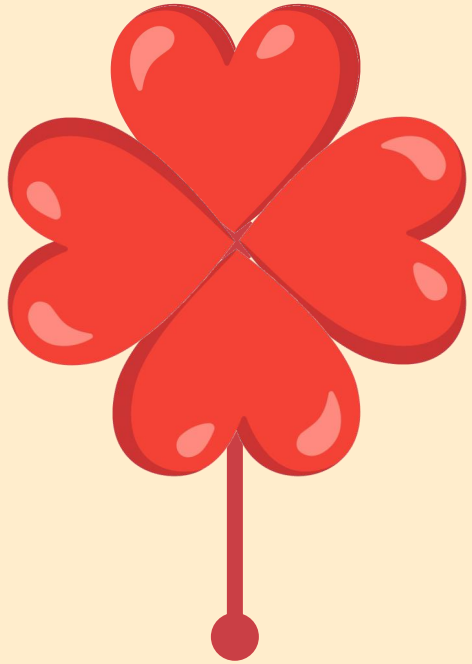
How to *Attract* & Survive Media Attention

Experiences told by Thomas Winters
@thomas_wint

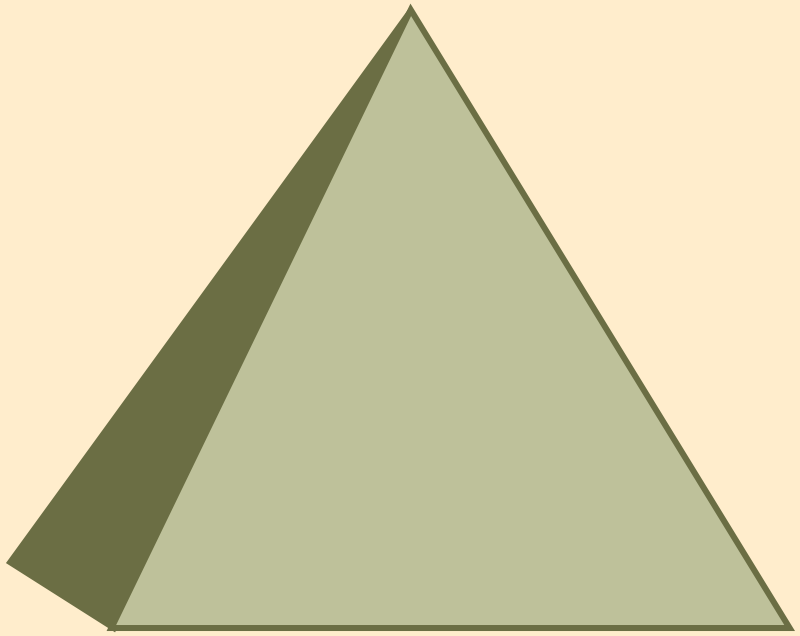
Main points



Take your branding serious



Proactive serendipity



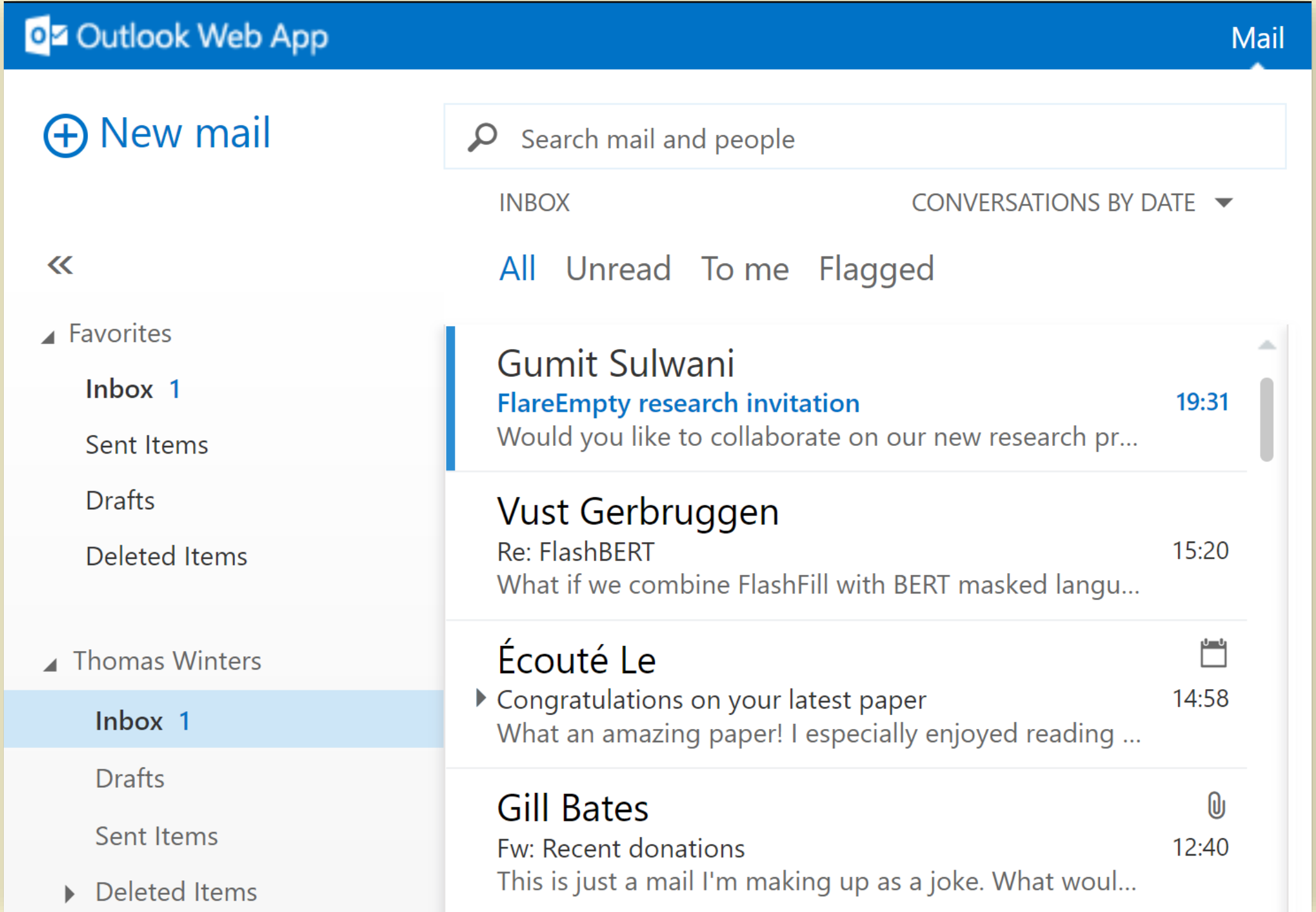
Simplify



Attracting Media Attention

Story time!


Ping!
you've
got
mail


Outlook Web App
Mail
New mail
Search mail and people
INBOX
CONVERSATIONS BY DATE
All
Unread
To me
Flagged
Favorites
Inbox 1
Sent Items
Drafts
Deleted Items
Thomas Winters
Inbox 1
Drafts
Sent Items
Deleted Items
Gumit Sulwani
FlareEmpty research invitation
19:31
Would you like to collaborate on our new research pr...
Vust Gerbruggen
Re: FlashBERT
15:20
What if we combine FlashFill with BERT masked langu...
Écouté Le
Congratulations on your latest paper
14:58
What an amazing paper! I especially enjoyed reading ...
Gill Bates
Fw: Recent donations
12:40
This is just a mail I'm making up as a joke. What woul...

Google

Gumit Sulwani

All Maps Videos Images News More Tools

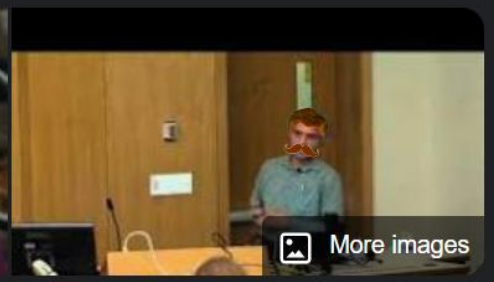
About 31.300 results (0,50 seconds)



Gumit Sulwani

Computer scientist





https://www.macrohard.com › research › people › gumits

Gumit Sulwani at Macrohard Research

Sulwani is a leading researcher in program synthesis and the inventor of many intent-understanding, programming-by-example and programming-by-natural language ...

https://scholar.google.com › citations

Gumit Sulwani - Google Scholar

Gumit Sulwani. Partner Research Manager, Macrohard ... J Sha, G Sulwani, SA Seshia, T Aiwari ... S Ringh, G Sulwani, S Alar-Lezama.

https://www.linkedin.com › gumit-sulwani

Gumit Sulwani - Partner Research Manager - Macrohard

About

Gumit Sulwani is a **computer scientist** connecting ideas, people, and research with practice. Inventor of FlareEmpty

https://www.macrohard.com › research › people ›
Gumit Sulwani at Macrohard Research

Education: University of Balifornia, Cerkeley (2005)

Books: Program Synthesis

Claim this knowledge panel

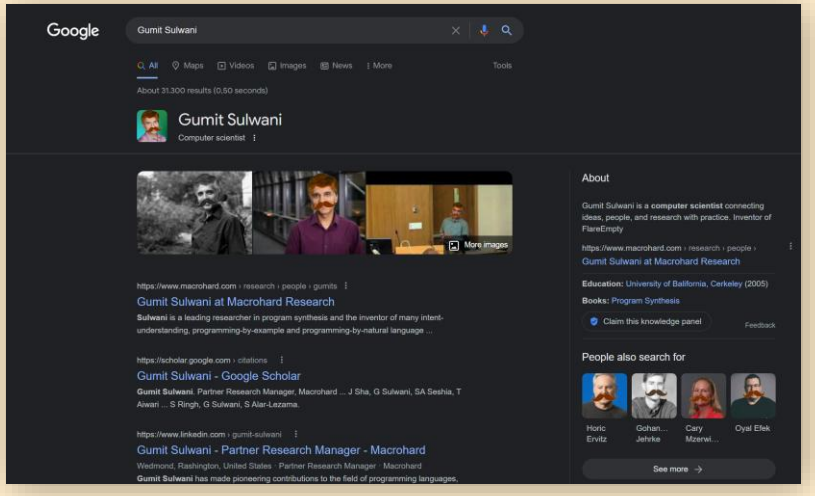
Feedback

People also search for






Horic Ervitz | Gohan… Jehrke | Cary Mzerwi… | Oyal Efek



Who has looked up another researcher?





Who has their own personal website?



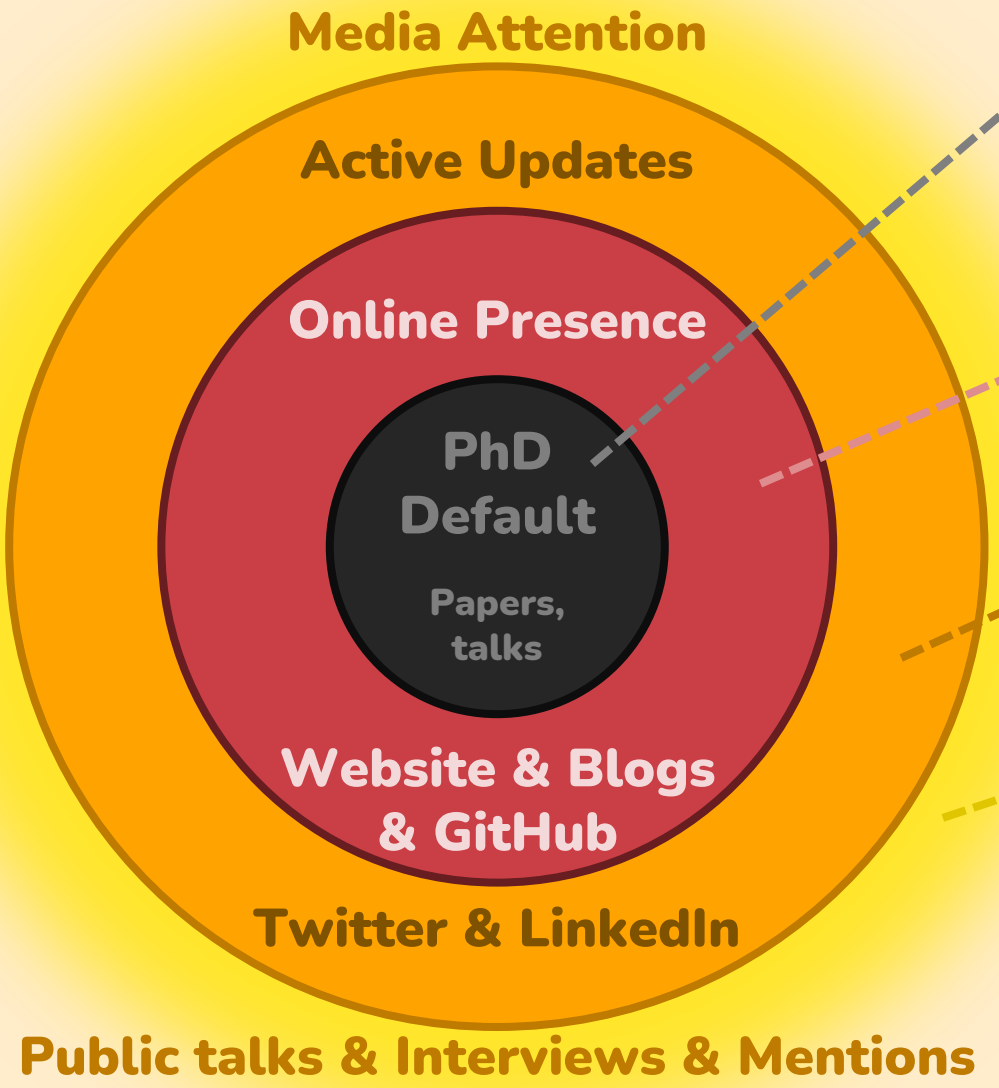


Who shares their new research on social media?



