CS772: Deep Learning for Natural Language Processing (DL-NLP)

Transformer start

Pushpak Bhattacharyya

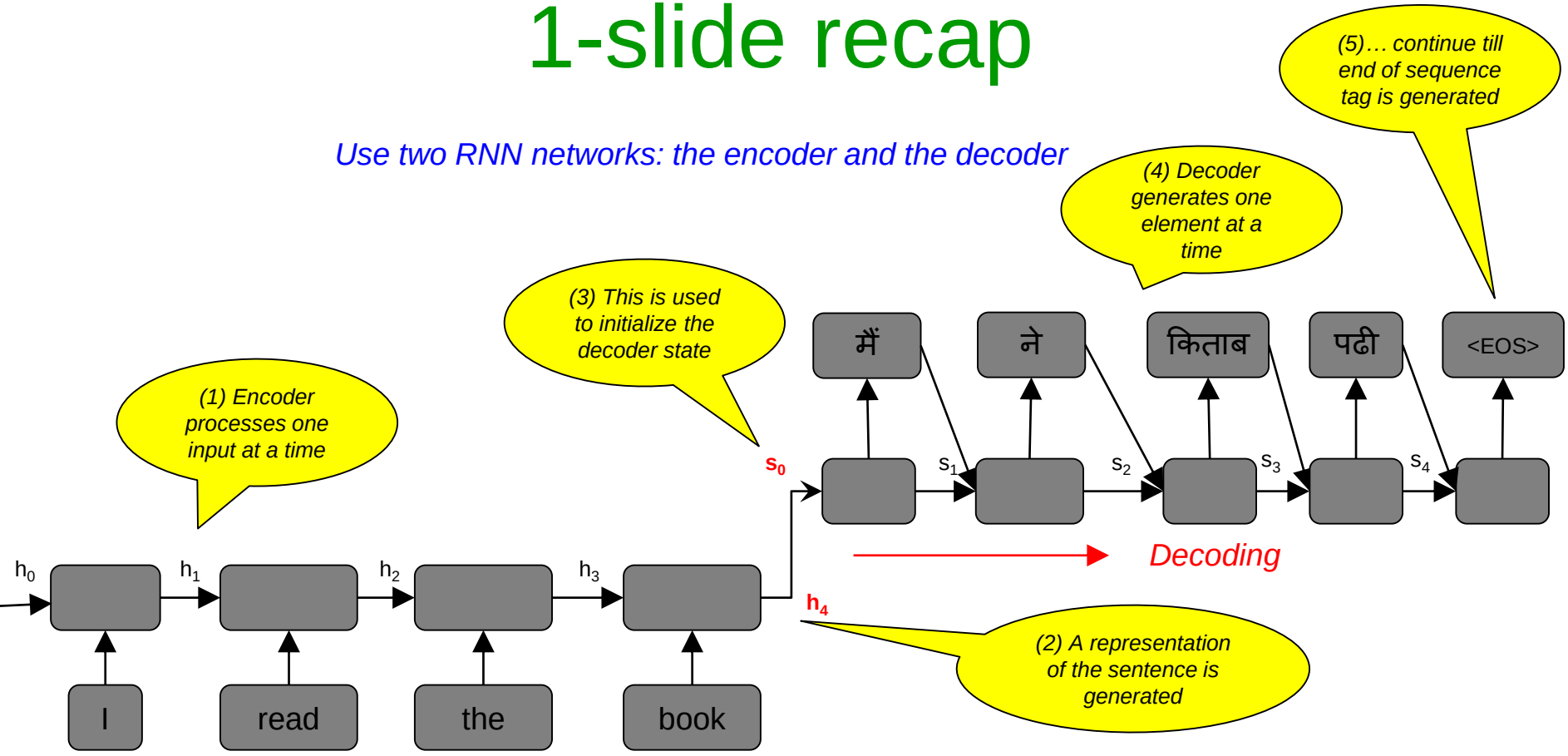
Computer Science and Engineering Department

IIT Bombay

Week 9 of 4mar24

1-slide recap

Use two RNN networks: the encoder and the decoder



https://developer.nvidia.com/blog/introduction-neural-machine-translation-gpus-part-2/

Sequence to Sequence Learning with Neural Networks Ilya Sutskever, Oriol Vinyals, Quoc V. Le. arxiv pre-print [link]

