Efficient Hierarchical Domain Adaptation for Pretrained Language Models

Alexandra Chronopoulou, Matthew E. Peters, Jesse Dodge

NAACL 2022

Presentation outline

- Motivation
- Proposed Approach
- Experiments
  - Few-domain setting
  - Many-domain setting
- Recap

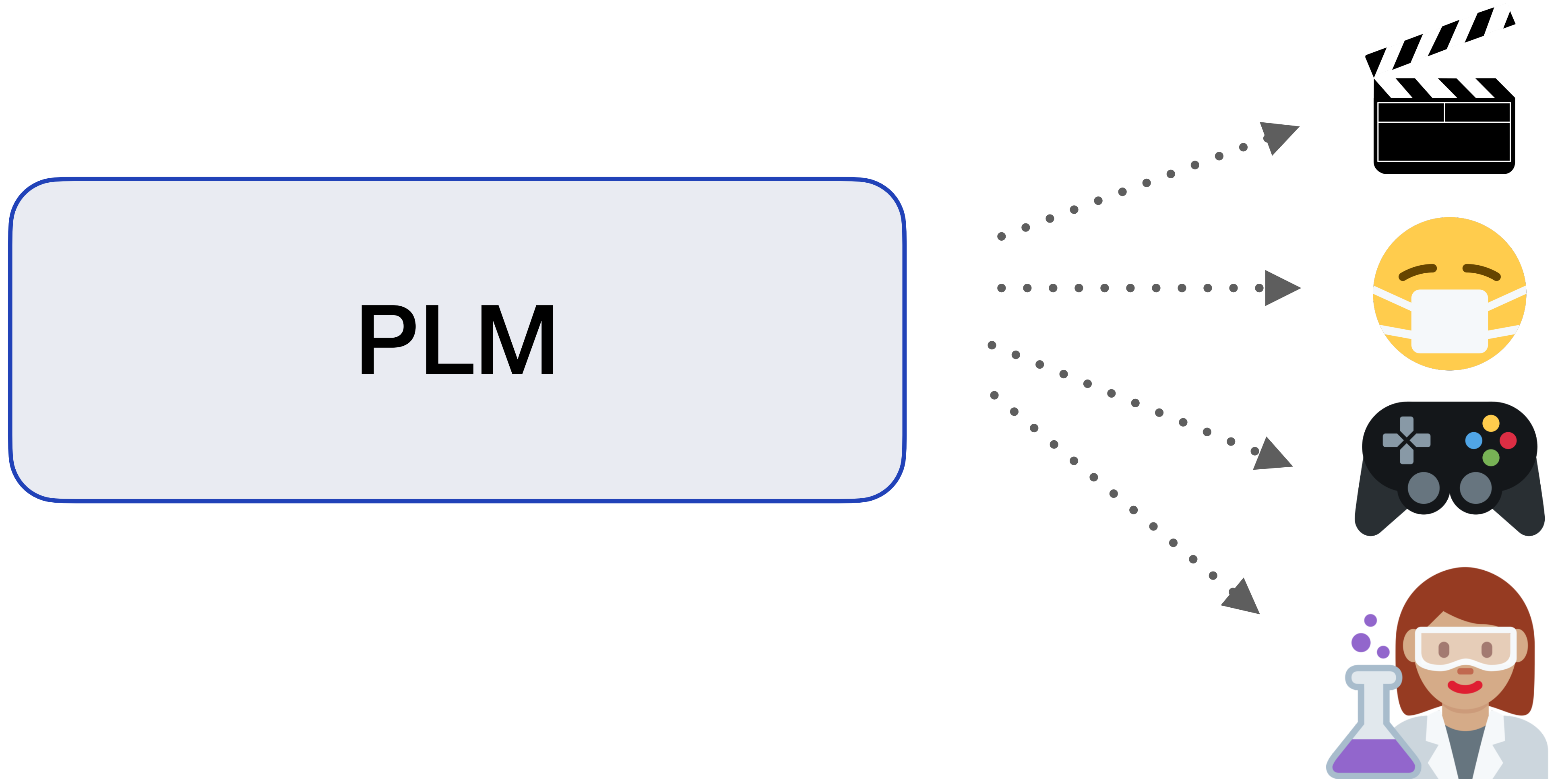
- **Motivation**
- Proposed Approach
- Experiments
  - Few-domain setting
  - Many-domain setting
- Recap

Domain adaptation of PLMs

L parameters — pre-training → PLM

Domain adaptation of PLMs

How can a PLM adapt to a new domain?



*train **all** parameters on domain Di*

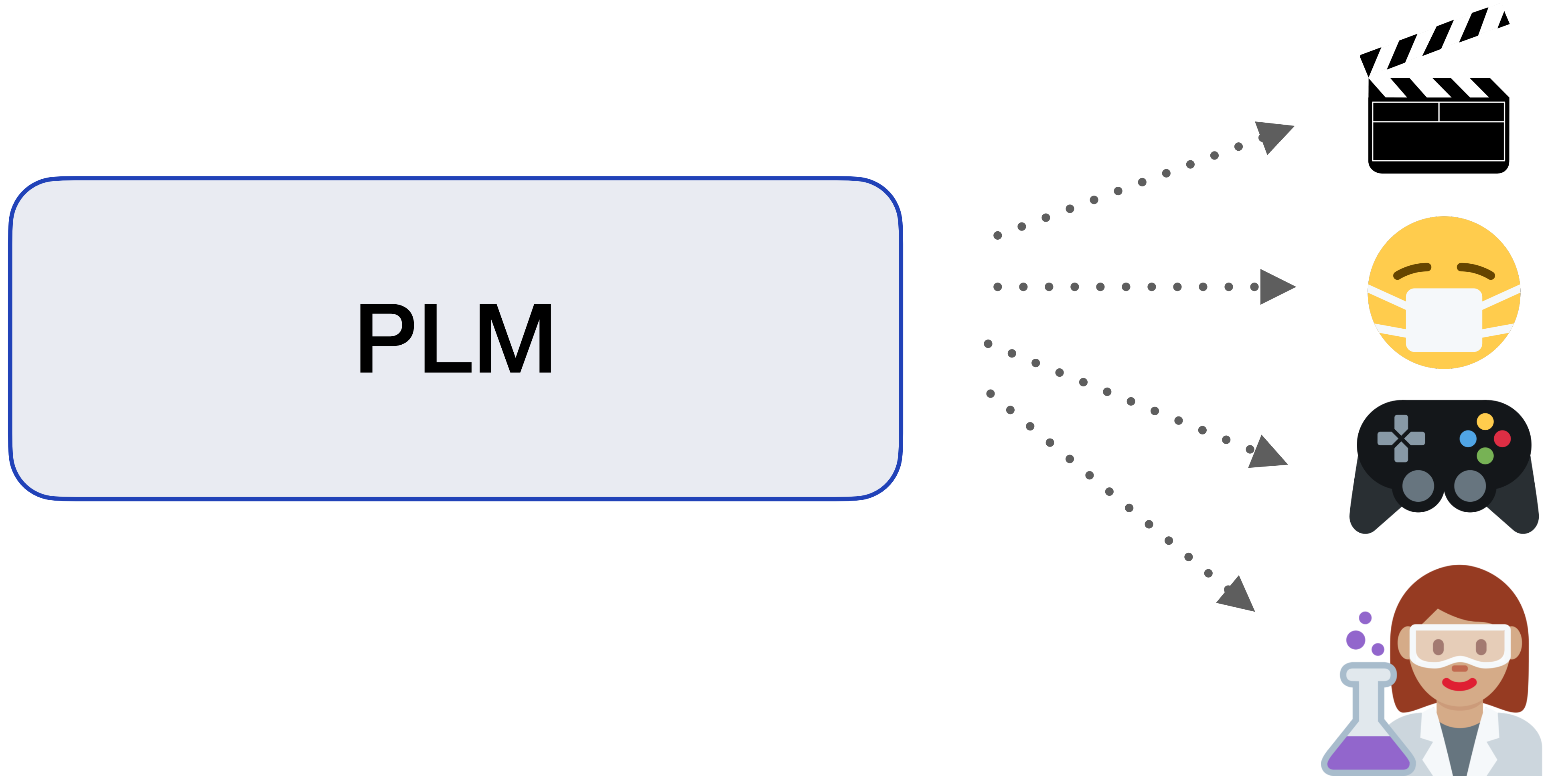
*train **all** parameters on domain Dj*

*train **all** parameters on domain Dk*

*train **all** parameters on domain Dl*

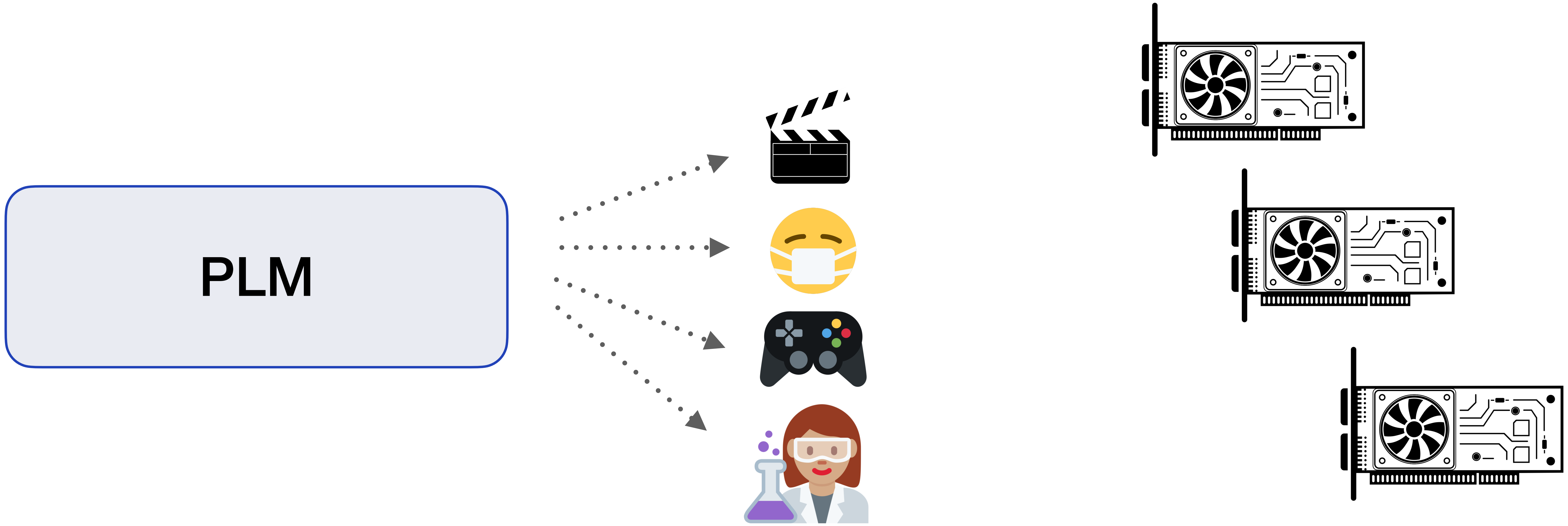
Domain adaptation of PLMs

Why is this problematic?



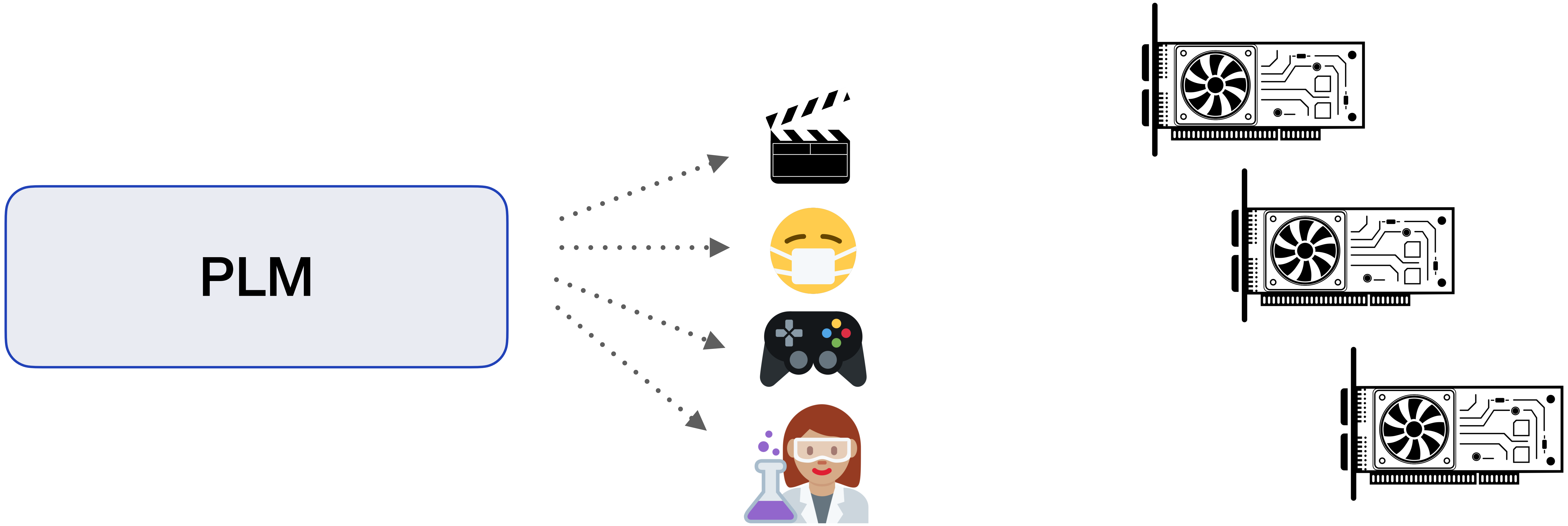
Domain adaptation of PLMs

Why is this problematic?



Domain adaptation of PLMs

Why is this problematic?



Computationally expensive!

- Han and Eisenstein: Unsupervised Domain Adaptation of Contextualized Embeddings for Sequence Labeling, EMNLP 2019.
- Gururangan et al.: Don’t Stop Pretraining: Adapt Language Models to Domains and Tasks, ACL 2020.
- Maronikolakis and Schutze: Multidomain Pretrained Language Models for Green NLP, AdaptNLP 2021.

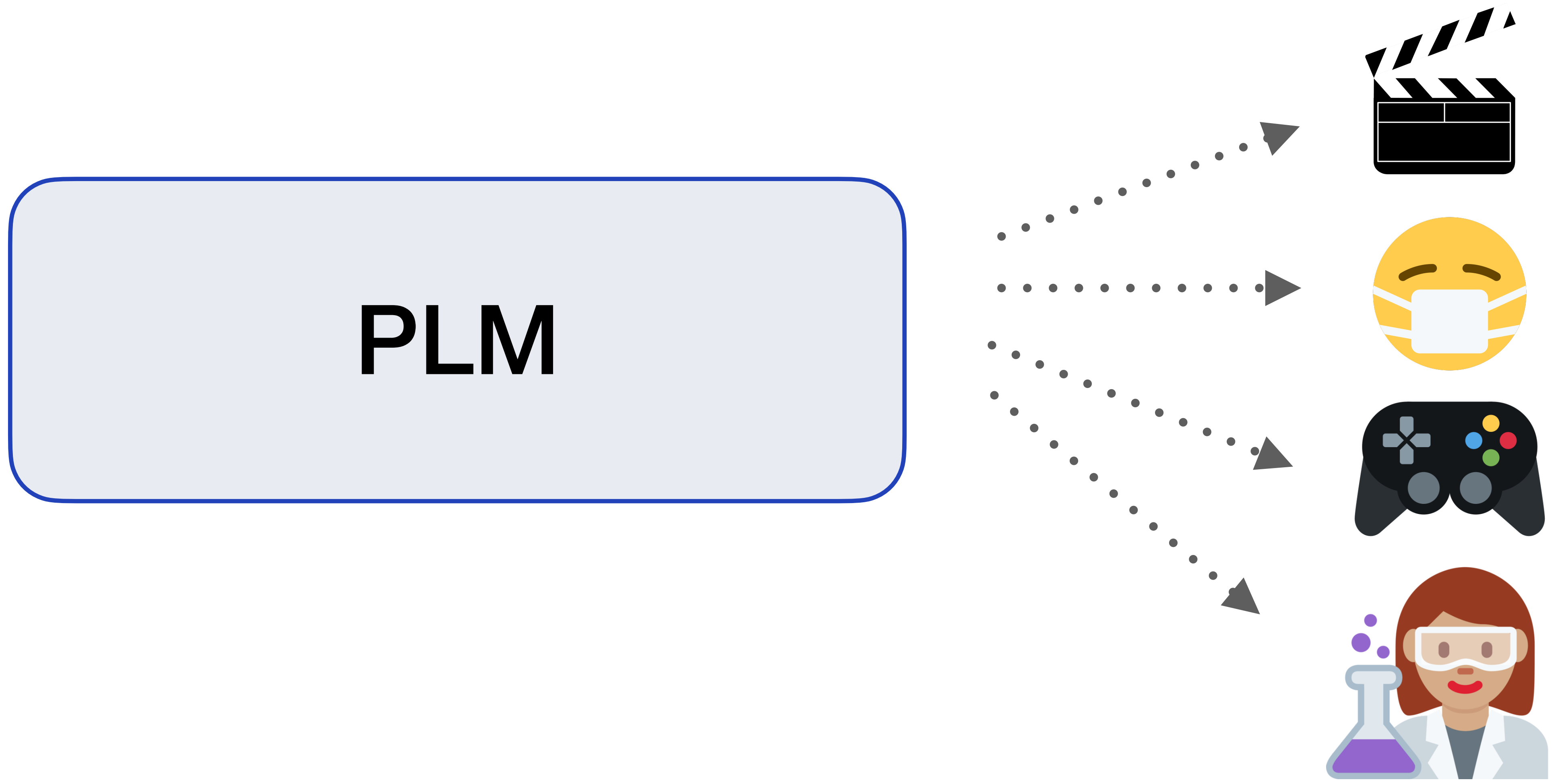
Domain adaptation of PLMs

Efficient alternative: instead of fine-tuning all the layers of the PLM, add modular components (like mixture-of-experts) to model specific domains

- Lepihkin et al.: GShard: Scaling Giant Models with Conditional Computation and Automatic Sharding, ICLR 2021
- Gururangan et al.: DEMix Layers: Disentangling Domains for Modular Language Modeling, NAACL 2022

