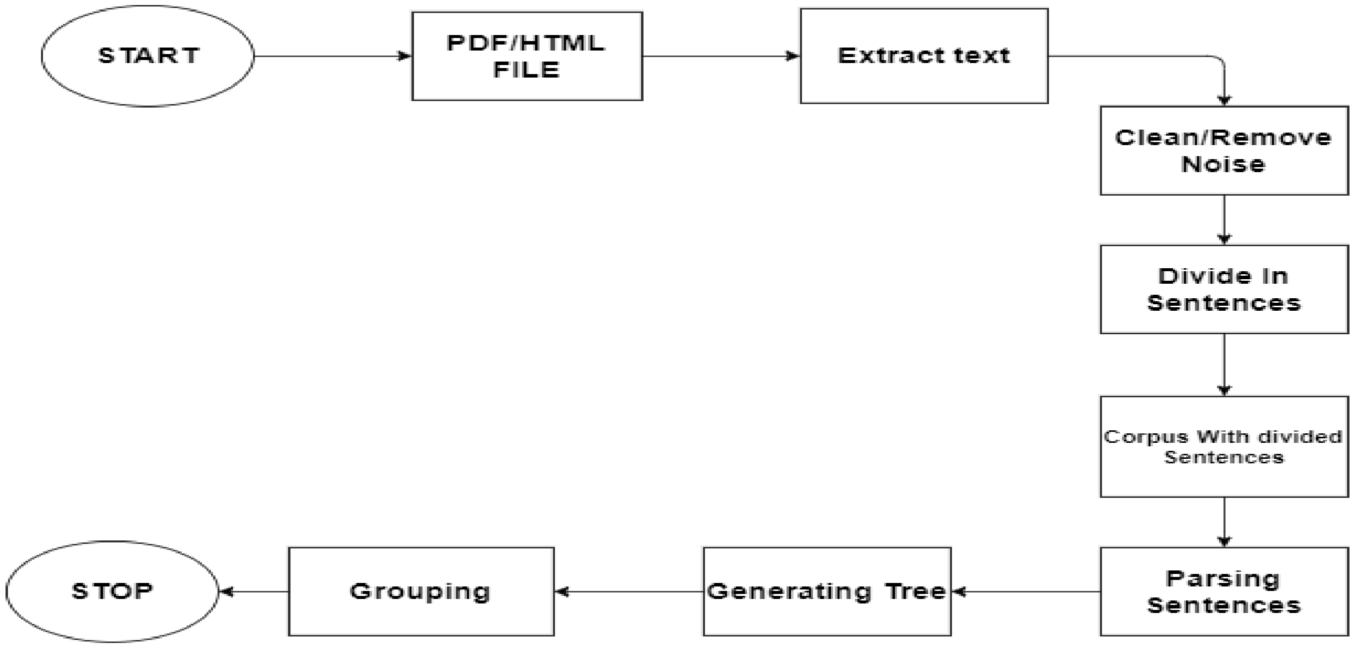
INTRODUCTION:

This is a research project in the domain of NLP, at Anusaarka Lab under LTRC center at IIIT Hyderabad. This project aims to modify the current method of training deep learning model from sentence level to word/phrase level. The Current method of training deep learning model for translation uses parallel corpus which is on sentence level. But this method requires a very large corpus to train the model which makes it difficult to get expected result from smaller set of data. So if we can convert the parallel corpus to word/phrase structure level it will be easier to train and get better result from a deep learning model.

So comparatively model will require small sized corpus for training, eventually it will reduce the time to train a model, as size of data is reduced. It will also give more human control over traditional model.

In this project we are working on the possibility of automating the task of alignment of words/phrases of translated sentence in order to map these words/phrases with the words/phrases with same meaning in source sentence i.e. to create a parallel corpus with word/phrase level structure.

