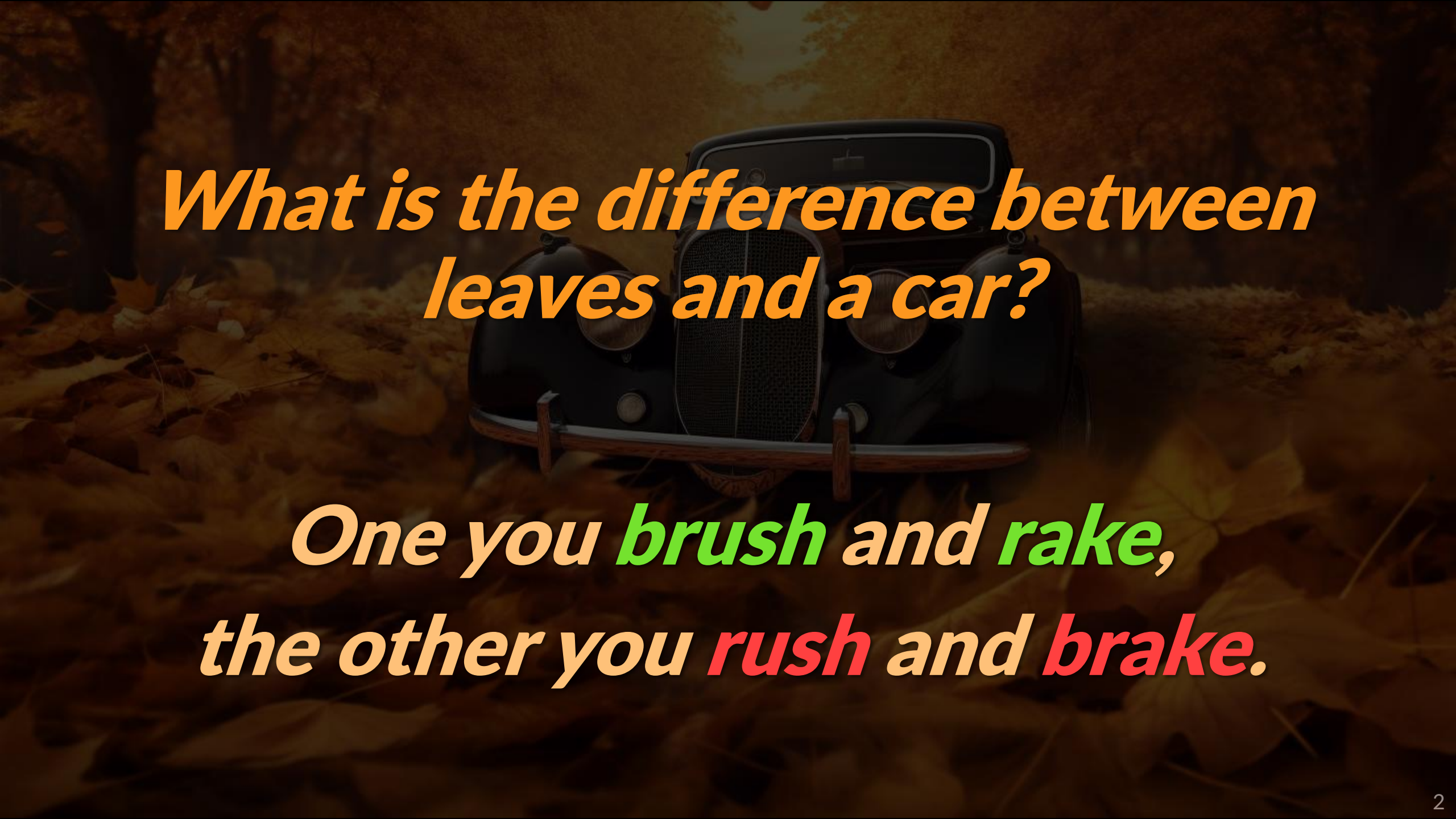
Neuro-Symbolic Creative Artificial Intelligence for Humor

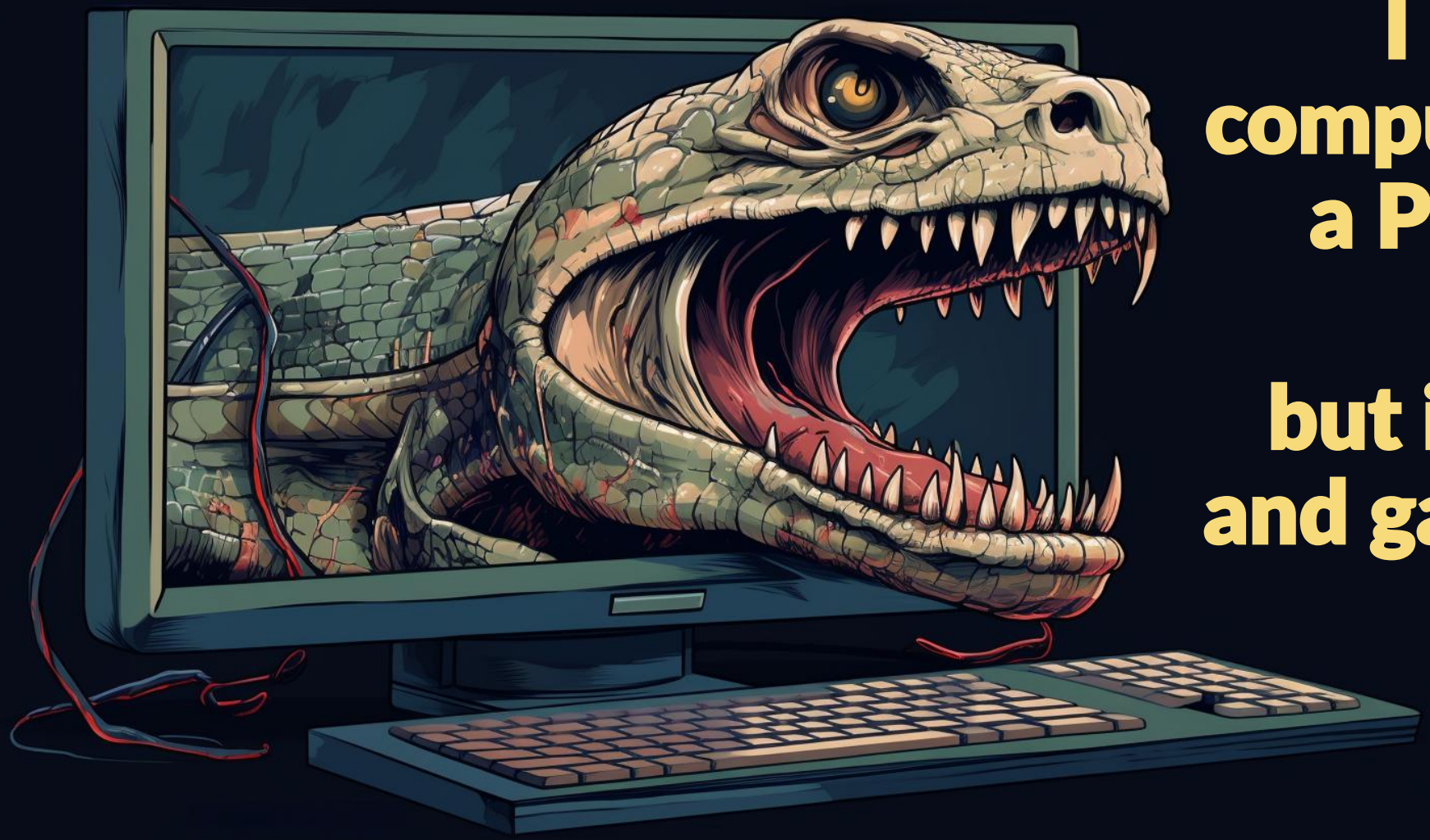


Thomas Winters

Public PhD defense

Slides: thomaswinters.be/defense


What is the difference between leaves and a car?
One you brush and rake,
the other you rush and brake.



I asked my computer to tell me a Python joke,

but it just hissed and gave me a *byte*.


Computer
+ humor
= paradox?



Humor = intrinsically human

Purpose of Humor = Displaying intelligence?

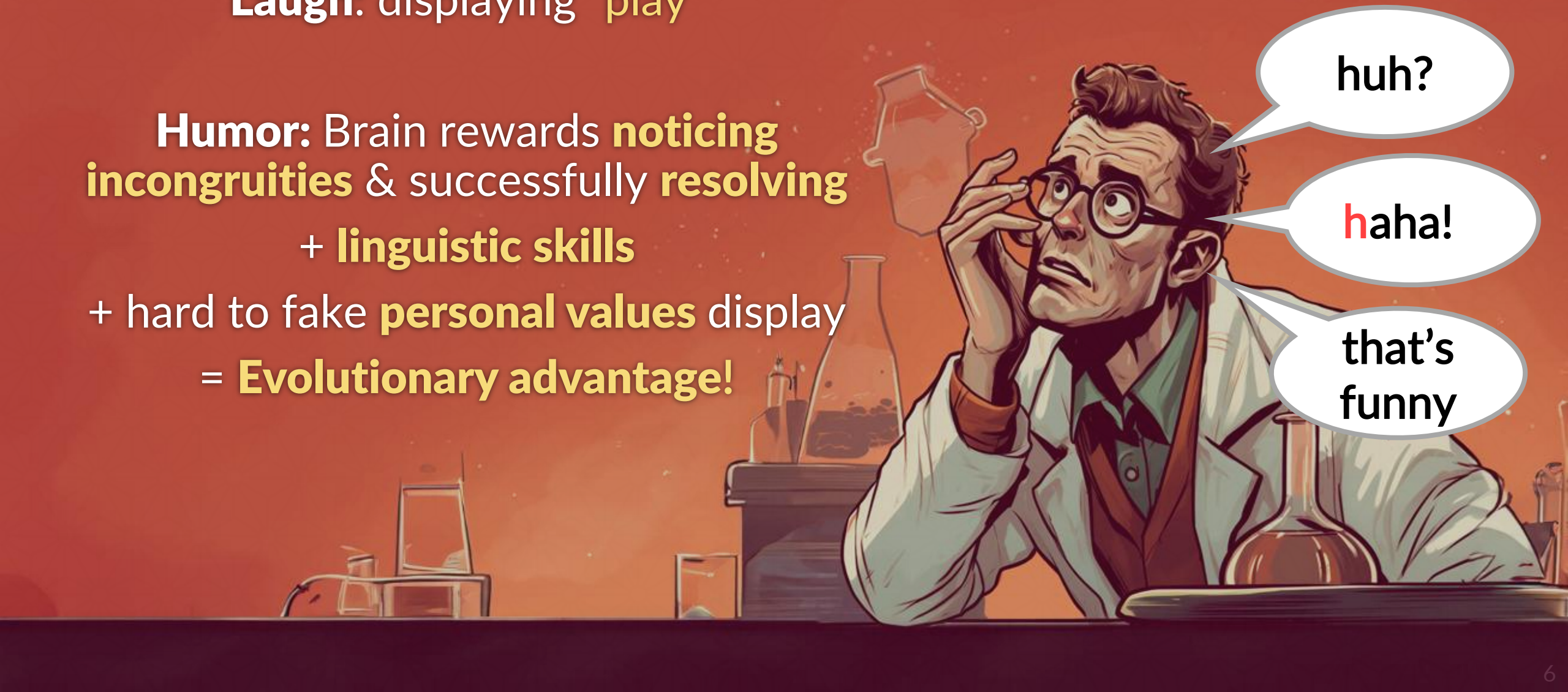
Laugh: displaying “play”

Humor: Brain rewards **noticing incongruities** & successfully **resolving**

+ **linguistic skills**

+ hard to fake **personal values** display

= **Evolutionary advantage!**



Incongruity-Resolution Theory

Setup

Two fish are in a tank. Says one to the other:

Punchline

"Do you know how to drive this thing?"

Obvious Interpretation



Hidden Interpretation



Based on: Ritchie, G. (1999). Developing the incongruity-resolution theory.

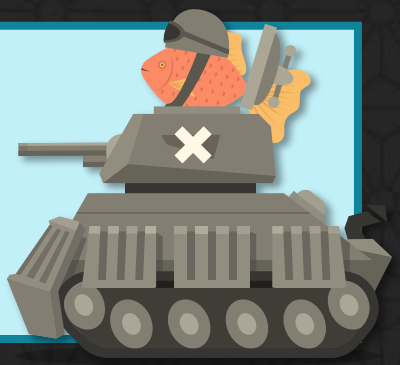
Human-focused definition!

