

LLMs in Psycholinguistics

Romance Lab

Goethe-Universität Frankfurt

Part II: 21st July 2025

Umesh Patil

Overview

1. Recap: Part I
2. Generate surprisal predictions using different LLMs from *Hugging Face* for:
 - a. Experiment 1: **German** Agreement Attraction
 - b. Experiment 2: **Spanish** Clitic Doubling
 - c. Experiment 3: **Portuguese** Semantic Congruency
 - d. Experiment 4: German Text Simplification
3. Q & A

Recap: LLM Concepts

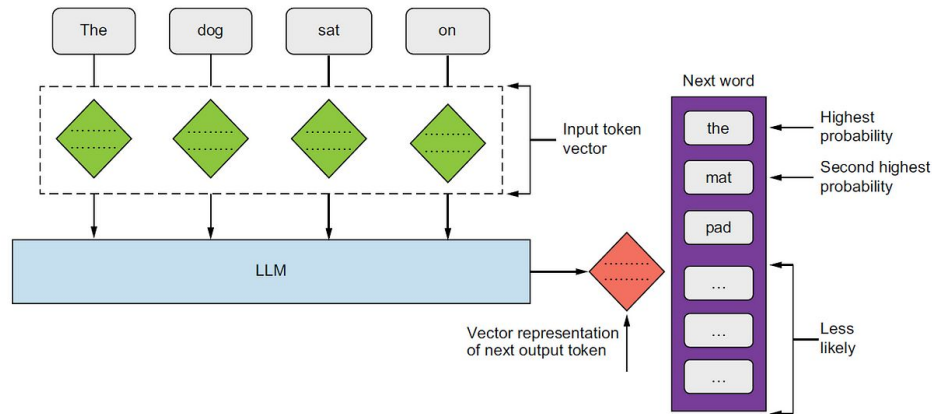
1. LLMs & Next Word Prediction

2. Tokenization

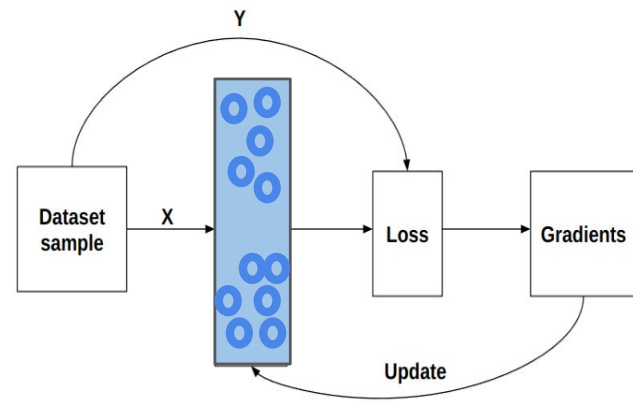
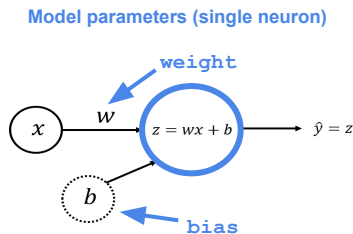
- Subword Tokens
- Vocabulary

3. Training

- Parameters (**Weights** & Biases)
- Loss
- Backpropagation
 - Gradients
- Gradient descent



minicons: Enabling Flexible Behavioral and Representational Analyses of Transformer Language Models