SYSTEM ARCHITECTURE:



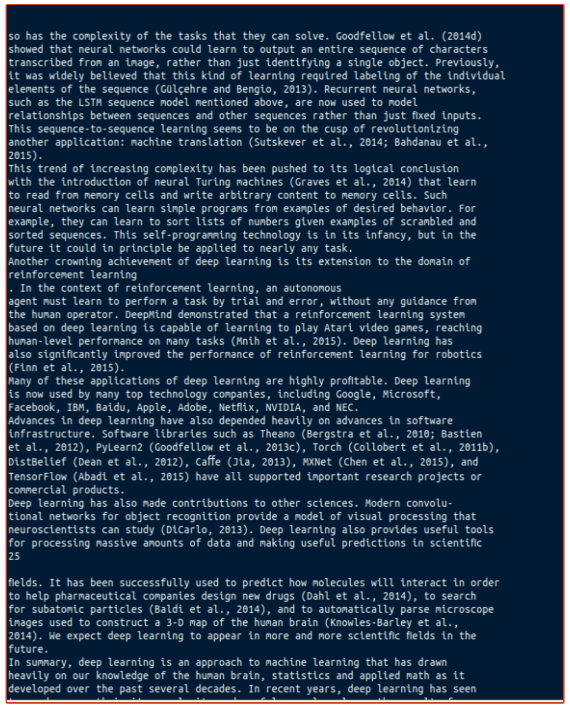
MODULES:

1.Parser(ACE & STANFORD)
2. LUCY Tree-bank
3. Sentence Grouping
4. Noise Remove

APPLICATIONS:

- Used in Natural Language Processing(NLP).
- Used in Automated Machine Translators.

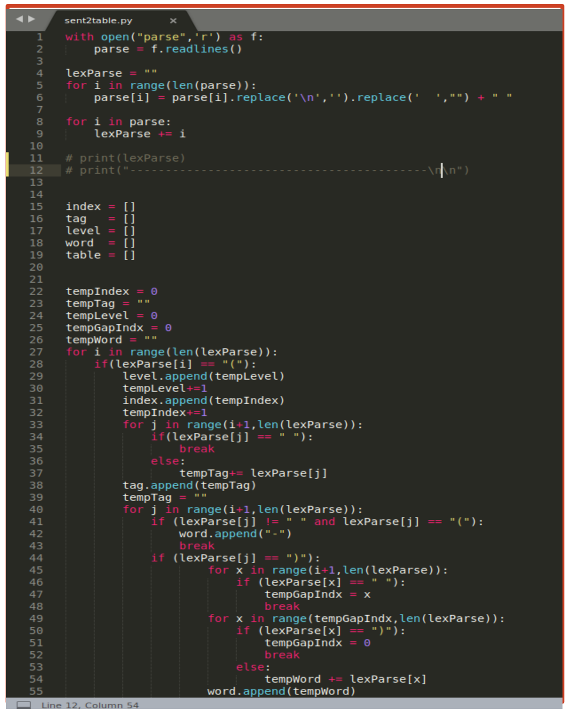
IMPLEMENTATION:



```
so has the complexity of the tasks that they can solve. Goodfellow et al. (2014d)
showed that neural networks could learn to output an entire sequence of characters
transcribed from an image, rather than just identifying a single object. Previously,
it was widely believed that this kind of learning required labeling of the individual
elements of the sequence (Gülçehre and Bengio, 2013). Recurrent neural networks,
such as the LSTM sequence model mentioned above, are now used to model
relationships between sequences and other sequences rather than just fixed inputs.
This sequence-to-sequence learning seems to be on the cusp of revolutionizing
another application: machine translation (Sutskever et al., 2014; Bahdanau et al.,
2015).
This trend of increasing complexity has been pushed to its logical conclusion
with the introduction of neural Turing machines (Graves et al., 2014) that learn
to read from memory cells and write arbitrary content to memory cells. Such
neural networks can learn simple programs from examples of desired behavior. For
example, they can learn to sort lists of numbers given examples of scrambled and
sorted sequences. This self-programming technology is in its infancy, but in the
future it could in principle be applied to nearly any task.
Another crowning achievement of deep learning is its extension to the domain of
reinforcement learning
. In the context of reinforcement learning, an autonomous
agent must learn to perform a task by trial and error, without any guidance from
the human operator. DeepMind demonstrated that a reinforcement learning system
based on deep learning is capable of learning to play Atari video games, reaching
human-level performance on many tasks (Mnih et al., 2015). Deep learning has
also significantly improved the performance of reinforcement learning for robotics
(Finn et al., 2015).
Many of these applications of deep learning are highly profitable. Deep learning
is now used by many top technology companies, including Google, Microsoft,
Facebook, IBM, Baidu, Apple, Adobe, Netflix, NVIDIA, and NEC.
Advances in deep learning have also depended heavily on advances in software
infrastructure. Software libraries such as Theano (Bergstra et al., 2010; Bastien
et al., 2012), PyLearn2 (Goodfellow et al., 2013c), Torch (Collobert et al., 2011b),
DistBelief (Dean et al., 2012), Caffe (Jia, 2013), MXNet (Chen et al., 2015), and
TensorFlow (Abadi et al., 2015) have all supported important research projects or
commercial products.
Deep learning has also made contributions to other sciences. Modern convolu-
tional networks for object recognition provide a model of visual processing that
neuroscientists can study (DiCarlo, 2013). Deep learning also provides useful tools
for processing massive amounts of data and making useful predictions in scientific
25

fields. It has been successfully used to predict how molecules will interact in order
to help pharmaceutical companies design new drugs (Dahl et al., 2014), to search
for subatomic particles (Baldi et al., 2014), and to automatically parse microscope
images used to construct a 3-D map of the human brain (Knowles-Barley et al.,
2014). We expect deep learning to appear in more and more scientific fields in the
future.
In summary, deep learning is an approach to machine learning that has drawn
heavily on our knowledge of the human brain, statistics and applied math as it
developed over the past several decades. In recent years, deep learning has seen
```