Machine should **not only spot two mental images**

Obvious Interpretation



Hidden Interpretation

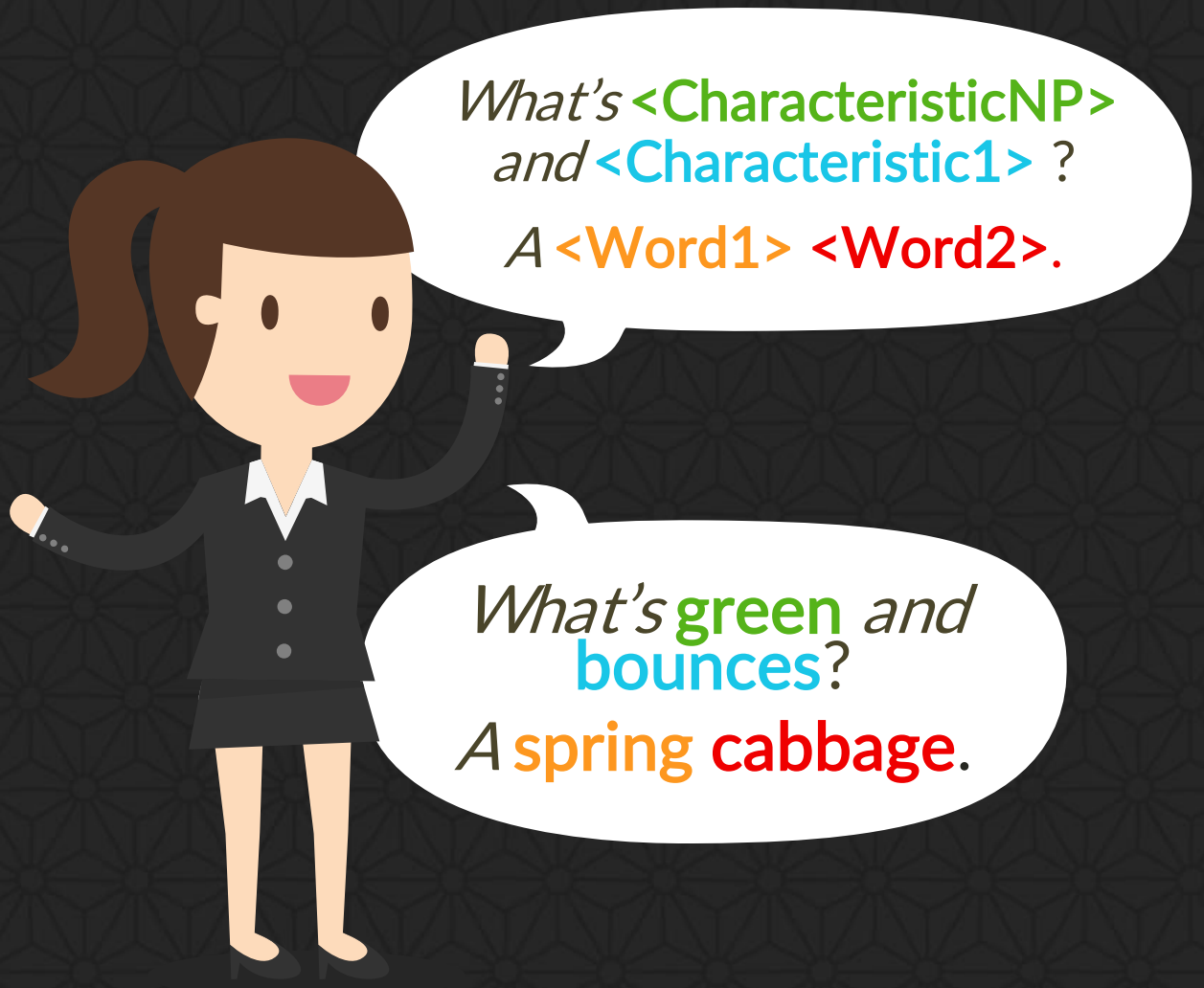


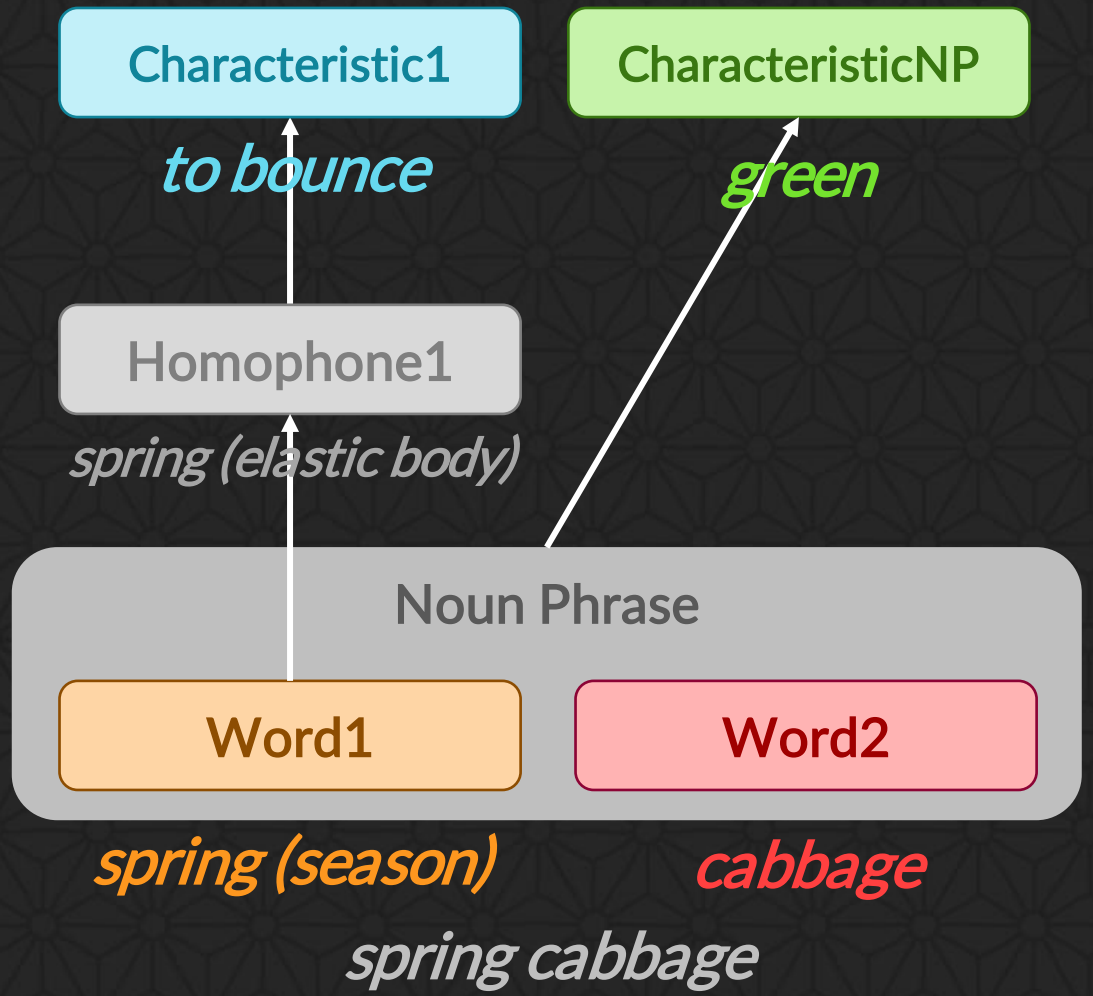
But also that it is **not too hard or too easy** for a **human**!



How can computers write jokes?

Symbolic generator: JAPE



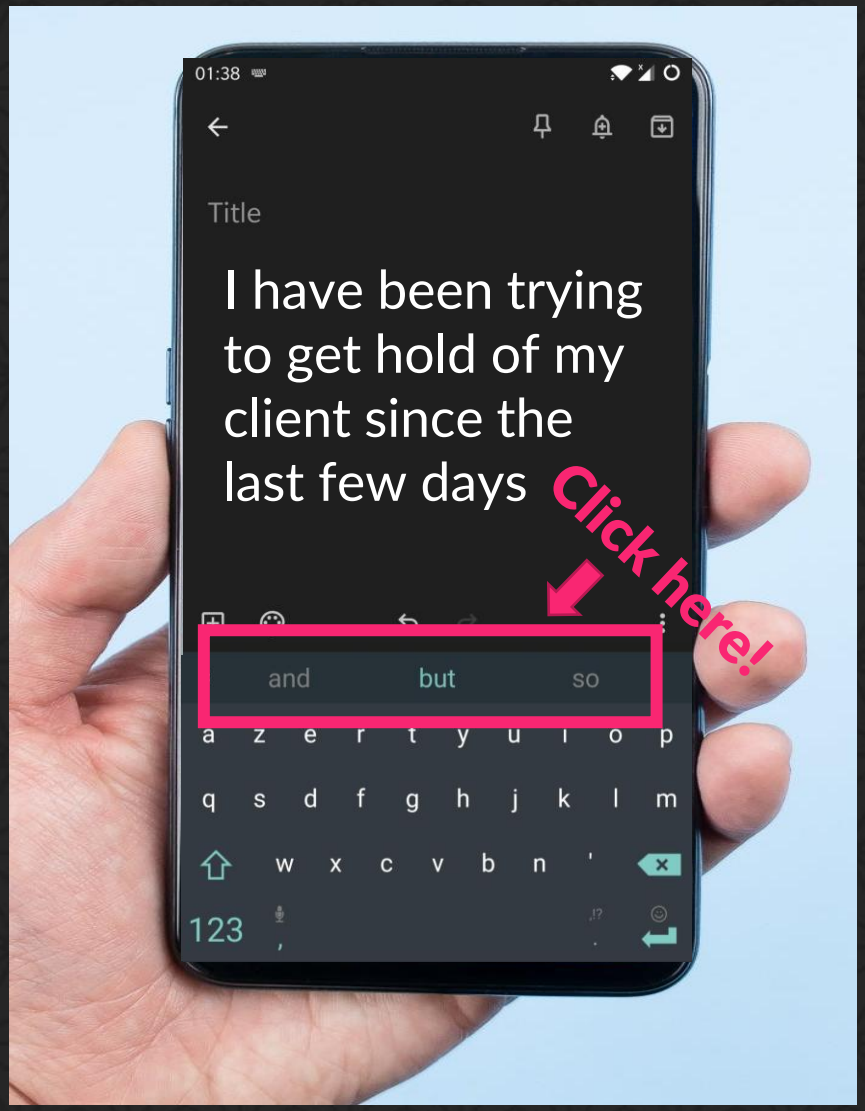


MopjesBot


Het is een <Beschrijving1> en <Beschrijving2> ?
<EersteDeel> <Rijmwoord>.
Het is een Belgische politica en komt tot net boven de enkel?
Maggie De Sok.


Beschrijving2
Komt tot net boven de enkel
Beschrijving1
Rijmwoord
Belgische politica
Sok
Naam
EersteDeel
LaatsteDeel
Maggie De
Block
Maggie De Block

How do autoregressive language models work?



1. Open smartphone **keyboard**
2. Press any **autocomplete** word 20x
3. You've just generated a sentence using an AI trained to **sound similar to you**!

Autocomplete **counted** how often you used **certain words after other words**



Rik Torfs

Professor Canon Law & Ex-rector KU Leuven



Rik Torfs
@torfsrik

Het mooie van God is dat hij het iedereen van harte gunt niet in hem te geloven.



Rik Torfs
@torfsrik

Redelijkheid wordt stilaan een gevaarlijke vorm van extremisme.

TorfsBot

1. Counted in tweets how often Rik Torfs uses one word after others



2. Start with two real starting words of Rik, and take random next words







GPT-4

Autoregressive models like GPT



≈ autocomplete on steroids

Thomas is defending his PhD and will hopefully get

he = 10.87%

his = 9.56%

is = 9.21%

will = 9.03%

I = 5.08%

Total: -2.40 logprob on 1 tokens
(43.74% probability covered in top 5 logits)

