Dr. Siddharth Ranjan Paper 1:

Brain's processing architecture - Cognitive tax on every word processed.

- Locality effects and processing costs
- How we process language in real time
- English - subject verb object language cat chase the mouse
  Hindi - subject object verb
- When we change the word order, we change the strategy the brain uses to predict the future
- Dependency locality theory
- Frequency effect (word level)
  - Frequent words are processed faster and read aloud than infrequent words.
  - Dual route cascade model (DRC)
  - Memory is like a massive interconnected grid of nodes (words)
  - Every word has a resting activation level (dimmer switch)
  - Priming effect, negative coefficient between frequency of word vs read time.
  - This is cognetive tax paid by brain for infrequent words.
- DLT

1) Integration cost (tax 1)
- Looking backward in a sentence - connect new word read to previous word it depends on.
  e.g. The boy [ate] the apple -①
  The boy who stood in the corner [ate] the apple -②
  when the word "ate" is read in both sentences the integration cost is more in 2nd sentence as it needs to think more on who ate, where he ate etc.

2) Storage cost (tax 2)
- Looking forward in a sentence - Mental burden for keeping a placeholder active in memory.
- Counting the number of predicted heads or incomplete predictions at any given point in the sentence.
- Higher storage costs = more reading time

Arguments vs adjuncts in a sentence.
↓
essential/core building blocks
extra flavour to sentence.

Arguments and adjuncts are processed symmetrically

Ideally Hindi anomaly :-
Arguments should be processed faster than adjuncts in Hindi
But its not much faster than english because since verbs comes last in Hindi we already can predict what the verb could be.

Information : Topic vs focus

↓ given information . ↘ New information .

The boy ate the choclate cake .

↓ ↓

Hypothesis old/familiar (should be faster) surprise (should be slower)

Actual findings 98.34 ms (slower) 94.36 ms . (faster)

Prediction vs familiarity

↓

Anchoring / imagening

(cognitively heavy)

novelty ≠ difficulty

Forward Surprisal

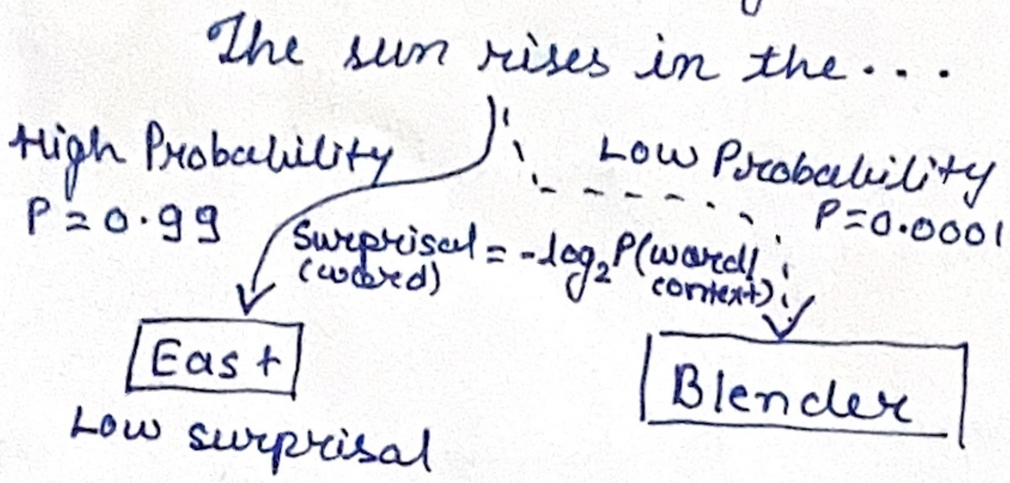
- Next word to be or yet to read dictates how much time we take for current word we are reading.
- If next word is difficult, our brain slows down the process of reading for current word.