Fig. A) Clean Corpus

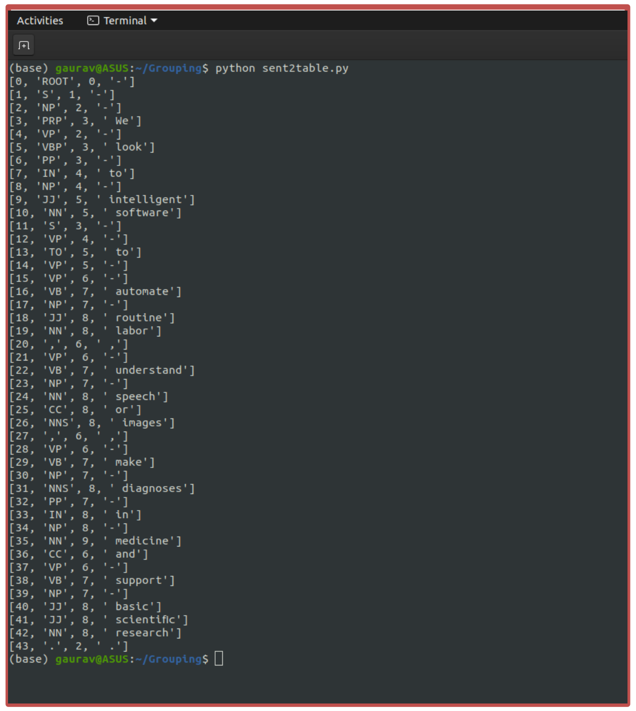
It is the output of program RemoveNoise.py, which is used to remove noises from the corpus.



```python
with open("parse",'r') as f:
    parse = f.readlines()

lexParse = ""
for i in range(len(parse)):
    parse[i] = parse[i].replace('\n','').replace('  ',"") + " "

for i in parse:
    lexParse += i

# print(lexParse)
# print("-----------------------------------\n\n")


index = []
tag   = []
level = []
word  = []
table = []


tempIndex = 0
tempTag = ""
tempLevel = 0
tempGapIndx = 0
tempWord = ""
for i in range(len(lexParse)):
    if(lexParse[i] == "("):
        level.append(tempLevel)
        tempLevel+=1
        index.append(tempIndex)
        tempIndex+=1
        for j in range(i+1,len(lexParse)):
            if(lexParse[j] == " "):
                break
            else:
                tempTag+= lexParse[j]
        tag.append(tempTag)
        tempTag = ""
        for j in range(i+1,len(lexParse)):
            if (lexParse[j] != " " and lexParse[j] == "("):
                word.append("-")
                break
            if (lexParse[j] == ")"):
                    for x in range(i+1,len(lexParse)):
                        if (lexParse[x] == " "):
                            tempGapIndx = x
                            break
                    for x in range(tempGapIndx,len(lexParse)):
                        if (lexParse[x] == ")"):
                            tempGapIndx = 0
                            break
                        else:
                            tempWord += lexParse[x]
                    word.append(tempWord)
```