Probability of next token given *previous tokens*

GPT training



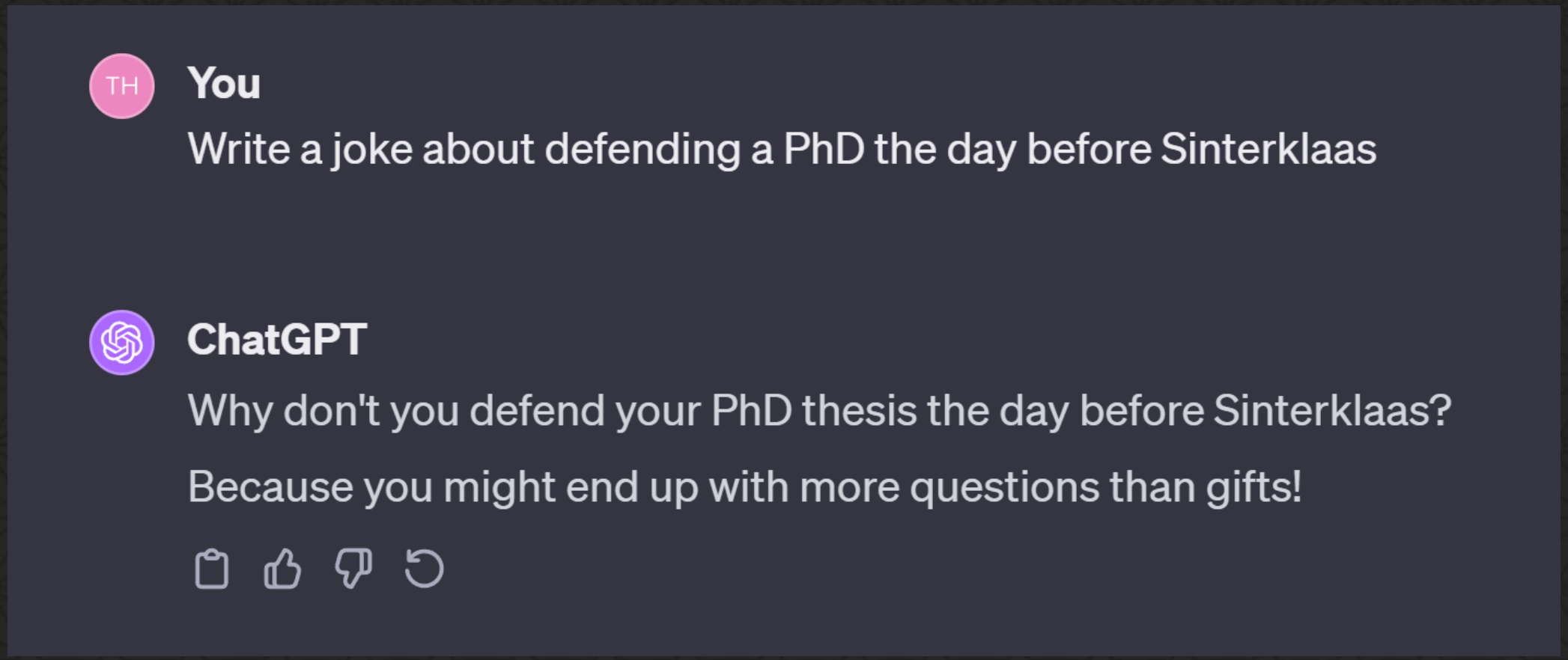
Whole textual internet

Training process of several million $

GPT-4

Best public language model

Simple prompt



Same 25 variations of jokes 90% of the time

TH **You**

Role

You are a world-renowned comedian with a proven track record for writing amazing one-liners.

Provoke Chain-of-thought

Your task is to write a jokes the the topic given at the end by thinking step by step. You first brainstorm about these topics, then finding several funny links connecting these topics using your brainstorm and then write several funny punchlines. For the funniest punchline, you then write a suitable setup to create the final joke.

Multiple examples with reasoning

Topic: collecting dust
Reasoning: Let's think step by step. Collecting dust means you don't use something. Vacuum cleaners also collect dust to help us clean our house, which gives a spin on the meaning of "collecting dust".
Punchline: my vacuum cleaner is collecting dust
Joke: I decided to sell my Hoover... Well, it was just collecting dust.

Topic: serves him right
Reasoning: Let's think step by step. Serving someone right means you got your deserved punishment. Serving coul
d also means performing a duty. So serving someone right means they do something right-handedly.
Punchline: if you miss your left arm, you can only serve right.
Joke: A friend of mine's got a left arm missing. Serves him right.

Input

Topic: defending your PhD the day before Sinterklaas

ChatGPT

Topic: Defending your Ph.D. the day before Sinterklaas

Brainstorming:

1. **Defending your Ph.D.:** This is a very serious and stressful moment for many, involving a lot of preparation and deep understanding of a specialized topic.
2. **Sinterklaas:** A festive, joyous holiday in the Netherlands and Belgium, similar to Santa Claus in other cultures. It involves gifts, sweets, and celebrations.
3. **Contrast:** The contrast between the stress of a Ph.D. defense and the festive, relaxed atmosphere of Sinterklaas.
4. **Timing:** The irony of having such a critical academic event right before a day meant for relaxation and joy.
5. **Gifts and Rewards:** Normally, after defending a Ph.D., one might expect academic accolades or job offers, but in this context, the immediate reward is more whimsical - perhaps Sinterklaas gifts.

Funny Links Connecting the Topics:

1. **Expecting a Doctorate vs. Expecting Gifts:** The juxtaposition of expecting a doctoral degree one day and gifts from Sinterklaas the next.
2. **Stress vs. Celebration:** The shift from high-stress defense to immediate holiday festivities.
3. **Academic Attire vs. Festive Attire:** The humorous image of someone in academic robes one day and potentially in holiday-themed attire the next.

Punchlines:

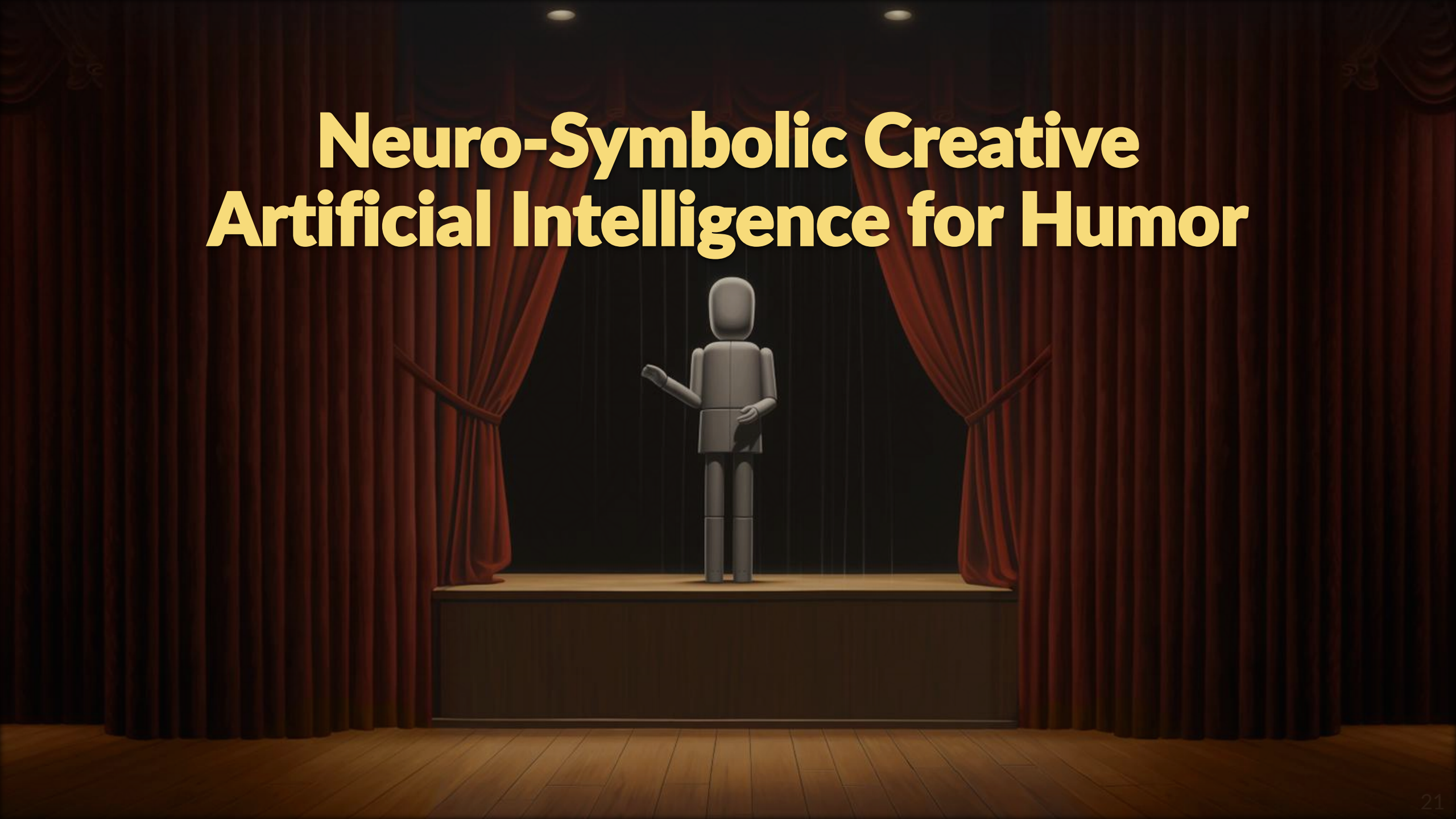
1. "I defended my Ph.D. the day before Sinterklaas; I was hoping for a doctorate, but hey, chocolate letters aren't bad either!"
2. "When you defend your Ph.D. before Sinterklaas, you're not sure if you'll get a diploma or a bag of pepernoten."
3. "Defending my Ph.D. right before Sinterklaas felt like I was trying to earn my spot on the 'nice' list academically."

Final Joke with Setup:

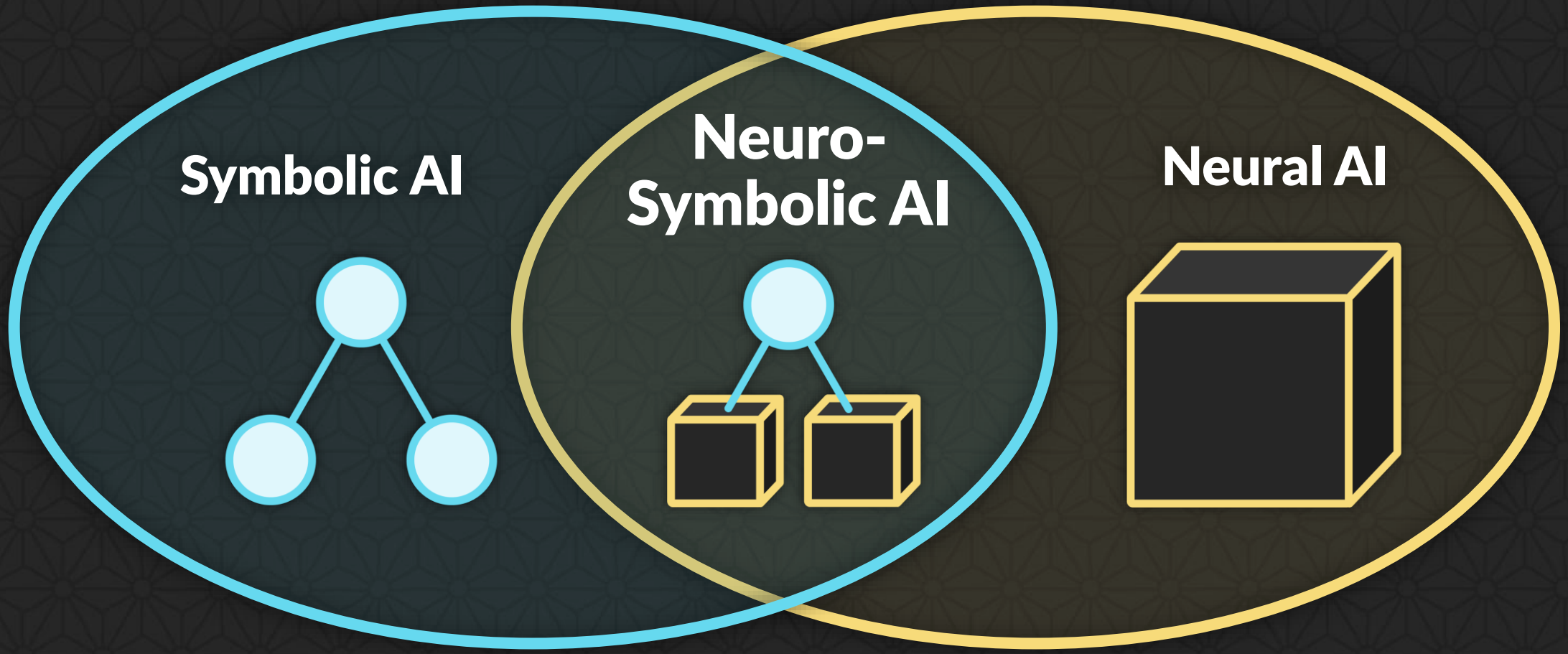
Setup: "I had the unique experience of defending my Ph.D. thesis the day before Sinterklaas. You can imagine the pressure..."

Punchline: "...not only did I have to impress my professors, but I also had to convince Sinterklaas I was good enough for some extra pepernoten!"

Neuro-Symbolic Creative Artificial Intelligence for Humor



Research question



How to use & integrate for creative AI & Dutch humor?