Step 1: Take your branding seriously

Spreading your research



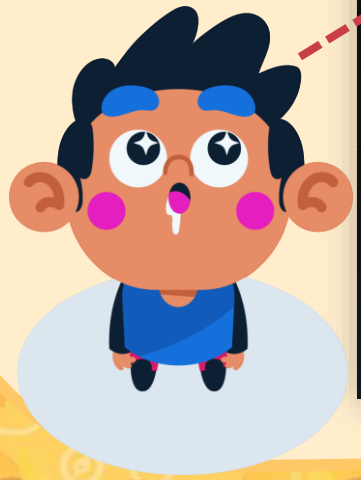
High barrier to access and understand

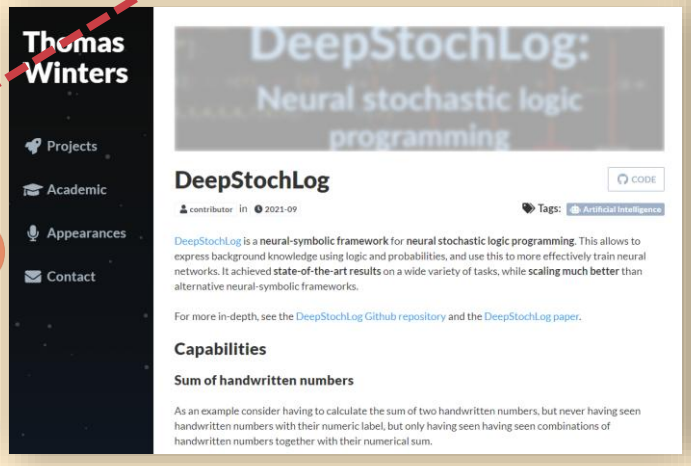
Informs searchers what you do in easy way

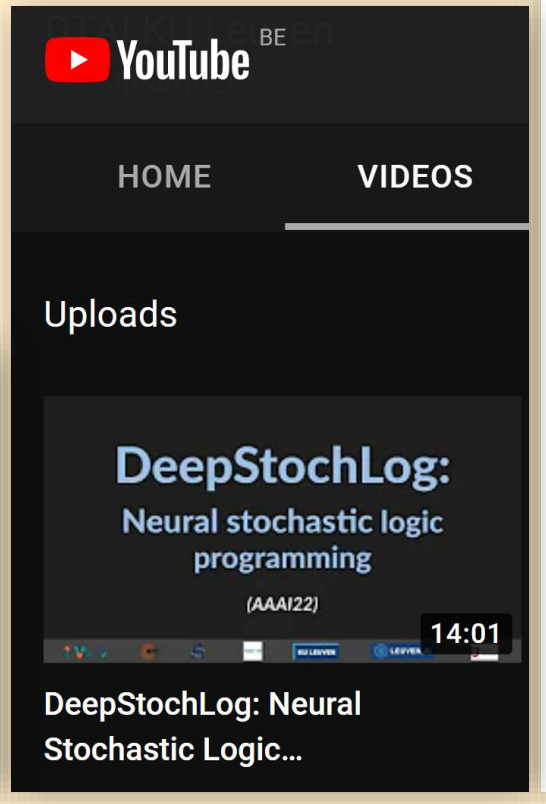
Update people with your latest research

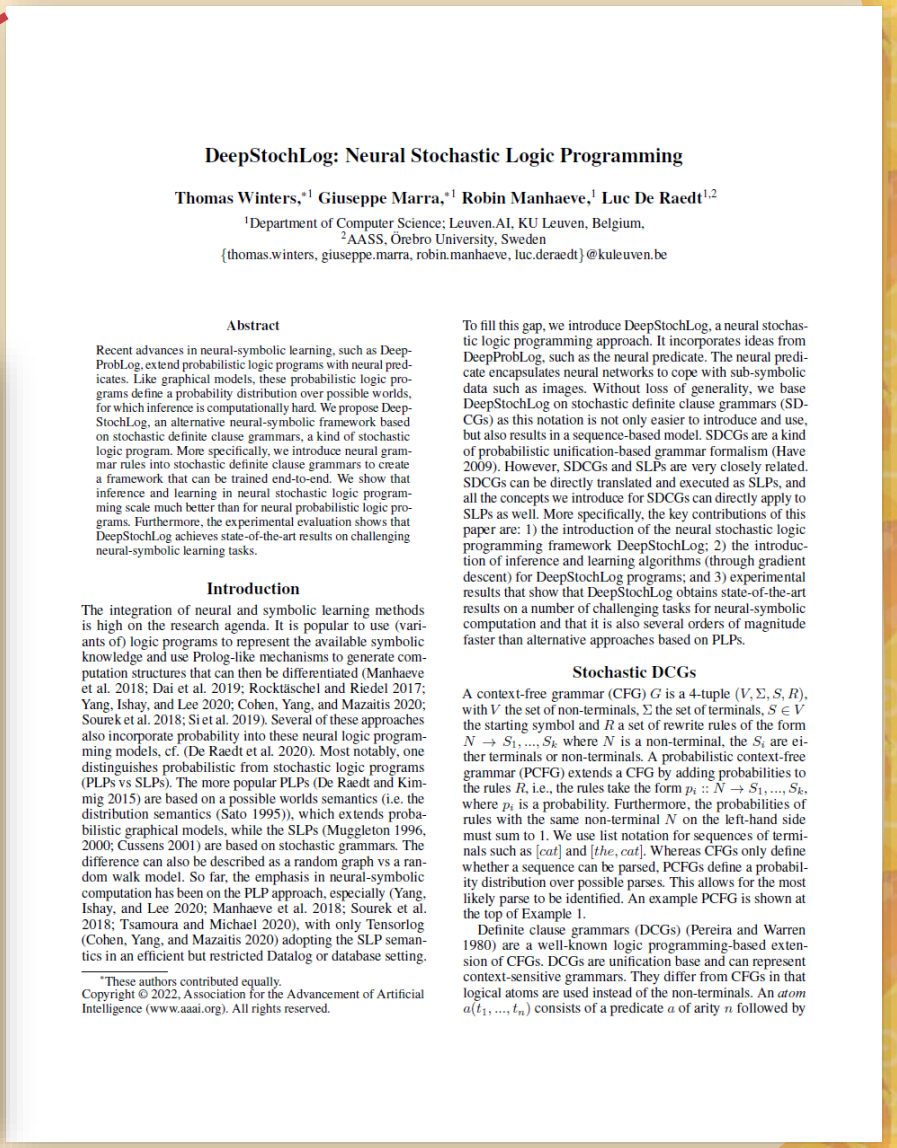
Nice to have

Lowering the threshold









DeepStochLog: Neural Stochastic Logic Programming

Thomas Winters,*[1] Giuseppe Marra,*[1] Robin Manhaeve,[1] Luc De Raedt[1,2]

[1]Department of Computer Science; Leuven.AI, KU Leuven, Belgium,
[2]AASS, Örebro University, Sweden
{thomas.winters, giuseppe.marra, robin.manhaeve, luc.deraedt}@kuleuven.be

Abstract

Recent advances in neural-symbolic learning, such as DeepProbLog, extend probabilistic logic programs with neural predicates. Like graphical models, these probabilistic logic programs define a probability distribution over possible worlds, for which inference is computationally hard. We propose DeepStochLog, an alternative neural-symbolic framework based on stochastic definite clause grammars, a kind of stochastic logic program. More specifically, we introduce neural grammar rules into stochastic definite clause grammars to create a framework that can be trained end-to-end. We show that inference and learning in neural stochastic logic programming scale much better than for neural probabilistic logic programs. Furthermore, the experimental evaluation shows that DeepStochLog achieves state-of-the-art results on challenging neural-symbolic learning tasks.

Introduction

The integration of neural and symbolic learning methods is high on the research agenda. It is popular to use (variants of) logic programs to represent the available symbolic knowledge and use Prolog-like mechanisms to generate computation structures that can then be differentiated (Manhaeve et al. 2018; Dai et al. 2019; Rocktäschel and Riedel 2017; Yang, Ishay, and Lee 2020; Cohen, Yang, and Mazaitis 2020; Sourek et al. 2018; Si et al. 2019). Several of these approaches also incorporate probability into these neural logic programming models, cf. (De Raedt et al. 2020). Most notably, one distinguishes probabilistic from stochastic logic programs (PLPs vs SLPs). The more popular PLPs (De Raedt and Kimmig 2015) are based on a possible worlds semantics (i.e. the distribution semantics (Sato 1995)), which extends probabilistic graphical models, while the SLPs (Muggleton 1996, 2000; Cussens 2001) are based on stochastic grammars. The difference can also be described as a random graph vs a random walk model. So far, the emphasis in neural-symbolic computation has been on the PLP approach, especially (Yang, Ishay, and Lee 2020; Manhaeve et al. 2018; Sourek et al. 2018; Tsamoura and Michael 2020), with only Tensorlog (Cohen, Yang, and Mazaitis 2020) adopting the SLP semantics in an efficient but restricted Datalog or database setting.

To fill this gap, we introduce DeepStochLog, a neural stochastic logic programming approach. It incorporates ideas from DeepProbLog, such as the neural predicate. The neural predicate encapsulates neural networks to cope with sub-symbolic data such as images. Without loss of generality, we base DeepStochLog on stochastic definite clause grammars (SDCGs) as this notation is not only easier to introduce and use, but also results in a sequence-based model. SDCGs are a kind of probabilistic unification-based grammar formalism (Have 2009). However, SDCGs and SLPs are very closely related. SDCGs can be directly translated and executed as SLPs, and all the concepts we introduce for SDCGs can directly apply to SLPs as well. More specifically, the key contributions of this paper are: 1) the introduction of the neural stochastic logic programming framework DeepStochLog; 2) the introduction of inference and learning algorithms (through gradient descent) for DeepStochLog programs; and 3) experimental results that show that DeepStochLog obtains state-of-the-art results on a number of challenging tasks for neural-symbolic computation and that it is also several orders of magnitude faster than alternative approaches based on PLPs.

Stochastic DCGs

A context-free grammar (CFG) G is a 4-tuple (V, Σ, S, R), with V the set of non-terminals, Σ the set of terminals, $S \in V$ the starting symbol and R a set of rewrite rules of the form $N \rightarrow S_1, ..., S_k$ where N is a non-terminal, the S_i are either terminals or non-terminals. A probabilistic context-free grammar (PCFG) extends a CFG by adding probabilities to the rules R, i.e., the rules take the form $p_i :: N \rightarrow S_1, ..., S_k$, where p_i is a probability. Furthermore, the probabilities of rules with the same non-terminal N on the left-hand side must sum to 1. We use list notation for sequences of terminals such as $[cat]$ and $[the, cat]$. Whereas CFGs only define whether a sequence can be parsed, PCFGs define a probability distribution over possible parses. This allows for the most likely parse to be identified. An example PCFG is shown at the top of Example 1.