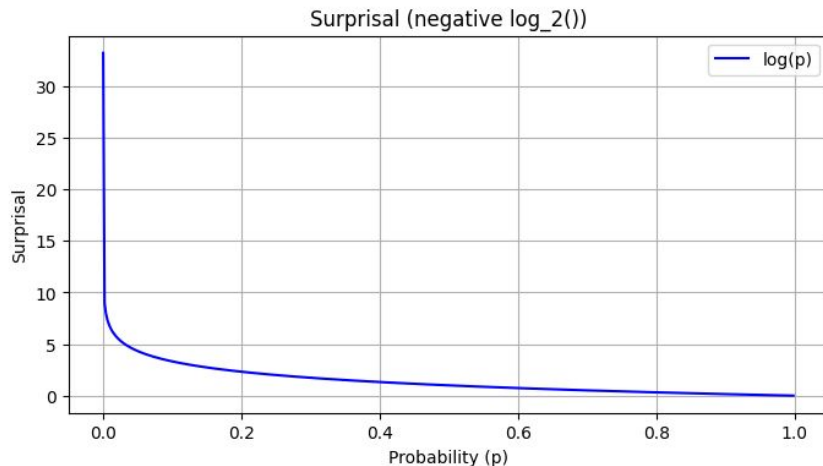


Recap: Surprisal

- **Surprisal:** Information conveyed by any given linguistic unit \mathbf{x} (e.g. phoneme, word, utterance) in context.
- Surprisal is:
 - high, when \mathbf{x} has a low conditional probability, and
 - low, when \mathbf{x} has a high conditional probability.
- Claim: Cognitive effort required to process a word is proportional to its surprisal (Hale, 2001).

$$\text{surprisal}(x) = \log\left(\frac{1}{P(x \mid \text{context})}\right) = -\log P(x \mid \text{context})$$

For LLMs: $\text{surprisal}(w_t) = -\log P(w_t \mid w_{1:t-1})$.



Recap: minicons functions

```
from minicons import scorer
```

```
model_name = "HuggingFaceTB/SmolLM2-135M"
```

```
lm_inc_scorer = scorer.IncrementalLMScorer(model_name)
```

The **Hugging Face** name of the LLM you want to use



Recap: minicons functions

The **Hugging Face** name of the LLM you want to use

```
from minicons import scorer
```

```
model_name = "HuggingFaceTB/SmolLM2-135M"  
lm_inc_scorer = scorer.IncrementalLMScorer(model_name)
```

```
lm_inc_scorer.token_score(  
    input,  
    bos_token=BOS,  
    prob=True/False,  
    surprisal=True/False,  
    bow_correction=True  
)
```

A list of surprisal (for `surprisal=True`) values or probabilities (for `prob=True`) for each token / word in the input text; if the input contains a list of texts then a list of list of probability or surprisal values (one list per text)

```
lm_inc_scorer.word_score_tokenized(  
    input,  
    bos_token=BOS,  
    prob=True/False,  
    surprisal=True/False,  
    bow_correction=True,  
    tokenize_function=tokenizer  
)
```

```
lm_inc_scorer.sequence_score(  
    input,  
    bos_token=BOS,  
    prob=True/False,  
    surprisal=True/False,  
    bow_correction=True  
)
```

A single *log* probability value for the entire input text denoting average log probability of a token in the input

```
*  
if "gpt2" in model_name or "pythia" in model_name or "SmolLM" in model_name:  
    BOS = True  
else:  
    BOS = False
```



<https://shorturl.at/lZCpm>

Experiment 1: German Agreement Attraction

gram_sg (GRAM, match):

Der Vertrag, den die Diplomaten feierlich unterzeichneten bei der Konferenz, hatte viele Seiten.

gram_pl (GRAM, mismatch):

Die Verträge, die die Diplomaten feierlich unterzeichneten bei der Konferenz, hatten viele Seiten.

ungram_sg (UNGR, match):

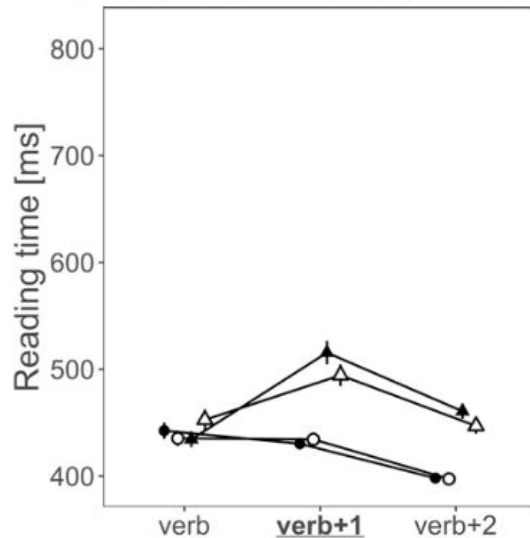
Der Vertrag, den der Diplomat feierlich unterzeichneten bei der Konferenz, hatte viele Seiten.