In the context of decoding, did A*, Viterbi and Beam Search

Long distance dependency is a reality

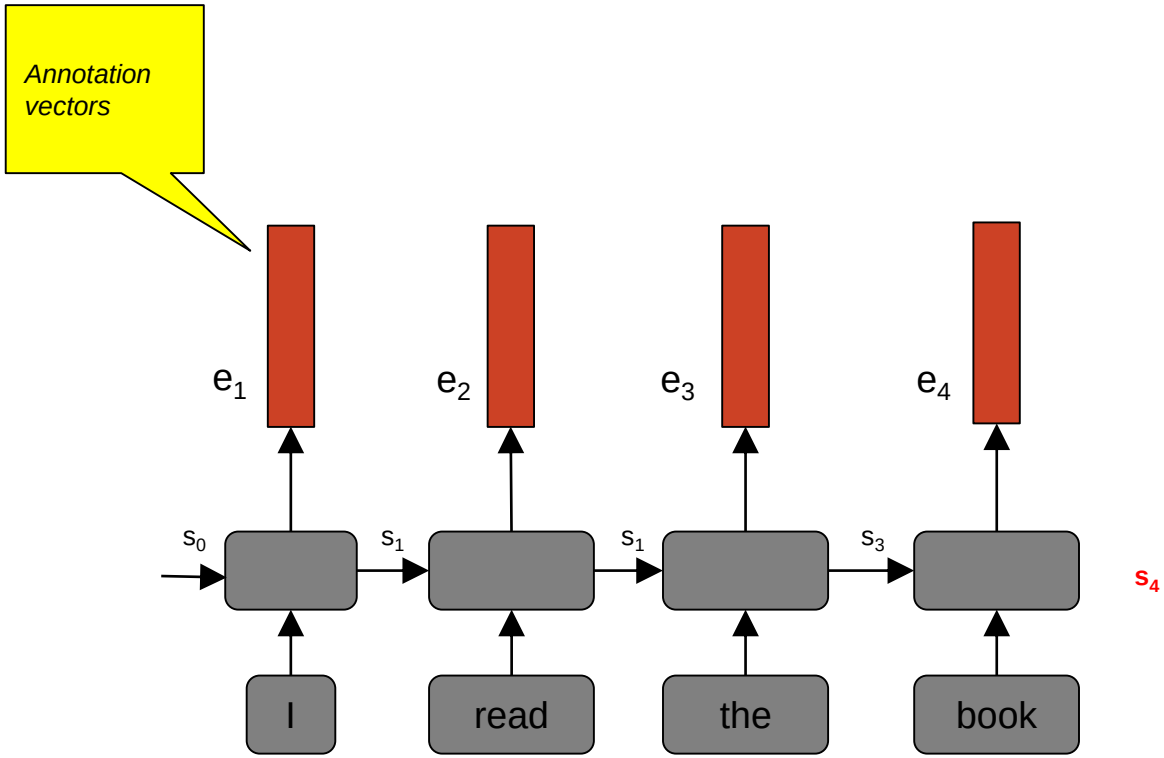
The *bank* that Ram used to visit 30 years before was closed due to the lockdown with the Govt. getting worried that crowding of people during the immersion ceremony on the river will aggravate the situation.

The entire source sentence is represented by a single vector

Problems

- Insufficient to represent to capture all the syntactic and semantic complexities
  - *Solution: Use a richer representation for the sentences*
- Long-term dependencies: Source sentence representation not useful after few decoder time steps
  - *Solution: Make source sentence information when making the next prediction*
  - *Even better, make **RELEVANT** source sentence information available*

Encode - Attend - Decode Paradigm



Represent the source sentence by the **set of output vectors** from the encoder

Each output vector at time *t* is a contextual representation of the input at time *t*

Let's call these encoder output vectors ***annotation vectors***

Bahdanau, Dzmitry, Kyunghyun Cho, and Yoshua Bengio. "Neural machine translation by jointly learning to align and translate." *ICLR 2015.*