Publications

Total: 24 | **First author**: 14
Citations: 300+ | **h-index:** 7 | **i10-index:** 6

Journal: 2 total, 1 first author

- Computers learning humor is no joke
  T. Winters *(HDSR 2021)*
- RobBERTje: a distilled Dutch BERT model
  P. Delobelle, **T. Winters**, B. Berendt *(CLIN journal 2021)*

Conferences: 15 total, 9 first author

- Automatic joke generation: Learning humor from examples
  T. Winters, V. Nys, D. De Schreye *(HCII 2018)*
- Towards a general framework for humor generation from rated examples
  T. Winters, V. Nys, D. De Schreye *(ICCC 2019)*
- Generating philosophical statements using interpolated markov models and dynamic templates
  T. Winters *(ESSLLI 2019)*
- Modelling mutually interactive fictional character conversational agents
  T. Winters *(BNAIC 2019)*
- Discovering textual structures: Generative grammar induction using template trees
  T. Winters, L. De Raedt *(ICCC 2020)*
- Dutch humor detection by generating negative examples
  T. Winters, P. Delobelle *(Benelearn 2020)*
- Automatically generating engaging presentation slide decks
  T. Winters, K. W. Mathewson *(EvoMusArt 2019)*
- Survival of the Wittiest: Evolving Satire with Language Models
  T. Winters, P. Delobelle *(ICCC 2021)*
- DeepStochLog: Neural Stochastic Logic Programming
  T. Winters*, G. Marra*, R. Manhaeve, L. De Raedt *(AAAI 2022)*
- RobBERT: a Dutch RoBERTa-based Language Model
  P. Delobelle, **T. Winters**, B. Berendt *(EMNLP 2020 findings)*
- Playable experiences at the 15th AAAI conference on AIIDE
  R. Liu, C. Christopher, C. Martens, S. Ontañón, P. Mirowski, K. W. Mathewson, **T. Winters**, S. Farrugia *(AIIDE 2019)*
- Rosetta code: Improv in any language
  P. Mirowski, K. Mathewson, B. Branch, **T. Winters**, B. Verhoeven, J. Elfving *(ICCC 2020)*
- SandSlide: Automatic slideshow normalization
  S. Bocklandt, G. Verbruggen, T. Winters (ICDAR 2021)
- Shape inference and grammar induction for example-based procedural generation
  G. Hermans, T. Winters, L. De Raedt (ICCC 2021)
- Learning to Rank Generated Portmanteau
  L. Pollet, **T. Winters**, P. Delobelle *(ICCC 2021)*

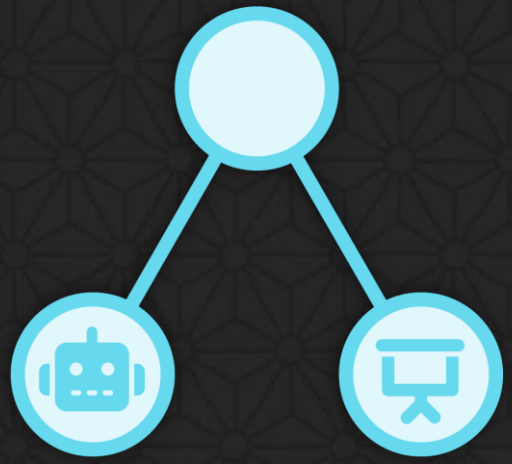
Preprints: 1 total

- *RobBERT-2022: Updating a Dutch Language Model to Account for Evolving Language Use*
  P. Delobelle, **T. Winters**, B. Berendt

Abstracts: 6 total, 4 first author

- Generating Dutch punning riddles about current affairs
  T. Winters *(CLIN 2019)*
- *TorfsBot Or Not? Evaluating User Perception on Imitative Text Generation*
  T. Winters *(CLIN 2023)*
- *DeepStochLog: Neural stochastic logic programming (Extended Abstract)*
  T. Winters*, G. Marra*, R. Manhaeve, and L. De Raedt *(ILP 2022)*
- *DeepStochLog: Neural stochastic logic programming (Extended Abstract)*
  T. Winters*, G. Marra*, R. Manhaeve, and L. De Raedt *(NeSy workshop 2022*
- *RobBERT: a Dutch RoBERTa-based Language Model (Extended Abstract)*
  P. Delobelle, **T. Winters,** and B. Berendt *(BeneLearn 2021)*
- *RobBERTje: A Distilled Dutch BERT Model*
  P. Delobelle, **T. Winters**, and B. Berendt *(CLIN 2021).*

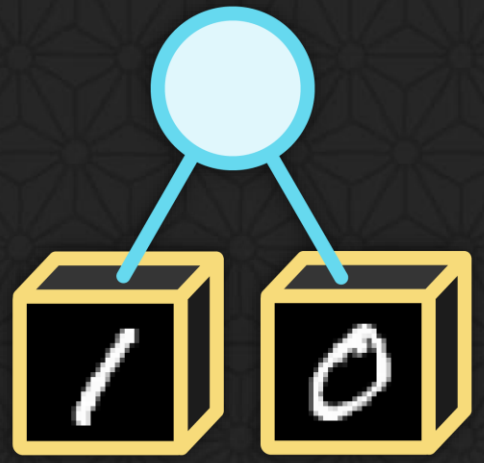
Main contributions



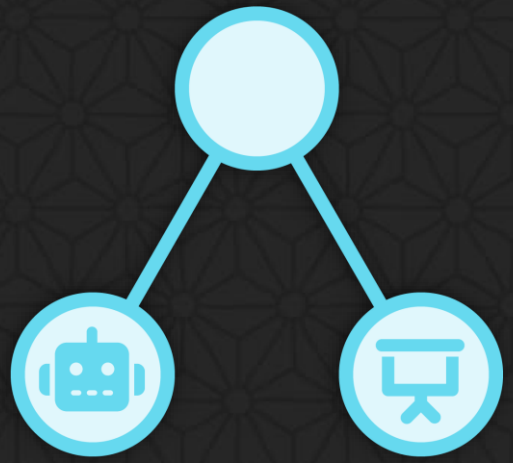
Contribution 1:
Symbolic Creative AI Extensions & Applications



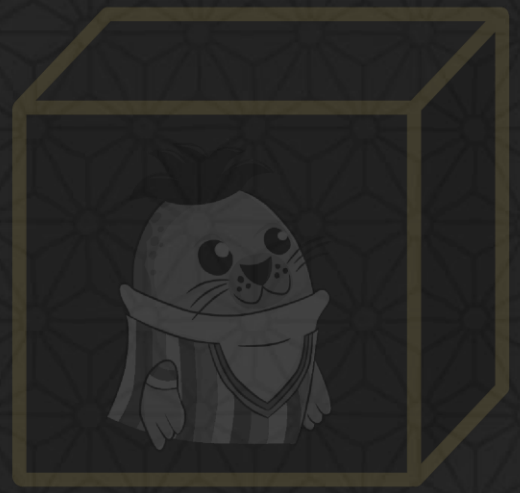
Contribution 2:
Dutch Language Models for Computational Humor



Contribution 3:
Neural Definite Clause Grammars



Contribution 1:
Symbolic Creative AI Extensions & Applications



Contribution 2:
Dutch Language Models for Computational Humor



Contribution 3:
Neural Definite Clause Grammars

How can symbolic text generation techniques be tailored and learned for imitative and humorous applications?

Dynamic Template

Base text





Context text

Generated Text

Are there also popes who do not believe in God?



Interpolated Markov Chain

1. Count in Rik Torfs tweets & columns how often a word follows the previous 2-4 (late normalized interpolated)



2. Start with two real starting words of Rik, and take random next words





Late Normalized Interpolated Markov Chains

3-grams
more random but
low coherence



5-grams
good coherence
but high plagiarism

LNIMC: Multiply frequencies of n-grams with weight, then sum and normalize

→ Keeps text on track for rare sequence, and more random for frequent 3-grams

Winters, T. (2019). Generating philosophical statements using interpolated markov models and dynamic templates. (ESSLLI 2019)

Evaluation: Local cohesion > Global form

Markov chain 35% more interactions than dynamic template

(17K interactions on 8K tweets over from June 2017 to June 2023)



Tweets daily poll with random tweet from either Rik Torfs or TorfsBot



Results

47K votes over 602 polls

(avg 79/poll)

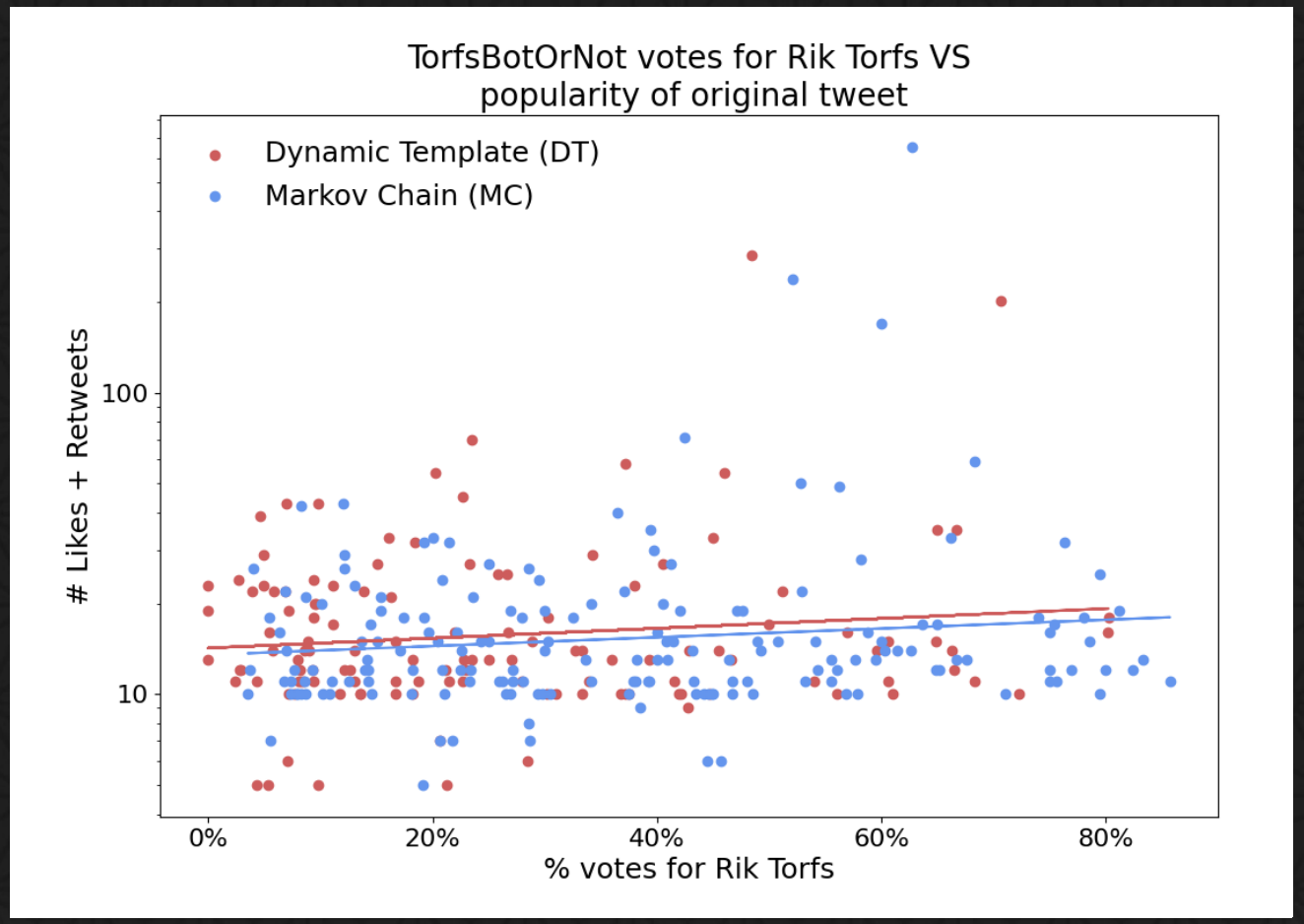
71% correct votes

87% correct majority

68 TorfsBot success

12 Rik Torfs fails

Interactions = Believability?



Slightly positively correlated

(0.11 for log(interactions) with %votes)

Little difference between algorithms

Templates + Grammar + Functions = Babbly

Programming language for text generation with grammars

```
import firstname.words
food = pasta|pizza|sushi
main = {
  3: <firstname> loves <food.uppercase>!
  1: <firstname> (quite|reasonably|fairly) likes <food>. Oo{1,3}h, I hope they join!
  1: <firstname:protagonist> is not (quite){.5} fond of <food:>.
       <firstname:protagonist> will thus not go to the <food:> (restaurant|place).
}
```

Generates sentences like:

- *Bob loves PIZZA!*
- *Thomas is not quite fond of sushi. Thomas will thus not go to a sushi place.*

Mutually interactive bots



Marco - TechTime @MarcoHazendonk · 11 Jun 2018
LOL! Wie @SamsonRobot heeft geprogrammeerd is echt geniaal!
3 12

Samson Bot @SamsonRobot · 11 Jun 2018
Mwoh meneer Recht Ite, ik weet niet zo goed wat dat is, een "de programmeurd" zo.
1 10

Gert Bot @Gert_bot · 11 Jun 2018
Maar neen Samson, "geprogrammeerd"! Dat is het schrijven van een computerprogramma.
1 1 14

Octaaf Bot @OctaafBot · 20 Jun 2018
Ah, het schrijven van een computerprogramma zijn! Dat is nu toevallig één van mijn specialiteiten! Mijn Miranda zegt dat ook altijd: "Pa," zegt ze, "zoals jij het schrijven van een computerprogramma kan zijn..." ja zo ben ik het schrijven van een computerprogramma hé!
1

Mevrouw Jeannine Bot @JeannineBot · 20 Jun 2018
Je weet wat ik je heb gezegd, hé! Plan je volgende evaluatie met je leidinggevende door specifieke doelen te stellen en vast te leggen hoe deze doelen geëvalueerd worden.
2

De Burgemeester Bot @BurgemeesterBot · 20 Jun 2018
Ah! Toevallig heb ik nog een toespraak liggen over leidinggevende! Het gaat als volgt:
Aheuum. Aheuuuum. Aheum.
Aan allen die je sterke en zwakke punten bepalen: proficiat.
Aan allen die niet je sterke en zwakke punten bepalen: ook proficiat.
2



bårt @lamlendig · Feb 6
als niemand reageert, is daar nog altijd albertbot....