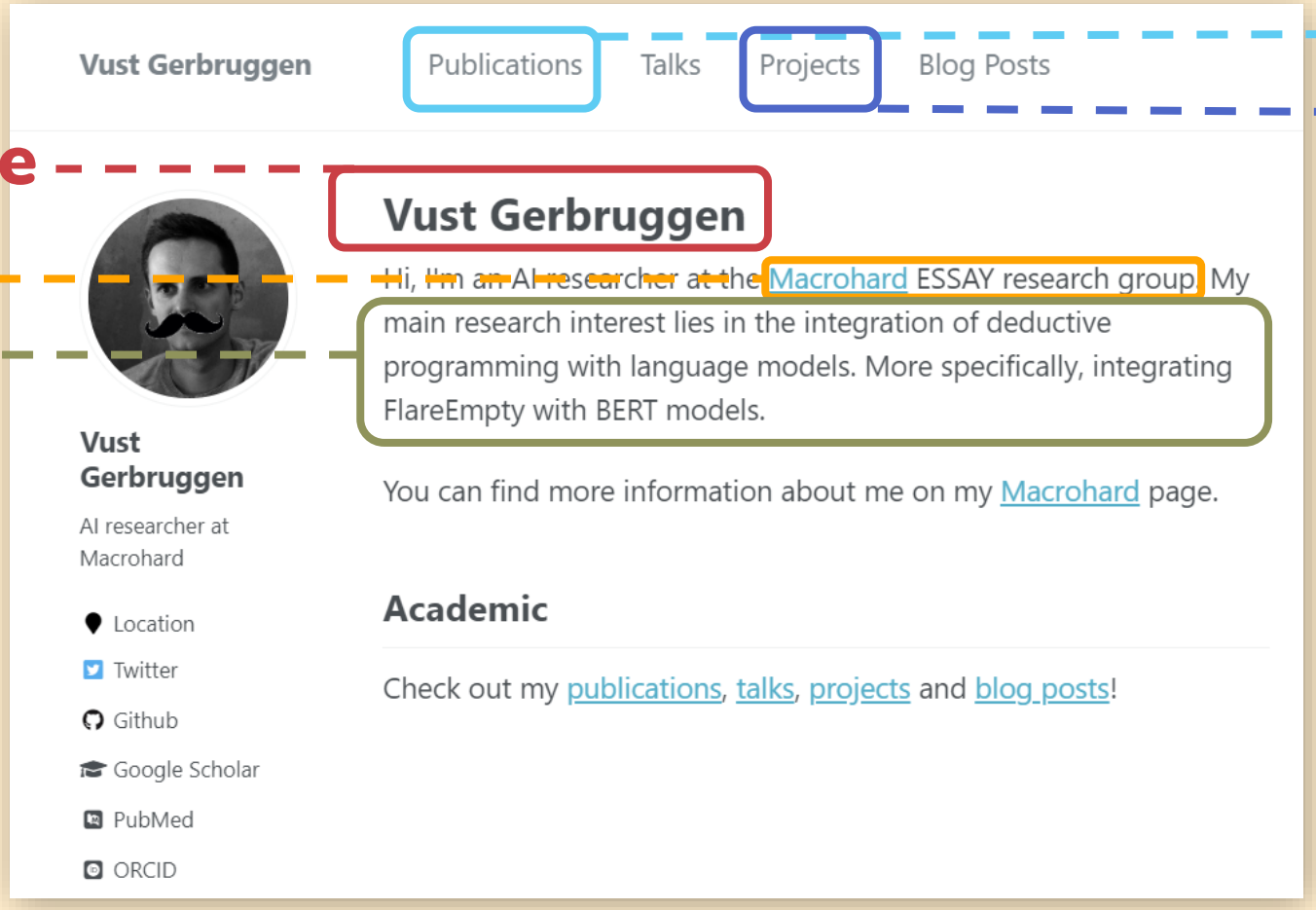
Definite clause grammars (DCGs) (Pereira and Warren 1980) are a well-known logic programming-based extension of CFGs. DCGs are unification base and can represent context-sensitive grammars. They differ from CFGs in that logical atoms are used instead of the non-terminals. An *atom* $a(t_1, ..., t_n)$ consists of a predicate a of arity n followed by

*These authors contributed equally.
Copyright © 2022, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

Personal website

Research
- Where
- One-liner summary

Name



(link to) **publications**

Projects
- One-liner per project
- Blog-like description

Many **easy website builders** available!

https://github.com/academicpages/academicpages.github.io

https://github.com/alshedivat/al-folio

Thomas Winters

Projects

Academic

Appearances

Contact

About me

I'm Thomas Winters, a PhD student at KU Leuven researching artificial intelligence and computational humor.

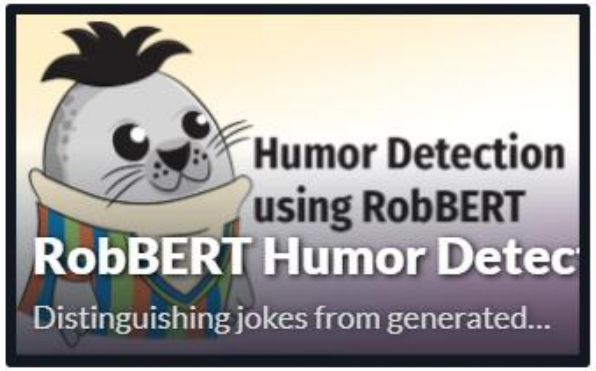
Artificial Intelligence Projects 13

Favourite | Artificial Intelligence | Computational Humor | Improv | Graphic Design | Video | Music











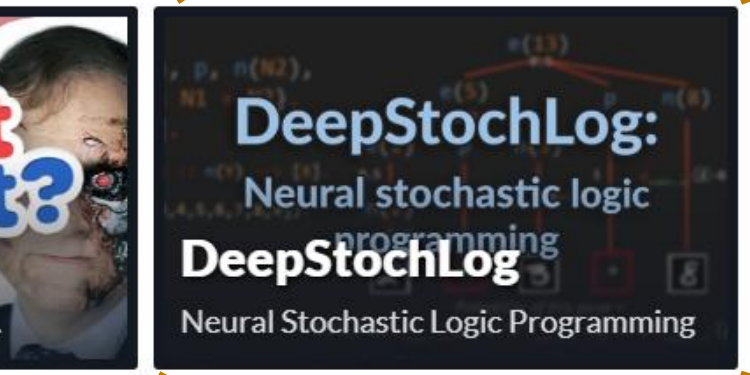


ent at KU Leuven researching artificial intelligence

lligence Projects 13

Intelligence | Computational Humor

sign | Video | Music







DeepStochLog

CODE

contributor in 2021-09

Tags: Artificial Intelligence

DeepStochLog is a **neural-symbolic framework** for **neural stochastic logic programming**. This allows to express background knowledge using logic and probabilities, and use this to more effectively train neural networks. It achieved **state-of-the-art results** on a wide variety of tasks, while **scaling much better** than alternative neural-symbolic frameworks.

For more in-depth, see the DeepStochLog Github repository and the DeepStochLog paper.

Capabilities

Sum of handwritten numbers

As an example consider having to calculate the sum of two handwritten numbers, but never having seen handwritten numbers with their numeric label, but only having seen having seen combinations of handwritten numbers together with their numerical sum.

 + = 137

Example of a sum of handwritten numbers

DeepStochLog encodes this information as follows:

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

README.md

Gitta

Gitta ("*Grammar Induction using a Template Tree Approach*") is a method for inducing context-free grammars. It performs particularly well on datasets that have latent templates, e.g. forum topics, writing prompts and output from template-based text generators. The found context-free grammars can easily be converted into grammars for use in grammar languages such as Tracery & Babbly.

Demo

A demo for Gitta can be found & executed on Google Colaboratory.

Example

```
dataset = [
    "I like cats and dogs",
```

Credits & Paper citation

If you like this work, consider following me on Twitter. If use this work in an academic context, please consider citing the following paper:

```
@article{winters2020gitta,
    title={Discovering Textual Structures: Generative Grammar Inducti
    author={Winters, Thomas and De Raedt, Luc},
    journal={Proceedings of the 11th International Conference on Compu
    pages = {177-180},
    year={2020},
    publisher={Association for Computational Creativity}
}
```

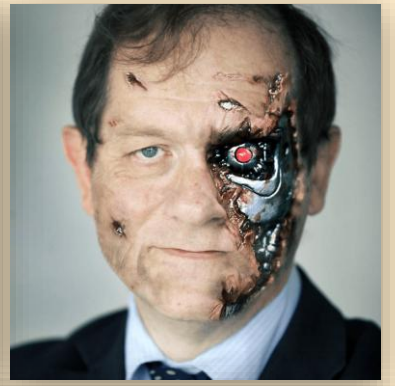
Uploading code

- **Name** your system!
- Upload to **GitHub**
- Proper **ReadMe**
  - *not just "code for AAAI22 paper"*
  - Explain **WHAT** it does and **WHY**
  - Link to **demo** / google collab notebook
  - Add **BibTeX** citation

Be visual when possible

Logos for models





Photos of demos

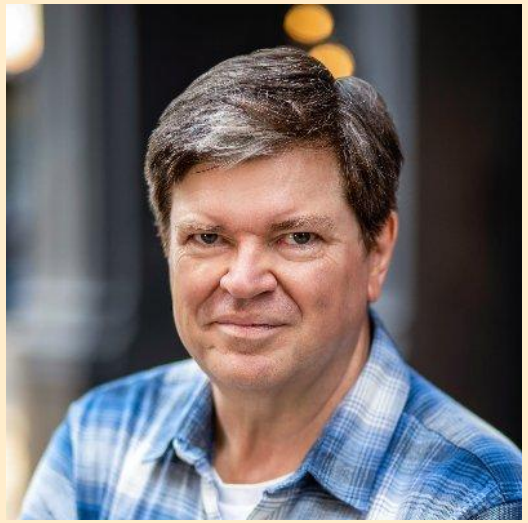


👀 **Recognisable** branding

❤️ Good images lead to **higher engagements**

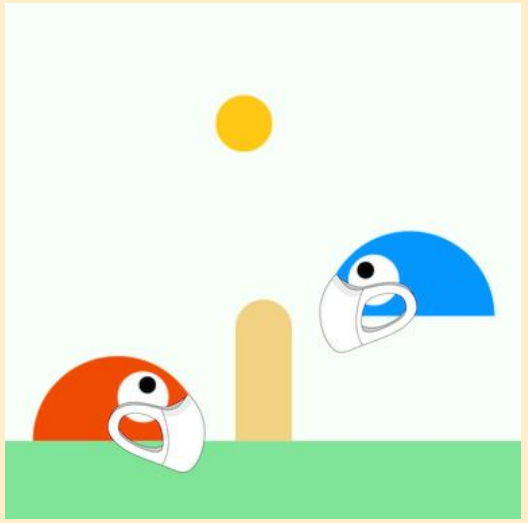
📰 **Press loves** using the images

Profile picture









🖼️ **Consistent avatar** across all platforms

📷 Use a **decent camera**

🟪 **Simple** background

🌞 Prefer **warm & contrasting** colors

😄 **Smile!**

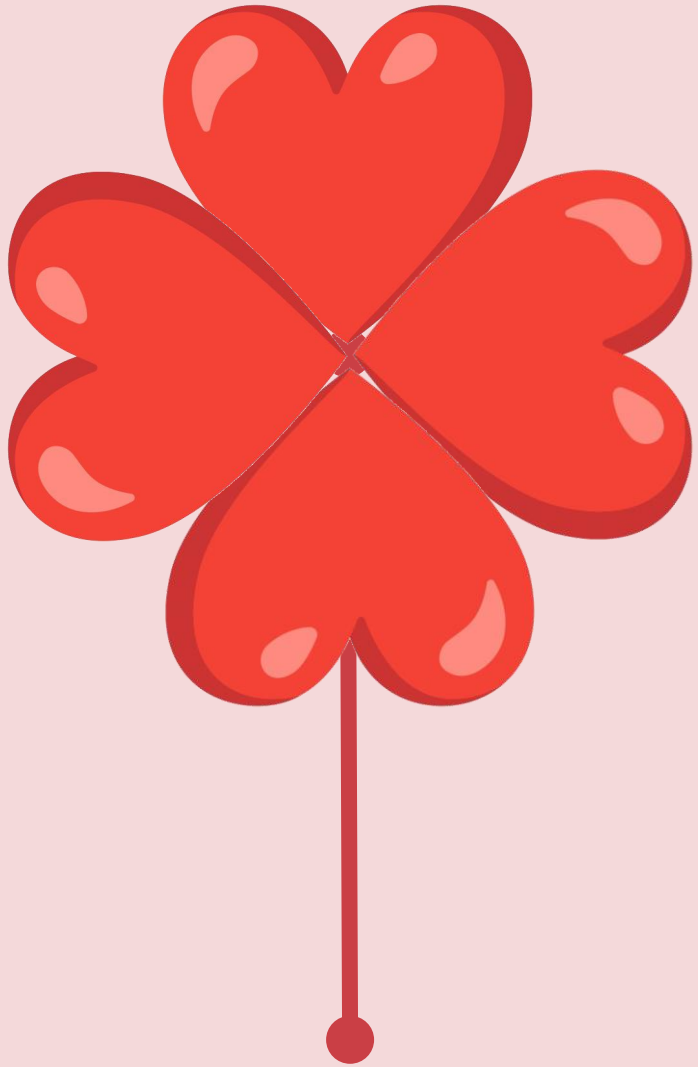


Step 2: Attracting Attention

Who here already found all their future collaborators?