ungram_pl (UNGR, mismatch):

Die Verträge, die der Diplomat feierlich unterzeichneten bei der Konferenz, hatten viele Seiten.

‘The treaty/treaties that the diplomat(s) solemnly signed at the conference had many pages.’

- GRAM, match ○ GRAM, mismatch
- ▲ UNGR, match △ UNGR, mismatch



Experiment 2: Spanish Clitic Doubling

Dative:

Laura dio flores a Rafael en el parque de las afueras de la ciudad.

Dative_doubled:

Laura le dio flores a Rafael en el parque de las afueras de la ciudad.

Accusative:

Laura encontró a Rafael en el parque de las afueras de la ciudad.

Accusative_doubled:

Laura lo encontró a Rafael en el parque de las afueras de la ciudad.

Experiment 3: Portuguese Semantic Congruency

Congruent:

O Glipi vai comer a bolacha.

Incongruent:

O Glipi vai comer a rosa.



Experiment 4: German Text Simplification

Standard German:

In der alten römischen Religion und Mythologie ist Janus der Gott des Anfangs und der Tore. Er hat eine doppelte Natur und wird gewöhnlich mit zwei Gesichtern dargestellt, da er in die Zukunft und in die Vergangenheit blickt. Janus überwachte den Beginn und das Ende von Konflikten und damit auch Krieg und Frieden. Die Türen seines Tempels waren in Kriegszeiten geöffnet und geschlossen in Zeiten des Friedens. Als Gott der Tore wurde er außerdem mit dem Betreten und Verlassen von Haustüren in Verbindung gebracht. Janus symbolisierte häufig Veränderungen und Übergänge, wie den Wechsel von einer Bedingung zur anderen, von einer Perspektive zur anderen und das Heranwachsen junger Menschen zum Erwachsenenalter. Deswegen wurde Janus zu Beginn der Saat- und Erntezeit verehrt, genauso wie zu Hochzeiten, Todesfällen und anderen Übergängen. Für Janus war kein spezieller Priester zuständig, sondern der höchste Priester leitete die Zeremonien selbst. Janus stellte den Mittelweg zwischen Barbarei und Zivilisation, ländlichem und urbanem Leben, Jugend und Erwachsenenalter dar. Die alten Griechen hatten kein Äquivalent zu Janus, den die Römer ganz für sich beanspruchten.

Simplified German:

Janus ist ein Gott. Er ist der Gott für den Anfang und die Tore. Er hat zwei Gesichter.

- Ein Gesicht schaut nach vorne.
- Ein Gesicht schaut nach hinten.

Janus passt auf den Beginn und das Ende von Kriegen auf.

Er passt auch auf den Frieden auf.

Im Krieg waren die Türen des Tempels offen.

Im Frieden waren die Türen des Tempels zu.

Er war auch der Gott für Türen.

- Er war zuständig, wenn man ins Haus kommt.
- Er war zuständig, wenn man das Haus verlässt.

Janus bedeutet Veränderung.

Das ist zum Beispiel:

- Wenn etwas neu wird.
- Wenn man von klein zu groß wird.
- Wenn man die Meinung ändert.

Janus wurde zu bestimmten Zeiten verehrt.

