Investigating the Relationship Between Surprisal and Processing in Programming Languages

Nicole Dodd[1], Skyler Reese[1], Emily Morgan[1]

[1]University of California, Davis

Comparing natural languages to code

Programming languages are like **natural languages** in some ways:

- we "read and write" code
- there are well-known code "idioms"
- code has formal syntactic rules

However, they are also *distinct* from natural languages:

- no "native" speakers of code
- very domain-specific use
- code has infinitely large lexicons yet limited syntax

Are the cognitive mechanisms recruited while processing programming languages similar to those used while processing natural languages?

We further test this question by investigating the relationship between **surprisal** and **predictability** while reading code.

Predictability and language processing

Predictable language facilitates processing in natural languages [5, 6]

- This effect is operationalized through **surprisal** (negative log probability of a word in context)
- high surprisal = more processing difficulty

Predictability appears to play a key role in code reading and writing:

- Code is **more cognitively demanding** to read than natural text [3]
- Programming languages are **more predictable** than natural languages [7, 8]
- Programmers **prefer more predictable code** [1]
- Predictable code **facilitates code comprehension** [2], but this facilitation is modulated by *programmer expertise*

No studies have yet analyzed how difficulty in code processing presents during incremental, on-line processing.

Current Study

Does surprisal predict processing difficulty in code?

We compared surprisal values of individual tokens (i.e., "words") in code to eye movement behaviors:

- prediction: high surprisal = longer total fixation times / increased proportion of regressive saccades

We also considered how programmer expertise might affect predictability effects.

If programming languages use a similar processing mechanism as natural languages, there should be **a positive correlation between surprisal and processing difficulty**.

Acknowledgements

This research was supported by the National Science Foundation CCF (SHF-MEDIUM) Grant to EM [grant number 2107592].

Presenter: Nicole Dodd

Email: ncdodd@ucdavis.edu

UC DAVIS DEPARTMENT OF LINGUISTICS

Methods

```
public class Rectangle {
      private int x1 , y1 , x2 , y2 ;
      public Rectangle ( int x1 , int y1 , int x2 , int y2 ) {
          this.x1 = x1 ;
          this.y1 = y1 ;
          this.x2 = x2 ;
          this.y2 = y2 ;
      }
      public int width ( ) { return this.x2 - this.x1 ; }
      public int height ( ) { return this.y2 - this.y1 ; }
      public double area ( ) { return this.width ( ) * this.height ( ) ; }
      public static void main { String [ ] args ) {
          Rectangle rect1 = new Rectangle ( 0 , 0 , 10 , 10 ) ;
          System.out.printIn ( rect1.area ( ) ) ;
          Rectangle rect2 = new Rectangle ( 5 , 5 , 10 , 10 ) ;
          System.out.printIn ( rect2.area ( ) ) ;
      }
}
```

Eye Movements in Programming (EMIP) Dataset [9]

- 216 programmers of different experience levels completing 2 code comprehension tasks
- Read two functions and answered a multiple-choice question about the output of each function
- Language options: Java (~96%), Scala (~2%), Python (~2%)
- Participants self-rated their expertise as ~~"none"~~, "low", "medium", or "high"

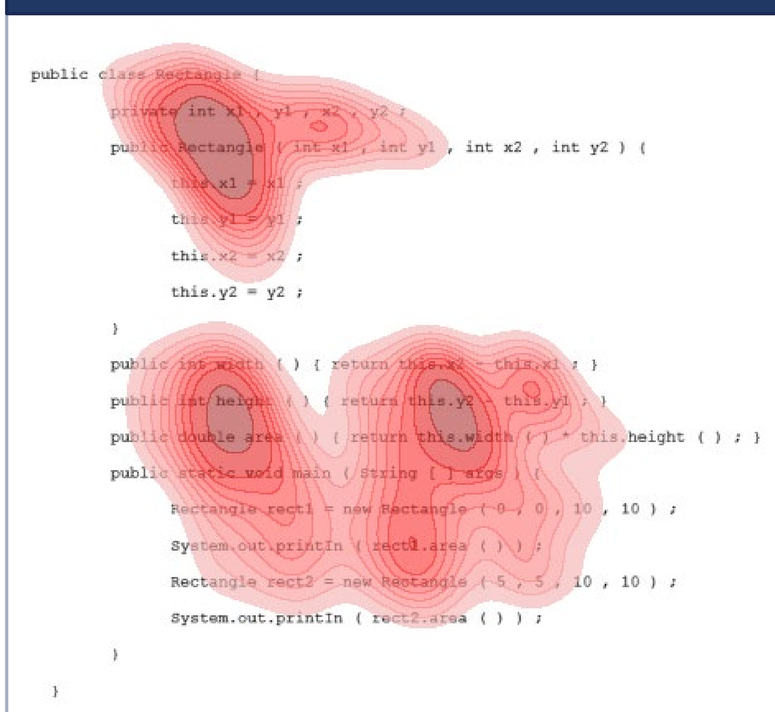
Tokens:

- Categorized tokens into open, closed, and punctuation classes
- Calculated surprisal of each token using OpenAI's Codex [10]

Eye movement measures:

- total fixation duration
- first pass regressive saccade
- overall regressive saccade

Results



Surprisal was not a significant predictor of fixation metrics across token classes, and neither was programming language expertise

Open class tokens were read significantly longer and closed class tokens were read significantly shorter

Programming language expertise had some effect on fixations metrics:

- shorter first fixation durations for high vs. low expertise programmers for open class tokens

On the other hand, **surprisal was a significant predictor of regressions** for open and closed class items; however, **in the opposite predicted direction:**

- tokens with high surprisal resulted in *fewer* first pass and overall regressions away from the token

Discussion and Conclusions

- Surprisal was not a significant predictor of fixation metrics; rather, the main determiner of difference in reading times was token class, and to some extent, programmer expertise
    - Open class tokens were read longer while closed class tokens were read shorter, suggesting that programmers pay more attention to items like variable names than repeated, template-like function terms
- Conversely, surprisal had a significant effect on regression measures: tokens with high surprisal resulted in significantly fewer first pass and overall regressions away from the token
    - Reading in code is much less linear than in natural text, so it is possible that readers make regressions for reasons other than processing difficulty
    - It is also possible that readers need to read more progressive tokens to contextualize a high-surprisal item before regressing away to something else
- *Limitations*: the stimuli were relatively short and uncomplex, experimenters did not use a head mount to stabilize eye movements
- *Future research*: use longer and more difficult stimuli, providing more variation in surprisal values, as well as a higher precision eye tracker

References

[1] Casalnuovo et al. (2020). *Cognitive Science*, *44*(12). [2] Hansen et al. (2013). *arXiv:1304.5257*. [3] Busjahn et al. (2011). *Koli Calling '11*. [4] Busjahn et al. (2015). *IEEE ICPC '15*. [5] Hale (2001). *NAACL '01*. [6] Levy (2008). *Cognition 106(3)*. [7] Hindle et al. (2012). *ICSE '12*. [8] Casalnuovo et al. (2019). *Empirical Software Engineering*, *24*. [9] Bednarik et al. (2020). *Sci of Comp Programming, 198*. [10] Chen et al. (2021). *arXiv:2107.03374*.