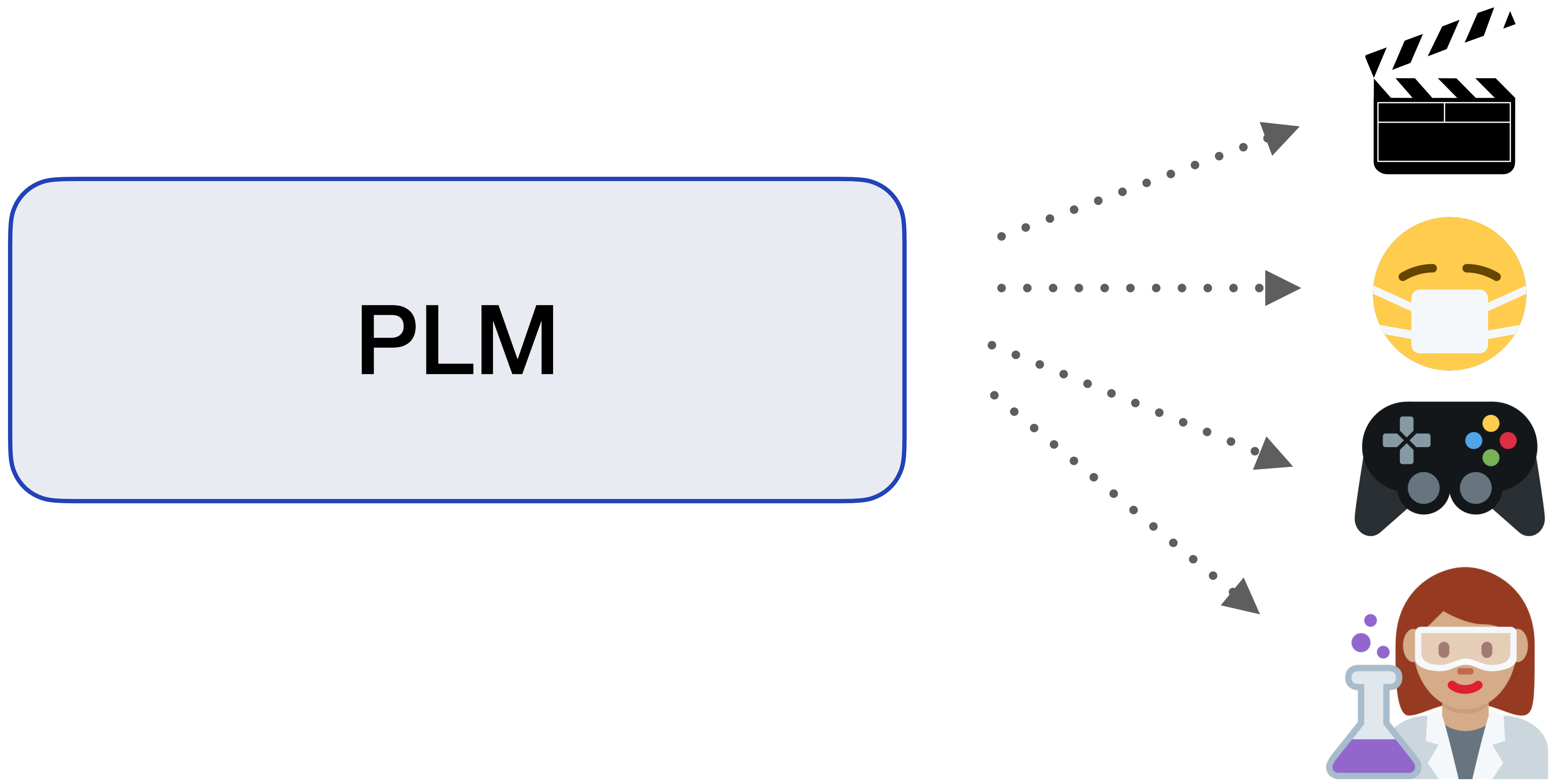
Domain adaptation of PLMs

Does this solve the problem?

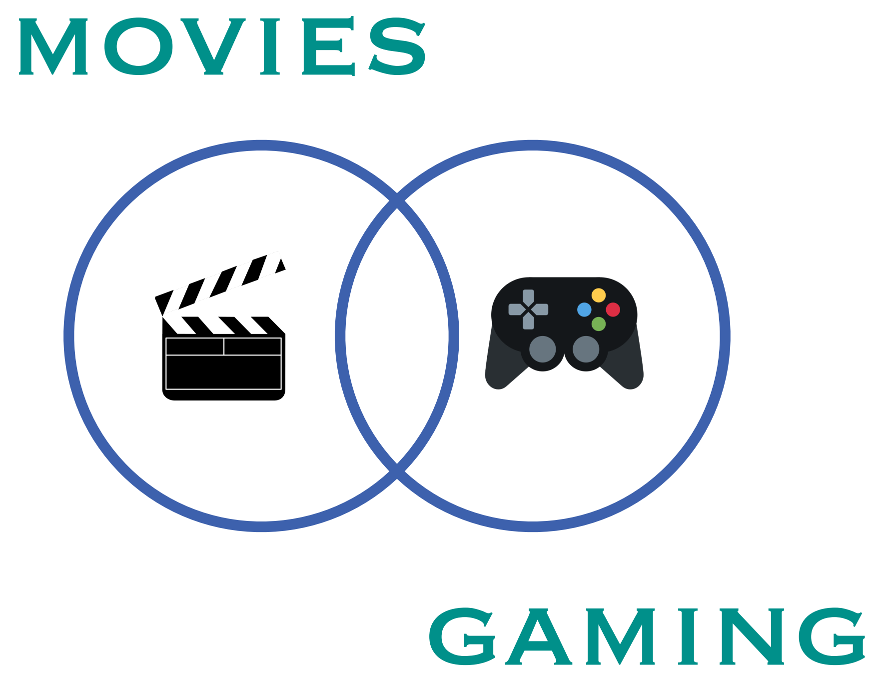


Domain adaptation of PLMs

Does this solve the problem?

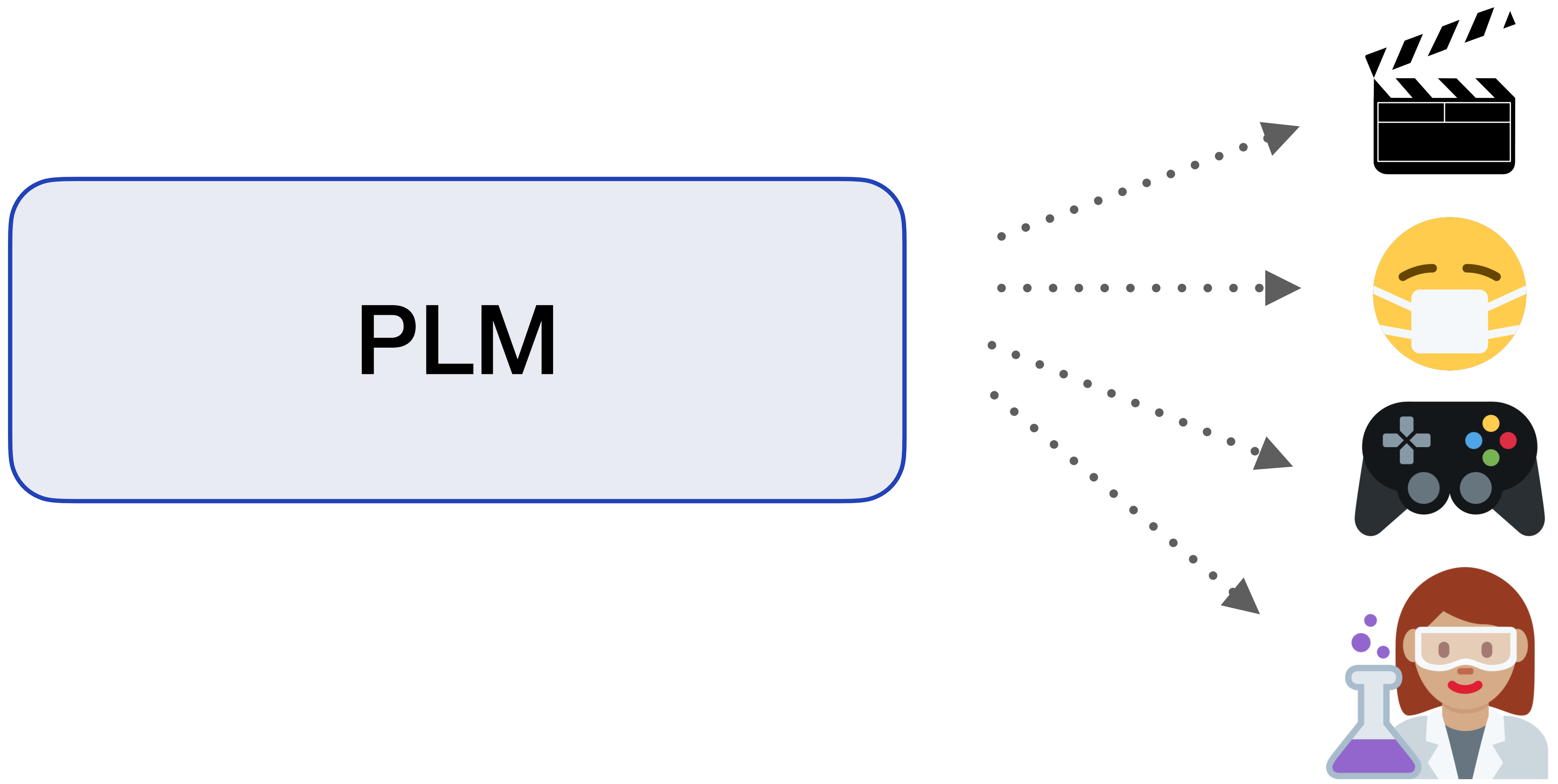


- Ignores **overlap** between domains

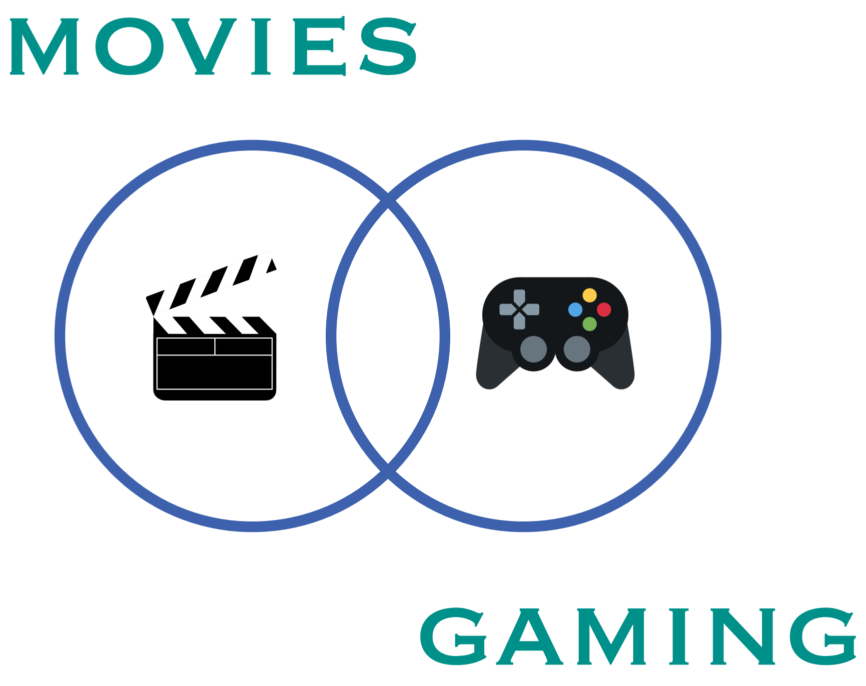


Domain adaptation of PLMs

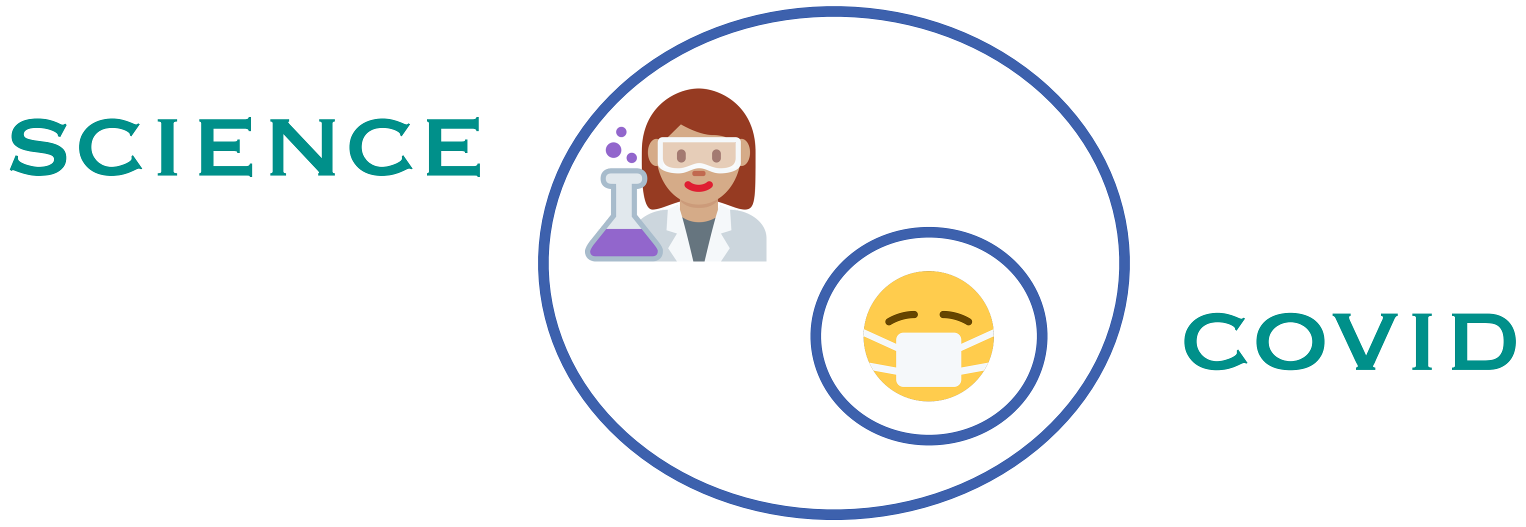
Does this solve the problem?



- Ignores **overlap** between domains



- Ignores **granularities** of domains



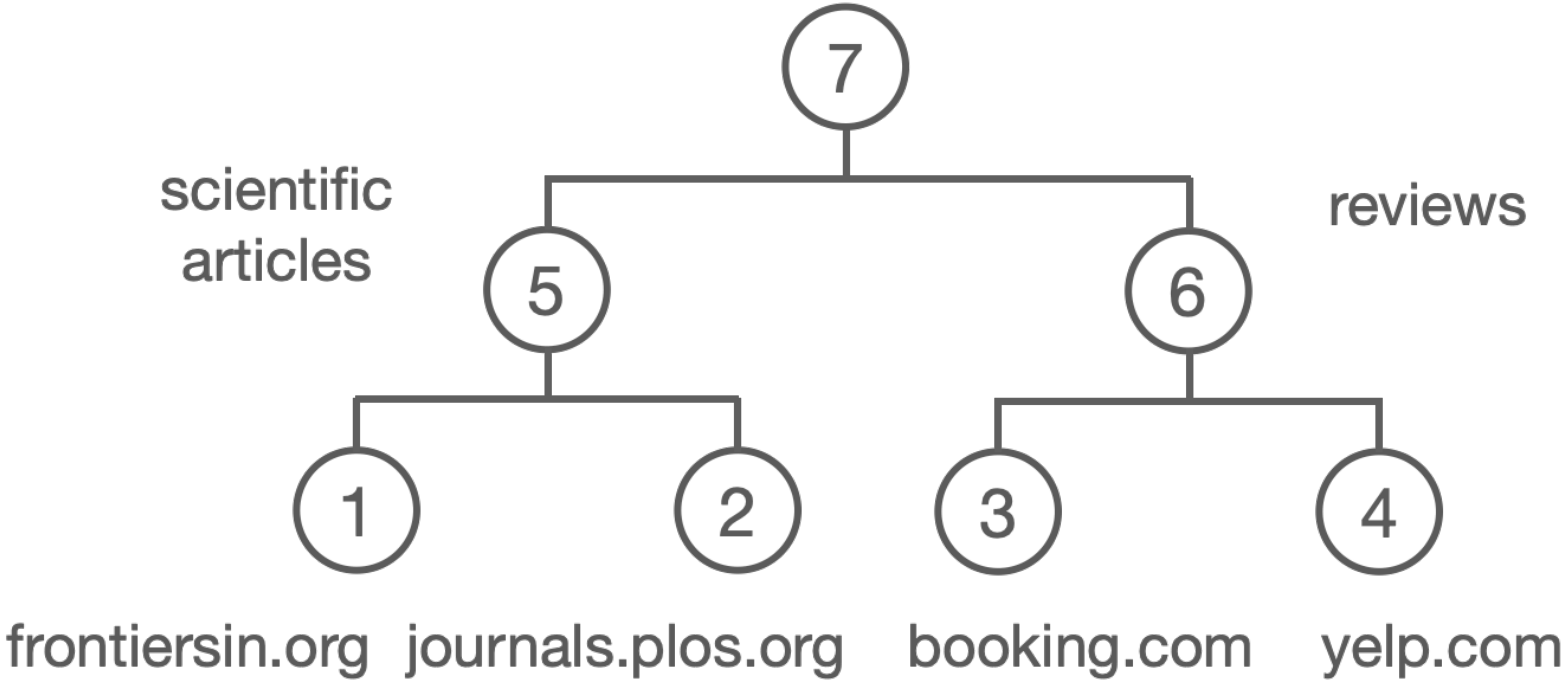
- Motivation
- **Proposed Approach**
- Experiments
  - Few-domain setting
  - Many-domain setting
- Recap

Hierarchical representation of domains

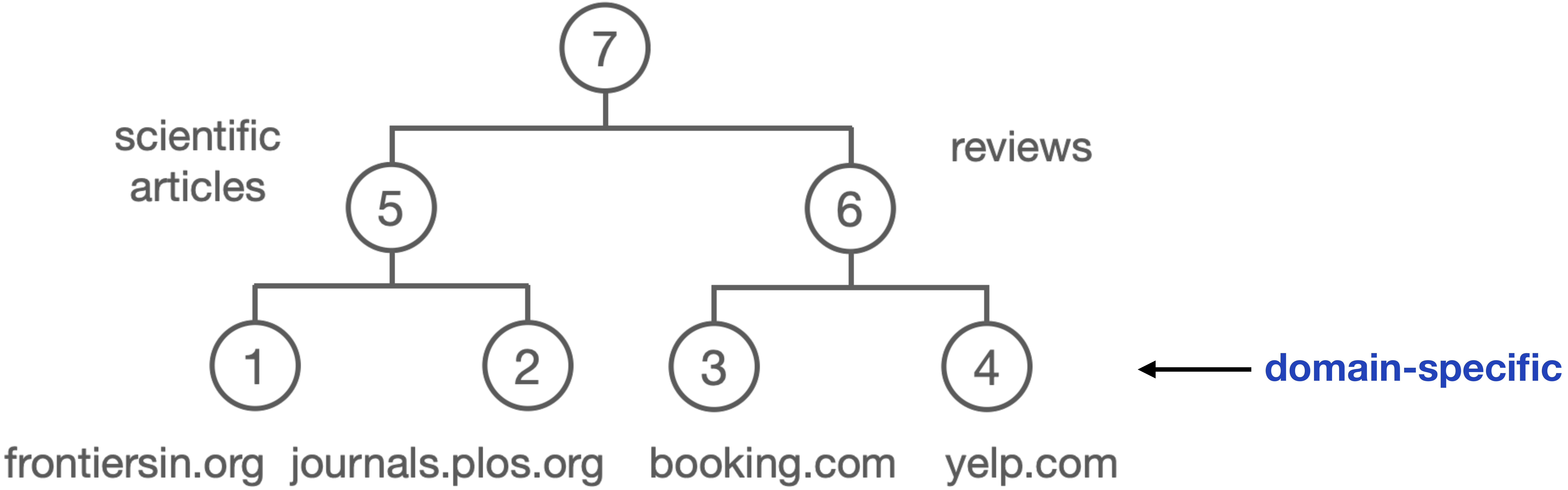
Idea: We represent domains with a **hierarchical** structure.

