NLP for ESG Goal Extraction

Shubham Pramod Suryawanshi  
shubham.suryawanshi@stud.fra-uas.de

***Abstract* —** **An annual ESG report is published by a company or organization about environmental, social and governance (ESG) impacts. It enables the company to be more transparent about the risks and opportunities. These ESG reports are generally lengthy, it would be helpful to automate the process of extracting goals set out by the organization to accelerate the process of analysing these ESG reports. In this project, I have leveraged the power of NLP techniques to extract ESG goals from lengthy ESG reports.**

Keywords — ESG, NLP, spaCy, Coreference Resolution, Name Entity Recognition, Part-of-speech tagging.

# Introduction

In nature there are many events which occur periodically, here we are trying to identify such time-series sequence which is associated entity/label and the end goal is to learn time-series with label and predict the label at given time.

The neocortex is the seat of intelligent thought in the mammalian brain. High level vision, hearing, touch, movement, language, and planning are all performed by the neocortex. Given such a diverse suite of cognitive functions, you might expect the neocortex to implement an equally diverse suite of specialized neural algorithms. This is not the case. The neocortex displays a remarkably uniform pattern of neural circuitry. The biological evidence suggests that the neocortex implements a common set of algorithms to perform many different intelligence functions [1].

The neocortex continually processes an endless stream of rich sensory information. It does this remarkably well, better than any existing AI computer. A wealth of empirical evidence demonstrates that cortical regions represent all information using sparse patterns of activity. To function effectively throughout a lifetime these representations must have tremendous capacity and must be extremely tolerant to noise. However, a detailed theoretical understanding of the capacity and robustness of cortical sparse representations has been missing [2] .

HTM provides a theoretical framework for understanding the neocortex and its many capabilities. To date we have implemented a small subset of this theoretical framework. Over time, more and more of the theory will be implemented. Today we believe we have implemented a sufficient subset of what the neocortex does to be of commercial and scientific value [1].

# LITERATURE SURVEY

## Transformer Architecture

A transformer is a special type of deep learning model that adopts the style of self-attention, differentially weighting the significance of each part of the input data. It is used primarily in the fields of natural language processing (NLP) and computer vision (CV). The Transformer is based solely on attention mechanisms dispensing with recurrence and convolutions entirely. [1]

## Coreference Resolution

Coreference resolution is the NLP procedure of finding all expressions that refer to the same entity in a piece of text. It is commonly performed step for a lot of higher level NLP tasks that involve natural language understanding such as document summarization, question answering, and information extraction. [2]

## Name Entity Recognition

Sequence of event that occur in given period of time.

## Part-of-speech tagging

The time-series which is associated with an event is called label i.e. power consumption (in KWh). Power consumption, we are using data set of gym’s power consumption on hourly basis.

## HTM

Hierarchical Temporal Memory (HTM) provides a flexible and biologically accurate framework for solving prediction, classification, and anomaly detection problems for a broad range of data types. HTM systems require data input in the form of Sparse Distributed Representations (SDRs). SDRs are quite different from standard computer representations, such as ASCII for text, in that meaning is encoded directly into the representation. An SDR consists of a large array of bits of which most are zeros, and a few are ones. Each bit carries some semantic meaning so if two SDRs have more than a few overlapping one-bits, then those two SDRs have similar meanings [3].

HTMs can be viewed as a type of neural network. By definition, any system that tries to model the architectural details of the neocortex is a neural network. However, on its own, the term “neural network” is not very useful because it has been applied to a large variety of systems. HTMs model neurons (called cells when referring to HTM), which are arranged in columns, in layers, in regions, and in a hierarchy. The details matter, and in this regard HTMs are a new form of neural network [1].

## *Sparse Distributed Representations (SDRs)*

Empirical evidence shows that every region of the neocortex represents information using sparse activity patterns made up of a small percentage of active neurons, with the remaining neurons being inactive. An SDR is a set of binary vectors where a small percentage of 1s represent active neurons, and the 0s represent inactive neurons. The small percentage of 1s, denoted the *sparsity*, varies from less than one percent to several percent. SDRs are the primary data structure used in the neocortex and used everywhere in HTM systems. There is not a single type of SDRs in HTM but distinct types for various purposes [4].

While a bit position in a dense representation like ASCII has no semantic meaning, the bit positions in an SDR represent a particular property. The semantic meaning depends on what the input data represents. Some bits may represent edges or big patches of color; others might correspond to different musical notes. Figure 1 shows a somewhat contrived but illustrative example of an SDR representing parts of a zebra. If we flip a single bit in a vector from a dense representation, the vector may take an entirely different value. In an SDR, nearby bit positions represent similar properties. If we invert a bit, then the description changes but not radically [4].

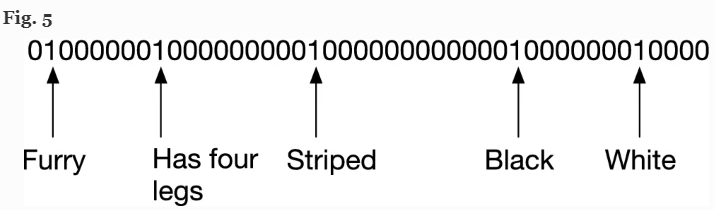


Figure 1 : Encoded data for an animal

# METHODOLOGY

In the experiment, the data set for power consumption was refined for any error and convert to segments. The datetime segments where then encoded using Scalar Encoder. The encoded data was then trained using algorithm called Multi-sequence Learning. Detailed explanation is given below.