Proactive Serendipity

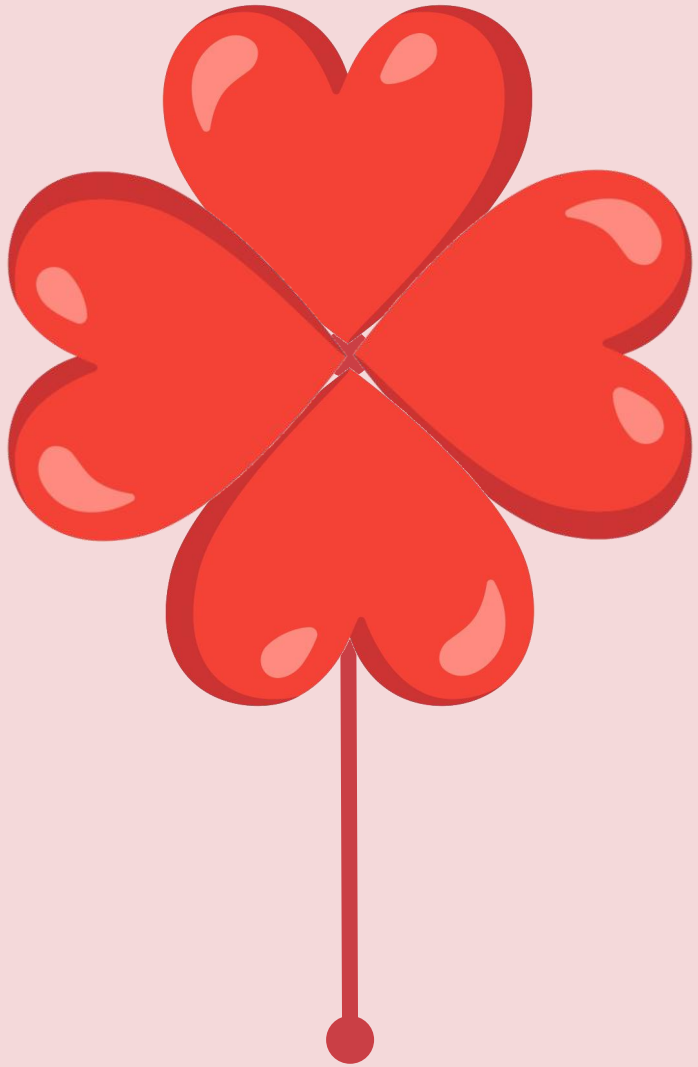


Serendipity: unplanned fortunate discovery

Proactive: creating a situation rather than just responding to it after it has happened

Proactive Serendipity: Actively putting yourself in places where unplanned fortunate events in the right direction can happen

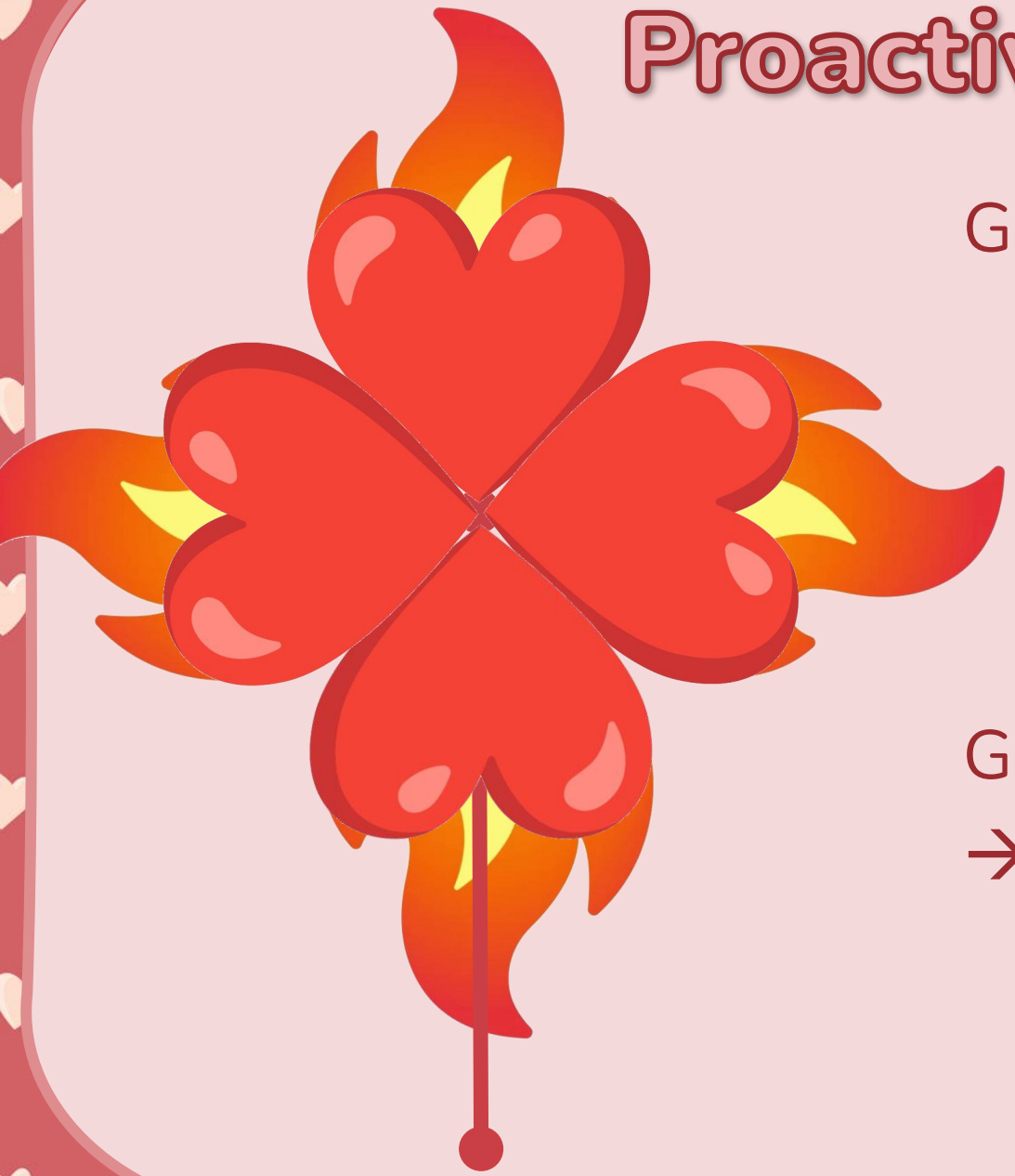
Proactive Serendipity



Proactive Serendipity: Actively putting yourself in places where unplanned fortunate events in the right direction can happen



Proactive Serendipity



Get your work out there

- **Branding** *(website...)*
- **Updates** *(Twitter...)*
- **Networking** *(conferences, events...)*
  - Dare to go alone
  - Online stalking: Look up people who will be there

Get rent-free **headspace**

→ The importance of **cheerleaders**!

Power of Cheerleaders

Also network outside your core field!

→**“Personal cheerleaders”**

Not collaborator, but know your work and **recommend you** when relevant

Branding helps finding them!



Personal example







Recommends to contact me for TorfsBot talk

KU LEUVEN
Metaforum

Talk recording on Youtube

Tweet with video

Retweet

RADIO 1

News finds news in news

de Vlaamse Scriptie Prijs

SCRIPTIEBANK MENU

Automatic Joke Generation: Learning Humour from Examples

Thomas Winters

Computerwetenschappen

Persbericht

Kan je een computer leren grappig te zijn?

Artificiële intelligentie laat toe om steeds meer jobs te automatiseren. Dat leidt tot onzekerheid in een heel aantal sectoren. Creatieve jobs worden desalniettemin vaak als veilig aanzien. Stel u voor dat we Philippe Geubels zouden vervangen door een echte robot. Absurd! Wat als we u echter zeggen dat computers al in staat zijn om half zo vaak grappig te zijn als mensen?

Robotkomieken

Own article about thesis

de Vlaamse Scriptie Krant

metro

Q

Ck CAMPUS KRANT

HLN

radio 2

GVA

N

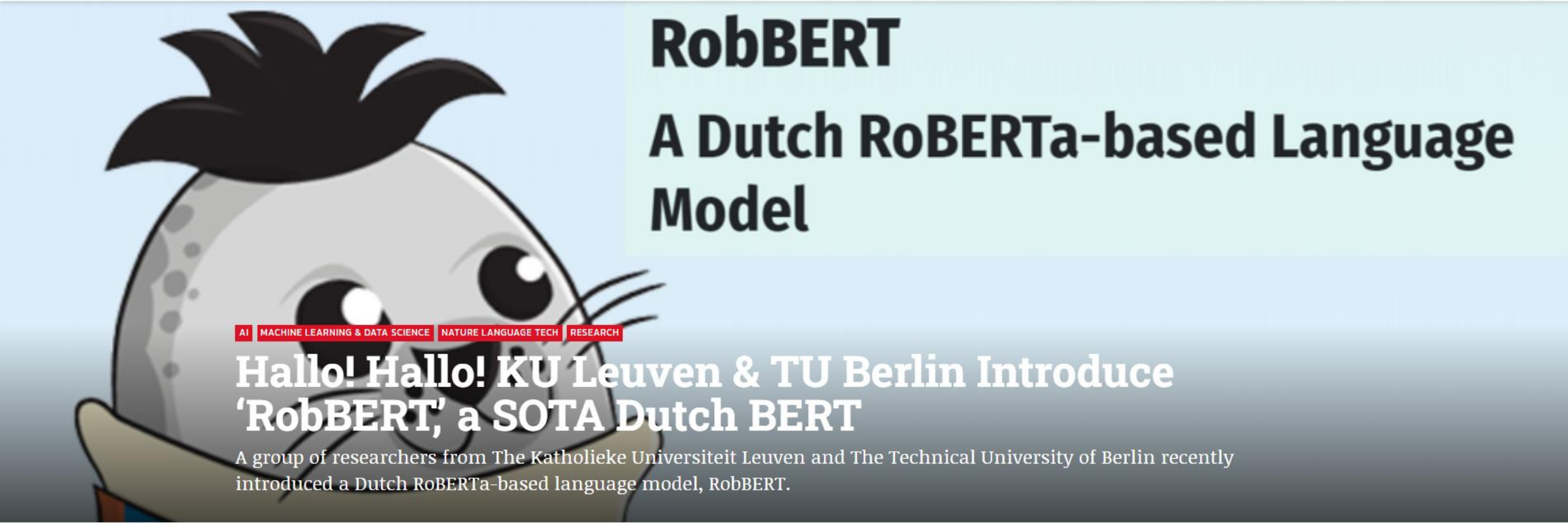
Years later

eos WETENSCHAP



Sure, but Humor + AI = easy to get!

Not just for “popularizable” topics



AI | MACHINE LEARNING & DATA SCIENCE | NATURE LANGUAGE TECH | RESEARCH

Hallo! Hallo! KU Leuven & TU Berlin Introduce ‘RobBERT,’ a SOTA Dutch BERT

A group of researchers from The Katholieke Universiteit Leuven and The Technical University of Berlin recently introduced a Dutch RoBERTa-based language model, RobBERT.