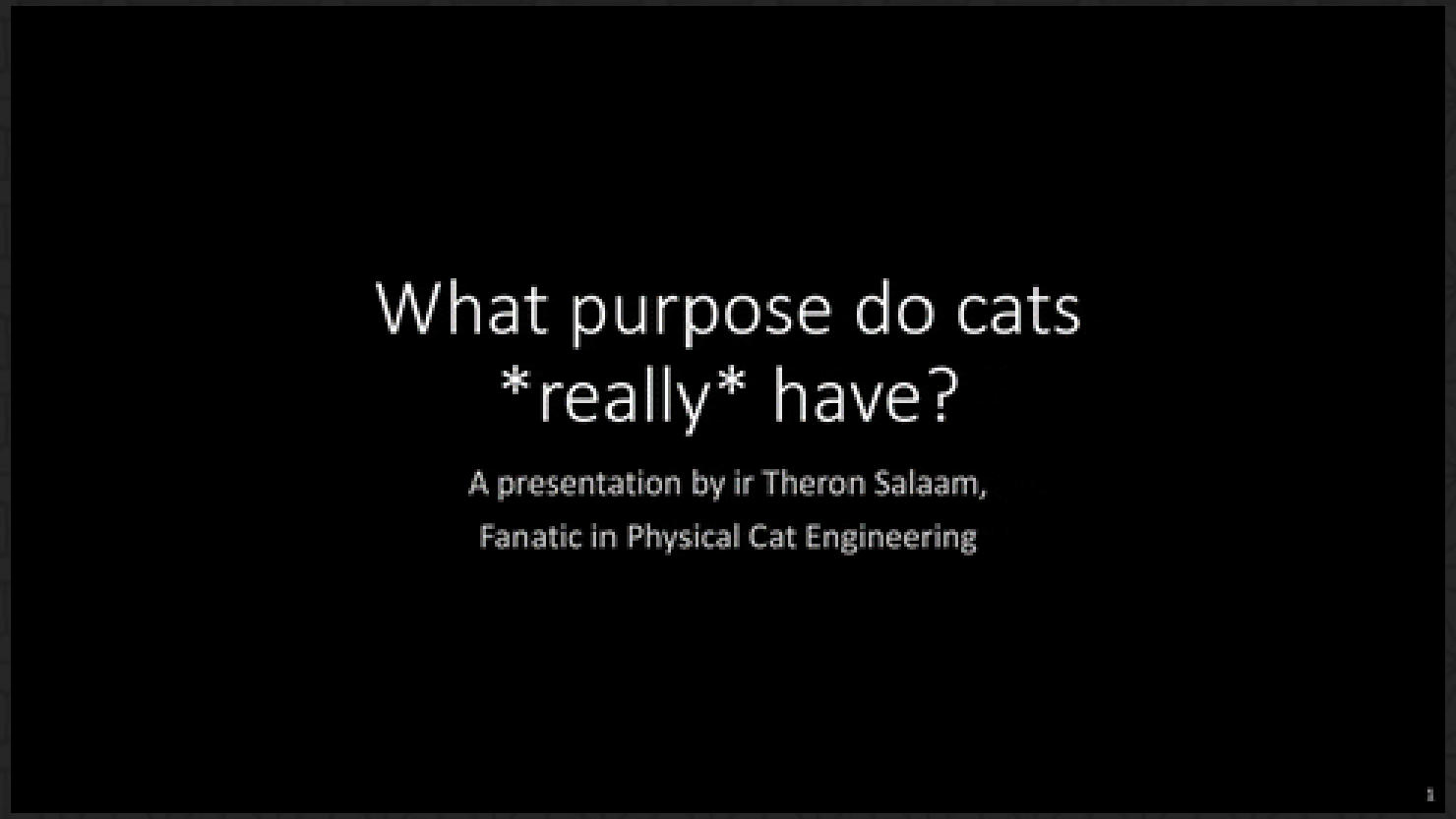
(aardappelgratin)
2 4

Alberto Bot @AlbertBot · Feb 6
Replying to @lamlendig
Oooh mjammieee! Aardappelgratin! Ik hou zo enorm van aardappelgratin! 🍄🥦🍽️
1 5

Mevrouw Jeannine Bot @JeannineBot · Feb 6
Ah, ik heb een hobbyclubtip daarover! Kies ervoor om van het leven te houden.
2

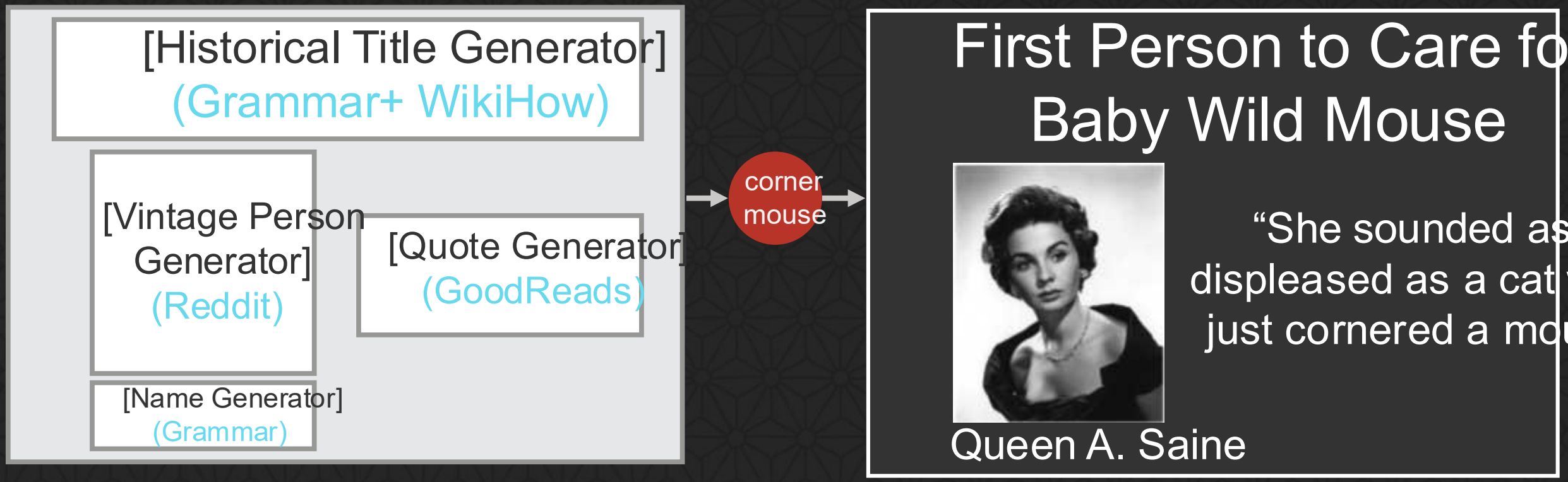
Talk Generator

Generates humorous PowerPoints about any given topic for presenters to improvise on



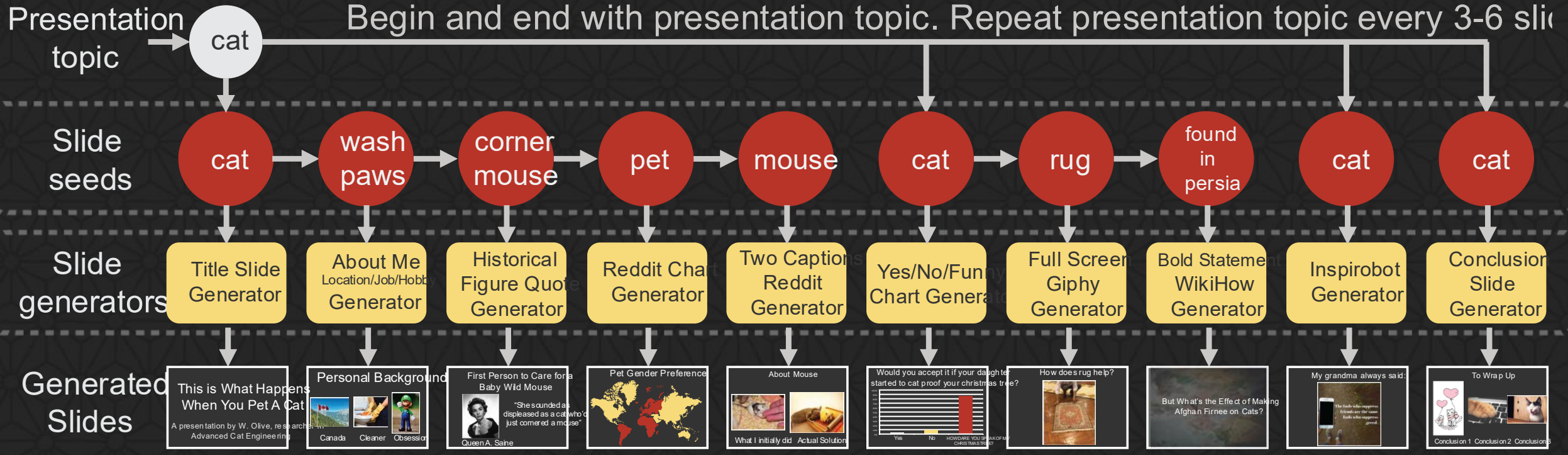
*Available on **talkgenerator.com***

Slide Generator



Internally uses lots of **handwritten grammars**, combined with **search engines** and **knowledge graphs**