Hierarchical representation of domains

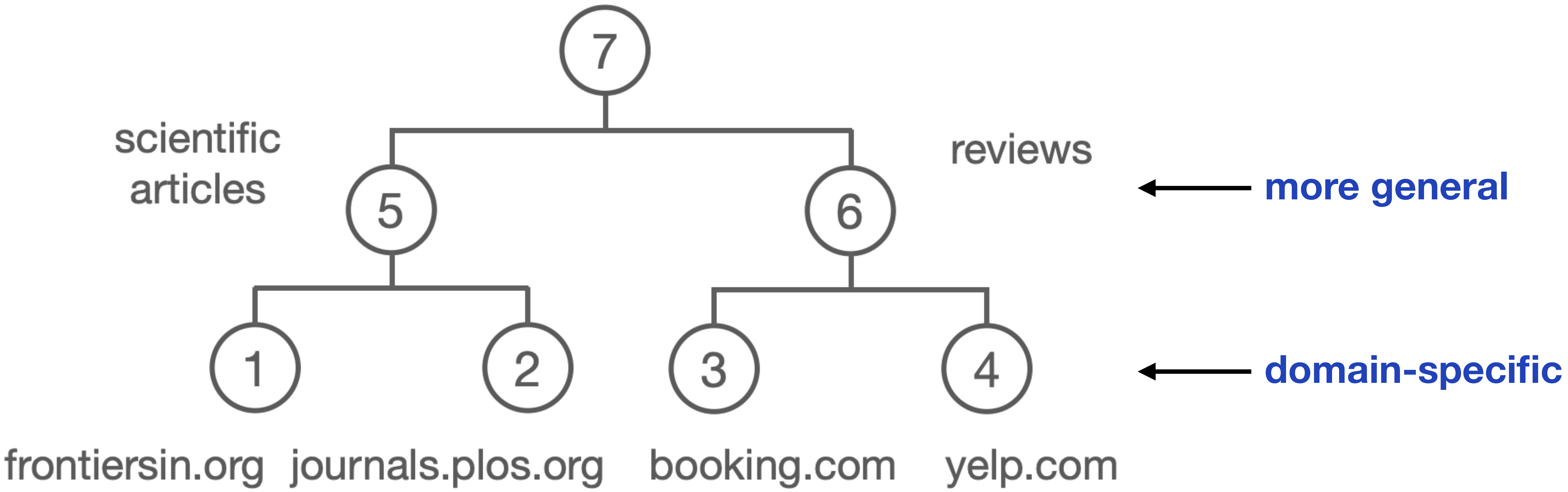
Idea: We represent domains with a **hierarchical** structure.



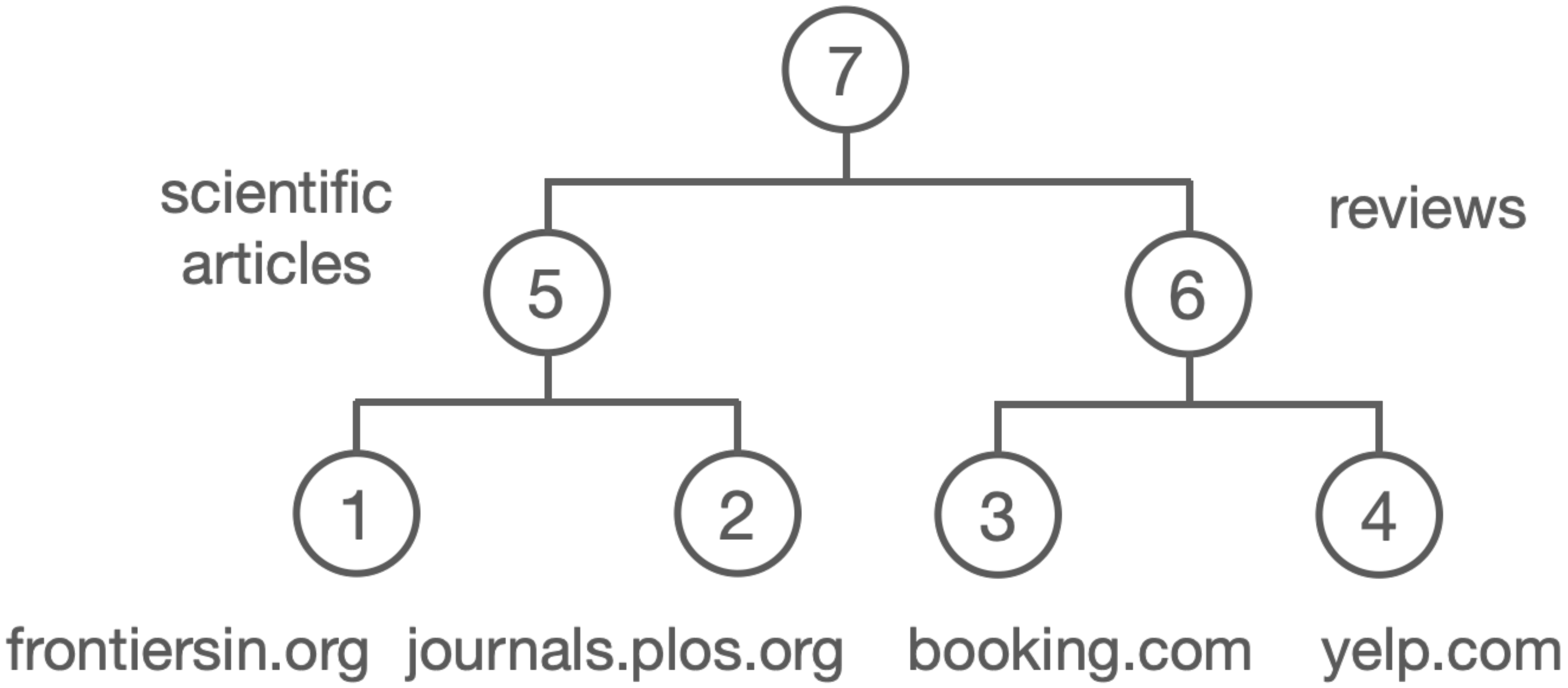
Hierarchical representation of domains



Hierarchical representation of domains



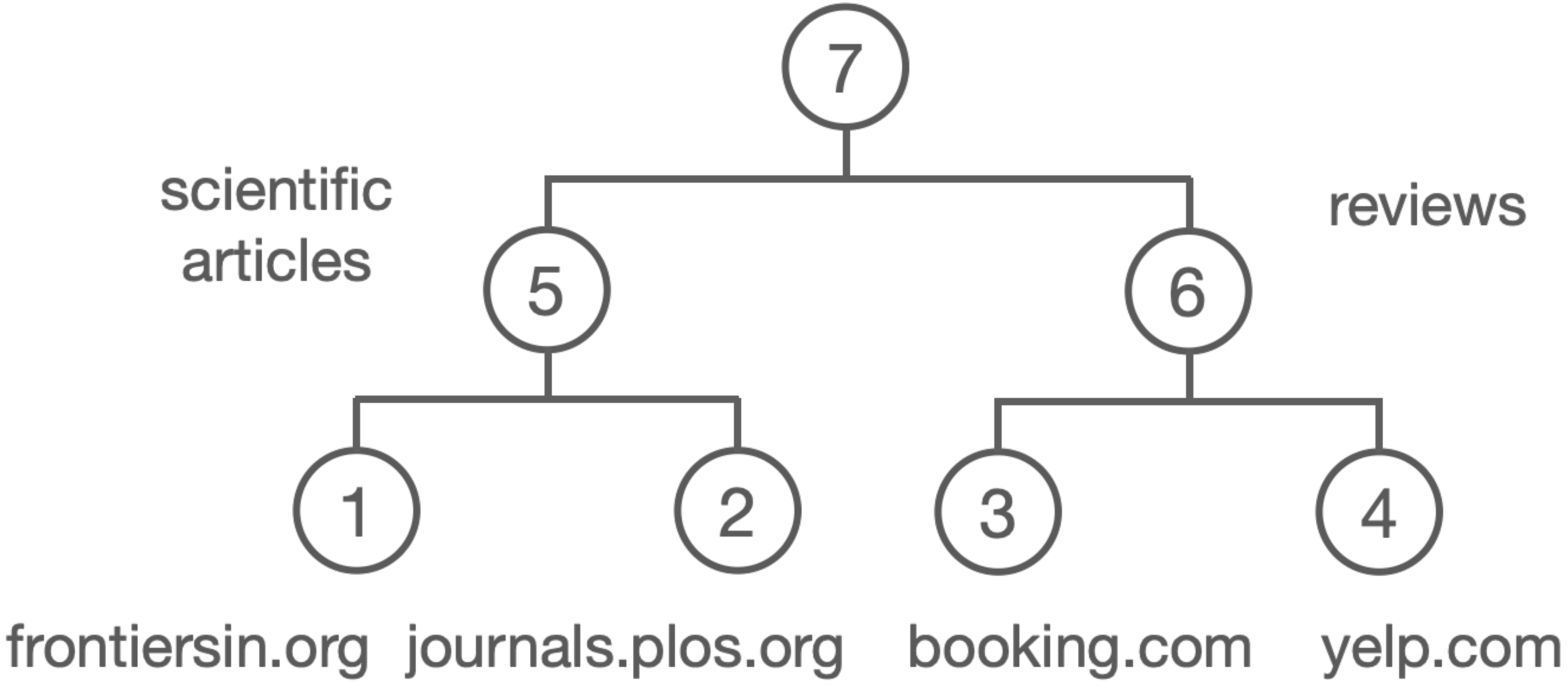
Hierarchical representation of domains



Our approach:

- **Automatically** clusters domains in a tree using PLM representations
- Specializes a PLM in *N* domains **efficiently**
- **Combines multiple paths** at inference time

Hierarchical representation of domains

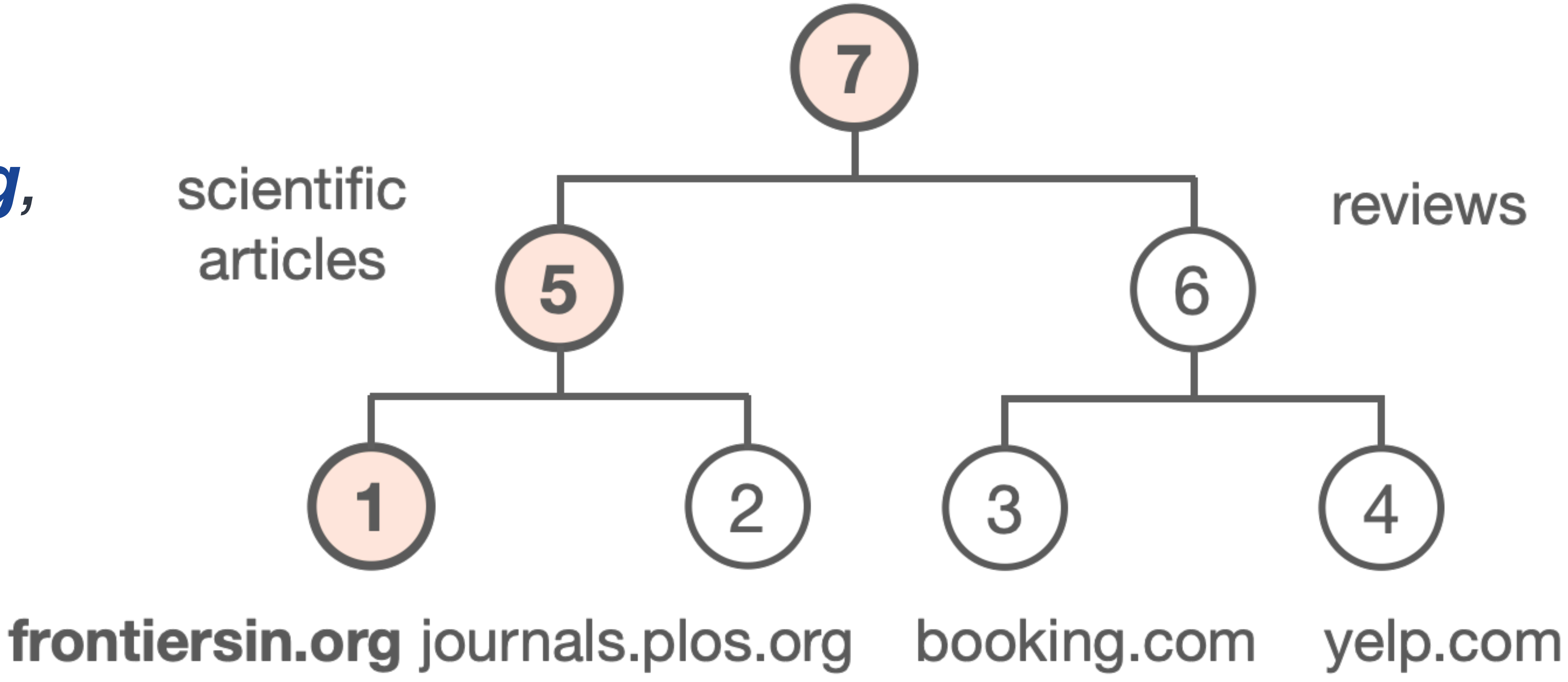


- We add this hierarchical structure to a frozen PLM
- Each node is an **adapter layer**

Hierarchical representation of domains

Training

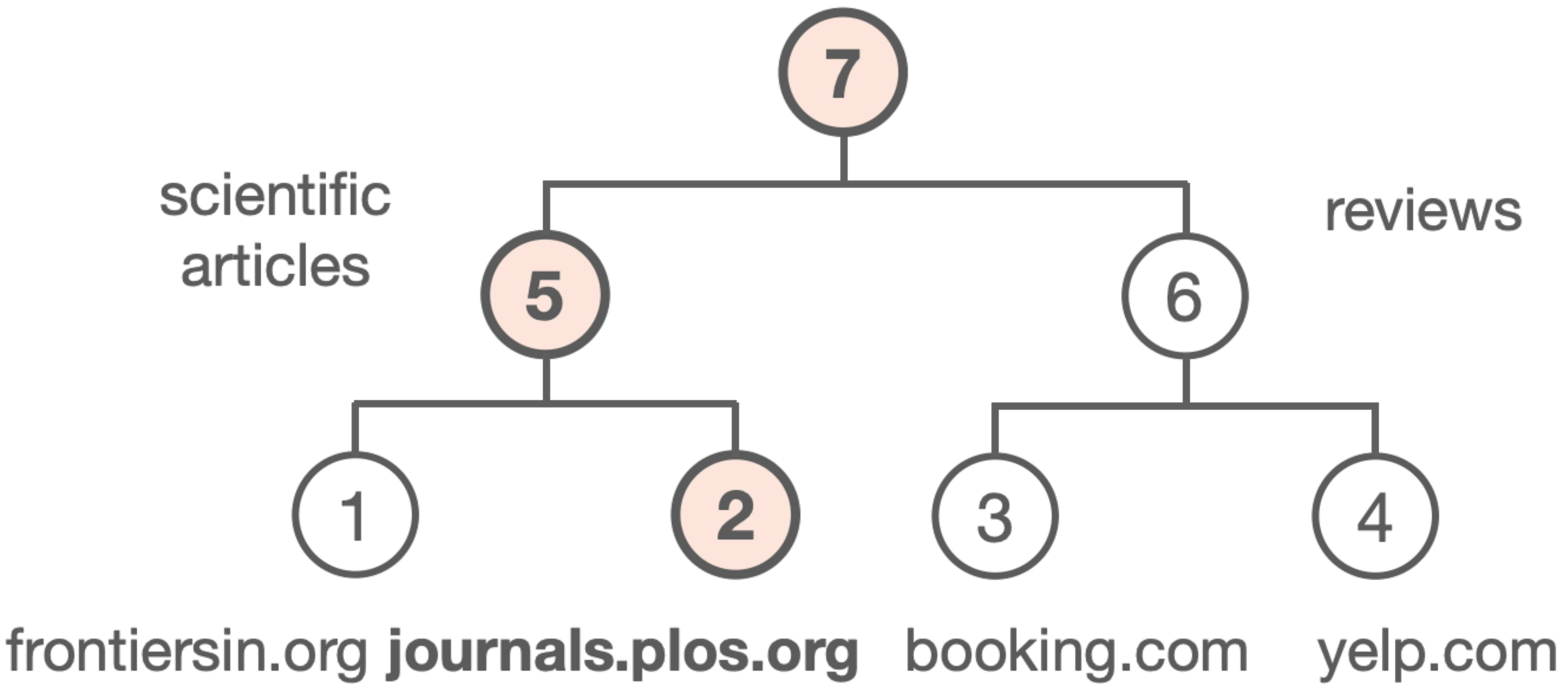
- Mini-batch from ***frontiersin.org***, (representation h_i)
- h_i is input to adapters **1, 5, 7**
- Outputs are averaged and passed to next layer



Hierarchical representation of domains

Training

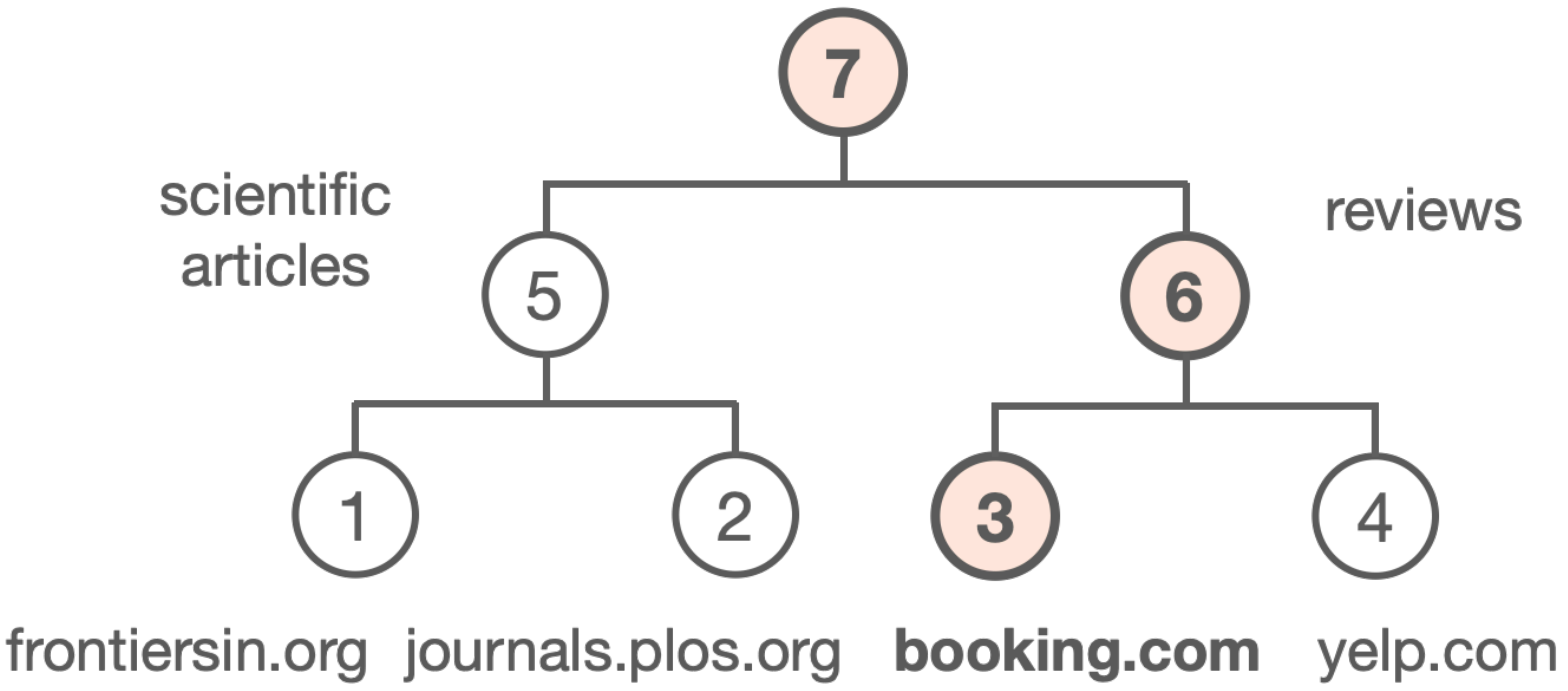
- Mini-batch from ***journals***, (representation h_i)
- h_i is input to adapters **2, 5, 7**
- Outputs are averaged and passed to next layer