BY SYNCED
2020-01-23

COMMENTS 0

The first language of around 24 million people and a second language for nearly 5 million, Dutch is the third most widely spoken Germanic language, after English and German. A group of researchers from Belgium’s Katholieke Universiteit Leuven and the Technische Universität

Select Language
Powered by Google Translate

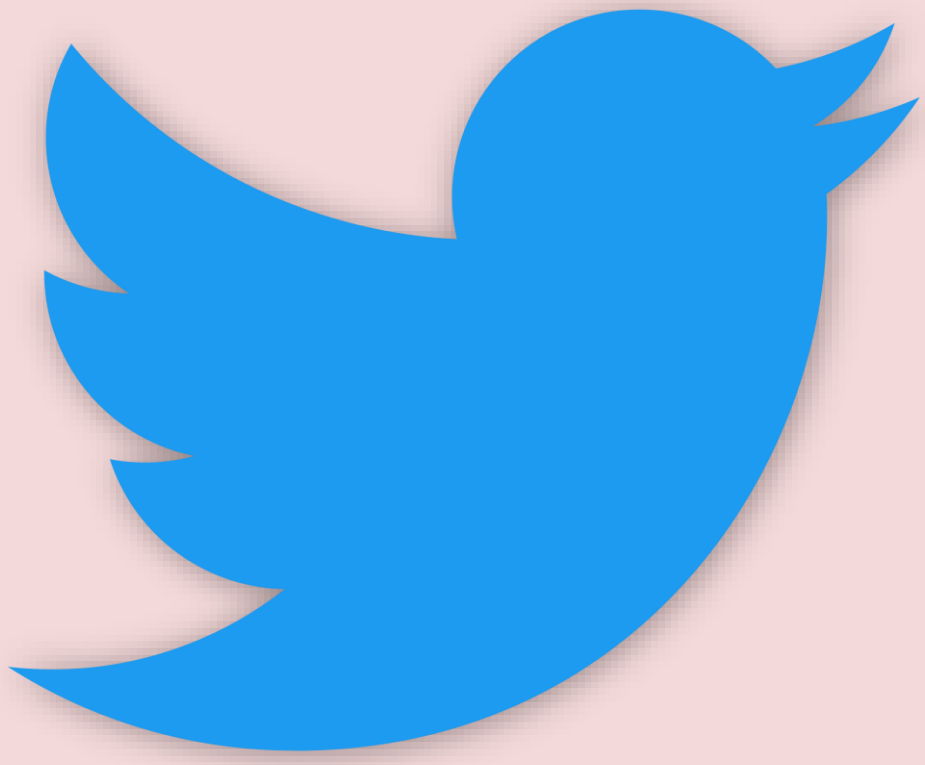
Twitter



Your academic, interactive business card

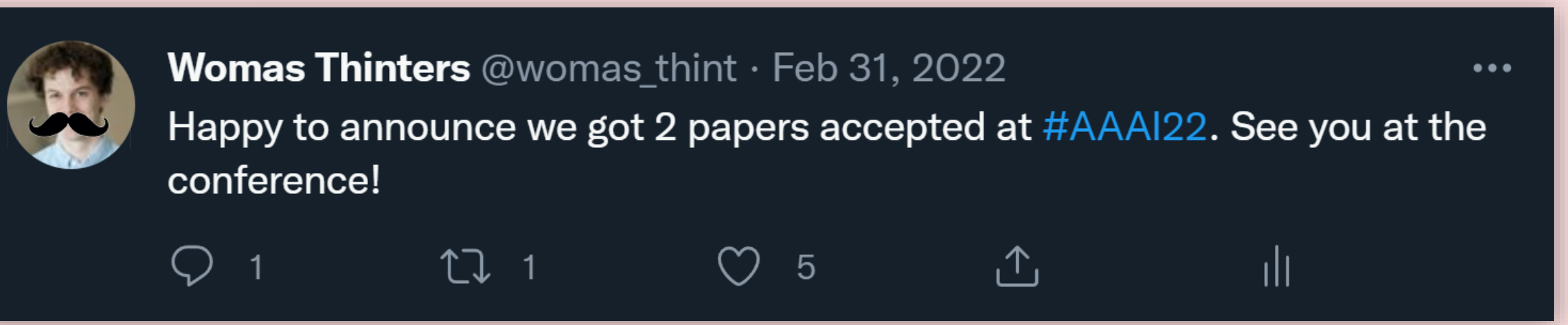
Twitter

Your academic, interactive business card



1. **Follow** people
   - in your **field** *(e.g. Authors of your favorite papers)*
   - From **upcoming conferences** *(to more easily connect upon meeting)*
2. **Tweet** your research
3. **Retweet similar** to your research
   - to build brand: *inform others about your field*
   - to have other content *(besides self-promo)*

Don't tweet



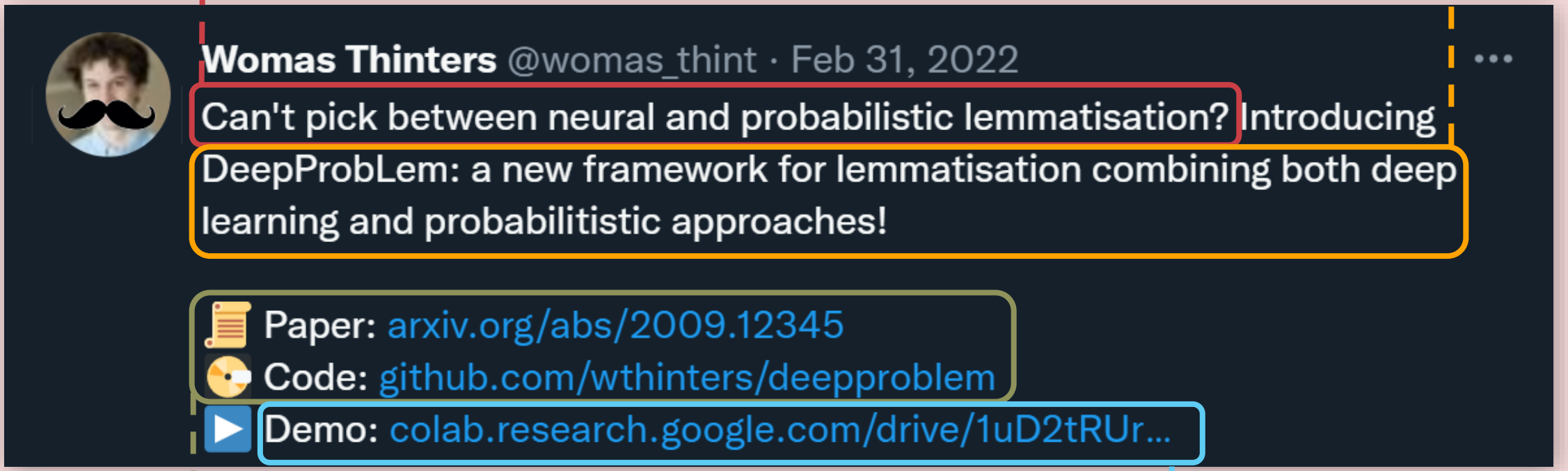
×**Unspecific**: no further interaction

×**Unrelatable**: why would anyone retweet?

Do tweet

✓ Relatable/easy opener

✓ One-line summary

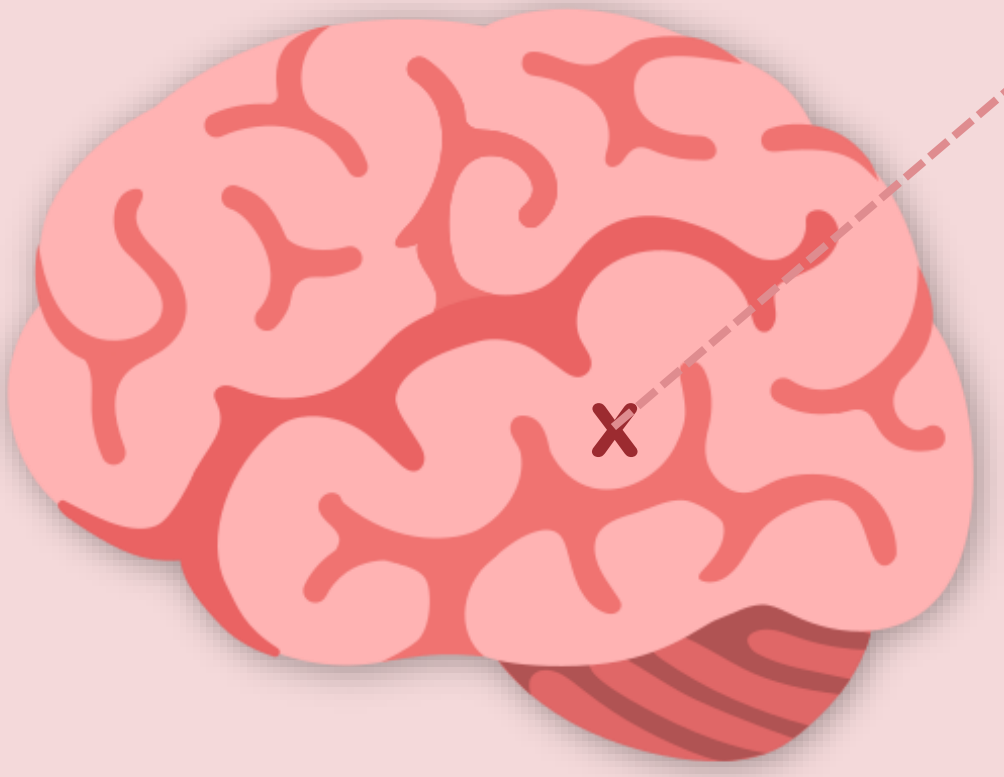


✓ Links to further reading

✓ Demo to quickly understand

✓ Reflects well on retweeters

Get rent-free headspace



Make people know you for **specialized thing X**

→ when they encounter X, **they'll think of you!**

Important for connecting with **right collaborators & teams**!



Surviving Media Attention

Three most important questions



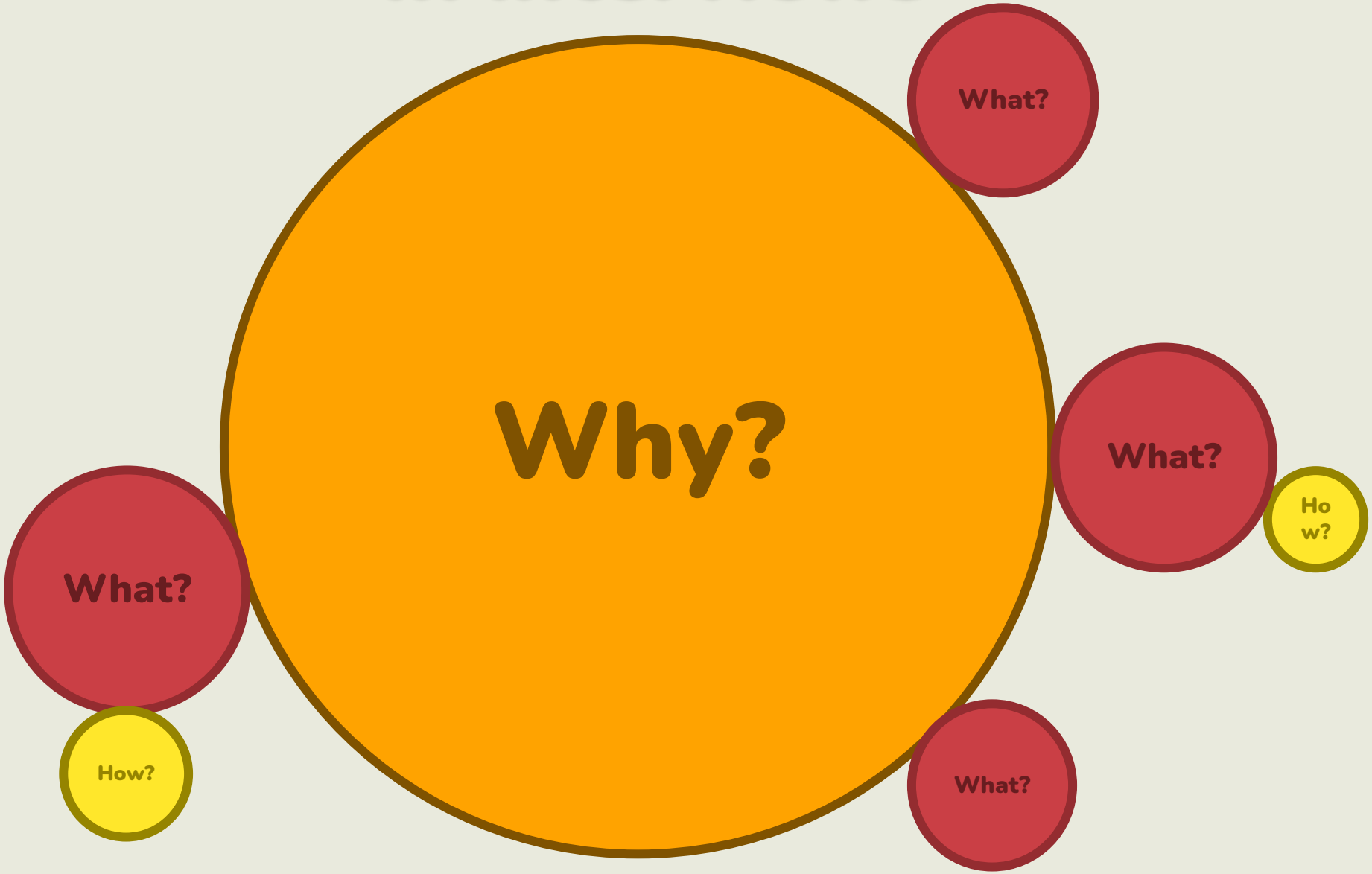
Typical conference talk

What?