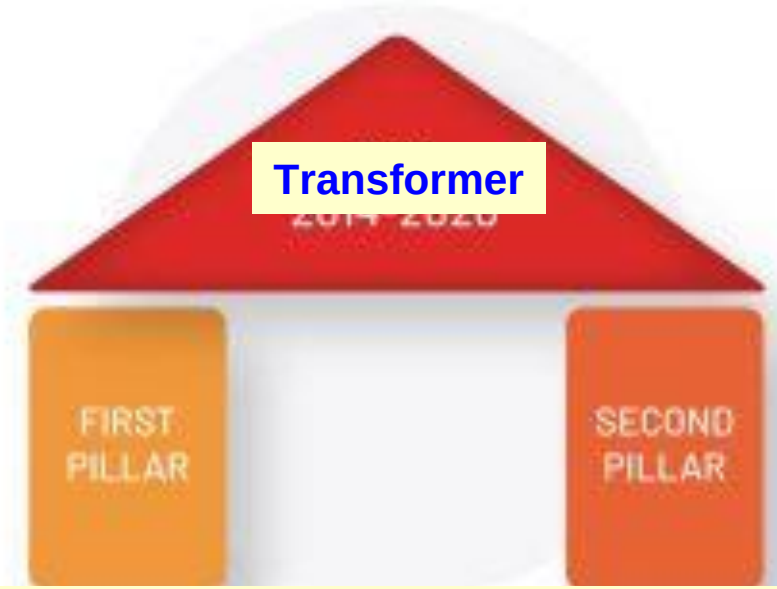
https://developer.nvidia.com/blog/introduction-neural-machine-translation-gpus-part-3/