Hierarchical representation of domains

Training

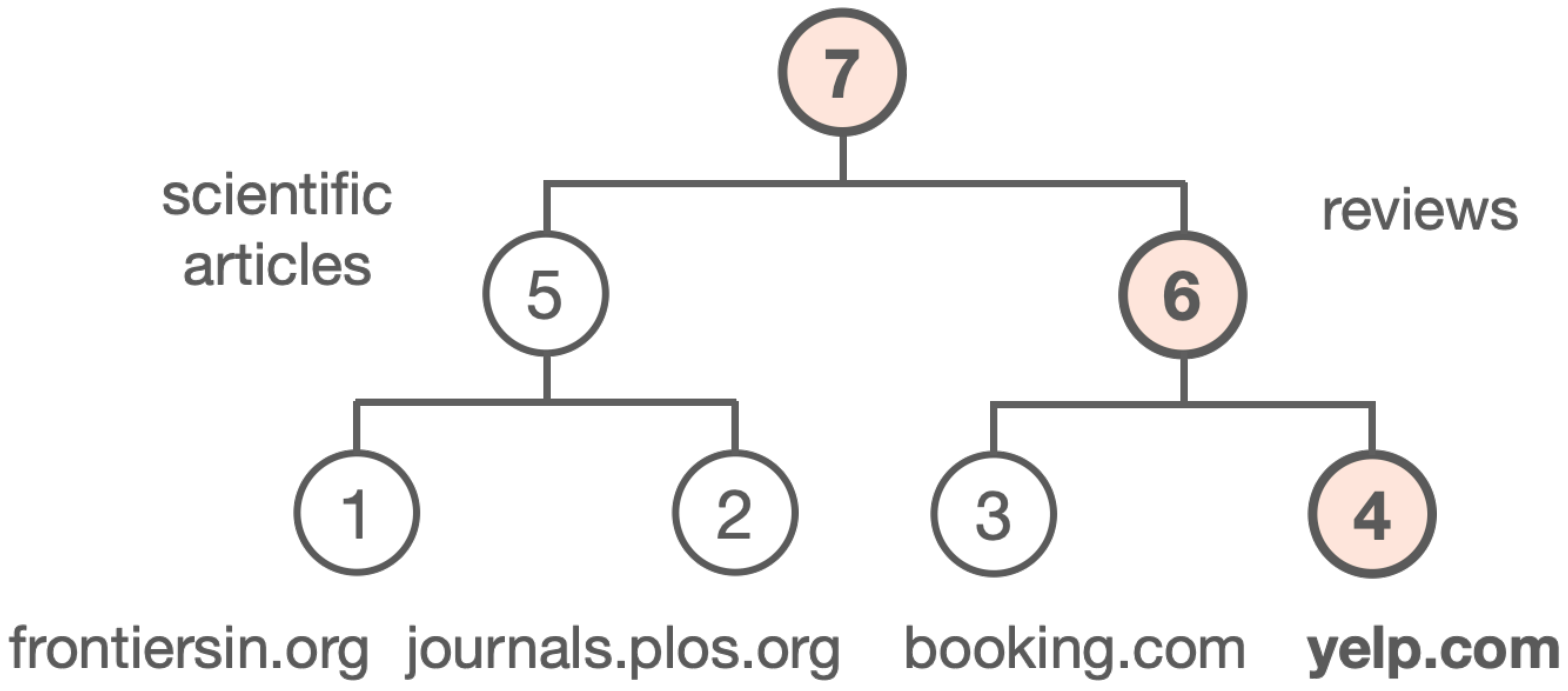
- Mini-batch from ***booking.com***, (representation h_i)
- h_i is input to adapters **3, 6, 7**
- Outputs are averaged and passed to next layer



Hierarchical representation of domains

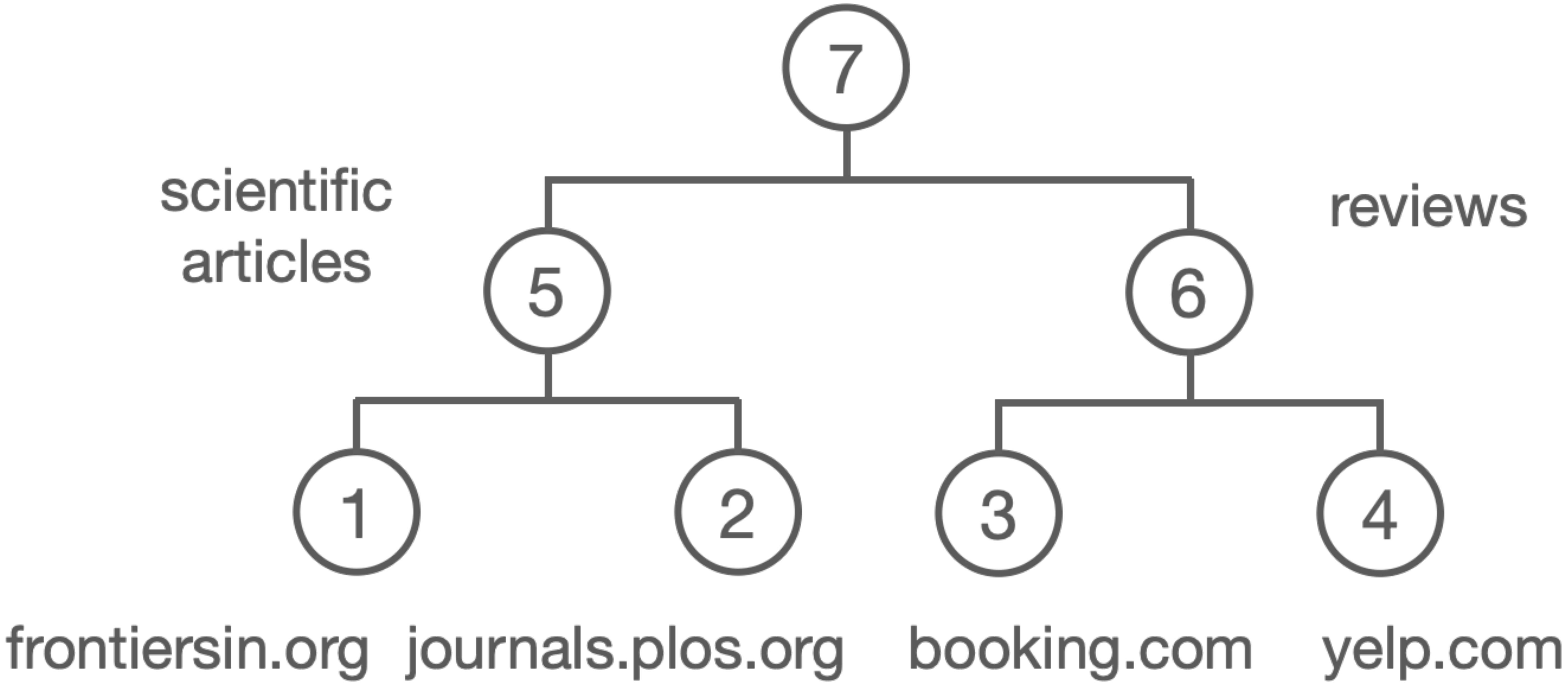
Training

- Mini-batch from ***yelp.com***, (representation h_i)
- h_i is input to adapters **4, 6, 7**
- Outputs are averaged and passed to next layer



Hierarchical representation of domains

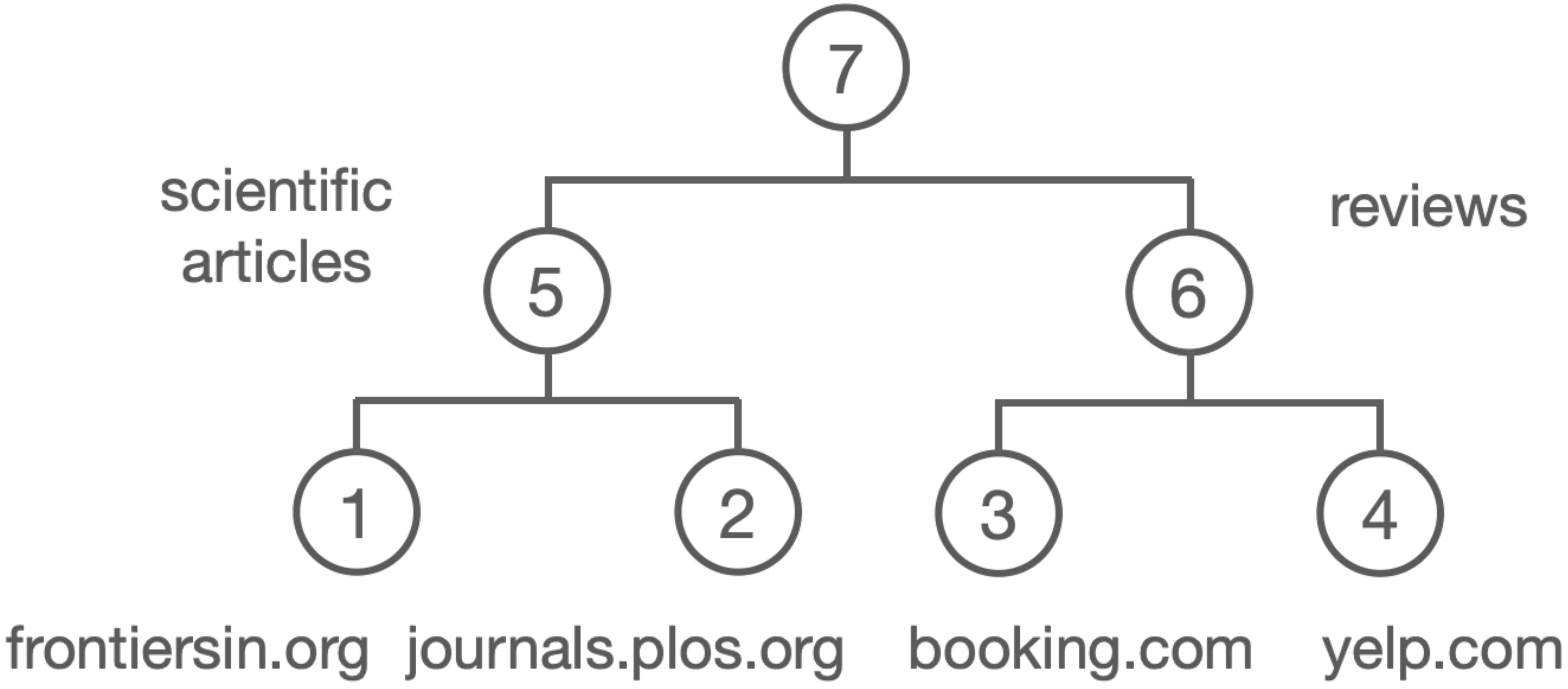
Evaluation (which path?)



Hierarchical representation of domains

Evaluation (which path?)

- **In-domain**

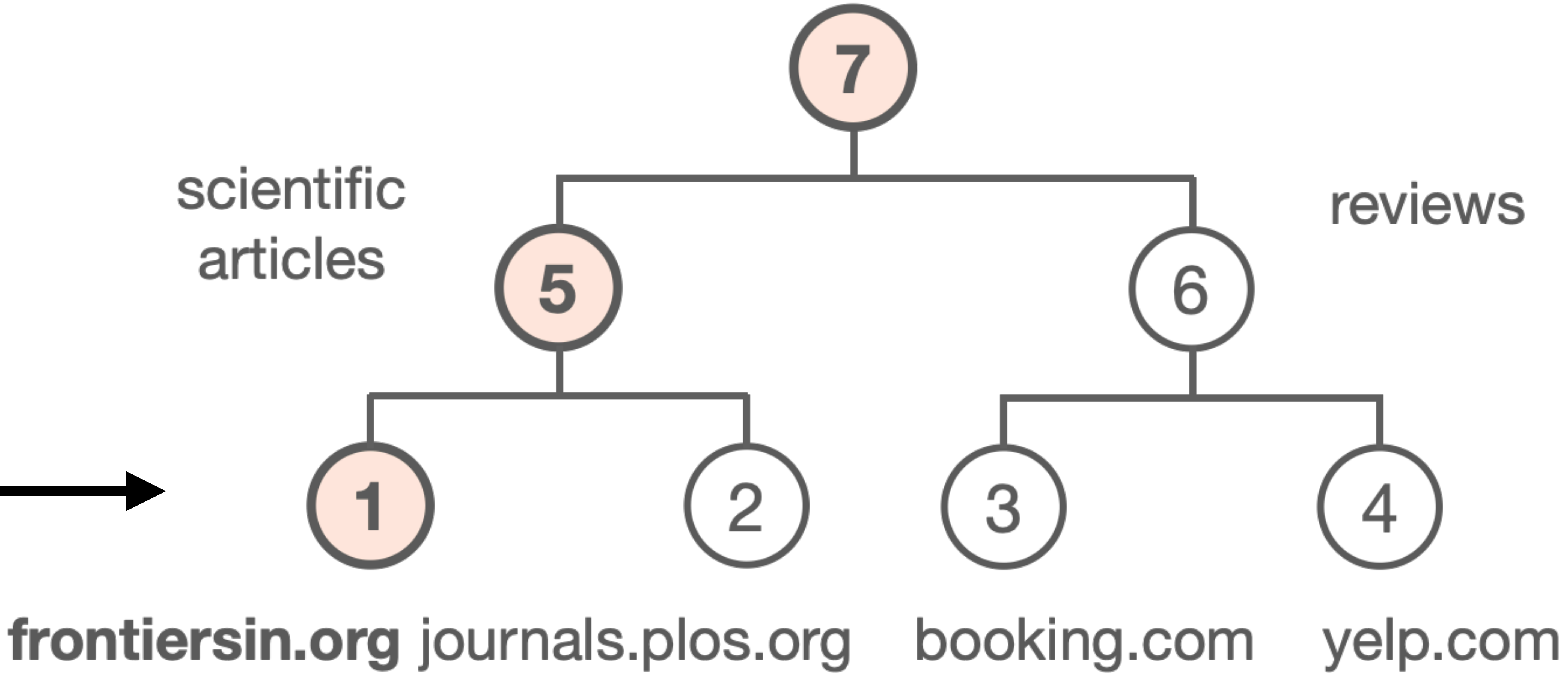


Hierarchical representation of domains

Evaluation (which path?)

- **In-domain**
  Same as training

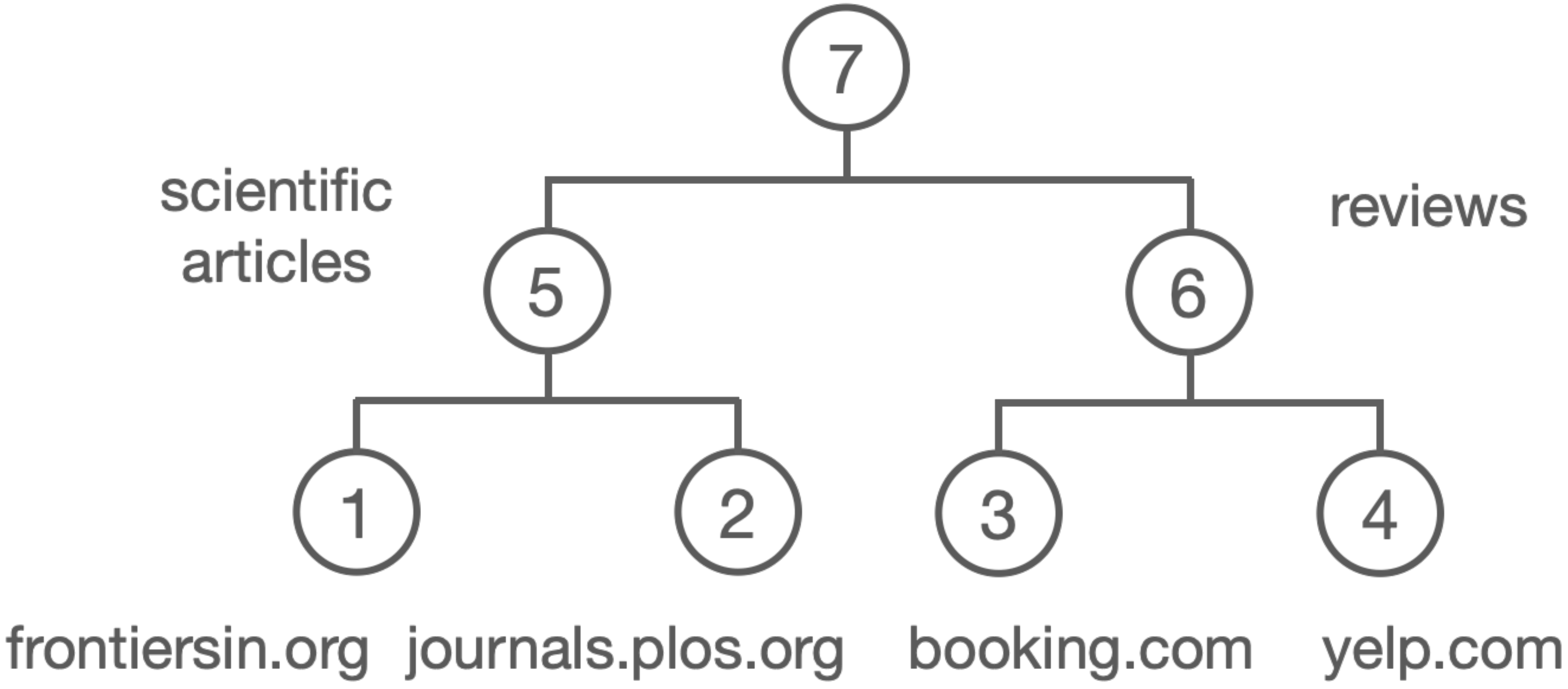
for ***frontiersin.org***, the path that leads to node assigned to this domain



Hierarchical representation of domains

Evaluation (which path?)