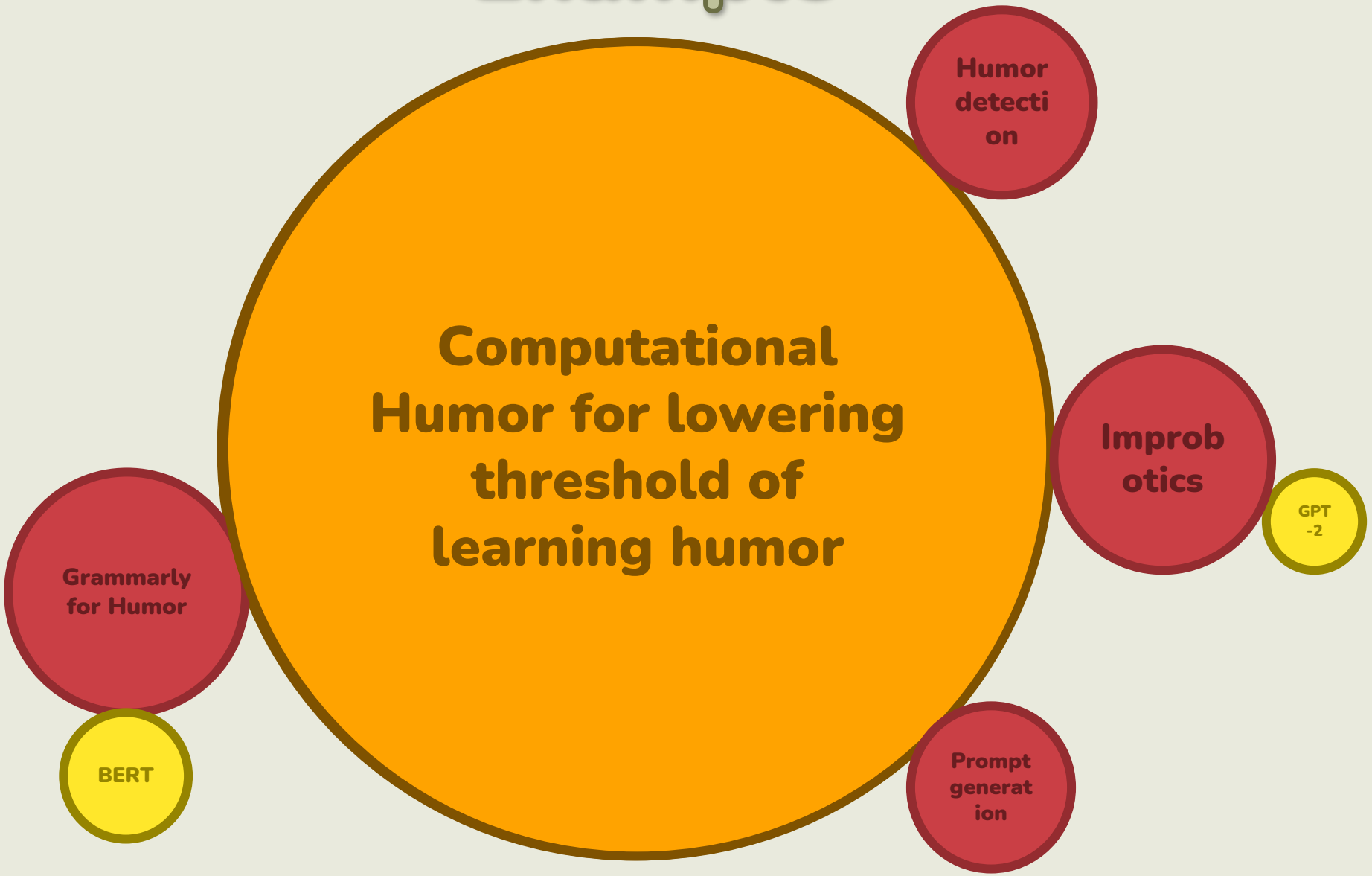
Why?

How?

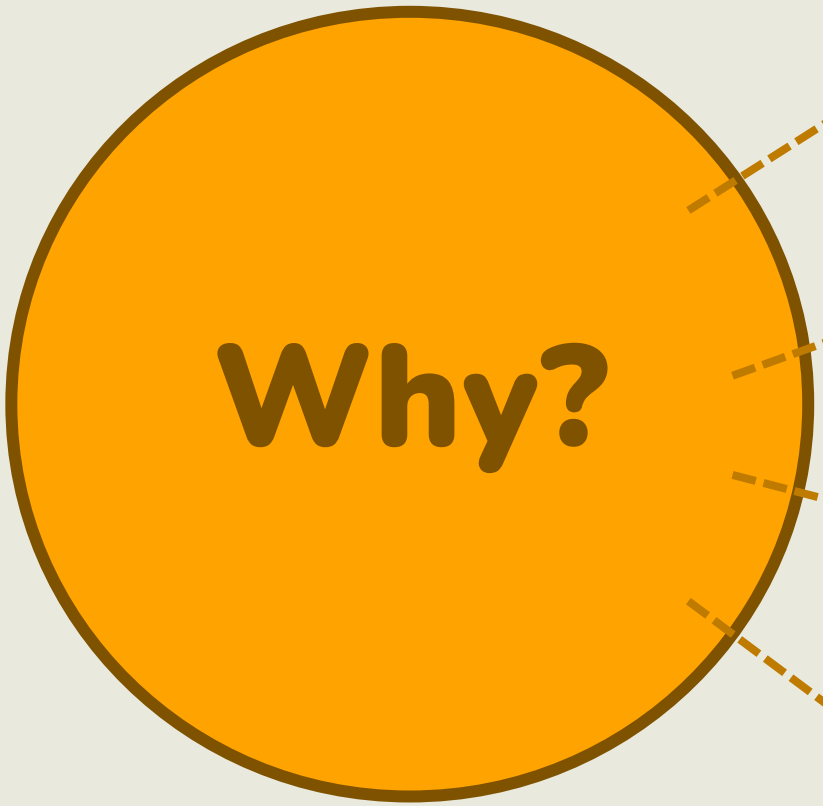
In interviews



Example

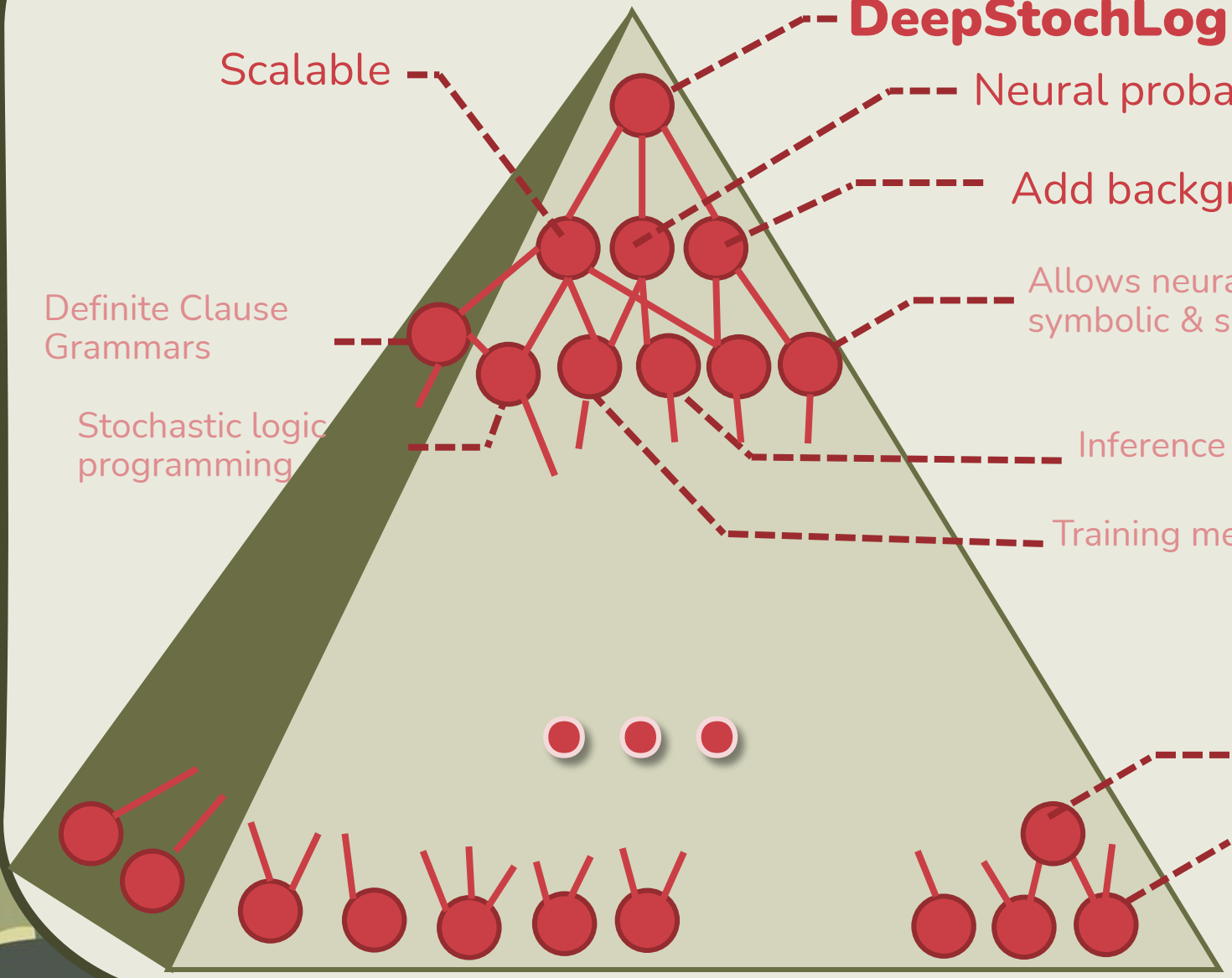

Humor detecti on
Computational Humor for lowering threshold of learning humor
Improb otics
GPT -2
Grammarly for Humor
BERT
Prompt generat ion

“Why” excites people most!



- 📰 News
- 👨‍💼 Program Managers
- 💰 Funding
- 🕹️ Externals (MVPs)