Paper 2 :- Predictibility effects of Content and Function words in Hindi Silent Reading -

Architecture of Sentence Processing -

- UID, surprisal, word order
- Why do we say "I gave him the book" instead of "I gave the book to him"
- Is syntax random or are we optimizing for listener's cognitive bandwidth

Bit and Brain : Refining Surprisal



Information as unexpectedness

- In psycholinguistics "Information is not meaning, It is unexpectedness"
- The Brain as a prediction Engine : we constantly pre-activate likely future inputs
- The last : When input matches prediction (Low surprisal), processing is cheap when it violates prediction (High surprisal) we pay a cognitive tax

Eye movement : Circadic masking (Brain literally cuts the video feed during eye movement) to prevent motion sickness from own eyeballs.

Reading : Error checking, prediction, massive memory management.

FPRT (First Pass reading time) :- 200-300 ms. lexical access. identifying shape color

TRT (Total Fixation time) :- sum of all time to read word.
captures confusion, integration time (cognitive cost) to integrate word into sentence

Surprisal theory (Claude Shanon's information theory)
Calculate surprisal for every word in sentence.

1. Backward lexical surprisal :- Given previous two words how likely is this new word
   looking at immediate neighbours.
   Do these 3 things belong together
   immediate familiarity check.
2. Forward surprisal :- Measures how much current word ~~measures~~ helps to predict next word.
   Peripheral vision already scanning next word.
3. Probabilistic context free grammer :- Structure surprise.
   Cares about category of word.
   Measures shock about surprise about structure / grammer in sentence.

DLT (Dependency locality theory)

1. Integration cost :- Integrating a word into sentence, Making connection of words to its noun.
2. Storage cost :- Energy required/cognitive load for holding thoughts about different words in mind like (chrome's open tabs)

Content words vs function words (glue words)

e.g. the cat eats the mouse

it was believed that content words are cognitively heavy function words are processed automatically

Hindi proves this false

Initially content words and function words are treated same but :-
- in hindi filler words are not just fillers they are important to understand structure of sentence.
- In hindi grammatically surprising function words. are crucial information "case markers", syntax markers.
- structural pivot.

(familiarity check, Backward surprisal (immediate word prediction))

Analogy :- function words are glue words in hindi like ने and को are like navigation signs on road which tells our brain which meaningful path to go while content words are like location or scenery on road.

- Brain doesn't slow down when brain encounters these function words even when grammatically surprising. It accelerates a structural rebuild.

Easy reader model

1) 0-50 ms - visual uptake, photons from words hit retina, discover lines, curves, contrast, etc. we don't even know its a word.
2) L1-stage - 75ms-100ms - familiarity check, backward surprisal, 3 gram check
   - is the shape familiar?
   - does it fits the probability of last two words we saw?

Backward surprisal and PCFG (grammer check) significantly affects the early measure (FPRT) - before we even know the dictionary definition of the word my brain is already checking its probability.

3) L2-stage -
   - lexical access - unlock the meaning
4) post lexical processing, saccade planning - where to move eyes next.
   - forward surprisal and storage cost only affects the late measure, total fixation time.
   - we don't worry about the future or heavy memory load of sentence until we confirm what current word is.

This shows how much ram we are consuming. we are not just passive receivers, we are active readers. we are constantly utilising our brain ram and compute for planning while reading every milliseconds.

How is this different from LLMs? (also word predictors)

- Hoffman :- Language models explain word reading times better than impirical predictibility.

Human wisdom scarce :- psychologists traditionally use the cloze test.
- 100 humans are asked to predict a word after 3 words in a sentence. This is then used to calculate probability of word based on total guesses of all humans.
- But LLMs can predict word better probability that matches ~~with~~ closer with human reading times.
- AI predicts it better that how fast a human can read a word than other humans can.