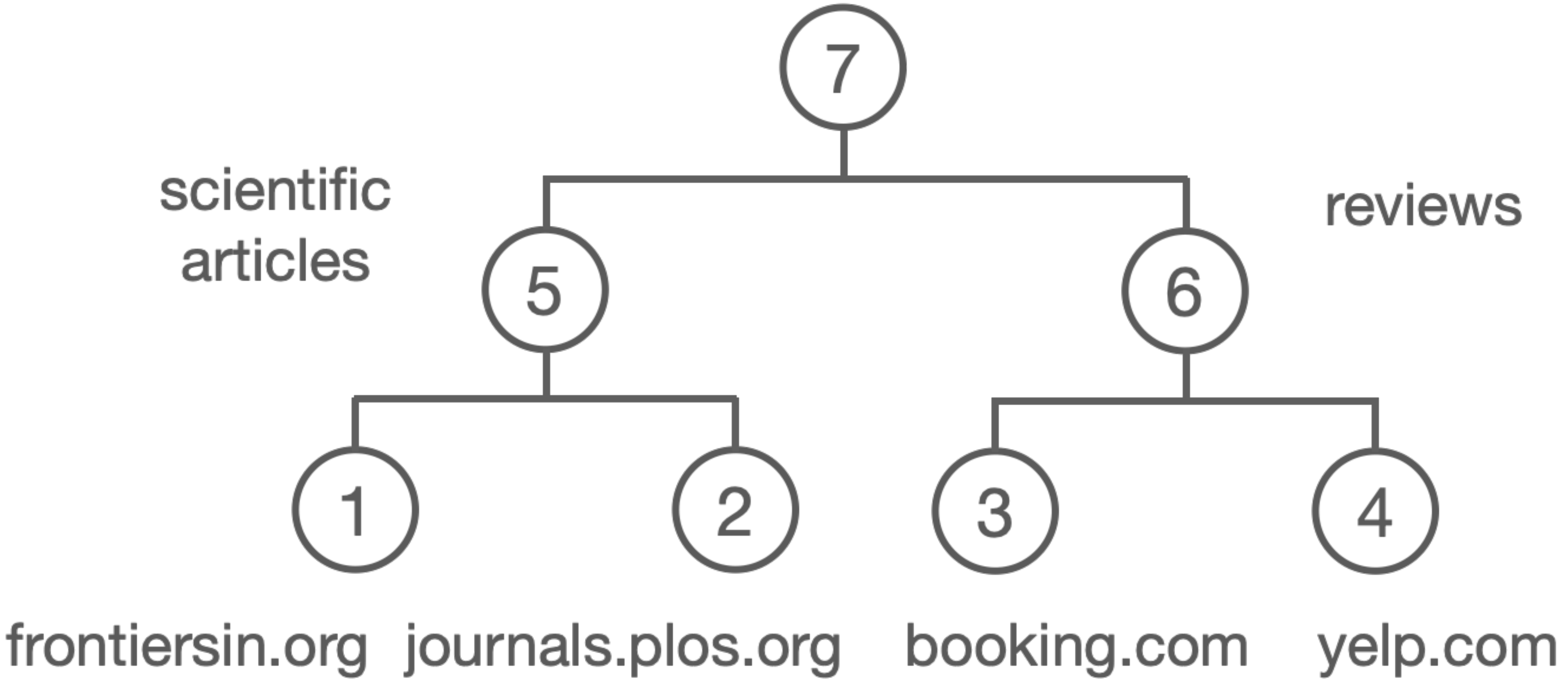
- **Out-of-domain**



Hierarchical representation of domains

Evaluation (which path?)

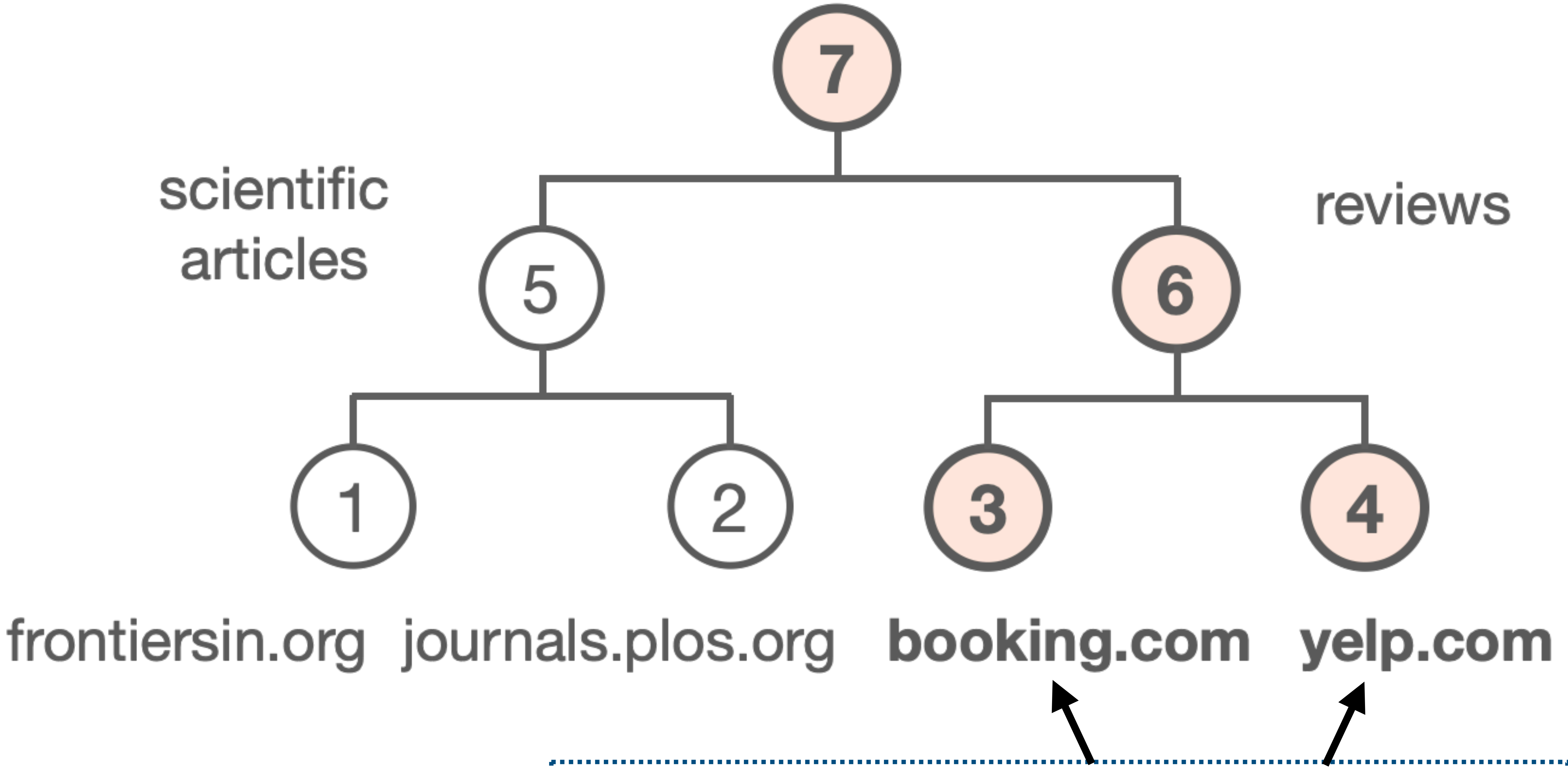
- **Out-of-domain**
  Not straightforward, choose based on domain similarity



Hierarchical representation of domains

Evaluation (which path?)

- **Out-of-domain**
  Not straightforward, choose based on domain similarity



for *tripadvisor.com*,
the path that leads to 3 (and / or 4)

- Motivation
- Proposed Approach
- **Experiments**
  - Few-domain setting
  - Many-domain setting
- Recap

Experiments

- **Few-domain setting: manually created tree**
- **Many-domain setting: automatically created tree**

What do we compare?

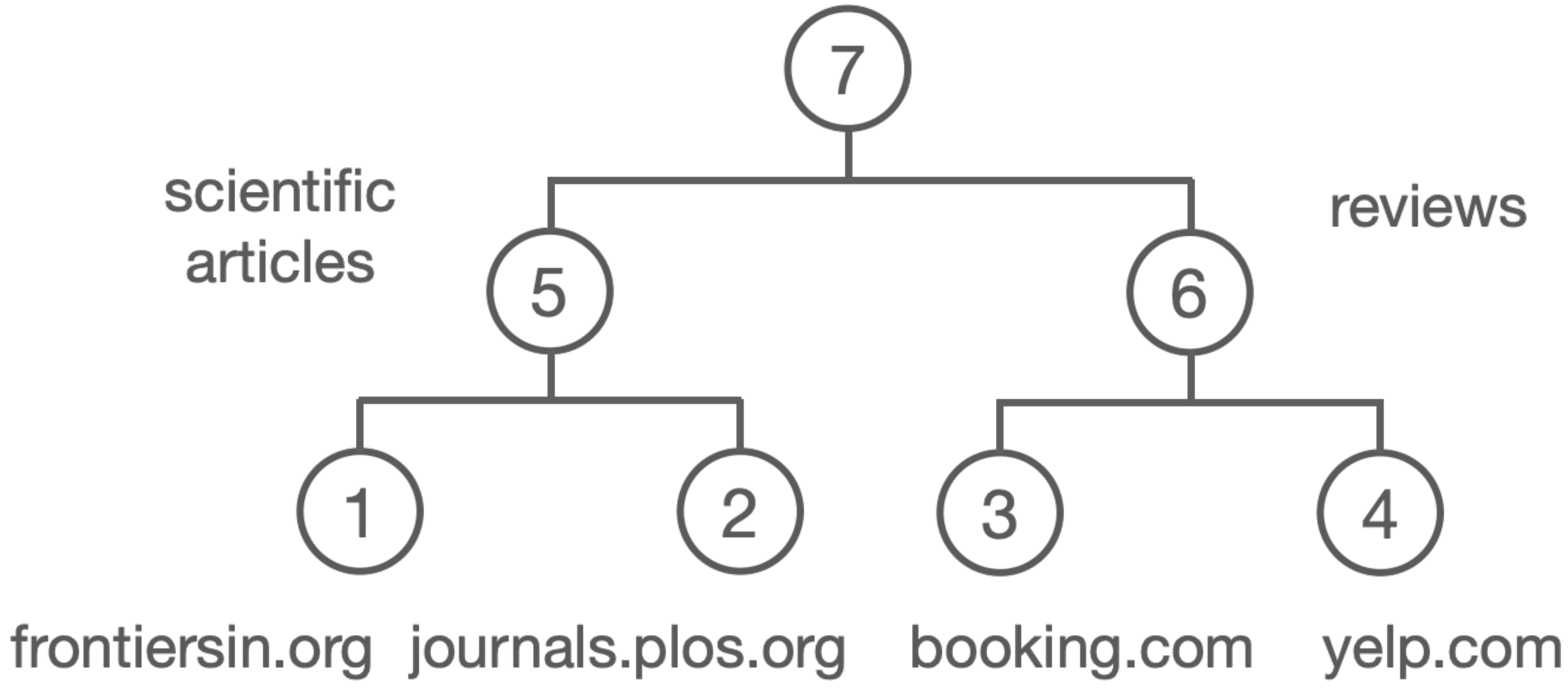
- **Hierarchical model**: GPT-2 (frozen) with a hierarchical structure of adapters
- **Baselines**
  - **Single adapters:** 1 adapter / domain
  - **Multi-domain adapters**: 1 adapter for **all** domains (*dense*)

- Motivation
- Proposed Approach
- **Experiments**
  - **Few-domain setting**
  - Many-domain setting
- Recap

Few-domain setting

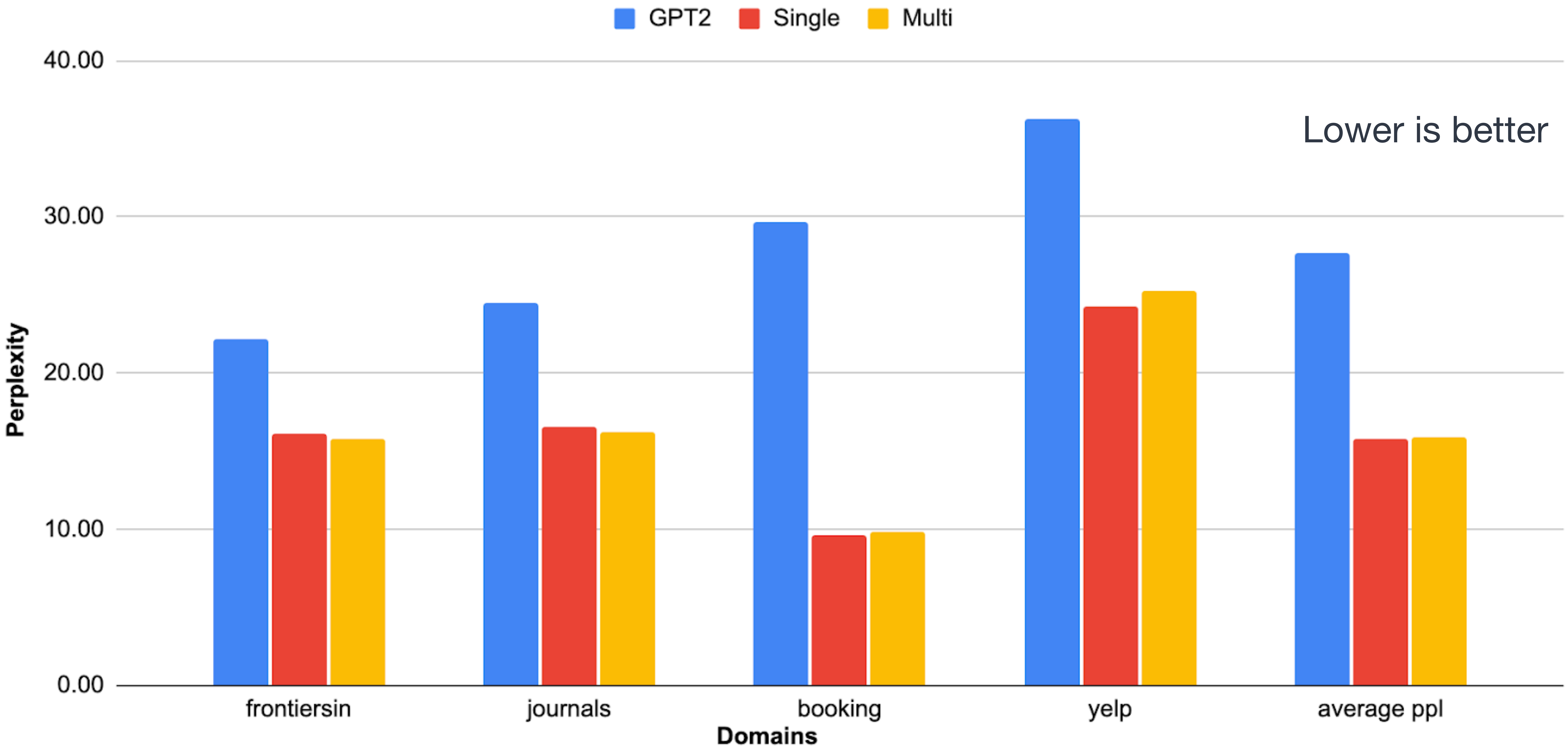
Data: text from 4 websites of C4

- 2 contain scientific articles
- 2 contain reviews
- equal amounts of data

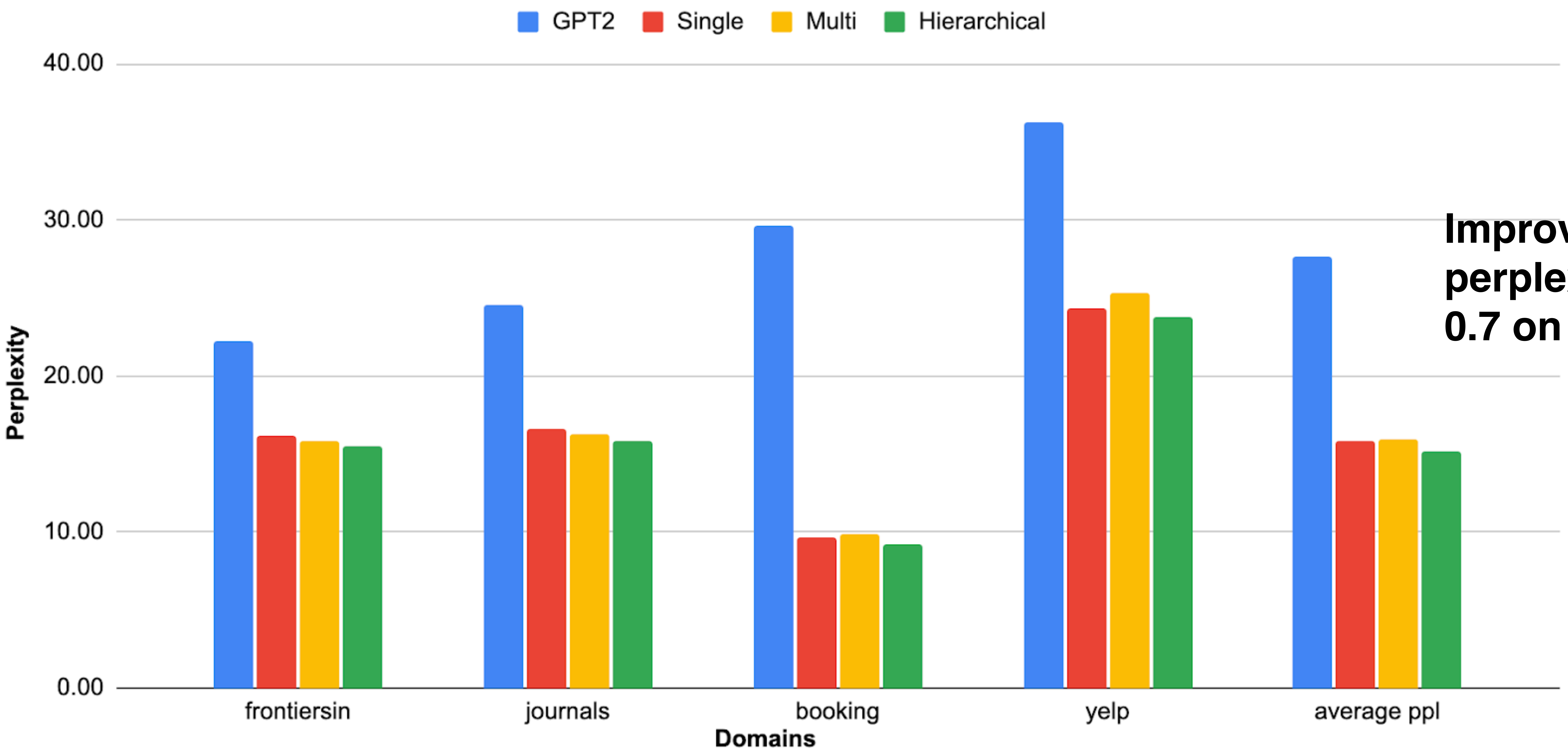


Few-domain setting - In-domain evaluation

Does it work?