Fig. B) Program Sent2table.py

It is the Snapshot of program Sent2table.py.



```
(base) gaurav@ASUS:~/Grouping$ python sent2table.py
[0, 'ROOT', 0, '-']
[1, 'S', 1, '-']
[2, 'NP', 2, '-']
[3, 'PRP', 3, ' We']
[4, 'VP', 2, '-']
[5, 'VBP', 3, ' look']
[6, 'PP', 3, '-']
[7, 'IN', 4, ' to']
[8, 'NP', 4, '-']
[9, 'JJ', 5, ' intelligent']
[10, 'NN', 5, ' software']
[11, 'S', 3, '-']
[12, 'VP', 4, '-']
[13, 'TO', 5, ' to']
[14, 'VP', 5, '-']
[15, 'VP', 6, '-']
[16, 'VB', 7, ' automate']
[17, 'NP', 7, '-']
[18, 'JJ', 8, ' routine']
[19, 'NN', 8, ' labor']
[20, ',', 6, ' ,']
[21, 'VP', 6, '-']
[22, 'VB', 7, ' understand']
[23, 'NP', 7, '-']
[24, 'NN', 8, ' speech']
[25, 'CC', 8, ' or']
[26, 'NNS', 8, ' images']
[27, ',', 6, ' ,']
[28, 'VP', 6, '-']
[29, 'VB', 7, ' make']
[30, 'NP', 7, '-']
[31, 'NNS', 8, ' diagnoses']
[32, 'PP', 7, '-']
[33, 'IN', 8, ' in']
[34, 'NP', 8, '-']
[35, 'NN', 9, ' medicine']
[36, 'CC', 6, ' and']
[37, 'VP', 6, '-']
[38, 'VB', 7, ' support']
[39, 'NP', 7, '-']
[40, 'JJ', 8, ' basic']
[41, 'JJ', 8, ' scientific']
[42, 'NN', 8, ' research']
[43, '.', 2, ' .']
(base) gaurav@ASUS:~/Grouping$
```

Fig. C)) Output of Sent2table.py

It is the Output of Sent2table.py, it extracts information from parse tree.

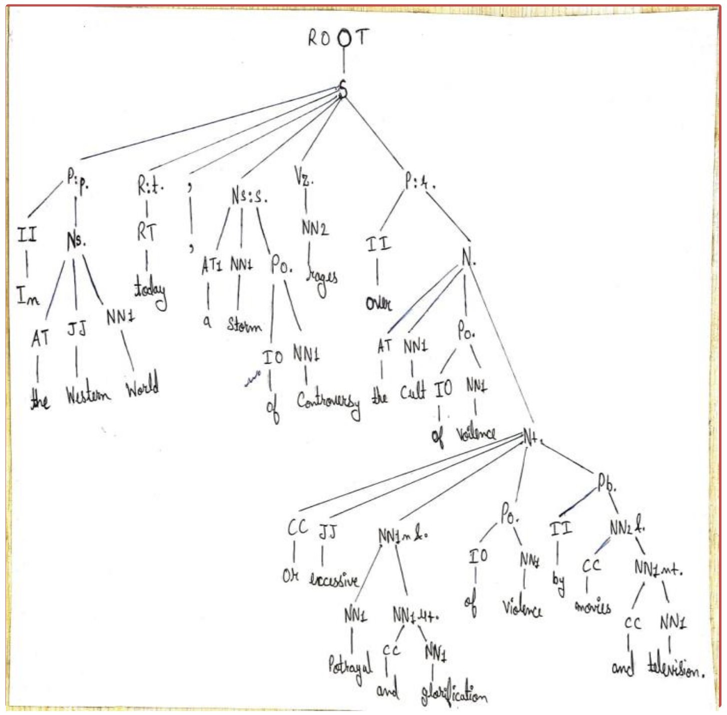


Fig. D) Manual Parse Tree

It is a parse tree we have drawn manually with the help of LUCY tree-bank.

REQUIREMENTS:

Hardware : Processor - Intel i3 or higher , RAM: 4 GB or higher , Disk Space: 1 GB of free disk

Software : Windows XP or later, Ubuntu , Python 3.6 or above , Java 8 or above

CONCLUSION:.

The work done till the grouping of sentences revealed that the parse generated by Stanford parser might be incorrect, which directly affects the process of grouping. The main goal of the project is to develop a tool/software which performs automatic word alignment of parallel corpus of different language. The work done by us during this project will be a stepping stone for others to carry this research work to achieve the ultimate goal.

REFERENCES:

[1] By Xin Lian, Kshitij Jain, Jakub Truszkowski, Pascal Poupart, and Yaoliang Yu (2020) “Unsupervised Multilingual Alignment using Wasserstein Barycenter”

[2] [Alaux et al., 2019] Jean Alaux, Edouard Grave, Marco Cuturi, and Armand Joulin. Unsupervised Hyper-alignment for Multilingual Word Embeddings. In ICLR,
2019.