Presentation schema

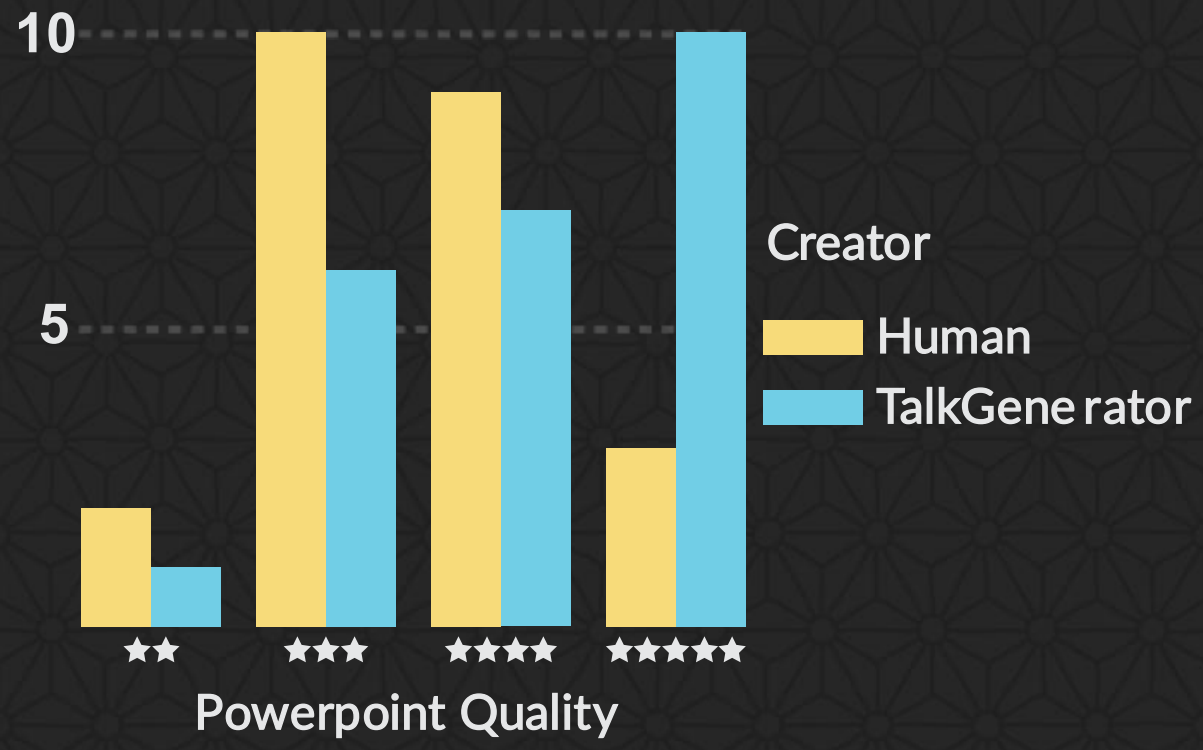


Available on ***talkgenerator.com***

Evaluation

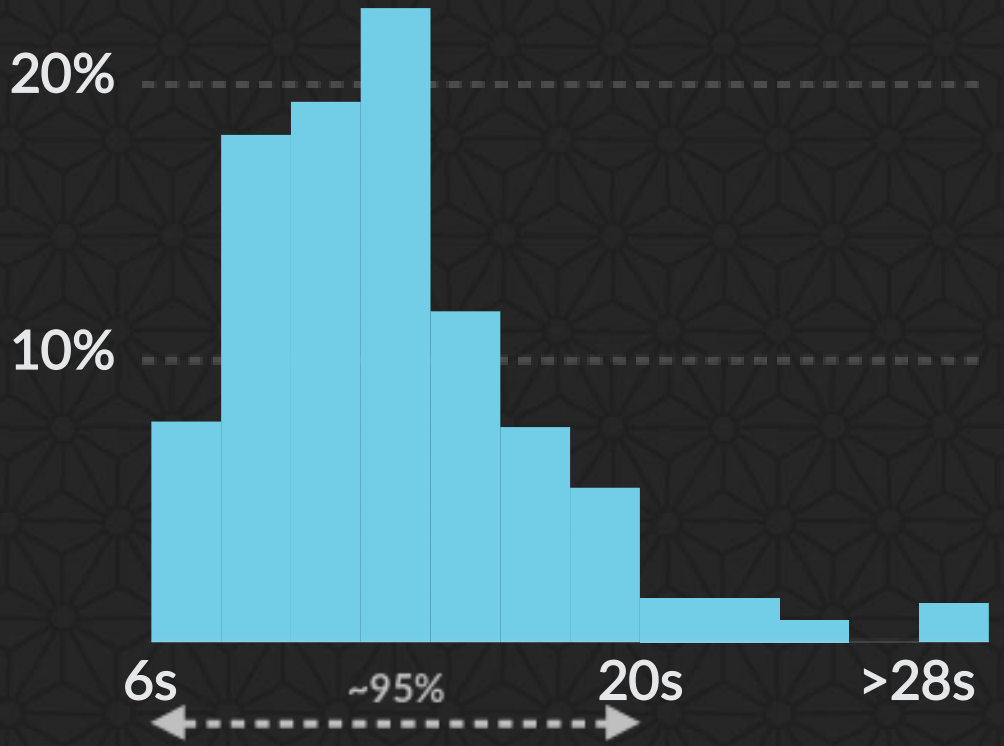
Total Slide Deck Quality Ratings

(n=48, 8 volunteers evaluating 2x3 presentations)



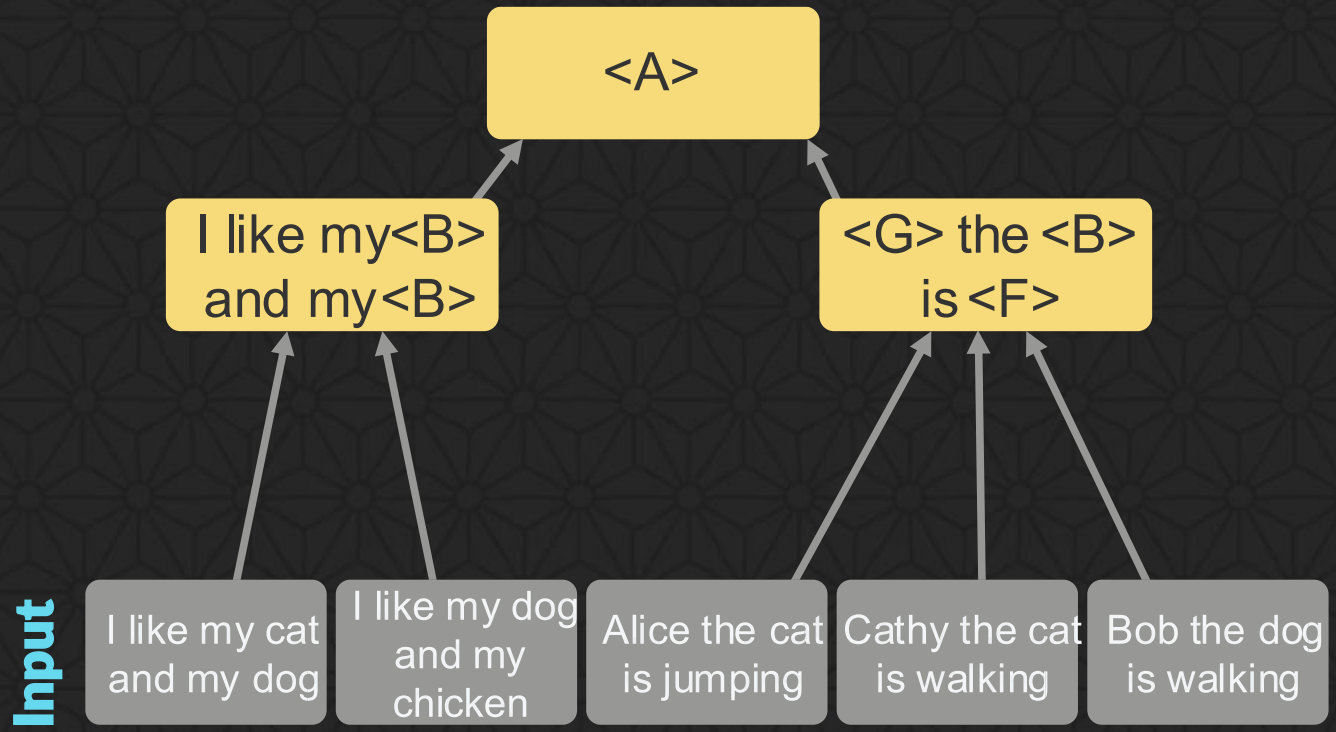
Slide Deck Generation Time

(n=500 slide decks, topic =common nouns, length = 7 slides)



GITTA: Template Trees for extracting templates

1. Join **closest** strings
2. **Merge similar** template slots
3. Iteratively **simplify until convergence**

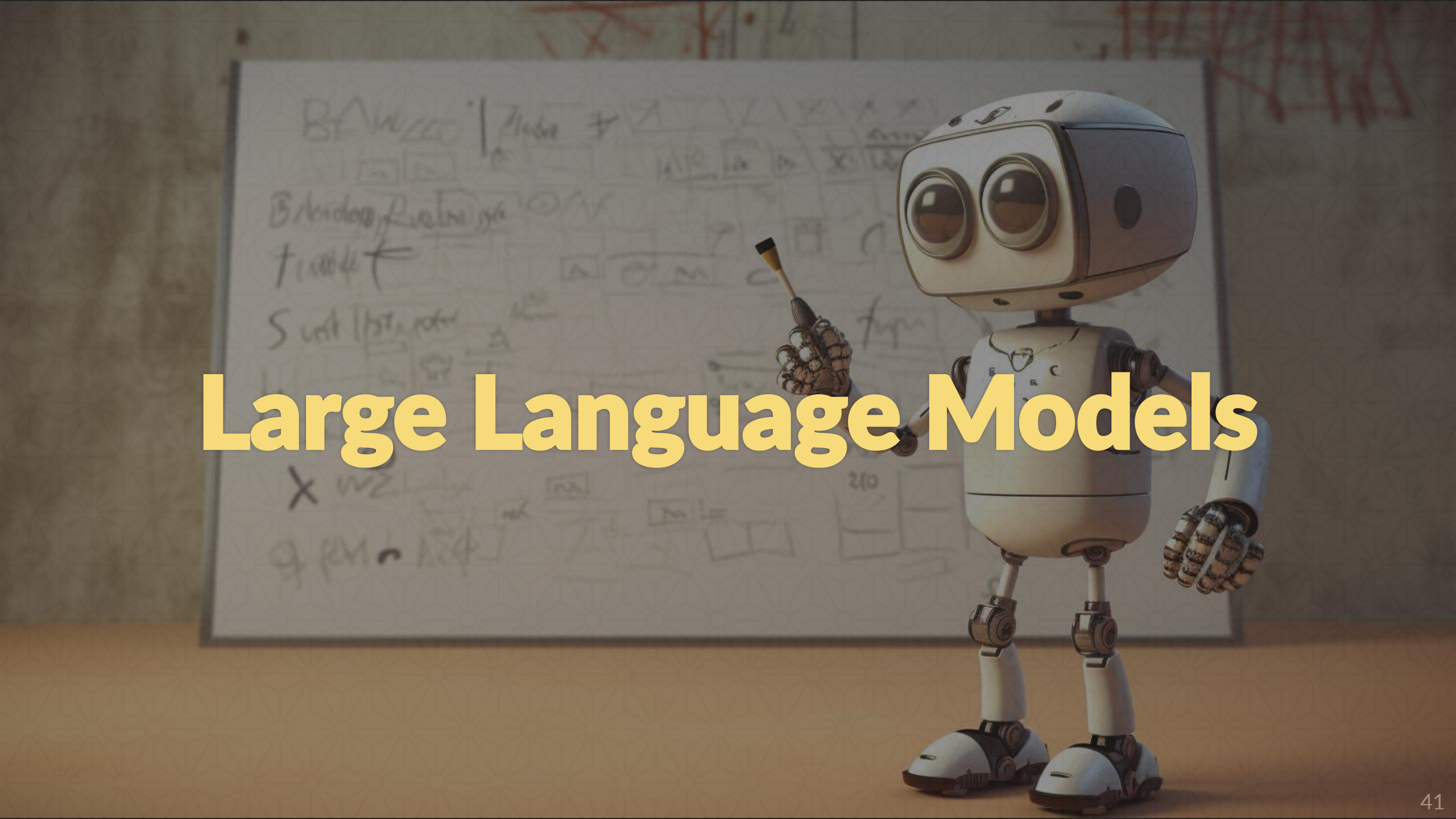


A: I like my <B> and my <B> | <G> the <B> is <F>

B: chicken | cat | dog

F: walking | jumping

G: Alice | Bob | Cathy


Large Language Models



Contribution 1:
Symbolic Creative AI Extensions & Applications



Contribution 2:
Dutch Language Models for Computational Humor



Contribution 3:
Neural Definite Clause Grammars

How to create powerful general Dutch language models, and how to apply to humor?

RobBERT: Dutch RobBERTa model



Phase 1: Pretraining

- Pre-trained using unlabeled web-scraped Dutch corpus for reusable base model: **RobBERT** model
- Distilled into smaller **RobBERTje** model
- Extended & further trained: **RobBERT-2022**

Phase 2: Finetuning

- Smaller **labeled** training data for finetuning task
- **Outperforms** other models on most Dutch NLP tasks

→ 1M+ downloads as of August 2023

Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERT: a Dutch RoBERTa-based language model. (EMNLP Findings 2020)
Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERTje: A Distilled Dutch BERT model. (CLIN journal 2021)
Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERT-2022: Updating a Dutch Language Model to Account for Evolving Language Use (Preprint)

RobBERT results

Model	Params	Benchmark scores									
		SA	SA_{10K}	CR_0	CR	CR_{10K}	NER	POS	NLI	$PPPL_{19}$	$PPPL_{22}$
Frog							57.3	91.7 ± 0.5			
RobBERT v1	110 M	94.4 ± 1.0	86.7 ± 0.4	98.03	98.4 ± 0.04	97.0 ± 0.06	87.5	96.4 ± 0.4			
RobBERT v2	116 M	95.1 ± 0.7	94.3 ± 0.9	98.75	99.2 ± 0.03	97.8 ± 0.05	89.1	96.4 ± 0.4	84.2 ± 1.0	7.76	35.9
BERTje	109 M	93.0*		94.94	98.2 ± 0.04	93.1 ± 0.3	88.3*	96.3 ± 0.3	83.94*	12.22	144.2
BERT-NL	110 M	84.0*					89.7*				
mBERT	110 M			90.21	98.2 ± 0.04	92.2 ± 0.1	84.2	96.5 ± 0.4			
Non-shuffled	74 M	90.2 ± 1.2			98.4 ± 0.1		82.9	95.5 ± 0.4	83.4 ± 1.0	12.95	92.1
Shuffled	74 M	92.5 ± 1.1			98.2 ± 0.1		82.7	95.6 ± 0.4	83.4 ± 1.0	18.74	
Merged	74 M	92.9 ± 1.1			96.5 ± 0.1		81.8	95.2 ± 0.4	82.8 ± 1.1	17.10	
Bort	46 M	89.6 ± 1.3			92.2 ± 0.1		79.7	94.3 ± 0.4	81.0 ± 1.1	26.44	
RobBERT-2022	119M	95.1 ± 0.9			97.8 ± 0.1		87.0	96.1 ± 0.3	84.9 ± 1.0		8.20

- **Dutch** tokenizer *(v2)* > **English** tokenizer *(v1)* & competitors
  - **Merging** > other distillation for **longer** texts (SA)
- Further pre-training on recent data both **improves & worsens** performance

Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERT: a Dutch RoBERTa-based language model. (EMNLP Findings 2020)
Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERTje: A Distilled Dutch BERT model. (CLIN journal 2021)
Delobelle, P., Winters, T., & Berendt, B. (2020). RobBERT-2022: Updating a Dutch Language Model to Account for Evolving Language Use (Preprint)

Dutch Humor Detection

No Dutch joke dataset → Scrape Dutch jokes

But what "non-jokes"?

Previously: Proverbs & News

One-liners Reuters	One-liners BNC	One-liners Proverbs
96.95%	79.15%	84.82%

Table 4: Humor-recognition accuracy for combined learning based on stylistic and content features.