Transfer essential

Transformer=

RNN + Attention + Positional_Encoding

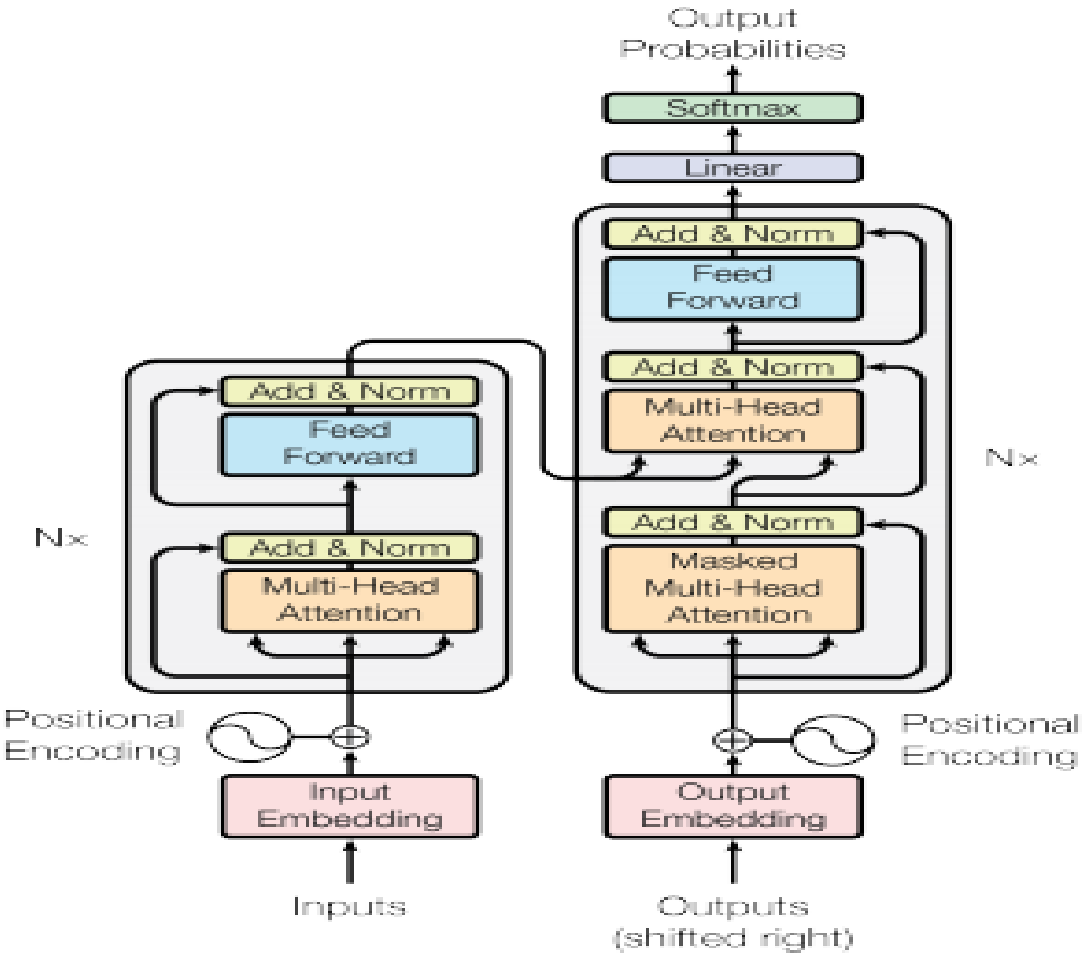
Two Pillars of Transformer



Attention + Positional Encoding

= Transformer

A classic diagram and a classic paper



Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. "Attention is all you need." NeurIPS (2017).

http://nlp.seas.harvard.edu/2018/04/03/attention.html
http://jalammar.github.io/illustrated-transformer/

Attention: Self, Multi-headed, Cross

From Bahadanu, 2015

“In order to address this issue, we introduce an extension to the encoder–decoder model which learns to align and translate jointly. Each time the proposed model generates a word in a translation, it (soft-)searches for a set of positions in a source sentence where the most relevant information is concentrated. The model then predicts a target word based on the context vectors associated with these source positions and all the previous generated target words.”

Czeck-English data

- [nesu] “I carry”
- [ponese] “He will carry”
- [nese] “He carries”
- [nesou] “They carry”
- [yedu] “I drive”
- [plavou] “They swim”

To translate …

- I will carry.
- They drive.
- He swims.
- They will drive.

Hindi-English data

- [DhotA huM] “I carry”
- [DhoegA] “He will carry”
- [DhotA hAi] “He carries”
- [Dhote hAi] “They carry”
- [chalAtA huM] “I drive”
- [tErte hEM] “They swim”

Bangla-English data

- [bai] “I carry”
- [baibe] “He will carry”
- [bay] “He carries”
- [bay] “They carry”
- [chAlAi] “I drive”
- [sAMtrAy] “They swim”

To translate … (repeated)

- I will carry.
- They drive.
- He swims.
- They will drive.

Foundation

- Data driven approach
- Goal is to find out the English sentence e given foreign language sentence f whose p(e|f) is maximum.
- Translations are generated on the basis of statistical model
- Parameters are estimated using bilingual parallel corpora

$$\tilde{e} = \underset{e \in e^*}{\operatorname{argmax}}\, p(e|f) = \underset{e \in e^*}{\operatorname{argmax}}\, p(f|e)p(e)$$

SMT: Translation Model

- $P(f|e)$: Probability of some f given hypothesis English translation e
- How to assign the values to $p(e|f)$?

$$p(f|e) = \frac{count(f,e)}{count(e)}$$ ← Sentence level

  - Sentences are infinite, not possible to find pair(e,f) for all sentences

- Introduce a hidden variable $\boldsymbol{a}$, that represents alignments between the individual words in the sentence pair

$$\Pr(\boldsymbol{f}|\boldsymbol{e}) = \sum_{\boldsymbol{a}} \Pr(\boldsymbol{f}, \boldsymbol{a}|\boldsymbol{e})$$ ← Word level