Few-domain setting - In-domain evaluation

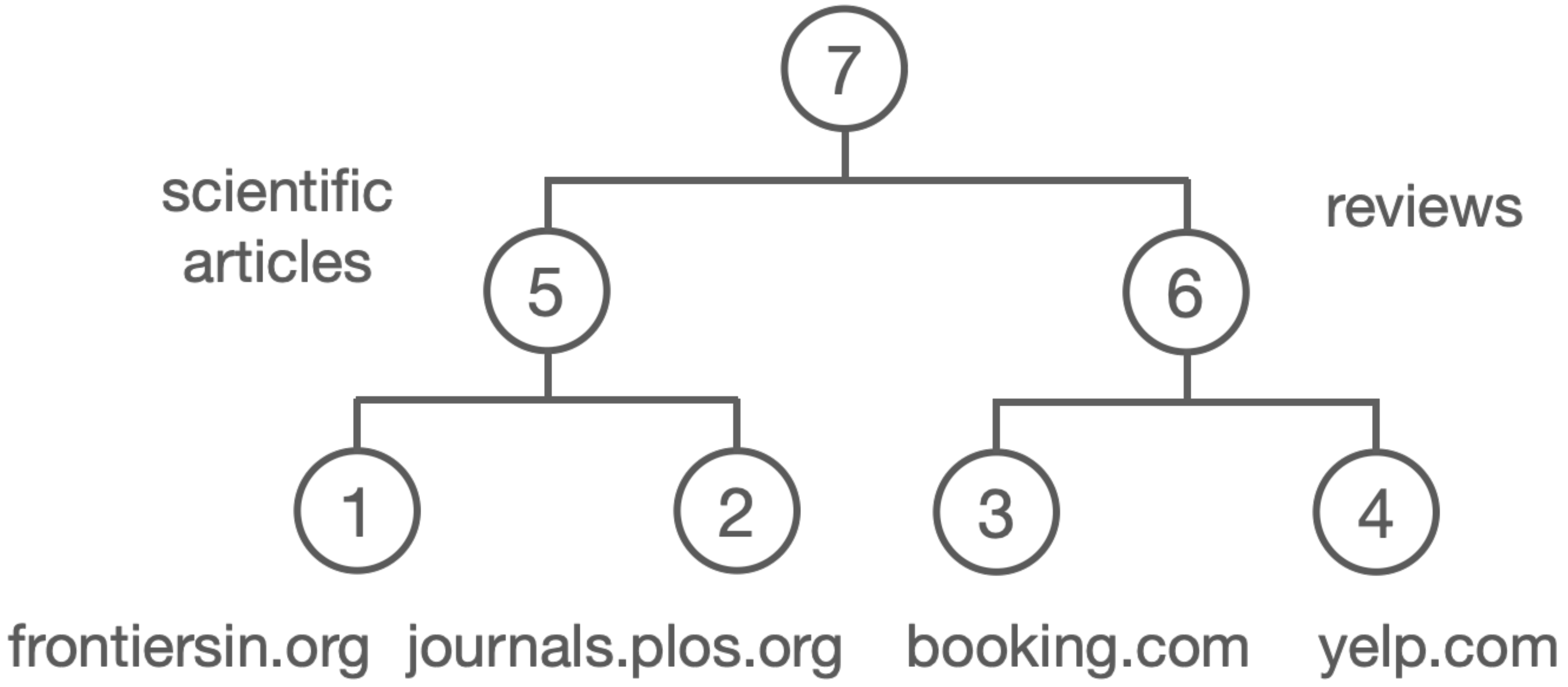


Improves perplexity by 0.7 on average!

Few-domain setting - Out-of-domain evaluation

Which single path through the tree should we activate?

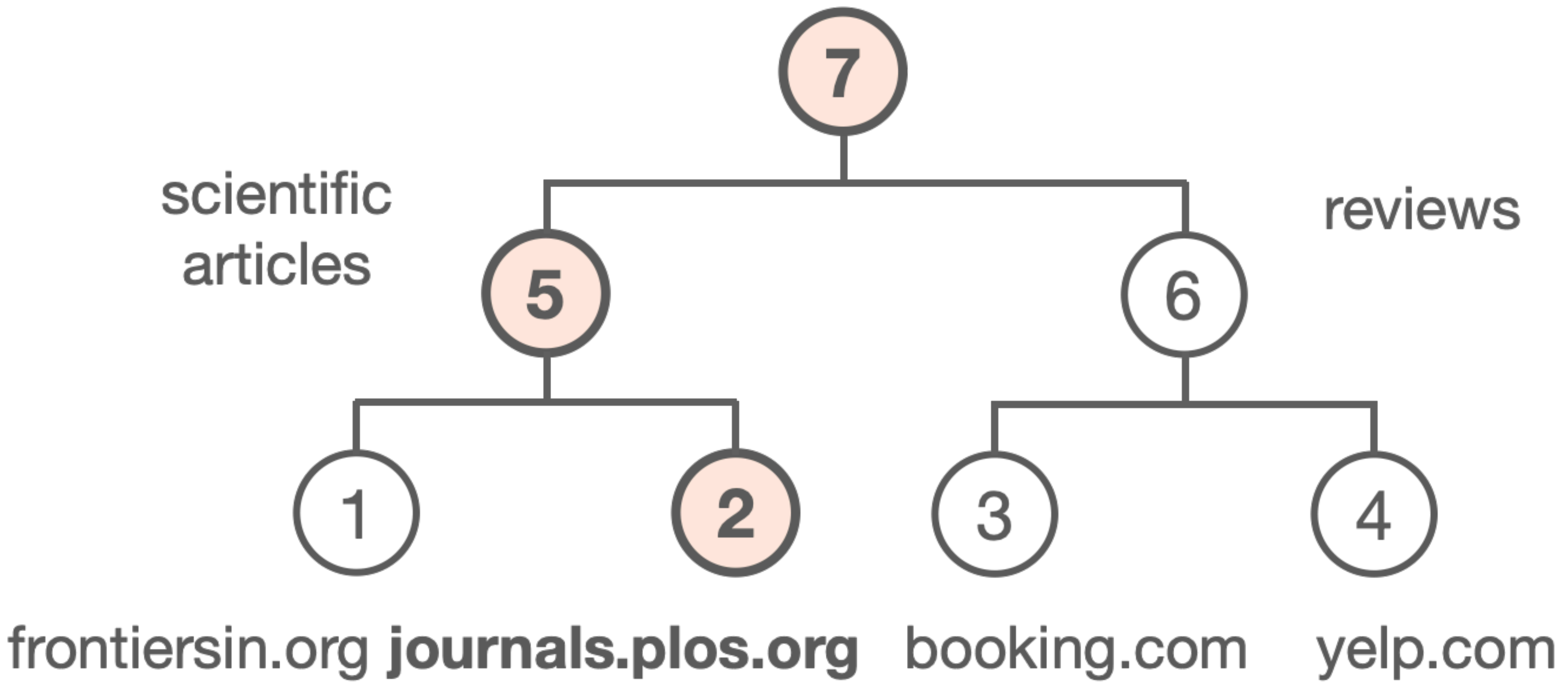
	1 path				2 paths	
	journals	frontiers	booking	yelp	science	reviews
ncbi	**17.6**	18.7	34.8	26.0	**17.3**	26.3
link.springer	**23.3**	**23.3**	37.0	33.1	**22.6**	31.8
scholars.duke	**20.7**	**20.7**	35.5	29.4	**19.9**	28.8
techcrunch	**27.7**	27.9	34.8	32.8	**27.1**	29.4
medium	**29.4**	**29.4**	35.9	36.2	**28.5**	30.6
tripadvisor	47.9	47.9	**37.0**	38.1	45.6	**26.0**
lonelyplanet	39.6	40.0	**25.5**	38.9	38.5	**25.3**
average	**29.5**	29.7	34.4	33.5	28.5	**28.3**



Few-domain setting - Out-of-domain evaluation

Which single path through the tree should we activate?

	1 path				2 paths	
	journals	frontiers	booking	yelp	science	reviews
ncbi	**17.6**	18.7	34.8	26.0	**17.3**	26.3
link.springer	**23.3**	**23.3**	37.0	33.1	**22.6**	31.8
scholars.duke	**20.7**	**20.7**	35.5	29.4	**19.9**	28.8
techcrunch	**27.7**	27.9	34.8	32.8	**27.1**	29.4
medium	**29.4**	**29.4**	35.9	36.2	**28.5**	30.6
tripadvisor	47.9	47.9	**37.0**	38.1	45.6	**26.0**
lonelyplanet	39.6	40.0	**25.5**	38.9	38.5	**25.3**
average	**29.5**	29.7	34.4	33.5	28.5	**28.3**

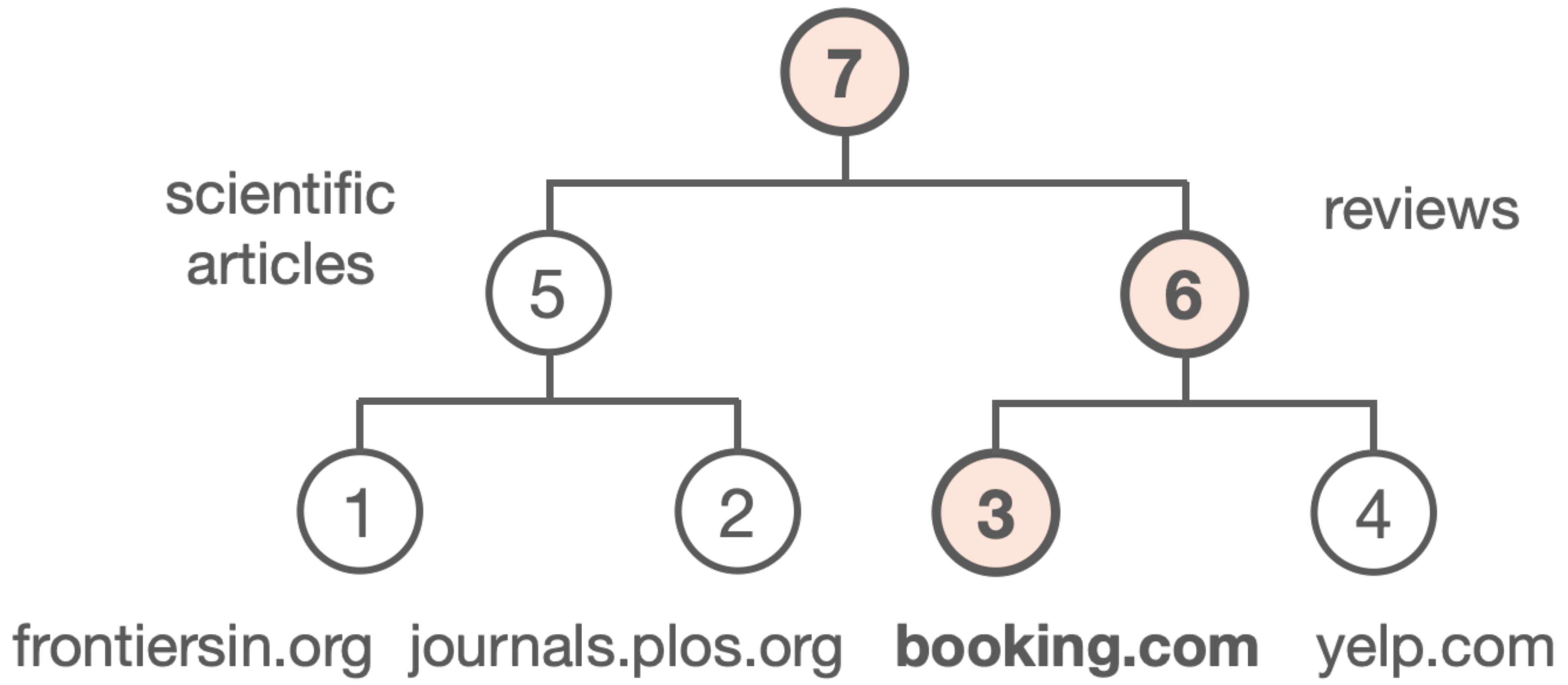


For out-of domain *medium:*
use path to ***journals***

Few-domain setting - Out-of-domain evaluation

Which single path through the tree should we activate?

	1 path				2 paths	
	journals	frontiers	booking	yelp	science	reviews
ncbi	**17.6**	18.7	34.8	26.0	**17.3**	26.3
link.springer	**23.3**	**23.3**	37.0	33.1	**22.6**	31.8
scholars.duke	**20.7**	**20.7**	35.5	29.4	**19.9**	28.8
techcrunch	**27.7**	27.9	34.8	32.8	**27.1**	29.4
medium	**29.4**	**29.4**	35.9	36.2	**28.5**	30.6
tripadvisor	47.9	47.9	**37.0**	38.1	45.6	**26.0**
lonelyplanet	39.6	40.0	**25.5**	38.9	38.5	**25.3**
average	**29.5**	29.7	34.4	33.5	28.5	**28.3**

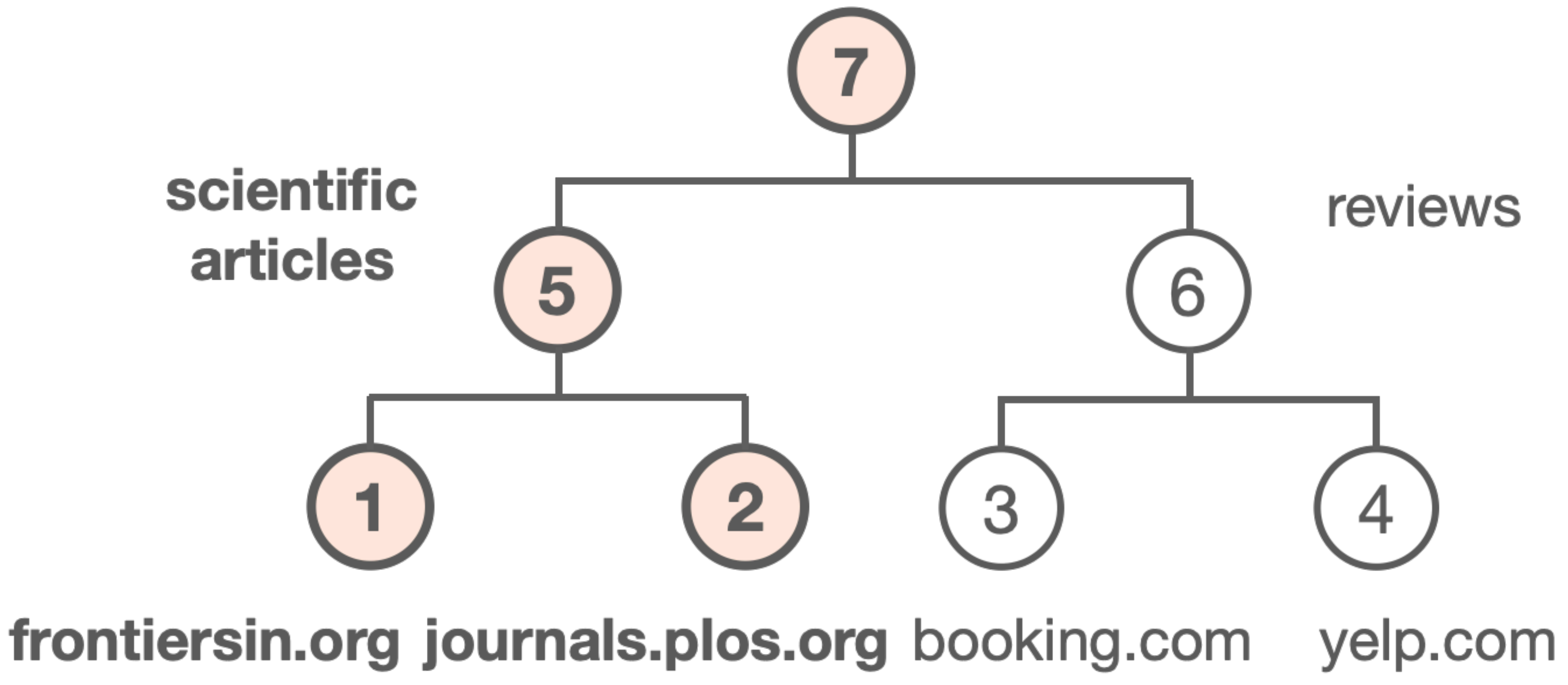


For out-of domain *tripadvisor:*
use path to ***booking***

Few-domain setting - Out-of-domain evaluation

Which 2 paths through the tree should we activate?

	1 path				2 paths	
	journals	frontiers	booking	yelp	science	reviews
ncbi	**17.6**	18.7	34.8	26.0	**17.3**	26.3
link.springer	**23.3**	**23.3**	37.0	33.1	**22.6**	31.8
scholars.duke	**20.7**	**20.7**	35.5	29.4	**19.9**	28.8
techcrunch	**27.7**	27.9	34.8	32.8	**27.1**	29.4
medium	**29.4**	**29.4**	35.9	36.2	**28.5**	30.6
tripadvisor	47.9	47.9	**37.0**	38.1	45.6	**26.0**
lonelyplanet	39.6	40.0	**25.5**	38.9	38.5	**25.3**
average	**29.5**	29.7	34.4	33.5	28.5	**28.3**

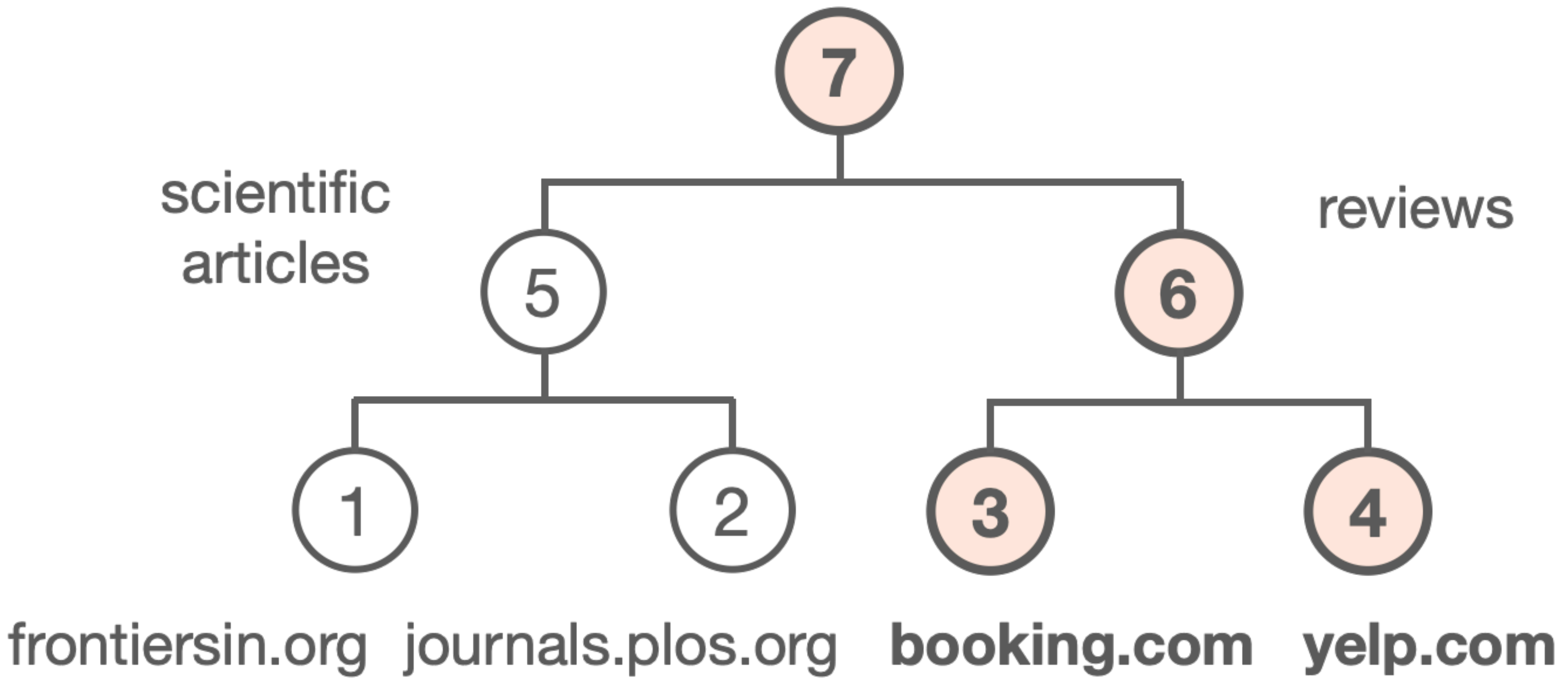


For held-out academic websites (like *link.springer, scholars.duke*): **scientific articles** paths

Few-domain setting - Out-of-domain evaluation

Which 2 paths through the tree should we activate?