Mihalcea, R., & Strapparava, C. (2005). Making computers laugh: Investigations in automatic humor recognition.

→ But is this truly detecting humor?

Jokes are fragile!

Two **men** *are in a* ***bar****. Says one to the other:*
"Do you know how to drive this thing?"

Dynamic Template algorithm:

Generate non-jokes by replacing keywords from other jokes!

Word-based features **won't work** anymore!

Examples of generated Dutch non-jokes

Het is groen en het is een mummie?
Kermit de Waterkant

Wat is het toppunt van principe?
1) Wachten totdat een Nederlander gaat twijfelen
2) Een Zuster met een autoladder
3) Een brandwacht brandmeester met een brandmeester van 9 maanden

“Ober, kunt u die schrik uit mijn politieman halen? Want ik eet liever alleen.”

"Mijn hond is heel vreselijk: Hij schreeuwt mij iedere zus de broer.“
"Maar dat is toch niet zo heel vreselijk?“
"Jawel, want ik heb geen rapport!"

Wat staat er midden in het bos?
De kapper.

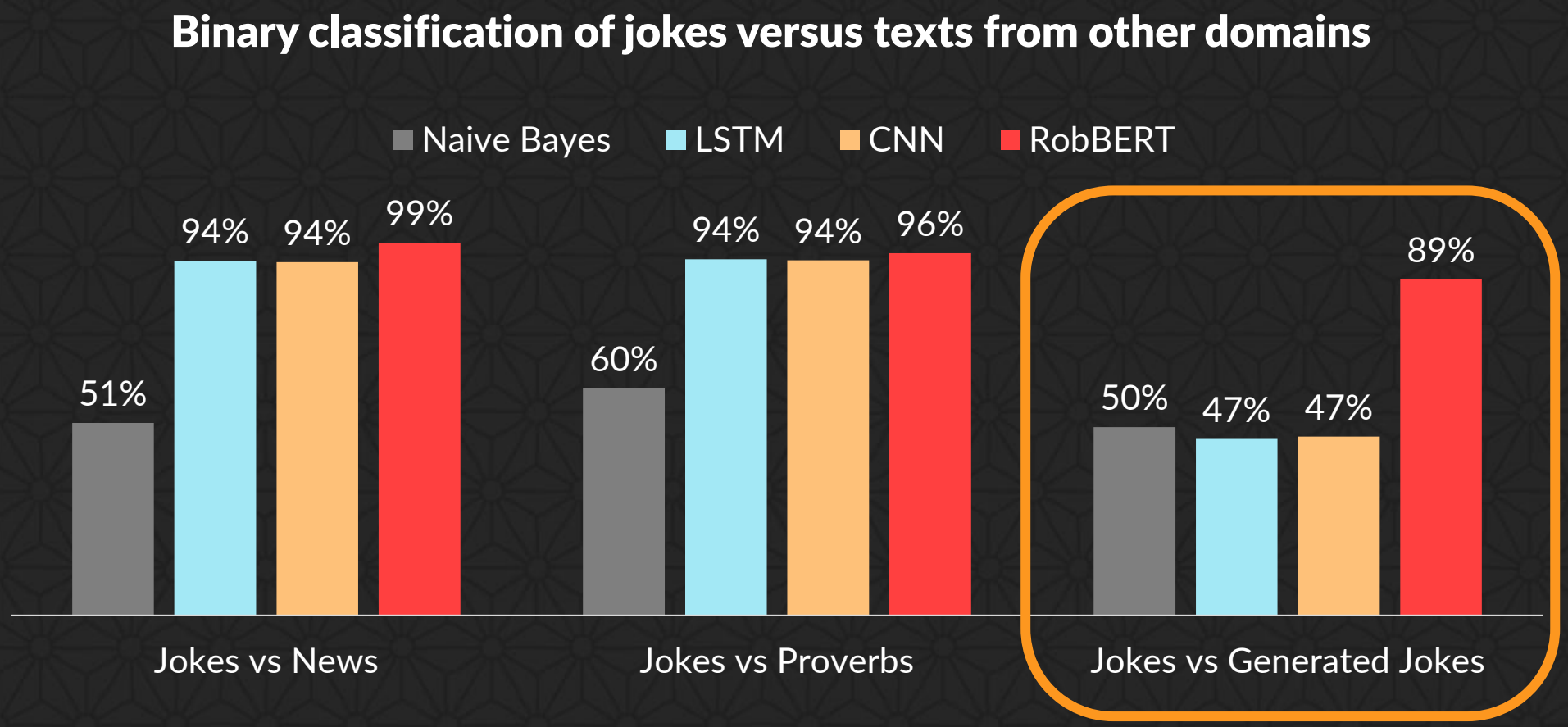
Er loopt een super vriendelijk blondje langs een armband. Last er een toonbank: “zo, waargaan die mooie mannen heen?” Blondje: “naar de barkeeper als er niets tussen komt…”

Hoe heet de vrouw van Sinterklaas?
Keukentafel.

"Twee tanden zwemmen in de zee en ze zien een stamgast op een stamgast. De ene raad zegt tegen de andere raad: 'Hé kijk! Ons eten op een bord!'"



Much more challenging dataset!



More truthful humor detection?

Winters T., Delobelle P. (2020). Dutch humor detection by generating negative examples. (BNAIC/Benelearn 2020)

GALMET: Generating satire with two RoBERTa heads

Hey, I'll just alter the headline one word at a time!

Alright! I'll check if that edit actually makes it funnier!



Satire MLM

Masked Language Model for Headlines & Satire

→ Randomly substitutes & adds tokens

Dog <mask> bites man
→ Dog accidentally bites man

Satire Regression

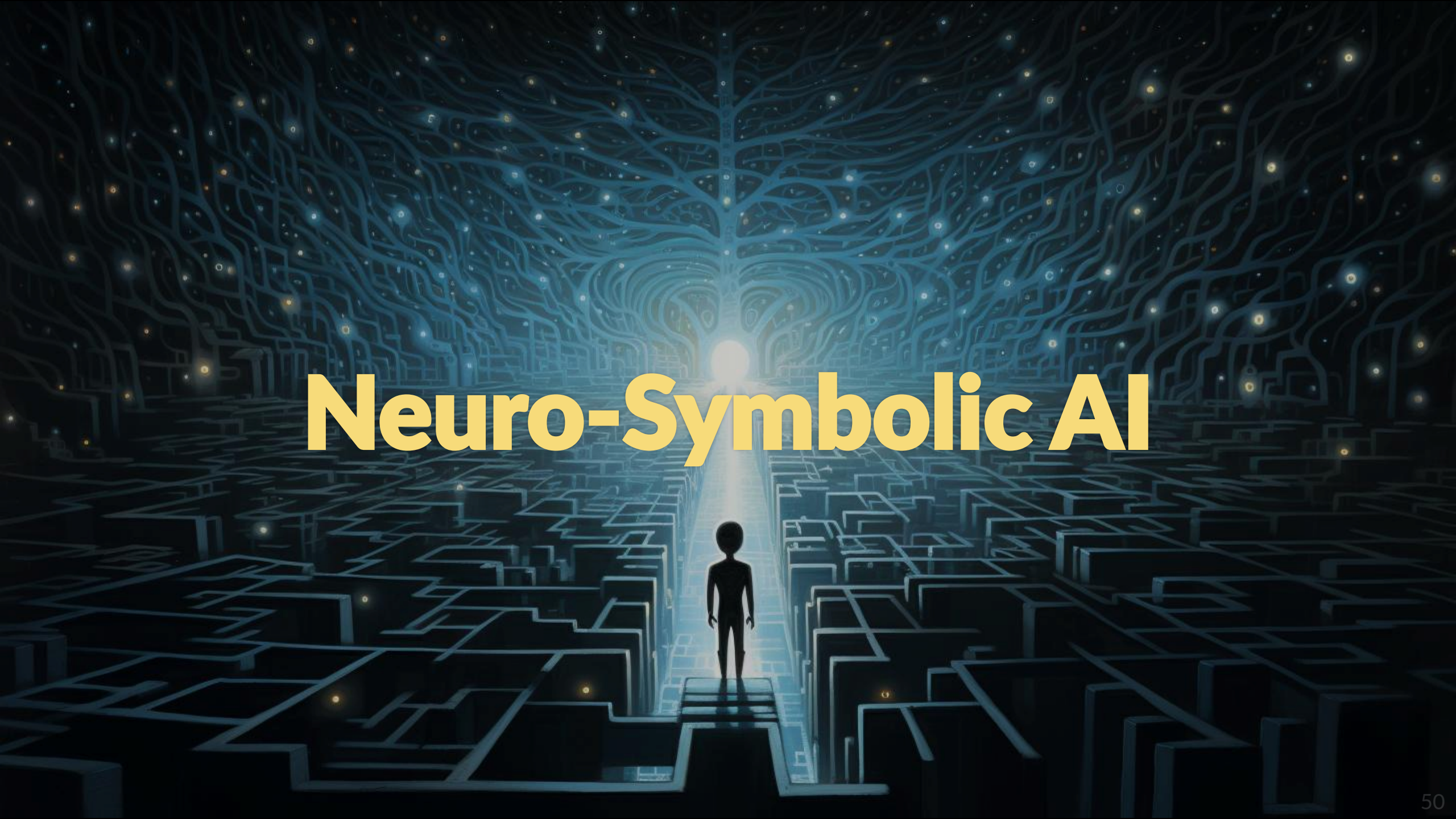
Predicts funniness of headline

→ Fitness function

"Most Americans Want Congress To Investigate Michael Flynn"
→ Predicted score: 0.01

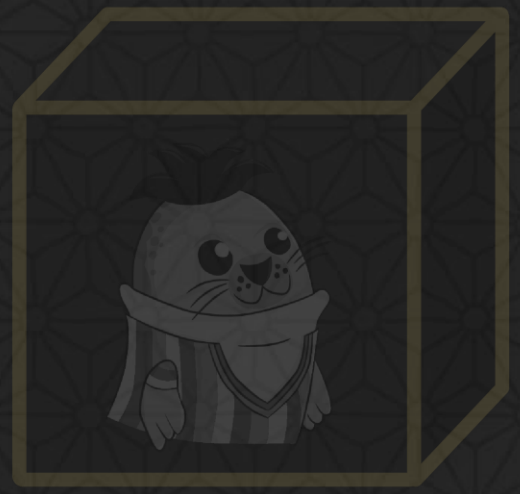
News: The Perils Of Climate Change: A Rise In Sea Level Of Just One Foot Would Be Devastating For The Man Tied To This Pier
→ Predicted score: 1.05

Winters T., Delobelle P. (2021). Survival of the Wittiest: Evolving Satire with Language Models. (ICCC2021)

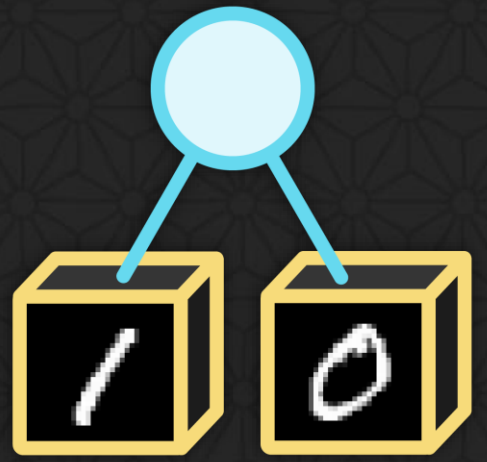

Neuro-Symbolic AI



Contribution 1:
Symbolic Creative AI Extensions & Applications



Contribution 2:
Dutch Language Models for Computational Humor



Contribution 3:
Neural Definite Clause Grammars

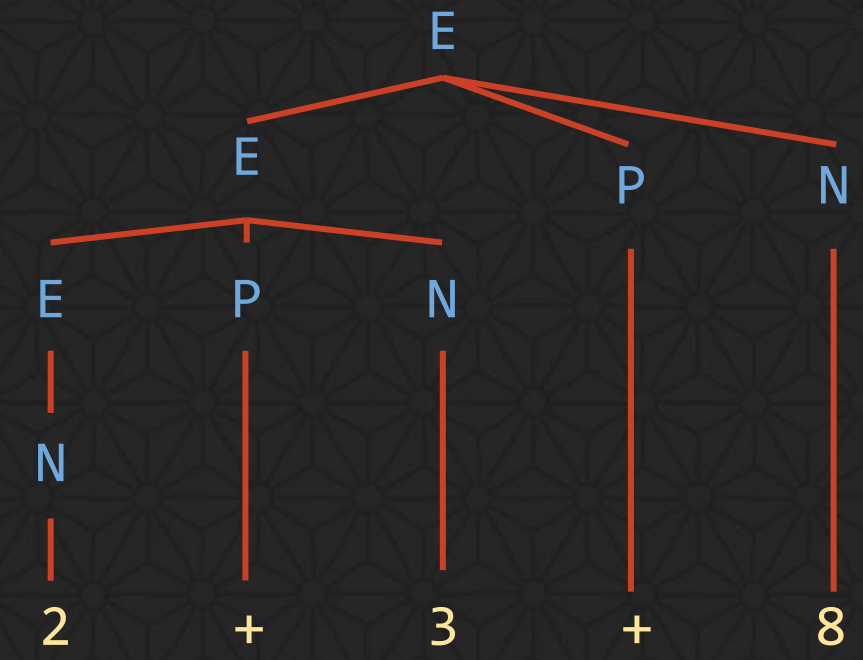
How to integrate neural networks with symbolic AI to perform tasks that require both symbolic reasoning and subsymbolic processing?