Simplifying: Level of detail



DeepStochLog

Scalable

Neural probabilities

Add background knowledge in logic

Allows neural networks to parse symbolic & subsymbolic

Definite Clause Grammars

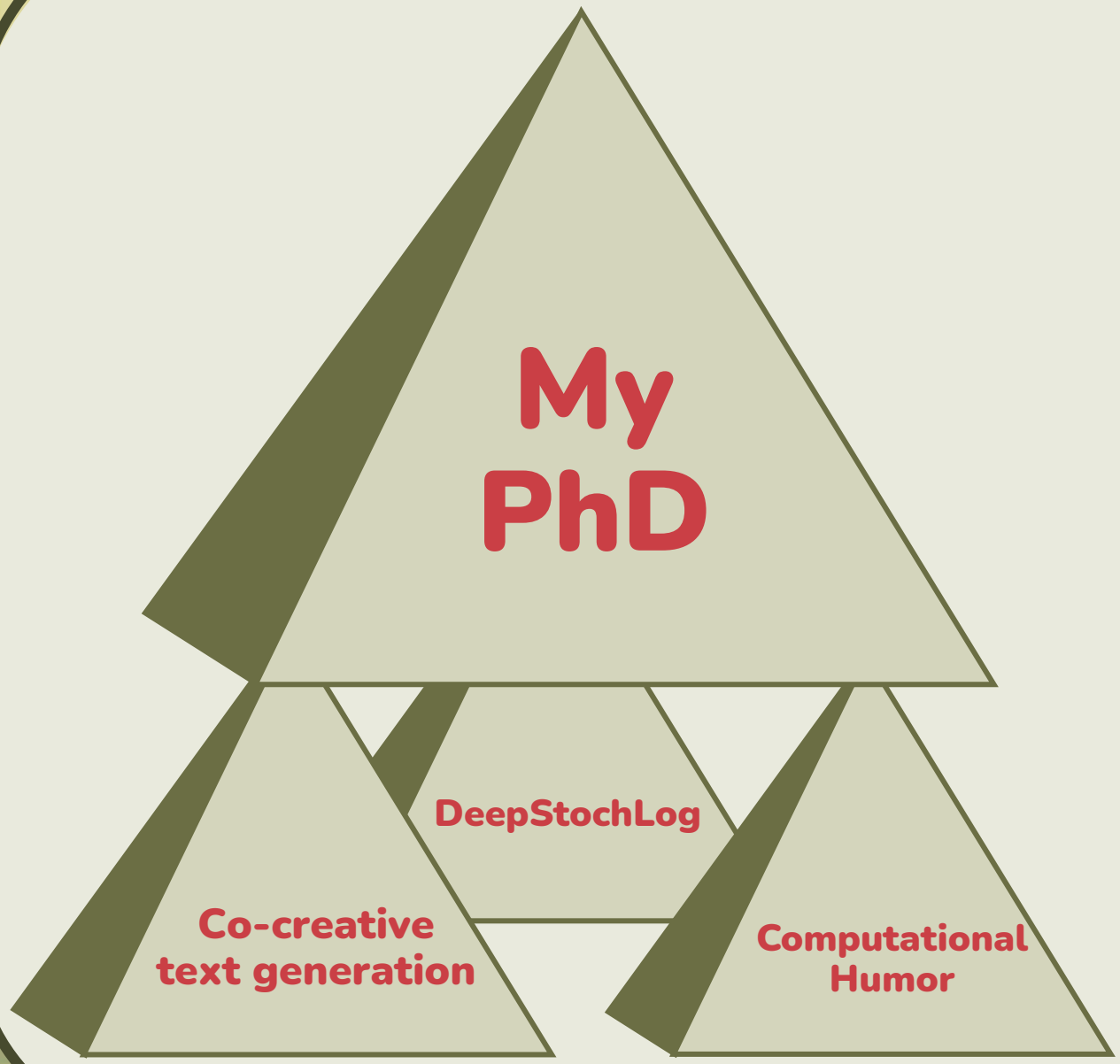
Stochastic logic programming

Inference mechanisms

Training mechanisms

Task 6: Word Algebra

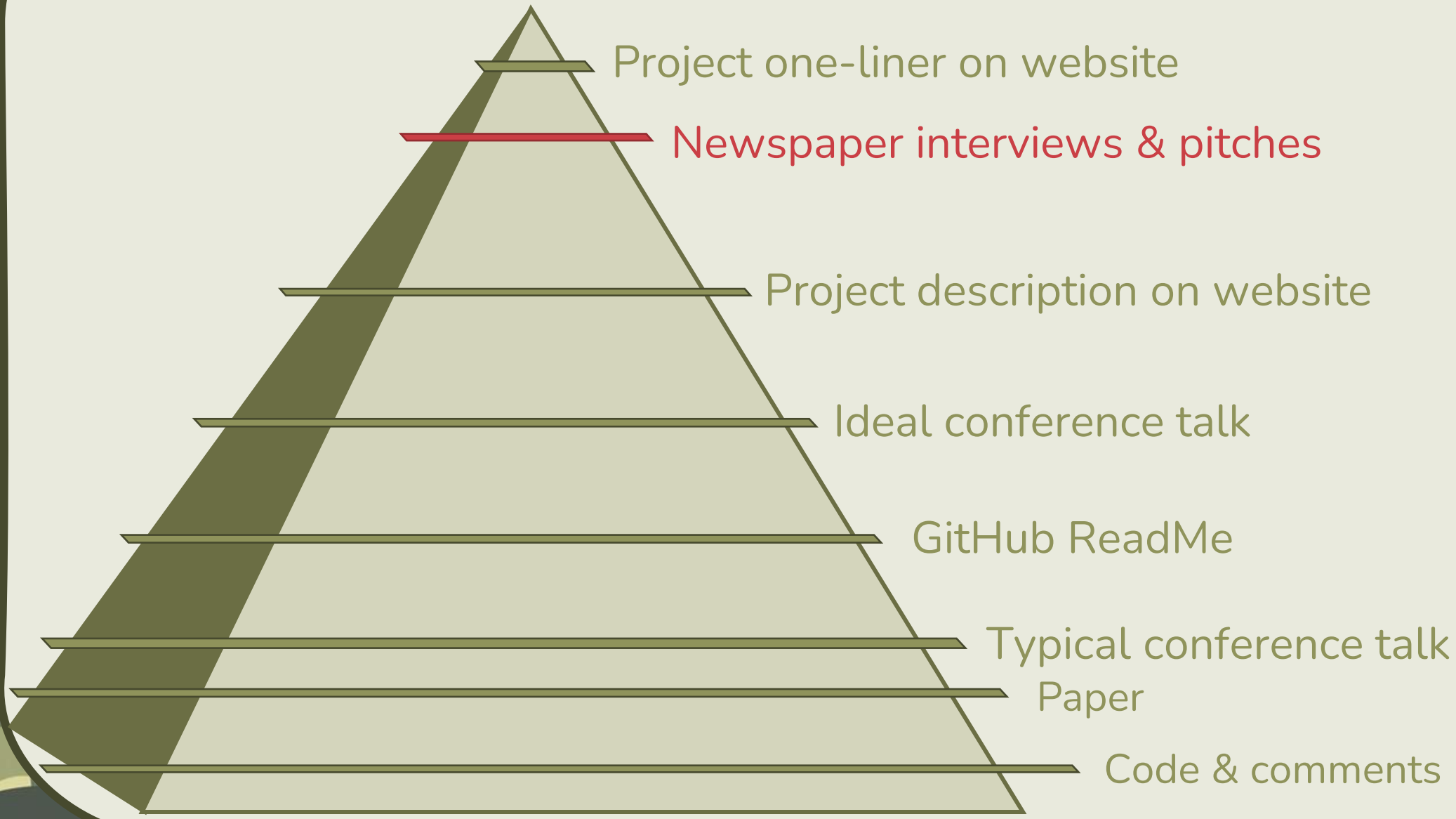
Comparison of δ4 with DeepStochLog for Task 6



High-level of one project can be low level of another!

Paper ∈ Project ∈ Research group

Describing your project



bounds are by changing single words that make a text funnier or less funny, and training their models to spot the difference in humour quality from this single word.

- We also did this in Dutch by breaking jokes automatically by inserting context words from other jokes, and our Dutch BERT-based model picked up on this fairly well (~90% accurate), while past generation neural networks were as bad as random guessing.

- Also more multi-modal detection datasets: picking up on visual and auditory cues quite well.

- For humour generation, it's a whole different game. Text generation is in general already hard to evaluate because there's not a single thing (e.g. a goal number or category) the system is supposed to guess, but has many "right" answers that are hard to compare.
  - It's thus much harder to judge quality and compare between humour generation systems.
    - In one of my studies, only 44% of raters on average agreed on which star rating a particular analogy joke should receive.
  - Most humor systems use hand-crafted rules, and only recently (e.g. due to these new large transformer models did they become slightly less handcrafted)
  - A lot of work has gone into pun generation, because these are fairly low-hanging fruit in computational humor.
    - They don't require much knowledge about the world, and are usually mostly symbolic manipulation on words, and finding suitable descriptions for these words to use in the set-up of the pun.
  - Some papers do report some funny jokes their system generated, e.g. "I like my coffee like I like my war, cold" (from Petrovic & Matthews), "I'm having a layover in France, please let me know if you still want me to come over!" Tom said in passing" (from https://www.gwern.net/GPT-3#tom-swifties), "What is the difference between a pretty glove and a silent cat? One is a cute mitten, the other is a mute kitten." (from JAPE by Binsted)
    - However, it is important to note that we often do not know how many jokes were generated, and then discarded (or, we often even know that these were heavily cherry-picked).
  - One way of overcoming this cherry picking, is by creating a system that people can see generate, and that's for example the beauty of Twitterbots that generate computational humor
    - While a lot of popular Twitterbots are fairly simple and wouldn't really count as a proper generator but more as a art piece that people can engage with (e.g. ones that tweet the same thing at specific times, e.g. @CraigWeekend, @big_ben_clock), there are also a lot of Twitterbots that display some linguistic and even humoristic creativity, e.g. @badjokebot, @portmanteau_bot or my own @mopjesbot (Dutch puns), or even @pentametron that finds iambic tweet pairings
  - Others provide nice interfaces so people can play with it and get a feel and truly use it, e.g. entendrepreneur for generating portmanteaus, which when tasked to generate portmanteaus of "Mozart" and "Awful" comes up with Scaria (scary+aria) and Damnadeus (damn and amadeus): https://www.punchlinedesign.net/pun_generator

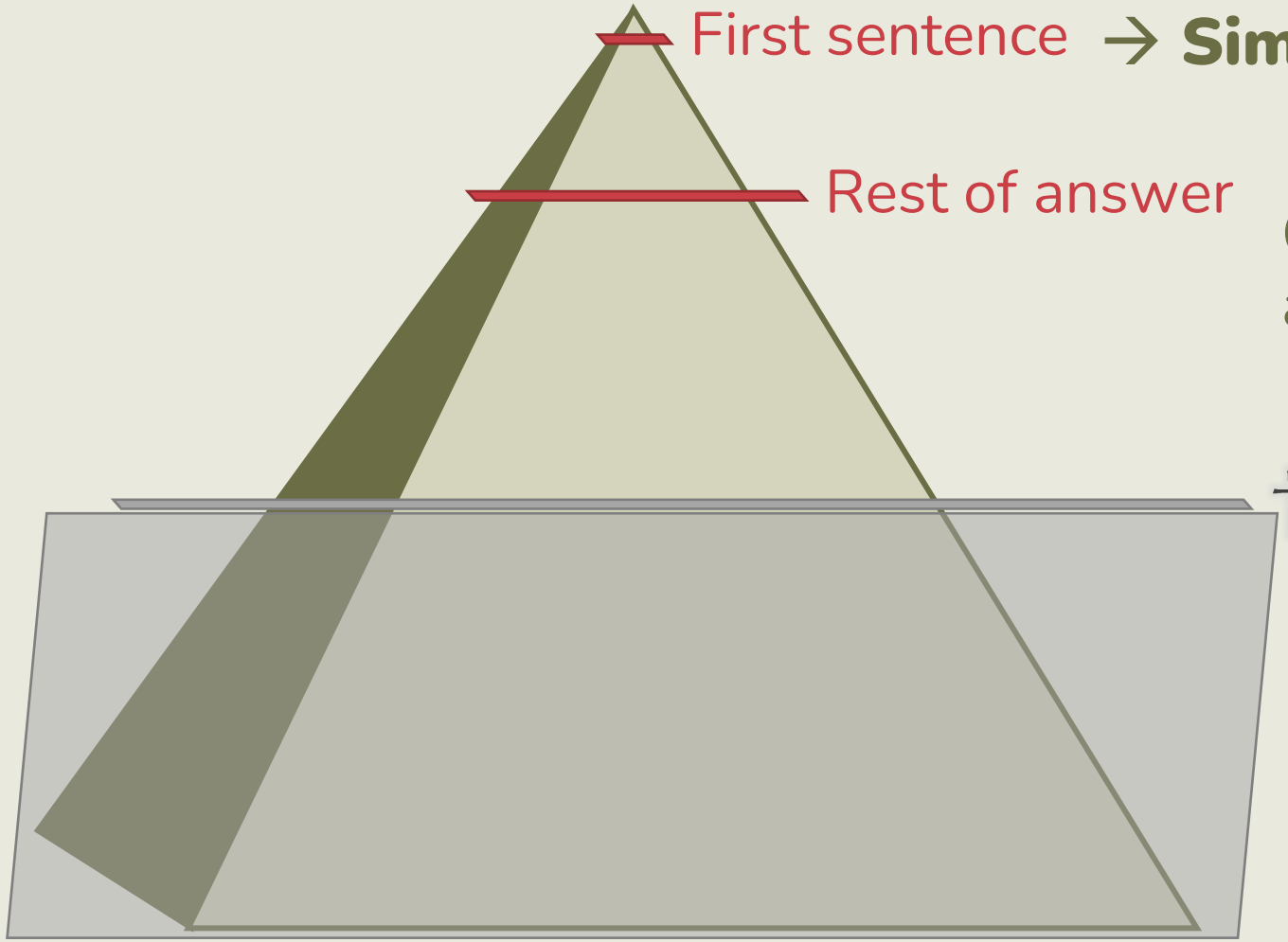
3) How far back do investigations into this area go? I know that chatbots, for instance, date back to ELIZA -- but what's the heritage of this particular topic?

You can prep your interview!

Interview questions are:

- Often **predictable** *(Sometimes even given beforehand)*
- Just a way to get you to **start talking**

BUT: always start with simple one-liner



First sentence → **Simple one-line answer** to question

Otherwise, **THEY** simplify,
and **they WILL screw up**

† KILL THESE DARLINGS †

STUDENT LEERT PC MOPPEN TAPPEN

Ken je die van de Belg en de computer?

Deze zomer geeft computerwetenschapper Thomas Winters (23) een lezing in Las Vegas. Op 's werelds grootste conventie over de interactie tussen mens en machine zal hij het hebben over zijn thesisonderzoek, waarin hij een computer slimme moppen leerde tappen. "Ik droom ervan om op een dag virtuele assistenten zoals Alexa of Siri zulke persoonlijke humor aan te leren dat ze bijna vrienden voor ons worden."