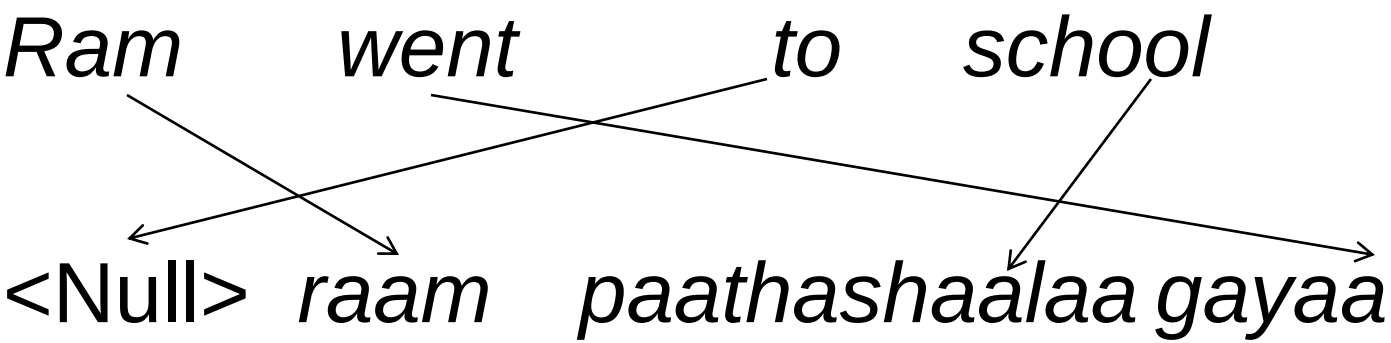
Alignment

- If the string, $e = e_1^l = e_1\ e_2\ \ldots e_l$, has l words, and the string, $f = f_1^m = f_1 f_2 \ldots f_m$, has m words,
- then the alignment, a, can be represented by a series, $\boldsymbol{a_1^m = a_1 a_2 \ldots a_m}$, of m values, each between 0 and l such that if the word in position j of the f-string is connected to the word in position i of the e-string, then
  - $\boldsymbol{a_j = i}$, and
  - if it is not connected to any English word, then $\boldsymbol{a_j = 0}$

Example of alignment

English: *Ram went to school*

Hindi: *raam paathashaalaa gayaa*



An “insight” matter

Positional Encoding

Limitation of RNN

- Encoder-decoder RNN generates a sequence of hidden states h_t, t varying from 0 to L, where L is the sentence length.
- Each h_t is a function of previous hidden state h_{t-1} and the input at position t.
- So, to process the input at t^{th} step, the encoder or decoder has to wait for $t-1$ steps.
- This sequential nature of RNN makes the training time very large.

Inspiration from Shakespeare

- “All the world's a stage,/ And all the men and women merely players”- As You Like It- Shakespeare

- All the sentence’s a stage./And all the words and punctuations are merely players”

“*children saw a big lion in the zoo in the morning*”

- main verb: *saw*;
- who (agent): *children*
- what (object): *lion*
- where (locative): *zoo*
- when (temporal): *evening*

Position Sensitivity: “Jack saw Jill” vs. “Jill saw Jack”

IF

the main verb (MV) is transitive and in past tense

THEN

the NP to the left of MV should get the ‘ ne’ postposition mark

and

The NP to the right of MV should get the ‘ko’ postposition mark

Transformer's major contribution- Positional Encoding (1/2)

- Word positions as additional disambiguation signals.
- Words influence one another by virtue of their properties and positions
- Such influences manifest in translations as morphological transformations, lexical choices, pragmatic markers and so on.
- Tenet of ML-NLP: *with sufficient data all these mutual influences can be learnt*.

Transformer's major contribution- Positional Encoding (2/2)

- Positions are encoded as embeddings and positional embeddings are supplied along with input word embeddings.

- The training phase teaches the transformer to condition the output by paying attention to not only input words, but also their positions.

Position Vector components

- Let the i^{th} component of the t^{th} position vector be denoted as *pos(t,i)*, *i* varying from 0 to *(d/2)-1*. Then

$$pos(t,2i) = \sin\left(\frac{1}{10000^{\frac{2i}{d}}}t\right)$$

$$pos(t,2i+1) = \cos\left(\frac{1}{10000^{\frac{2i}{d}}}t\right)$$

Why Sine and Cosine? (1/2)

Foundational Observation-1:

Let *S* be a set of symbols. Let *P* be the set of patterns the symbols create. If |P|>|S|, then there must exist patterns in *P*, that have repeated symbols.

Binary set $\{0, 1\}$ $|S| = 2$

$\Sigma^* = \{\phi, 0, 1, 01, 10, 11, 00, \ldots\}$

$|P| = \infty$

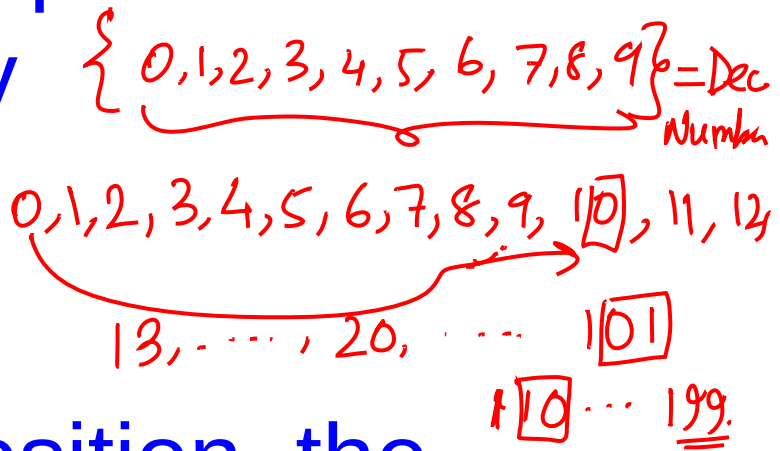
00111

Why Sine and Cosine? (2/2)