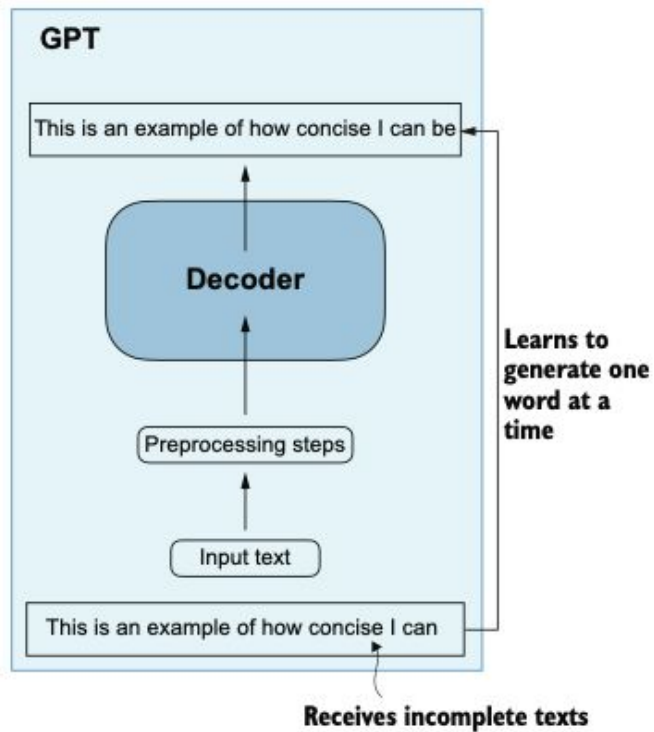
- Zu Beginn der Saat- und Erntezeit.
- Bei Hochzeiten.
- Bei Todesfällen.
- Bei anderen Übergängen.

Für Janus gab es keinen eigenen Priester. Der höchste Priester machte die Zeremonien. Janus zeigt den Weg zwischen.

- wild und zivilisiert.
- Land und Stadt.
- jung und erwachsen.

Die alten Griechen kannten Janus nicht. Die Römer kannten Janus.

Types of (transformer-based) LLMs: Decoder-only (aka autoregressive)