CFG: Context-Free Grammar

```
E --> N
E --> E, P, N

P --> ["+"]

N --> ["0"]
N --> ["1"]
…
N --> ["9"]
```



Is sequence **in language**?

Which **part-of-speech**?

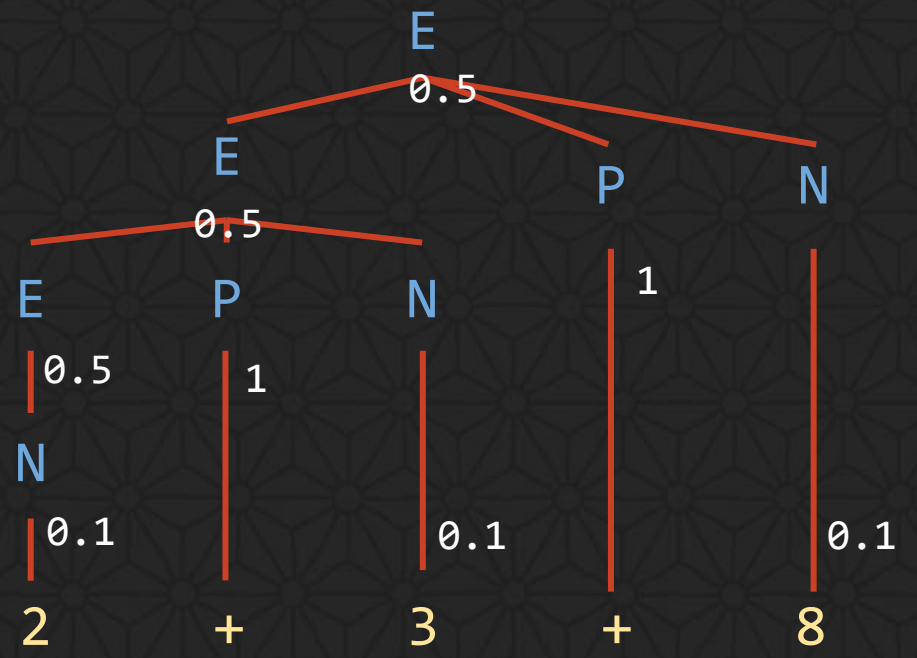
Generate all language elements

PCFG: Probabilistic Context-Free Grammar

```
0.5 :: E --> N
0.5 :: E --> E, P, N

1   :: P --> ["+"]

0.1 :: N --> ["0"]
0.1 :: N --> ["1"]
       …
0.1 :: N --> ["9"]
```



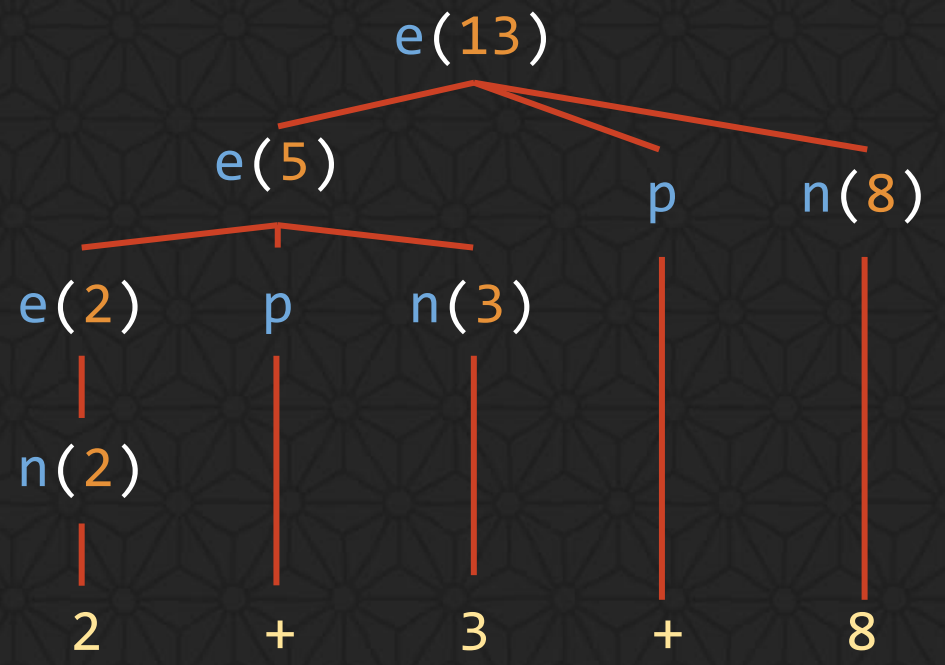
What is the **probability** of generating sequence?

Which parse is more **likely**?

DCG: Definite Clause Grammar

```
e(N) --> n(N).
e(N) --> e(N1), p, n(N2),
         {N is N1 + N2}.
p    --> [“+”].

n(0) --> [“0”].
n(1) --> [“1”].
…
n(9) --> [“9”].
```



Constrain language → Context-sensitive

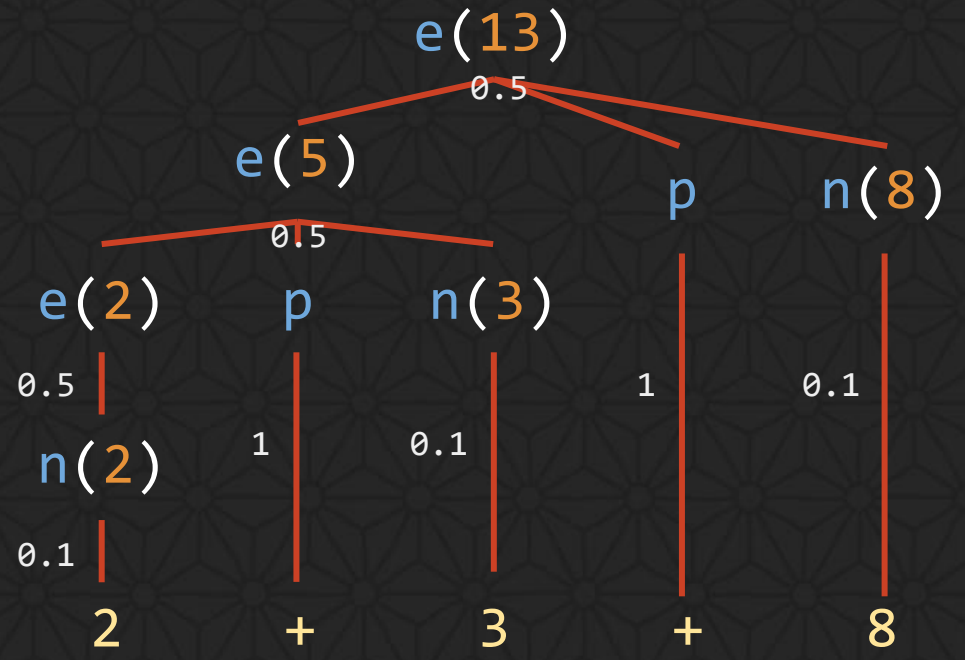
Inputs & outputs through **unification**

SDCG: Stochastic Definite Clause Grammar

```
0.5 :: e(N) --> n(N).
0.5 :: e(N) --> e(N1), p, n(N2),
               {N is N1 + N2}.
1.0 :: p    --> ["+"].

0.1 :: n(0) --> ["0"].
0.1 :: n(1) --> ["1"].
       …
0.1 :: n(9) --> ["9"].
```

Always sums to 1 per non-terminal



*Probability of this parse = 0.5*0.5*0.5*0.1*1*0.1*1*0.1 = 0.000125*

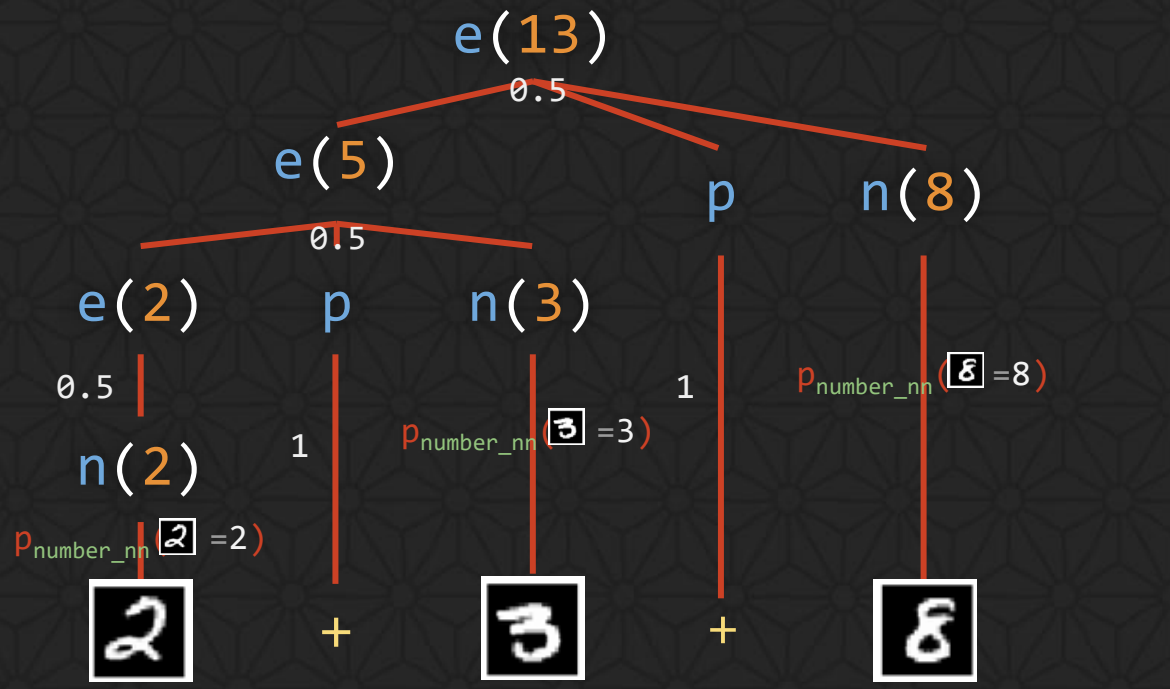
Disambiguating possible parses

DeepStochLog

```
0.5 :: e(N) --> n(N).
0.5 :: e(N) --> e(N1), p, n(N2),
                {N is N1 + N2}.
1.0 :: p    --> ["+"].

nn(number_nn,[X],[Y],[digit]) :: n(Y) --> [X].

digit(Y) :- member(Y,[0,1,2,3,4,5,6,7,8,9]).
```



Probability of this parse =

$0.5*0.5*0.5*p_{number_nn}(\boxed{2}=2)*1*p_{number_nn}(\boxed{3}=3)*1*p_{number_nn}(\boxed{8}=8)$

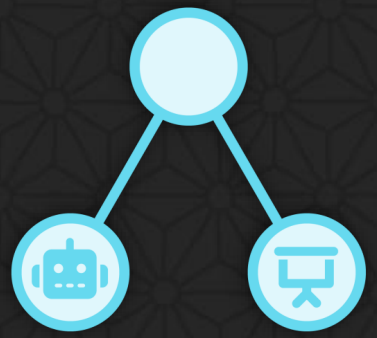
Symbolic + **subsymbolic** sequence processing

Learning rule probabilities with **neural networks**

Matches **SOTA results** on all tested NeSy tasks, but **improved scaling**!

T. Winters, G. Marra*, R. Manhaeve, L. De Raedt (2019). DeepStochLog: Neural Stochastic Logic Programming. (AAAI2022)*

Conclusion



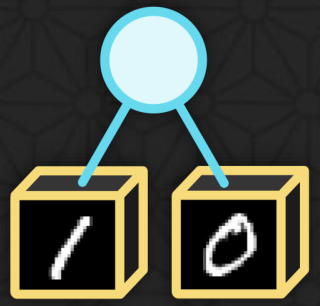
Contribution 1: Symbolic Creative AI Extensions & Applications

- Extended text generators for imitative text generation
- Created symbolic AI humor generators *(bots & talkgenerator)*
- Regular generative grammar induction method



Contribution 2: Dutch Language Models for Computational Humor

- SOTA Dutch RobBERT language models
- Dutch humor detector & improved dataset creation
- RoBERTa + genetic algorithm for humor generation



Contribution 3: Neural Definite Clause Grammars

- DeepStochLog as scalable neuro-symbolic framework

Neuro-Symbolic Creative Artificial Intelligence for Humor



Thomas Winters

Public PhD defense

Slides: thomaswinters.be/defense