	1 path				2 paths	
	journals	frontiers	booking	yelp	science	reviews
ncbi	**17.6**	18.7	34.8	26.0	**17.3**	26.3
link.springer	**23.3**	**23.3**	37.0	33.1	**22.6**	31.8
scholars.duke	**20.7**	**20.7**	35.5	29.4	**19.9**	28.8
techcrunch	**27.7**	27.9	34.8	32.8	**27.1**	29.4
medium	**29.4**	**29.4**	35.9	36.2	**28.5**	30.6
tripadvisor	47.9	47.9	**37.0**	38.1	45.6	**26.0**
lonelyplanet	39.6	40.0	**25.5**	38.9	38.5	**25.3**
average	**29.5**	29.7	34.4	33.5	28.5	**28.3**



For *tripadvisor* and *lonelyplanet*:
reviews paths

Few-domain setting - Out-of-domain evaluation

Which 2 paths through the tree should we activate?

	1 path				2 paths	
	journals	frontiers	booking	yelp	science	reviews
ncbi	**17.6**	18.7	34.8	26.0	**17.3**	26.3
link.springer	**23.3**	**23.3**	37.0	33.1	**22.6**	31.8
scholars.duke	**20.7**	**20.7**	35.5	29.4	**19.9**	28.8
techcrunch	**27.7**	27.9	34.8	32.8	**27.1**	29.4
medium	**29.4**	**29.4**	35.9	36.2	**28.5**	30.6
tripadvisor	47.9	47.9	**37.0**	38.1	45.6	**26.0**
lonelyplanet	39.6	40.0	**25.5**	38.9	38.5	**25.3**
average	**29.5**	29.7	34.4	33.5	28.5	**28.3**

No *a priori* criterion to choose !!!

How can we automatically find the best path(s)?

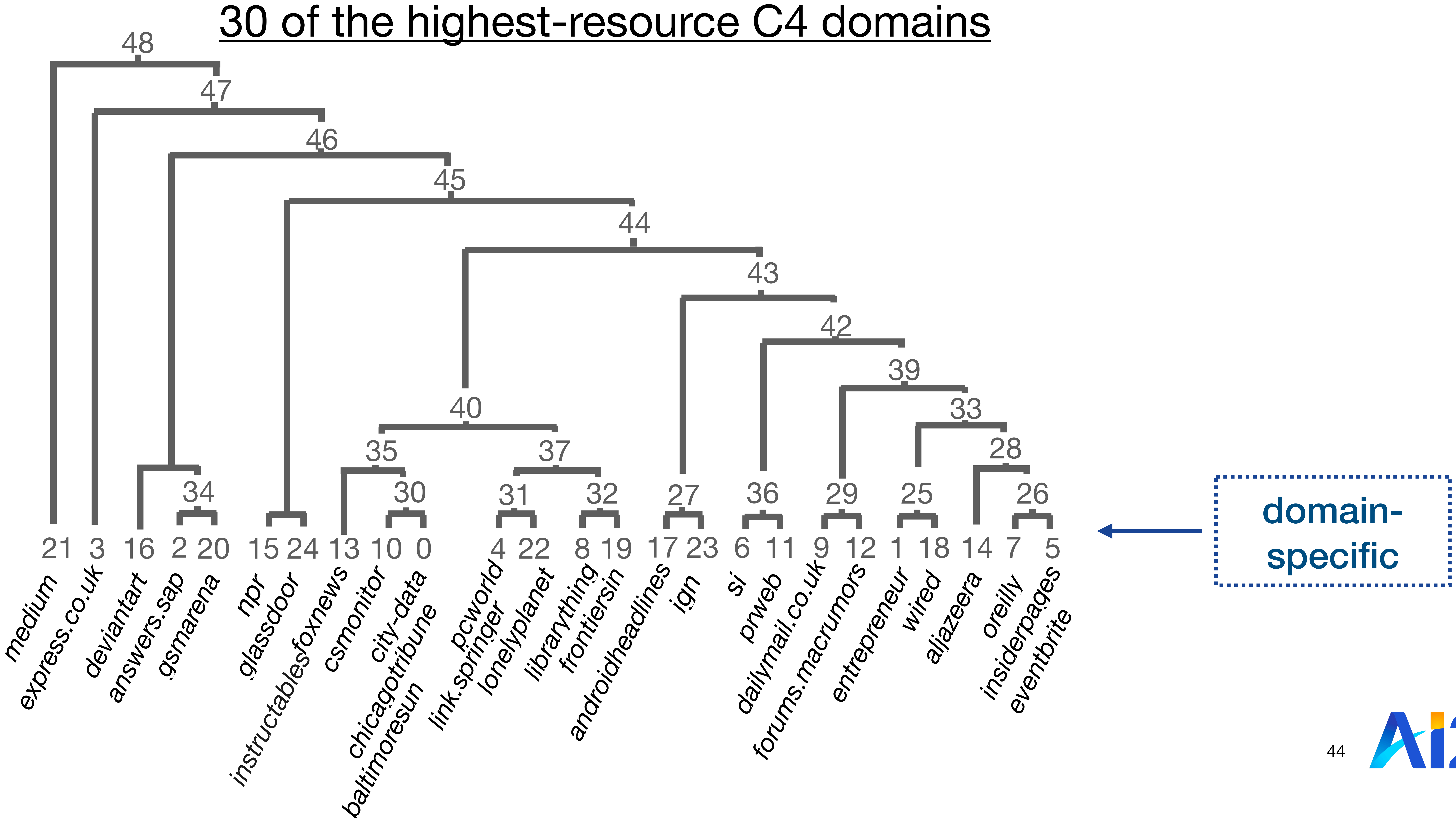
- Motivation
- Proposed Approach
- **Experiments**
  - Few-domain setting
  - **Many-domain setting**
- Recap

Many-domain setting

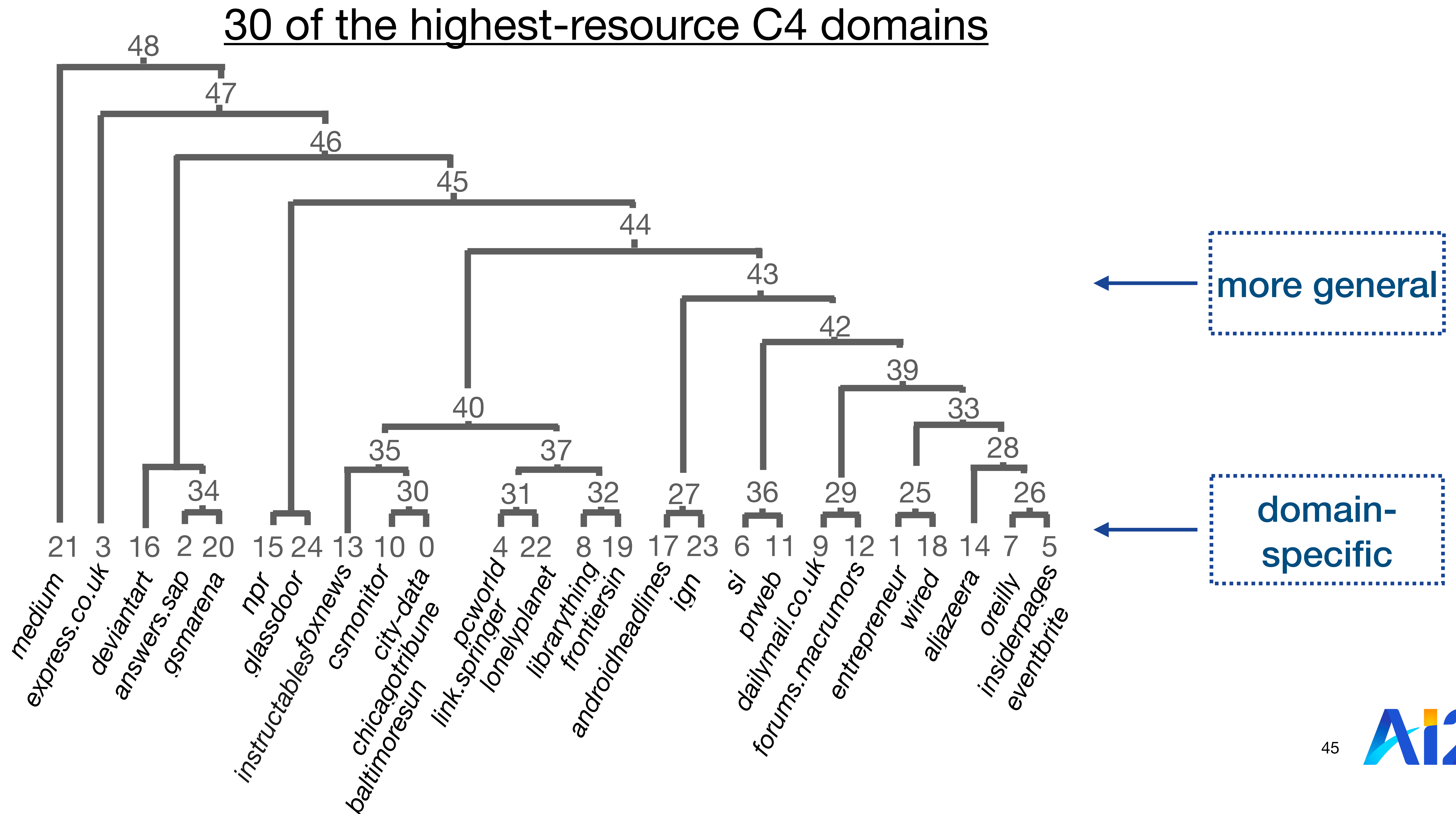
How do we infer the hierarchy?

- GPT-2 representations of 30 websites
- Fit a Gaussian Mixture Model (GMM) with 30 components
- **Hierarchical clustering of the GMM using symmetrized KL divergence as a distance metric**

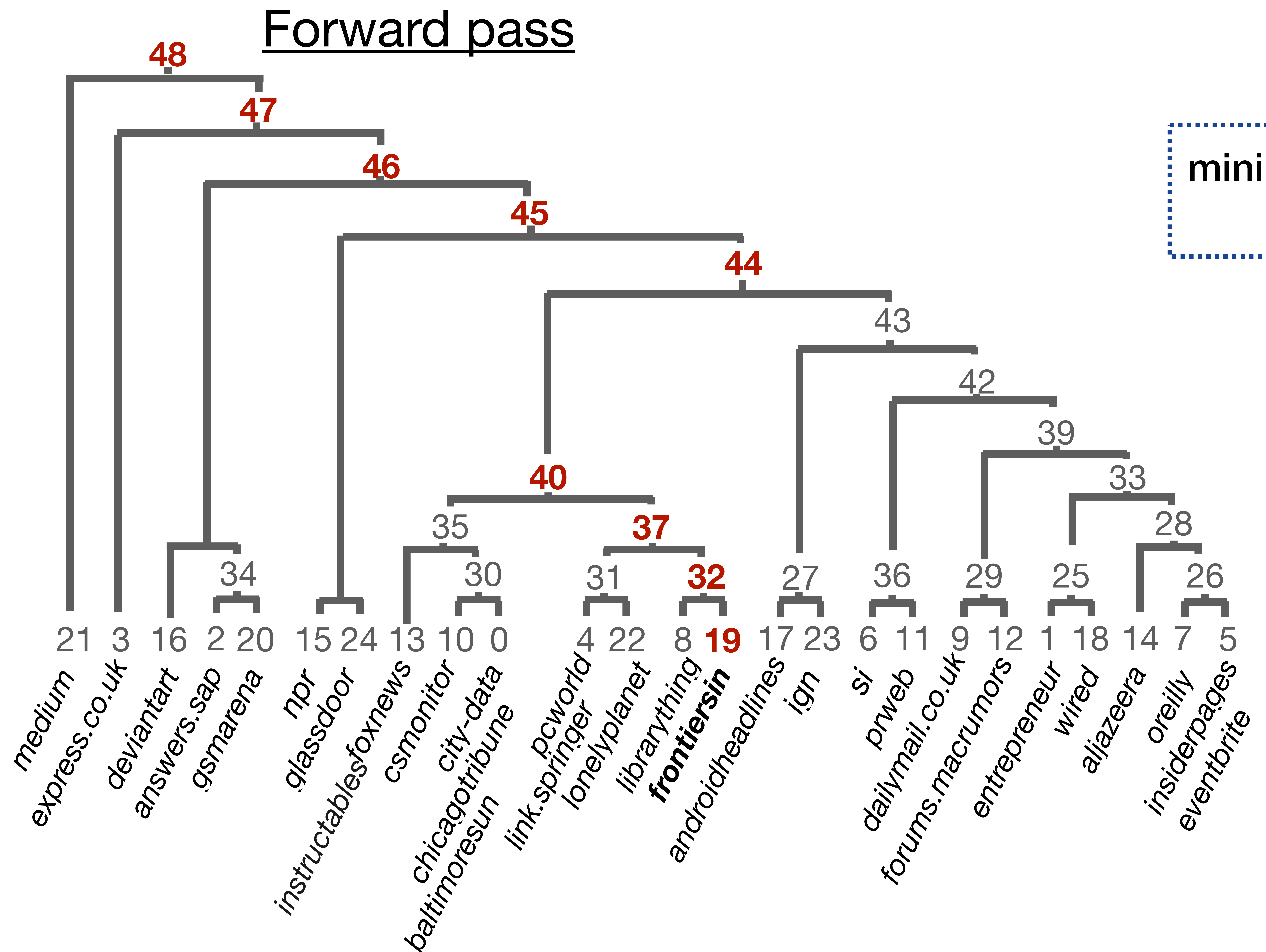
Hierarchical representation of domains



Hierarchical representation of domains

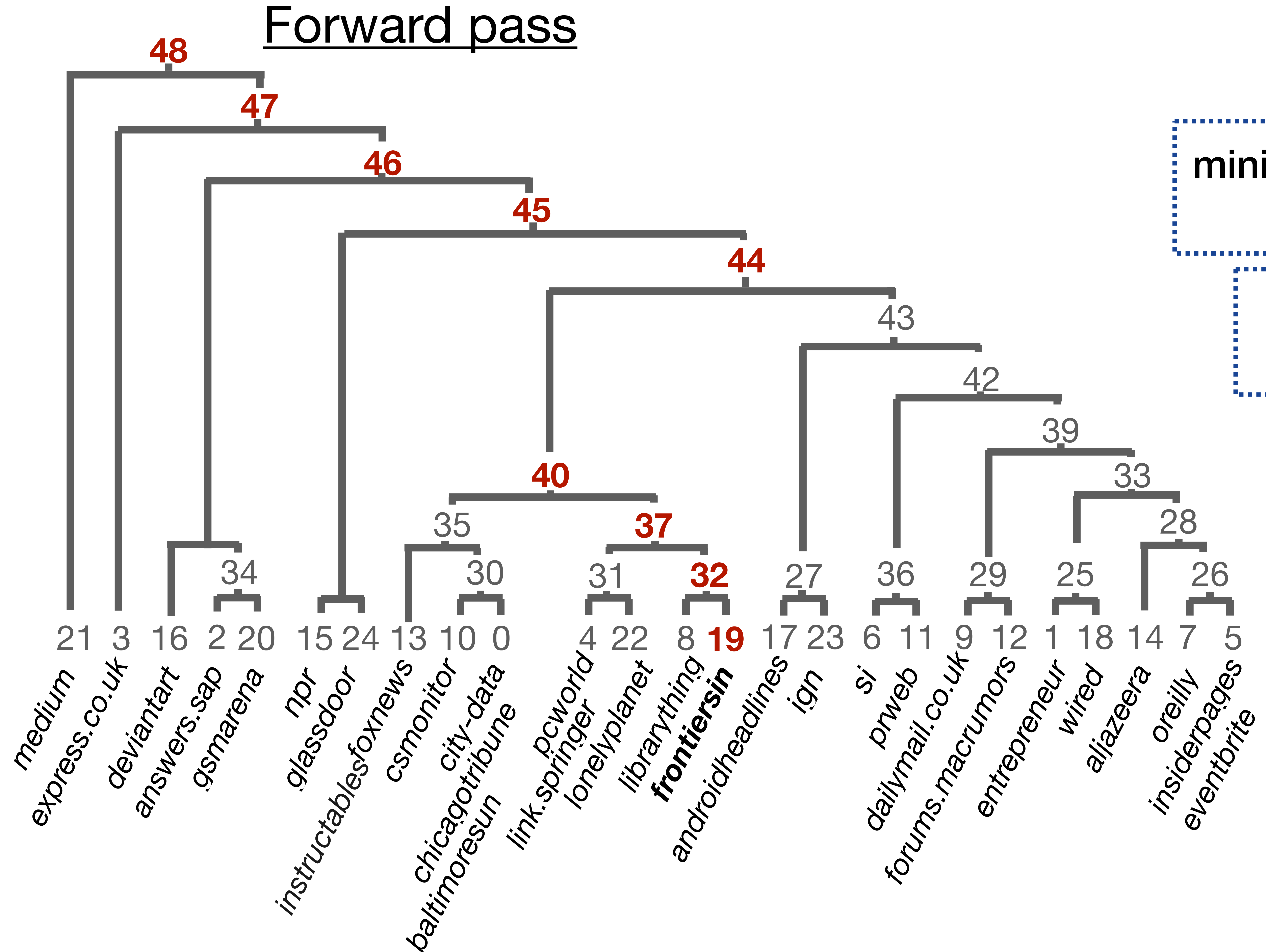


Hierarchical representation of domains



mini-batch from frontiersin.org
(representation hi)