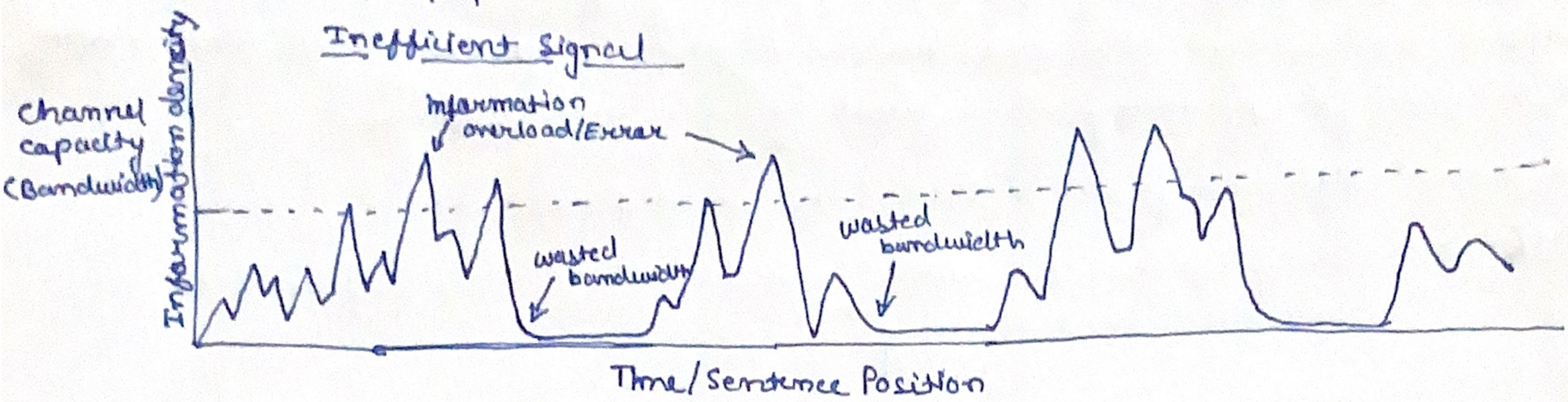
AI models with simple n-grams are excellent at capturing short range lexical access (familiarity check) at L1 stage (forward surprisal)
complex AI models have much better memory they can look back farther back in the sentence. They are much better at early preprocessing for predicting next word.
our brains aren't running a single program but multiple complex algorithms at once.
A simple scanning algorithm for ~~prediction~~ immediate word
A much more complex neural network for the context in future.

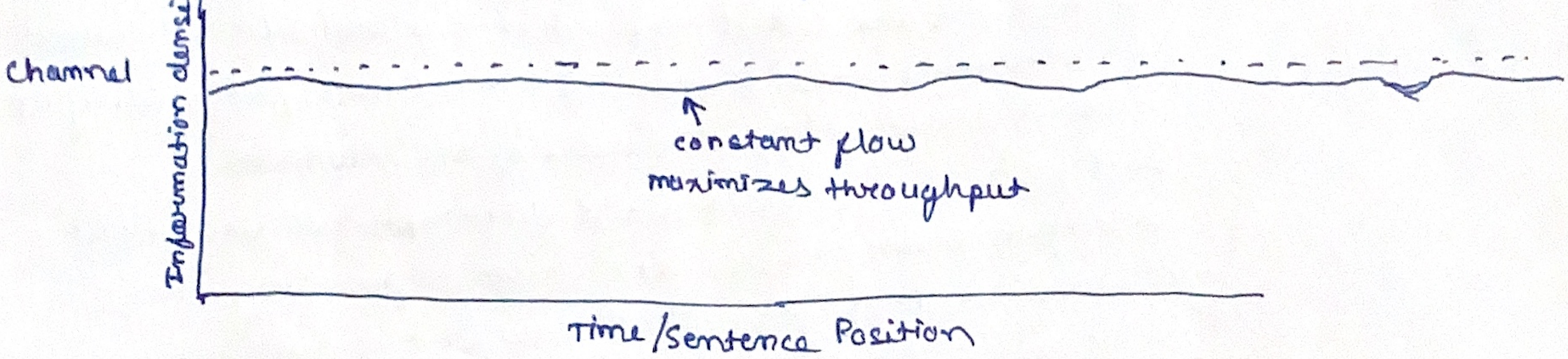
The logarithm of our reading time correlates perfectly with surprisal which is -ve log probability is a proof that a fundamental level our brain is running on probability statistics (betting on what comes next)
Are ai models getting better at understanding the process of reading than humans who is actually doing the reading?
And more fundamentally, what real difference between predicting next word and genuine understanding?
That is the modern version of chinese rail!- Does machine actually get the story or does it have incredibly sophisticated statistical map that tells it water is very likely to follow drink?
Exactly where the comprehension? I think the Hindi study actually gives us a little hint about the difference.
Current AI models, especially the big transformer models, treats all ~~words~~ tokens more or less equal vectors in a high dimensional space. our brains clearly do this complex hierarchical ~~restructure~~ restructuring.
Is AI actually building the grammatical structure the way we are? or is it just brute forcing the probability so well that it looks the same to outside? AI may be missing structural