Foundational Observation-2:

IF

the patterns can be arranged in a series with equal difference of values between every consecutive pair,



THEN

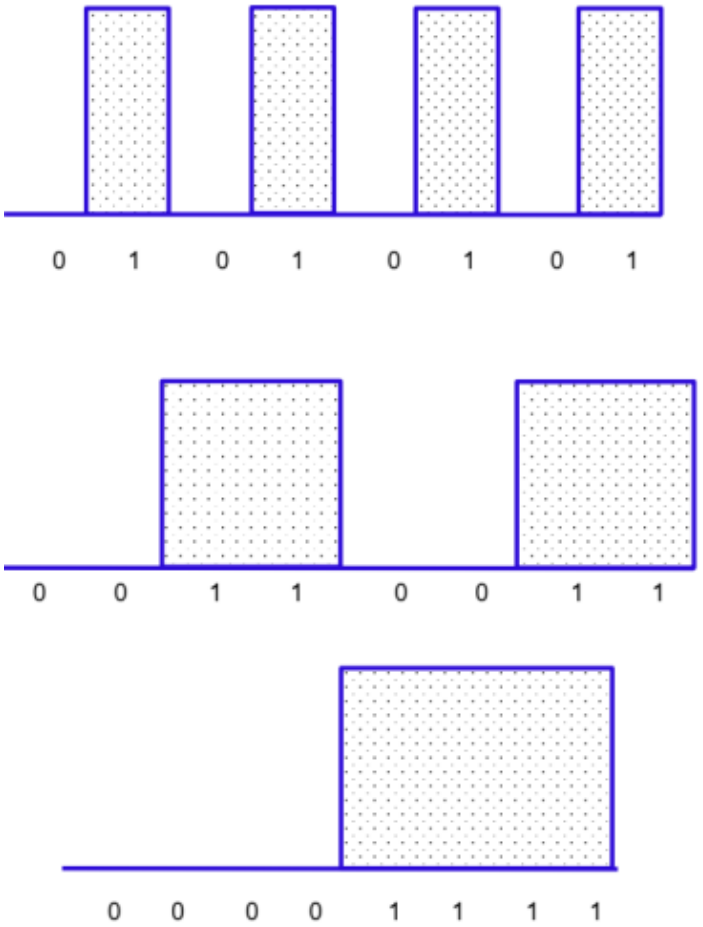
at any given position, the symbols at different positions of the pattern strings must REPEAT,

Periodicity and Decimal Integers

- 10 symbols called digits: *0, 1, 2, 3, 4, 5, 6, 7, 8, 9*.
- In the sorted list of integers in ascending order, the string length of the integer goes on increasing
- The digits repeat after every 10 numbers in the lowest significant position, after every 100 numbers in the next lowest position, after every 1000 numbers in the next to next lowest and so on.

Binary Numbers

0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1



Challenges in designing PEs

- Cannot append decimal integers as position values; words later in the sentence will dominate, by the force of their positions being large integers
- Cannot normalize too: Word relations changing with the length of sentences- linguistically untenable
- "*Oh, what a beautiful day!!*"- which expresses (i) delight, (ii) the nature of the '*day*' being '*beautiful*', (iii) '*Oh*', being an exclamatory prefix to the rest of the phrase and so on, should be invariant with respect to the sentence length

Binary values also will not do!

- *0*s will contribute nothing, and *1*s will influence completely.
- Such *black-and-white* (0-1) hard decisions go against the grain of NLP whose other name is ambiguity.
- A language object represented by a vector must allow *soft choices* in its components, preferably represented by values in the closed range [0,1].

Criteria PEs should satisfy

- Should be *added* component by component to the word vector.
- Components should range from 0 to 1, both included.
- Components should be periodic, since they represent consecutive integers.
- Ingenious on the part of the creators of transformers to spot that *sine* and *cosine* functions meet the above requirements.