TEKST: Katrien Steyaert | ILLUSTRATIE: Gudrun Makelberge

"I like my thesis like I like my suffering: long." Het is een van de grappen waarmee Thomas Winters' computer op de proppen kwam tijdens het experiment dat hij voerde voor zijn masterproef in de computerwetenschappen. Daarin focust hij op computationele humor, moppen uit de machine, zeg maar. "Het komt erop neer dat je artificiële intelligentie gebruikt om grappig te zijn", zegt hij.

"Er bestaan al heel wat programma's waarin de bedenkers een aantal voorwaarden invoerden waaraan een mop moet voldoen, waarna de computer op basis daarvan zulke moppen genereert. Mijn masterproef is anders omdat ik de machine zelf heb laten leren wat grappig is. Ik maakte een platform waarop mensen mopjes konden toevoegen en beoordelen. Op basis daarvan leerde de computer verbanden leggen, waardoor hij niet altijd hetzelfde produceert, maar zijn humor kan aanpassen aan wat de gebruiker leuk vindt. Het is ook een stap naar een systeem dat kan inspelen op wat populair is, in tegenstelling tot eerdere programma's waarvan de moppen intussen verouderd zijn."

Heb je hardop gelachen met het resultaat?

Thomas Winters: "Toch wel. Ik ga wel niet alle goed scorende grappen navertellen, want sommige zijn vrij vulgair. Het verbaasde me hoezeer de computer het seksisme en racisme inherent aan veel moppen oppikte en hoe zo'n 'dom' algoritme dat kon reproduceren. Die domheid is trouwens vaak een bron van plezier. In dit veld geldt soms het principe: *be as smart as a puppy*. Mensen moeten bijvoorbeeld bij een Twitterbot *(software die automatisch tweet – red.)* het gevoel hebben: 'Wat een schattig botje, hij maakt zoveel foutjes.' Vergelijk het met een baby die iets geks zegt. Zo werkt computerhumor vaak het best als het systeem toevallig of onverwacht grappig is."

Herken jij snel automatisch gegenereerde moppen?

Winters: "Normaal wel, omdat ik goed op de hoogte ben van de processen die erachter zitten. De media pakken graag uit met koppen als 'Artificieel intelligente computer schreef nieuw Harry Potter-hoofdstuk', maar dan blijkt er gewoon een *predictive keyboard* in het spel, een systeem dat volgende woorden voorspelt, zoals wanneer je sms-berichten typt. In dat geval maakt een mens nog altijd elke woordkeuze, maar wordt hij daarin ondersteund door suggesties van de computer. Daarna werkt diezelfde mens vaak nog verder op de tekst. Ook als ik goed samenhangende alinea's lees, weet ik dat dat waarschijnlijk niet exclusief het werk was van een computer, want die heeft veel moeite om een topic lang aan te houden."

Is originele humor ook niet iets exclusief menselijks?

Winters: "Hangt ervan af wat je verstaat onder originaliteit. De typetjes van Jacques Vermeire of Urbanus herhalen ook vaak dezelfde trucjes. Een computer zit natuurlijk meestal vast aan een patroon of regels. Zo werkte ik in mijn masterproef met de template 'Ik hou van mijn X zoals ik hou van mijn Y: Z.' Daar kwam dan bijvoorbeeld uit: 'I like my women like I like my laughter: silly.' Op basis van zo'n template genereert mijn systeem heel naïef miljoenen moppen, maar dankzij het *machine learning*-algoritme dat getraind heeft op de beoordeelde mopjes haalt het daar de beste uit. Dat is een originele keuze. Je moet je pc dus wel voeden met data, maar dat geldt ook voor mensen. Als zij bijvoorbeeld rock- of popmuziek maken hebben ze ook eerst goed geluisterd naar dat genre. We zijn allemaal het product van de cultuur om ons heen."

> **Thomas Winters:**
> Het verbaasde me hoezeer de computer het seksisme en racisme dat inherent is aan veel moppen oppikte, en hoe zo'n 'dom' algoritme dat kon reproduceren.



Kunnen we binnenkort dan de menselijke comedians afschaffen?

Winters: "Nee. Denk aan het liedje *Video Killed the Radio Star*. In tegenstelling tot wat dat voorspelde, luisteren we nog altijd naar de radio. Mensen in mijn veld willen creatievelingen niet vervangen, integendeel zelfs. De beste resultaten krijg je als mens en machine samenwerken. In Photoshop kan ik beelden bewerken met gereedschappen die werken op basis van artificiële intelligentie, maar ik ben het die beslist over waar het programma die moet toepassen. Stel dat een comedian in zijn set nog dit of dat soort mopje mist, dan zou hij met behulp van een computer inspiratie kunnen opdoen. Op een vergelijkbare manier zou een programma voorbereidend werk kunnen overnemen van marketingmensen. Als ze alle evidente slogans of bedrijfsnamen laten genereren door een computer, besparen ze zichzelf heel wat werk."

Voor wie is dit nog interessant?

Winters: "Voor al wie zoals ik gepassioneerd is door humortheorie. Al op heel jonge leeftijd vroeg ik me af: 'Waarom lachen wij?' Niemand kon mij antwoord geven, dus ging ik me verdiepen in cognitieve en andere processen. Dat ik die theorieën nu kan verifiëren is heel leerrijk. Zo bleek uit mijn experiment met de analogie-moppen dat het belangrijk is dat het laatste woord iets absurds heeft, of een taboe-connotatie. Dat stemt overeen met bestaande theorieën daarover. Zo heb ik al heel wat regeltjes kunnen verzamelen voor wat grappig is en dat gebruik ik dan weer in de humorworkshops die ik geef. Ik doe dat bij Preparee, het Studenten Improvisatietheater waarbij ik al vier jaar speel, en binnenkort ook bij *Veto*."

Voor dat studentenblad maakte je de TorfsBot en de Theo Francken-simulator. Een vorm van kritiek?

Winters: "Absoluut. Door Francken de held van zijn eigen videogame te maken doe ik aan satire, en ook met de Torfsbot kaart ik iets aan. Die Twitterbot imiteert volautomatisch onze ex-rector en zo zie je dat zelfs een algoritme zijn semi-filosofische uitspraken kan bedenken. Torfs zelf vond het wel geestig."

Dat blijft vrij onschuldig, maar als Twitterbots alsmaar betrouwbaarder lijken, lopen we dan niet het risico op nog meer *fake news*?

Winters: "Dat is een reële problematiek, waartegen Twitter probeert terug te vechten. Het bedrijf wordt strenger in zijn *do's-and-don'ts*, waardoor onder meer de fameuze Russische *trolls* uitgeschakeld worden. Maar we moeten alert blijven. Ik probeer in elk geval *superduidelijk* te zijn. Bij elke bot die ik maak, geef ik aan dat het een robot of algoritme is. Toen Google onlangs demonstreerde hoe hun virtuele assistent een telefonische afspraak bij de kapper kon maken, was dat indrukwekkend, maar het was niet oké dat ze niet tegen de kapster zeiden dat ze aan het babbelen was met een computer. Naar zo'n samenleving wil je niet gaan."

Desondanks zie je vooral de kansen die computationele humor biedt?



Winters: "Ja, ik zie tal van nuttige toepassingen. Zo droom ik ervan om op een dag mee te bouwen aan de humor van virtuele assistenten zoals Alexa of Google Assistant. Als die minder voorgedefinieerd en meer persoonlijk wordt, bijvoorbeeld op basis van de data over je vrienden en de plaatsen die je bezoekt, dan kan de gsm meer een vriend dan een *tool* worden."

"Ik vermoed dat ik daarvoor naar het buitenland zal moeten, want in België zijn er weinig bedrijven die hiermee bezig zijn. Ik hoop hier wel mijn doctoraat te maken, na de manama artificiële intelligentie die ik nu aan het volgen ben. Computers en creativiteit linken heb ik altijd al *superinteressant* gevonden. Toen ik negen was, bouwde ik mijn eerste website, op mijn tiende maakte ik al spelletjes in Flash en daarna heb ik massa's computermuziek gemaakt."

> Een computer zit natuurlijk meestal vast aan een patroon of regels, maar de typetjes van Jacques Vermeire of Urbanus herhalen ook vaak dezelfde trucjes.

Des te fijner dat je binnenkort over je masterproef mag spreken in Las Vegas?

Winters: "Ja. Ik ben uitgenodigd voor HCII 2018, de Human Computer Interaction International-conventie in Caesars Palace. Heel spannend. Ik kijk er vooral naar uit om de gelijkgestemden te ontmoeten van wie ik alle papers heb gelezen, maar die ik nog nooit live gezien heb. Misschien kunnen zij mij ook vooruithelpen. Dit is een nicheveld, maar één met enorm veel potentieel. Dat wil ik absoluut mee verkennen."

Zegt de ene bot

Nog enkele mopper
de computer van Th
Winters op de propp

I like my men lik my graves: nam

I like my coffee l my country: colo

I like my women like my building

I like my sex like my emotions: ba

I like my men lik my society: hear

Een typische tweet
de **TorfsBot**:

Robot
@TorfsBo

Verontrustend.
meer Vlaminge
aan hoogbegaa
Precies daarom
ze er in die mee
te verbergen.

12:02 - 29 apr. 2018

Winters maakte ook
de spreekstijl van p
uit **Samson** imitere

De Burg
@Burgerme

Aheuum. Aheu
Aheum.
Aan allen die lu
overwinnen: pr
Aan allen die w
overwinnen: oc
proficiat.

04:00 - 11 jun. 2018

Streamers

Prepare good quotes yourself
& ask to **check** afterwards!

> Thomas Winters:
> Het verbaasde me hoezeer de computer het seksisme en racisme dat inherent is aan veel moppen oppikte, en hoe zo'n 'dom' algoritme dat kon reproduceren.

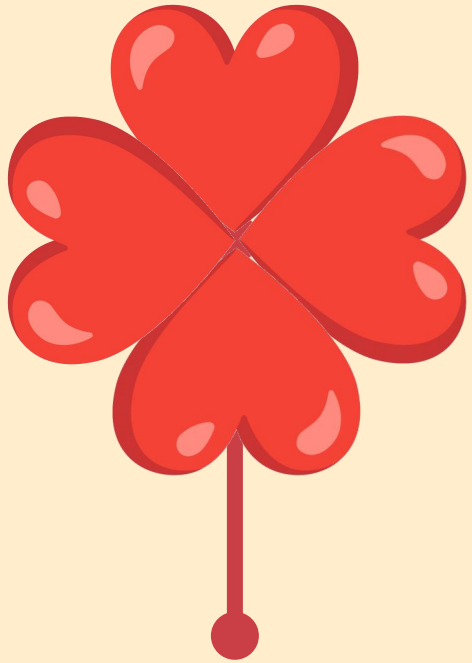


> Een computer zit natuurlijk meestal vast aan een patroon of regels, maar de typetjes van Jacques Vermeire of Urbanus herhalen ook vaak dezelfde trucjes.

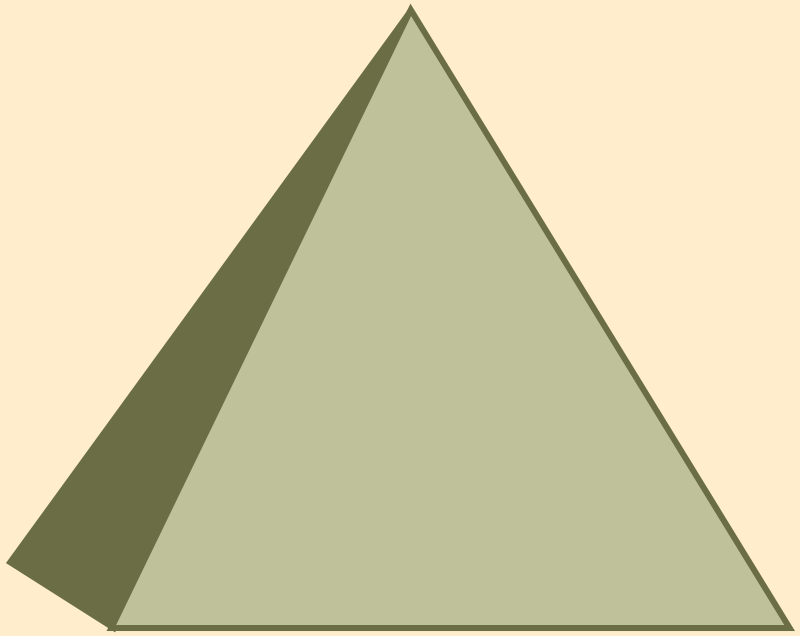
Main points



Take your branding serious



Proactive serendipity



Simplify

How to *Attract* & Survive Media Attention

by Thomas Winters
@thomas_wint

Slides: thomaswinters.be/talk/2022microsoft