cognition evident in Hindi

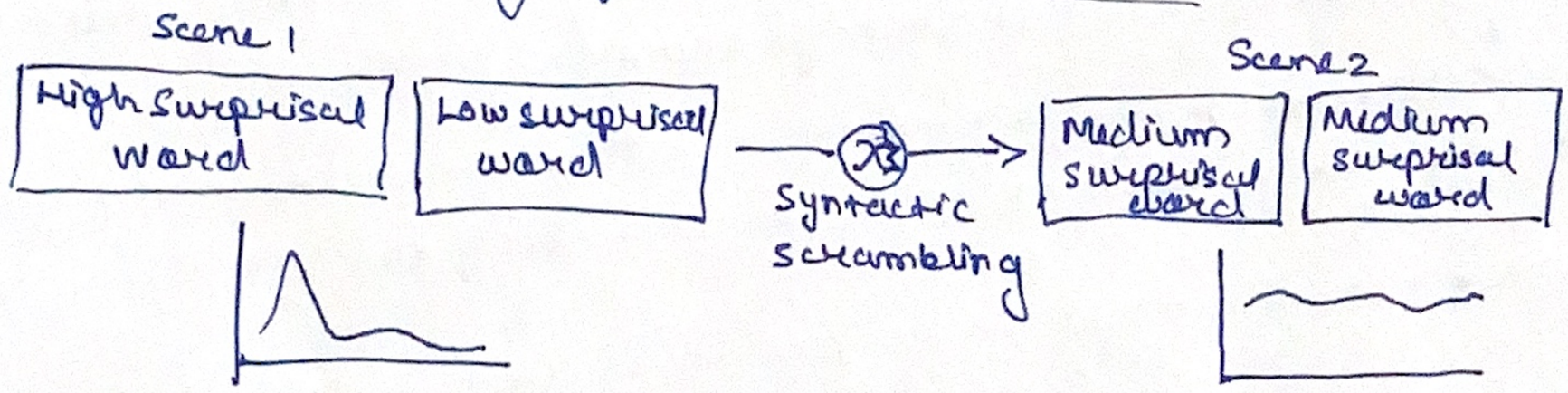
Paper 3

UID (2018) paper.