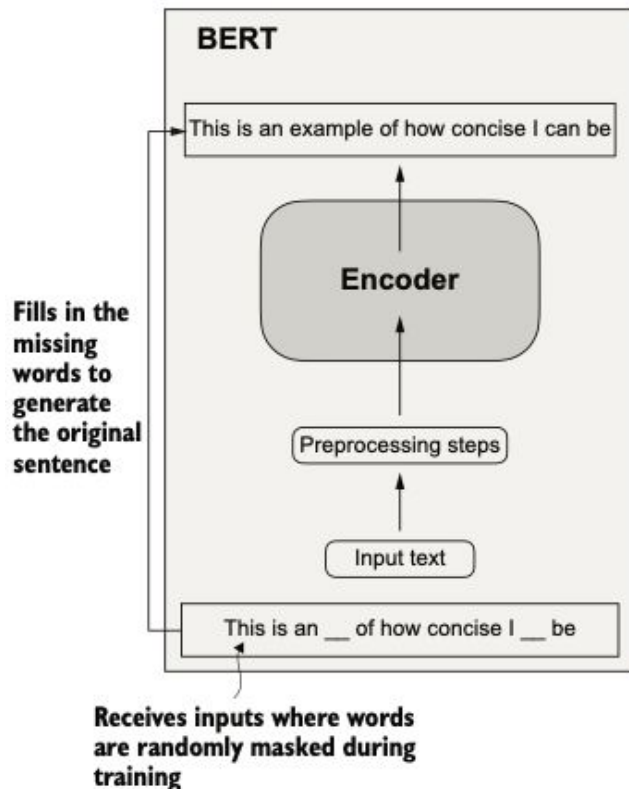


⇒ Ideal for calculating *surprisal*

using minicons:

```
scorer.IncrementalLMScorer()
```

Types of (transformer-based) LLMs: Encoder only

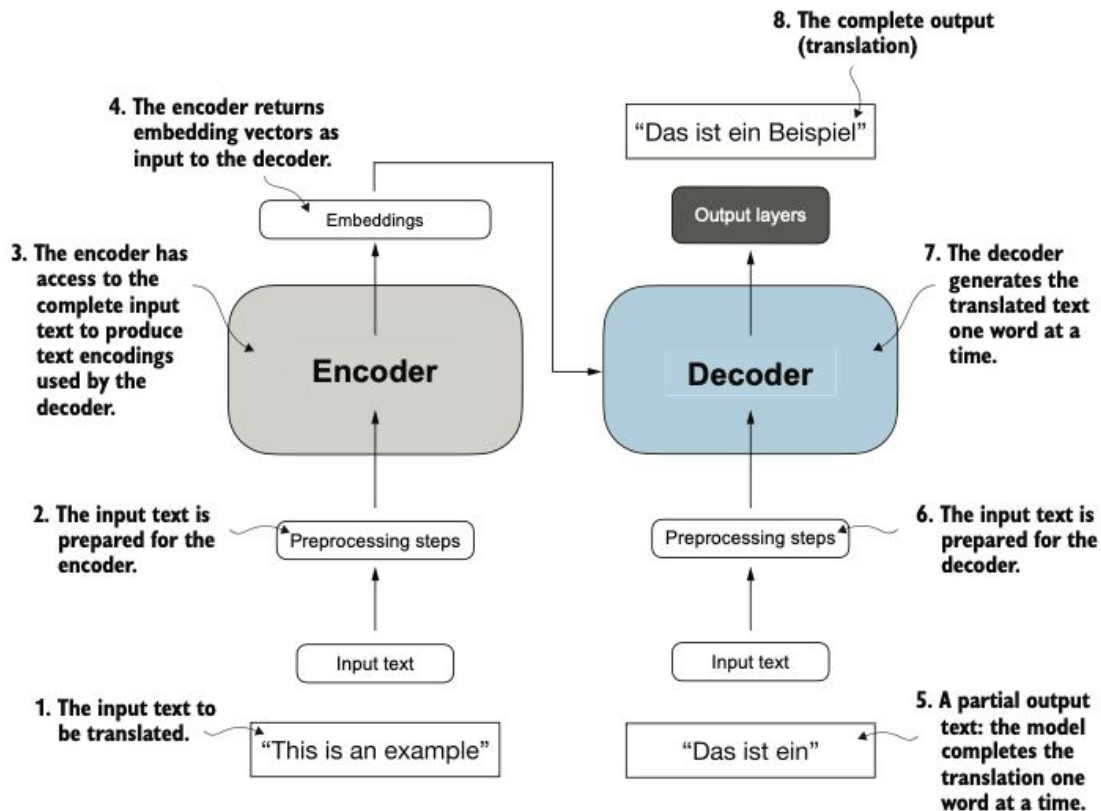


⇒ Can be used for calculating *surprisal*

using minicons:

```
scorer.MaskedLMScorer()
```

Types of (transformer-based) LLMs: Encoder-Decoder



Types of (transformer-based) LLMs

```
from helper_functions import get_model_architecture_type

model_name = "HuggingFaceTB/SmolLM2-135M"
get_model_architecture_type(model_name)
```

⇒ 'decoder-only'

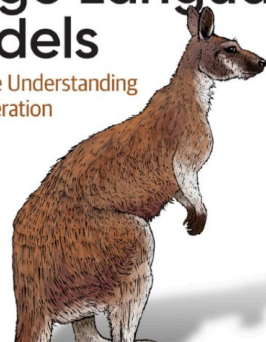
Resources

- Talk: Kyle Mahowald's HSP-talks
→ LLMs' relevance for psycholinguistics
- Book: Hands-On Large Language Models
– Jay Alammar, Maarten Grootendorst
→ More details about how to use LLMs
- Book: Build a Large Language Model (From Scratch)
– Sebastian Raschka
→ More technical... for learning the internals of LLMs

O'REILLY™

Hands-On Large Language Models

Language Understanding
and Generation



BUILD A

Large Language Model

Sebastian Raschka



FROM
SCRATCH