Hierarchical representation of domains

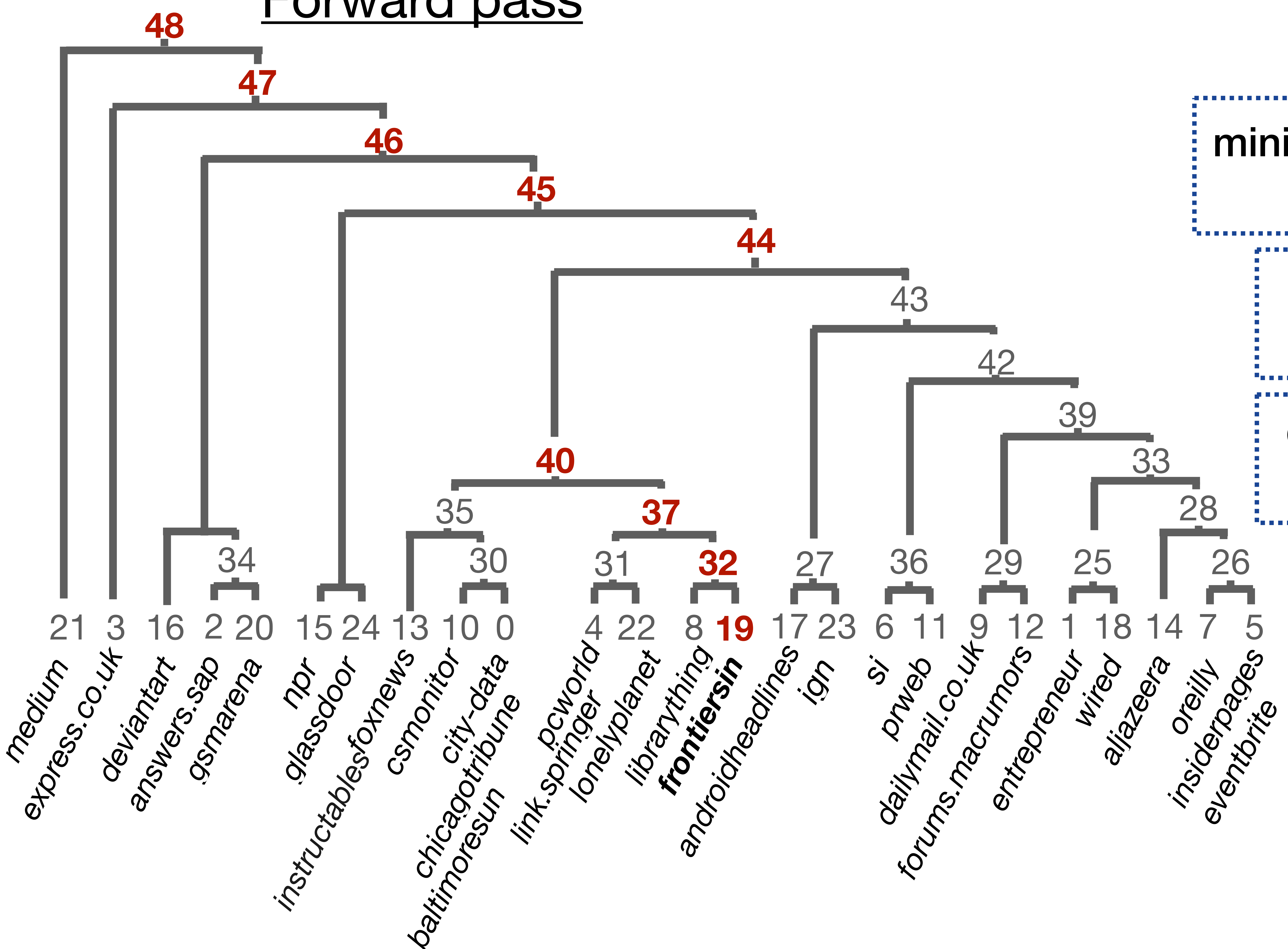


mini-batch from frontiersin.org (representation hi)

hi is input to the adapters in the path shown in red

Hierarchical representation of domains

Forward pass



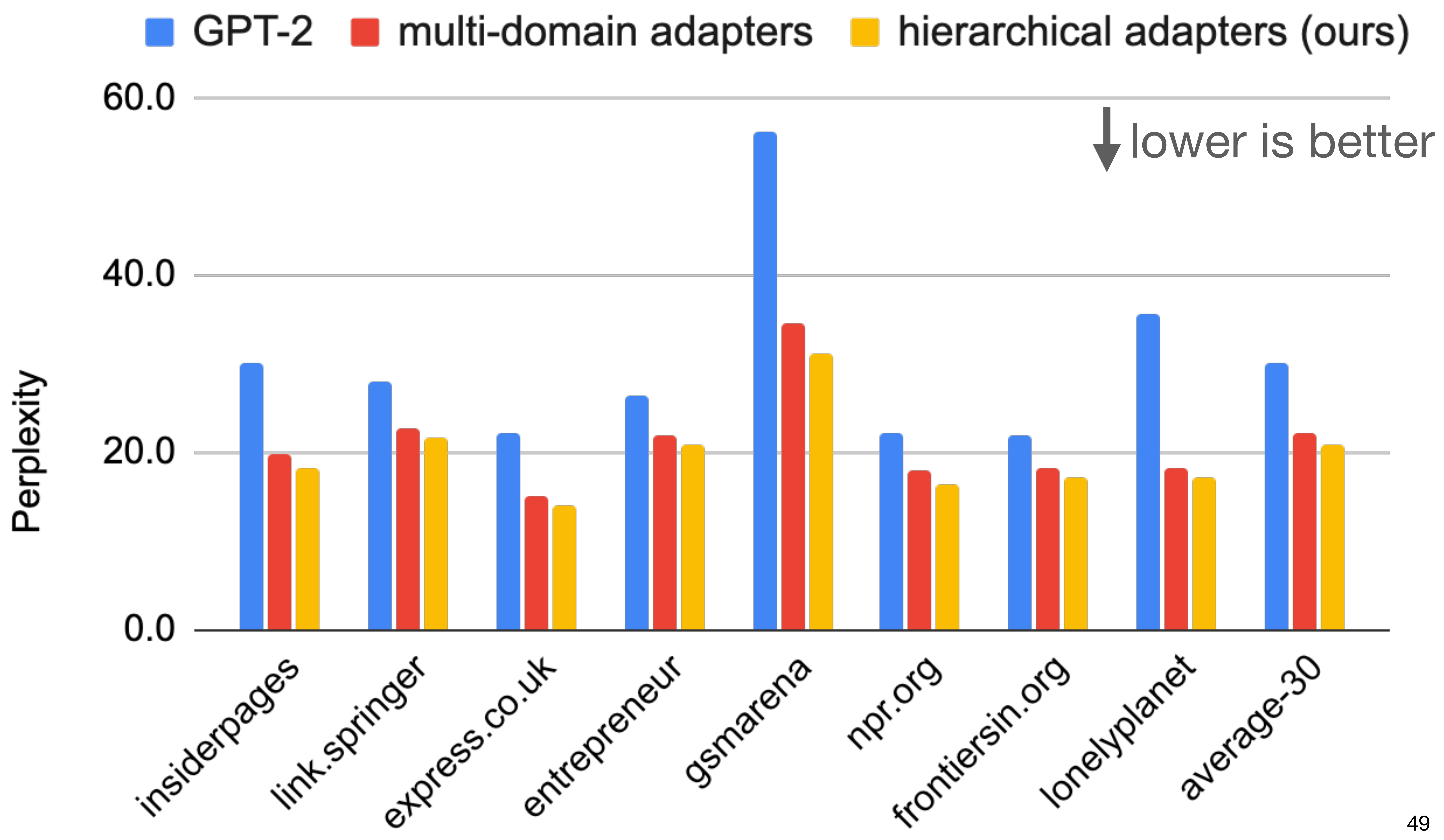
mini-batch from frontiersin.org (representation hi)

hi is input to the adapters in the path shown in red

Outputs are averaged and passed to the next layer

In-domain results

Main results: our approach consistently outperforms the baselines (on <u>**every**</u> domain)



Out-of-domain results

- We use the already fitted GMM to assign the probability of *N* sequences from a held-out website belonging to each cluster

- For each held-out domain, we use the **path to the training domain** (cluster) where the **majority** of sequences gets mapped to!

- No more parameters need to be trained!

Out-of-domain results

Main results:

when we activate 2 paths in the tree, we get better results than the baselines.

	GPT-2	multi-domain	hierarchy (1 path)	hierarchy (2 paths)
tripadvisor.com	40.4	34.8	35.9	**33.8**
dailystar.co.uk	20.7	13.9	**12.2**	**12.2**
techcrunch.com	27.7	21.5	21.8	**20.1**
scholars.duke.edu	22.6	20.7	**20.3**	**20.3**
booking.com	29.7	22.9	24.5	**22.0**
github.com	32.8	**30.3**	30.6	30.6
average (38)	26.8	$\boxed{22.3}$	23.0	**21.7**

Out-of-domain results

Paths used for out-of-domain evaluation

	Path 1	Path 2
tripadvisor.com	insiderpages	**lonelyplanet**
dailystar.co.uk	**express.co.uk**	**dailymail.co.uk**
techcrunch.com	wired	**entrepreneur**
scholars.duke.edu	**link.springer**	**frontiersin**
booking.com	insiderpages	**lonelyplanet**
github.com	oreilly	answers.sap

- Motivation
- Proposed Approach
- Experiments
  - Few-domain setting
  - Many-domain setting
- **Recap**

Recap

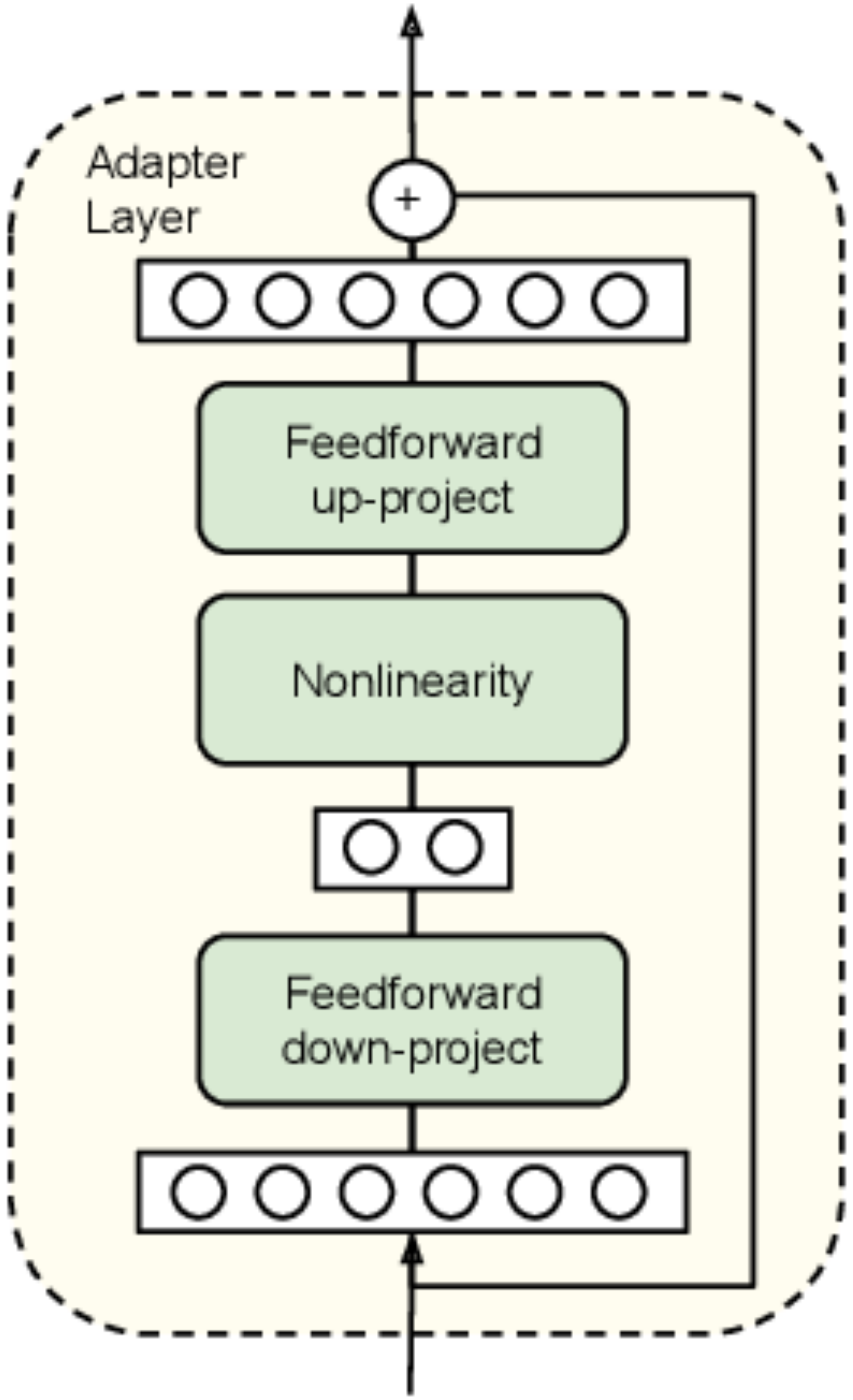
- We presented an approach that encodes the relations between domains using a **hierarchical structure**
- In-domain: across-the-board improvements
- Out-of-domain: better when activating 2 paths in the tree
- Efficiency: we train adapters added on top of a PLM sparsely

Thank you!

Bonus Slides

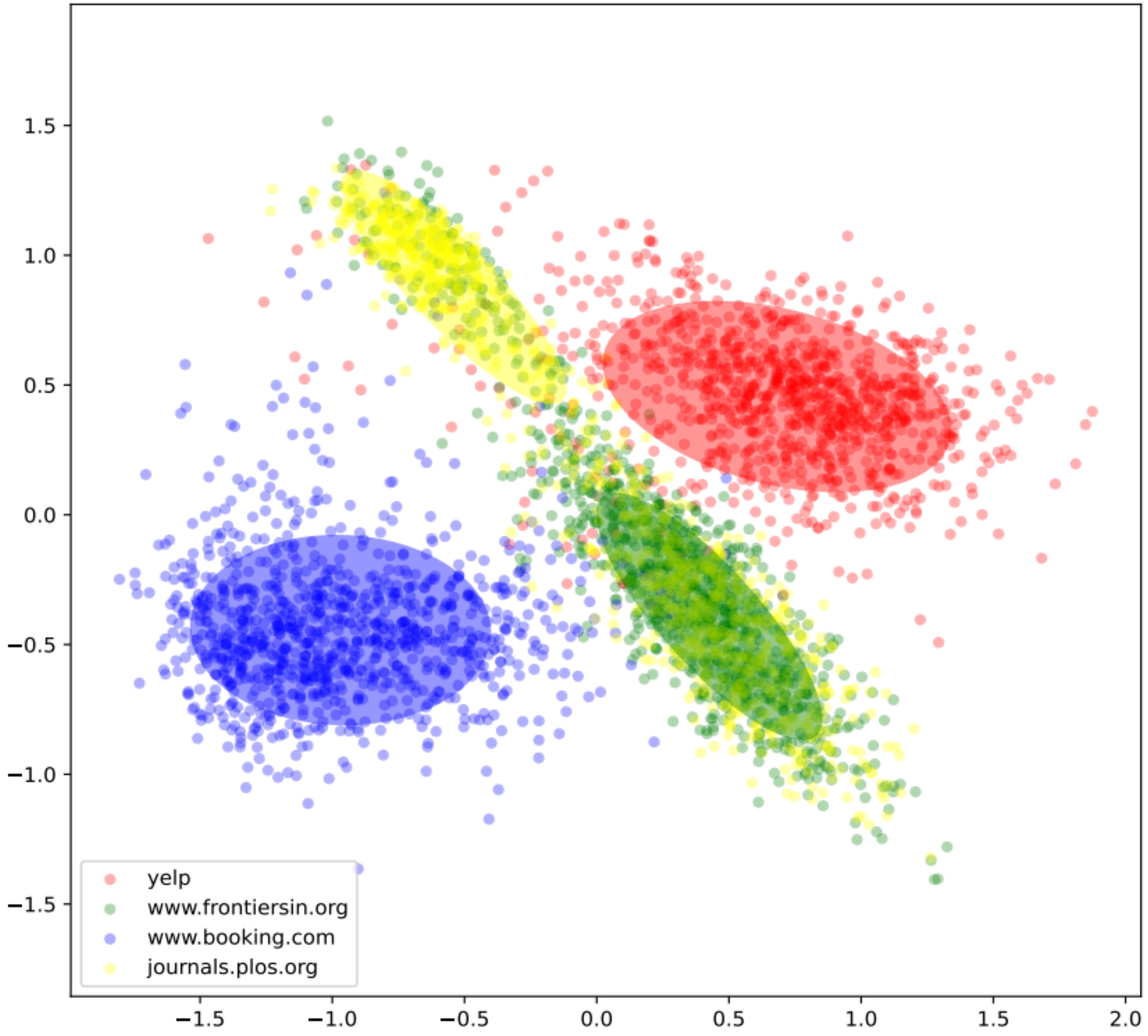
Adapter Layer

- Project hidden vector of i-th layer ($\mathbf{h}_i$) of dimension $\boldsymbol{d}$ to a dimension $\boldsymbol{m}$ ($m < d$)
- Non-linear activation (ReLU)
- Project back to $\boldsymbol{d}$ + residual connection



Bonus Slides

- GMM fitted on 4 domains-websites
- We do the same with 30 domains, then based on a distance metric find out their hierarchical structure (agglomerative clustering)



Bonus Slides

- **KL-divergence**

$$D_{KL}(\mathcal{N}_0 \| \mathcal{N}_1) = \frac{1}{2}\mathrm{tr}\left(\Sigma_1^{-1}\Sigma_0) + \ln\left(\frac{\det \Sigma_1}{\det \Sigma_0}\right)\right) + \frac{1}{2}\left((\mu_1 - \mu_0)^{\mathrm{T}}\Sigma_1^{-1}(\mu_1 - \mu_0) - N\right)$$

- Averaged **KL-divergence**

$$D_{KLsym}(\mathcal{N}_0, \mathcal{N}_1) = \frac{1}{2}\left(D_{KL}(\mathcal{N}_0 \| \mathcal{N}_1) + D_{KL}(\mathcal{N}_1 \| \mathcal{N}_0)\right)$$

Bonus Slides

Adapter Layer

- 2x inserted in each Transformer (see right Figure, Houlsby et al., 2019)
- Following approaches only used 1x Transformer layer (Bapna and Firat, 2019)
- The (pretrained) Transformer stays frozen, only the adapters are fine-tuned