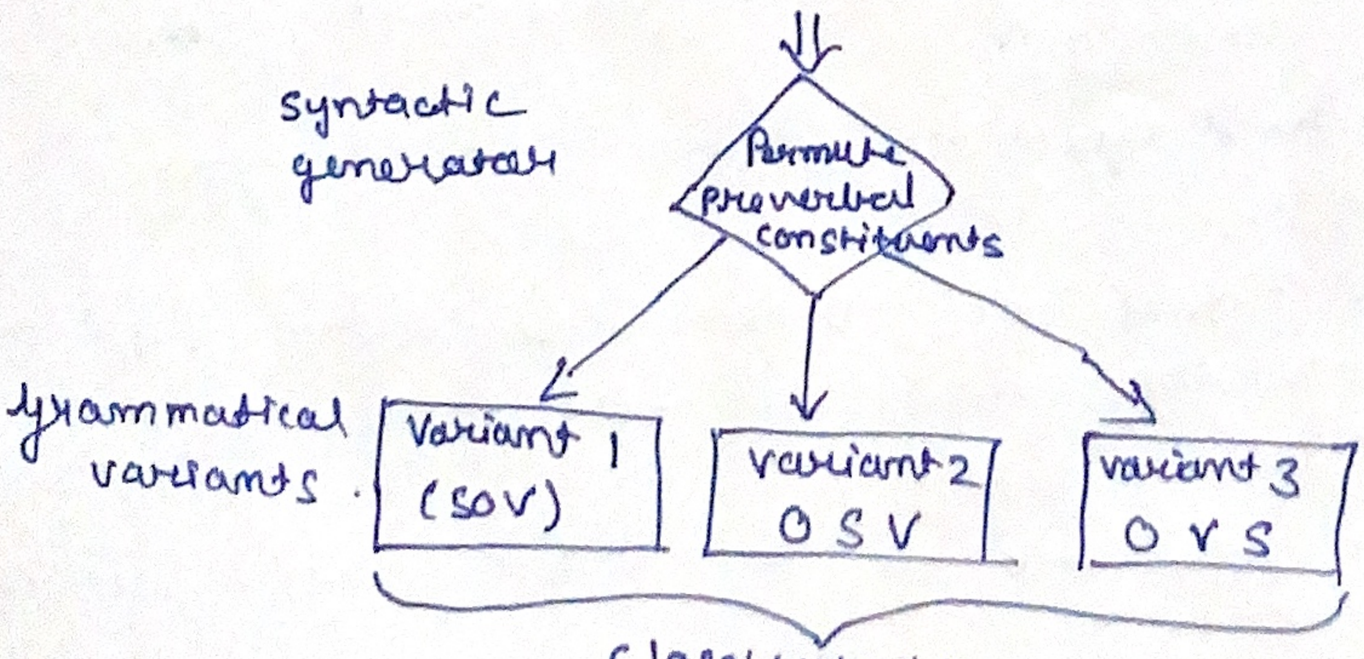


Mechanism: Smoothing signal via word order.



If a word is highly unexpected, UID predicts we should:
1. Move it later in sentence (allowing context to build up)
2. Insert functional words (like 'that') to dilute the density
To test this we need a language ~~needed~~ with massive flexibility
we need Hindi SOV scrambling

generating counterfactual.



Methodology -
- we cannot understand a choice unless we know what was rejected
- Dataset:- Hindi sentence vs English variants
  - variants (175,801)
  - variants (8,385)

Hypothesis
- If UID is true, real sentence should be mathematically smoother than artificial variants

Feature Engineering : Quantifying Smoothness.

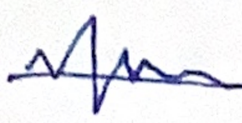
To test Hypothesis, we extract statistical features for every variant.

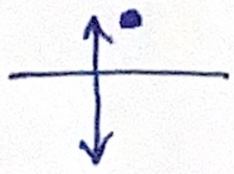
Mathematical goal : we define Smoothness as standard deviation of surprisal

Prediction : Co-efficient for "standard Deviation" should be negative and Significant (As variance increases, probability of being "Real" sentence decreases)

Sentence Feature vector.

1) Σ Sum of Surprisal
   Total information content

2)  Standard deviation of surprisal
   Crucial Metric : A lower SD means a flatter, smoother line.

3)  Delta to mean
   How far individual words deviate from average.