Two methods were used to create refined segments as part of the experiment.

## Datatset Preprocessing

The dataset consists of two columns called timestamp and power consumption (in KWh). The data was for taken down for a time period of six months at an interval of one hour. Slice of the used data set can be seen in Figure 2. The whole dataset has total of 4416 rows.

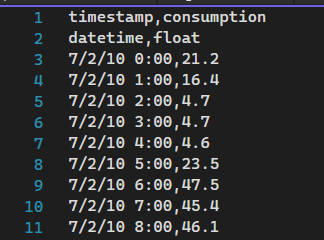


Figure 2 : Dataset

The datetime in the timestamp was not formatted and did not have proper formatting. Figure 3 show the transformation from raw data to segmentation of datetime.

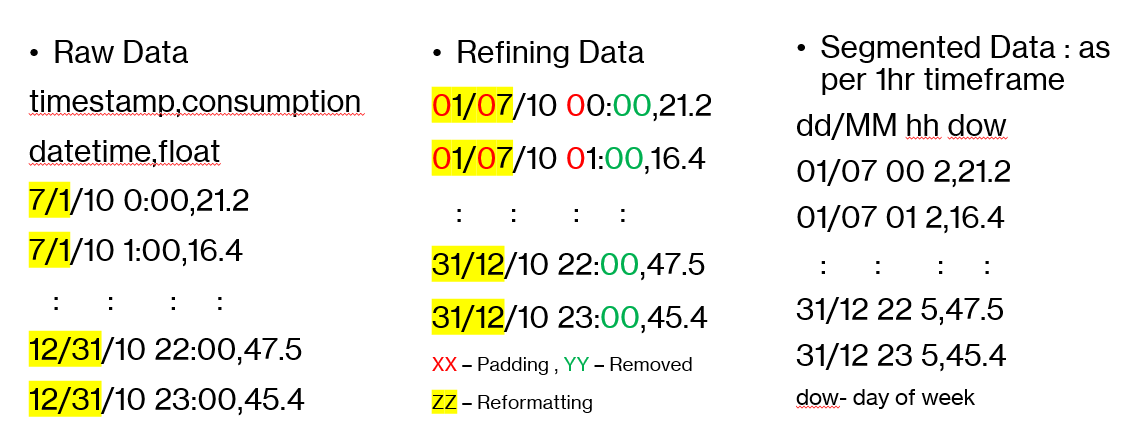


Figure 3 : Segment of dataset

The segments of datetime are done as per one hour time frame as shown in Figure 3.

## Creating multiple sequences

For the experiment, whole dataset was not taken into consideration. Only a part of data was taken which is 744 rows out of 4416 rows. There are three ways or configuration in which the time-series can be taken up and create multiple sequences which are by day, by week, by month. If considered a sequence for a single day then 24 segment create a time-series.

### By day

When the configuration of “by day” is chosen on total segments of 744 in which each time segment is of 1 hour then results into 24 segments of each sequence and total of 31 sequences. This configuration is used for our experiment.

### By week

When the configuration of “by week” is chosen on total segments of 4416 in which each time segment is of 1 hour then results into 168 segments of each sequence and total of 26 sequences.

### By month

When the configuration of “by day” is chosen on total segments of 4416 in which each time segment is of 1 hour then results into 720 segments of each sequence and total of 6 sequences.

## Encode datetime/segment

Encoding is done using Scalar Encoder of NeoCortexApi. For segmenting there are two methods as show below to create and reformat.

### Method 1

In this method for experiment, the segmented

REFERENCES

|  |  |
| --- | --- |
| [1] | "Attention Is All You Need"  Ashish Vaswani, Noam Shazeer, Niki Parmar,  Jakob Uszkoreit, Llion Jones, Aidan N. Gomez,  Lukasz Kaiser, Illia Polosukhin - 6 December 2017.  Available: https://arxiv.org/abs/1706.03762. |
| [2] | "Deep Reinforcement Learning for Mention-Ranking Coreference Models"  - Kevin Clark and Christopher Manning, 2016 Available:https://nlp.stanford.edu/static/pubs/clark2016deep.pdf. |
| [3] | S. purdy, "arxiv.org," Arxiv, 18 February 2016. [Online]. Available: https://arxiv.org/abs/1602.05925. [Accessed 22 March 2022]. |
| [4] | K. j. H. &. S. Ahmad, "link.springer.com," SpringerLink, 20 July 2021. [Online]. Available: https://link.springer.com/article/10.1007/s42452-021-04715-0#Sec14. [Accessed 22 March 2022]. |
| [5] | "Dataset: https://github.com/numenta/nupic/blob/master/  src/nupic/datafiles/extra/hotgym/rec-center-hourly.csv". |
| [6] | "NeoCortexApi : https://github.com/ddobric/neocortexapi". |