Hindi Case Study :-

- Unlike English (Rigid SVO), Hindi is (flexible SOV) language
- Arguments in hindi can scramble (more extensively before the verb)
- If UID drives word order, Hindi Speakers have perfect toolkit to optimise it. we should see massive evidence of smoothing here.
- But in reality UID fails to provide significant predictors
- "Smoothest" sentence are not the ones humans speak. "Real" sentences have "bursts" of information. The 'smooth pipe' analogy fails empirical scrutiny in Hindi. * classification accuracy : lexical Surprisal Model (89.96%.) vs UID measures (50%.)

Superior Model : Maximising local predictability -

- Data suggests 'Trigram surprisal' is primary driver
- Interpretation : 'Brain is greedy'. It does not plan the global contour of paragraph. It optimises for immediate next word
  we prefer a predictable step now over a smooth path later

Secondary Driver : Dependency length Minimisation (DLM)

"Keep related things close" long dependencies burden working memory

Finding : DLM matters, but it is secondary to surprisal

Trade off : we will tolerate a longer dependency distance if it makes next word more predictable. we use "fast and frugal" heuristics.

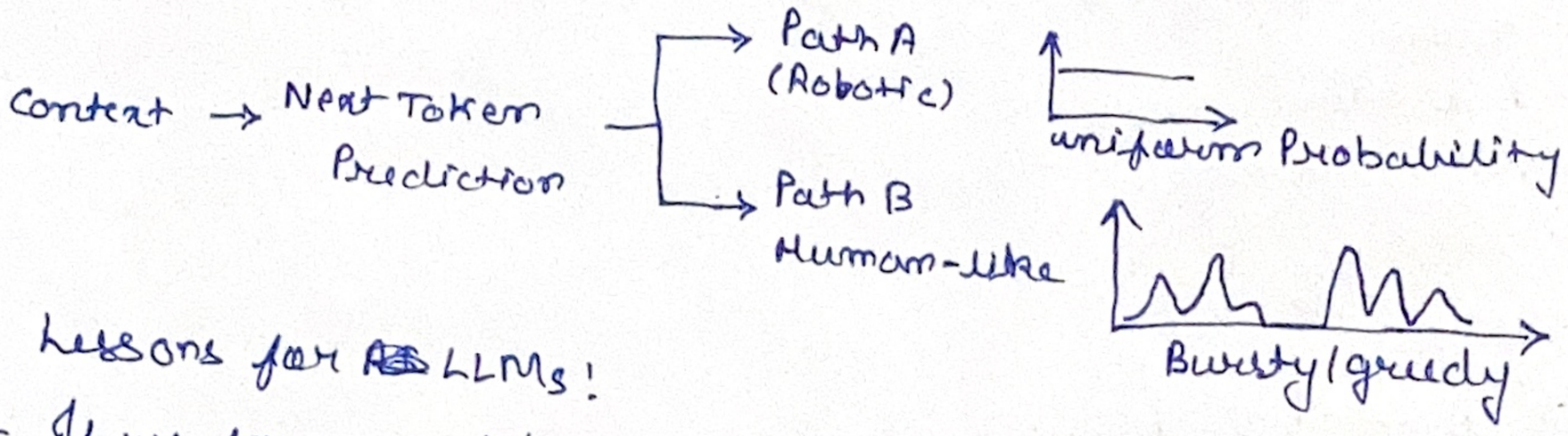
Across topologically diverse languages, pressure for local predictability outweighs pressure for global uniformity

Role of Case Markers in Hindi

Ram ने → किताब को

- Hindi needs strict smoothing because:
  - Explicit tagging: Case markers (ने, को) act as signposts, explicitly labelling who did what to whom.
  - When a language has explicit tagging, brain doesn't need to rely on word order for understanding the sentence structure and composition, case markers absorb the cognitive load.

Implications for AI: Decoding Natural Generation.



Lessons for LLMs:
- If we force models to generate 'smooth' information density (low variance) output becomes monotonous.
- Natural language is bursty. Good generation requires managing local transition probabilities (Attention), not forcing global uniformity
The next token prediction objective aligns with human psycholinguis