LREC-COLING 2024

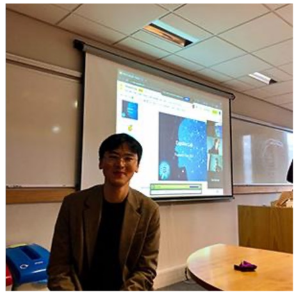
ELRA European Language Resources Association

ICCL International Committee on Computational Linguistics

Neuro-symbolic AI Lab

Formal Semantic Controls over Language Models

Danilo Silva de Carvalho, Yingji Zhang, André Freitas



MANCHESTER 1824 The University of Manchester

idiap RESEARCH INSTITUTE

CANCER RESEARCH UK | Manchester Institute

Motivation

Some language interpretation tasks require additional levels of **safety and control**.

While Language Models (LMs) have provided a flexible foundation for addressing a diverse spectrum of tasks, can we develop language representation/models with more granular levels of **control and interpretability**?

Provocative question: Is it sufficient to assume that LMs will build rigorous representation of reality and language use?

Motivation

Critical applications: medicine, law, decision support, etc.

But also: end-user facing applications.

Patients living in the San Francisco area with ErbB2+ breast cancer, a body weight > 60 kg, and a history of treatment with Cyclophosphamide in the last year, are eligible for this clinical trial.

$\models$

Q: How do models represent these concepts?
Q: Do they deliver consistent conceptual inference?

Clinical Trial Report - Eligibility Criteria

Inclusion criteria

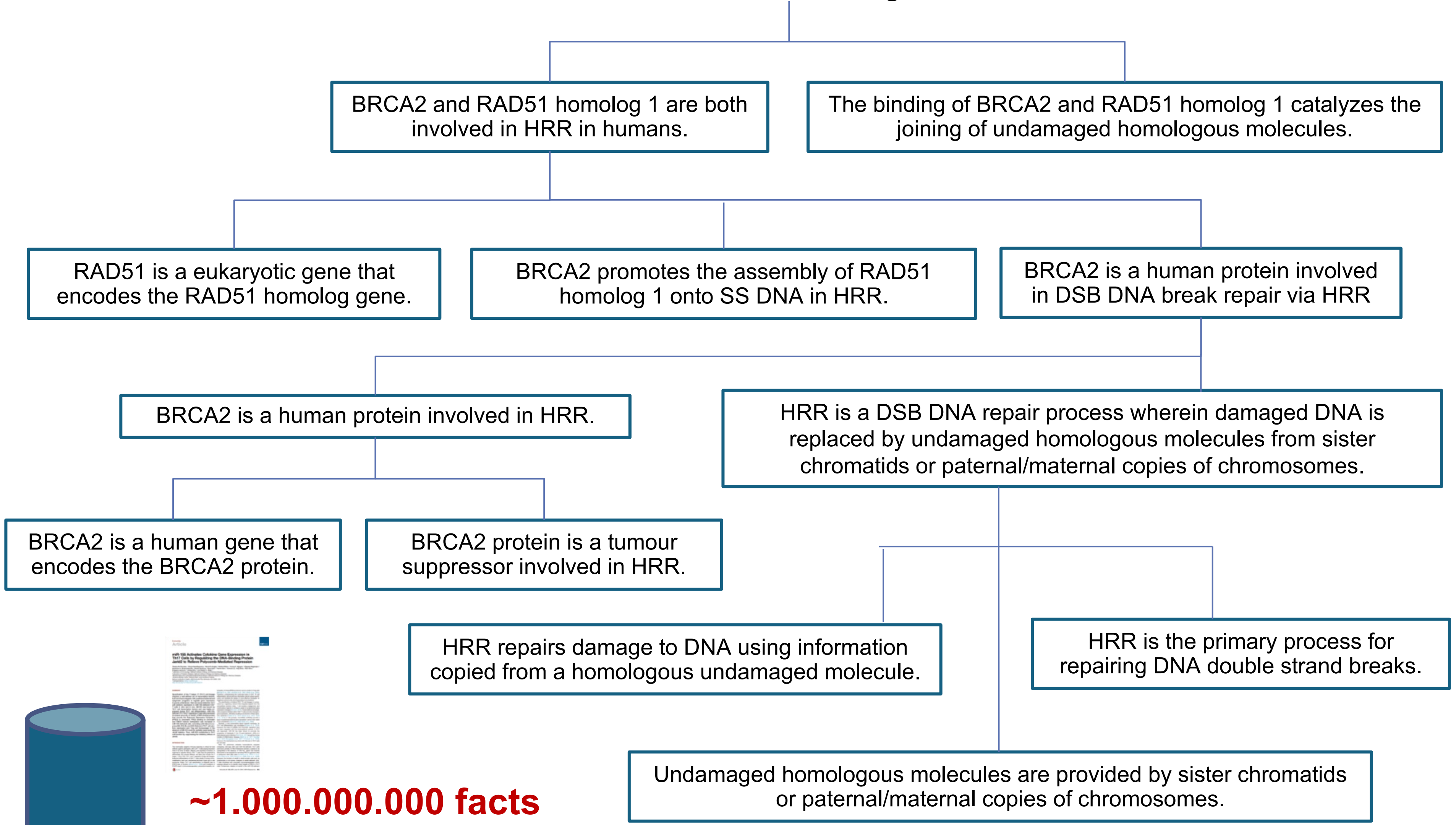
- Patients with a history of chemotherapy treatment within the last 24 months.
- Age $\geq$ 60 years
- HER2-positive T1 histologically confirmed invasive carcinoma of the breast.
- Body weight > 110 lbs
- Patients be California residents

Exclusion criteria

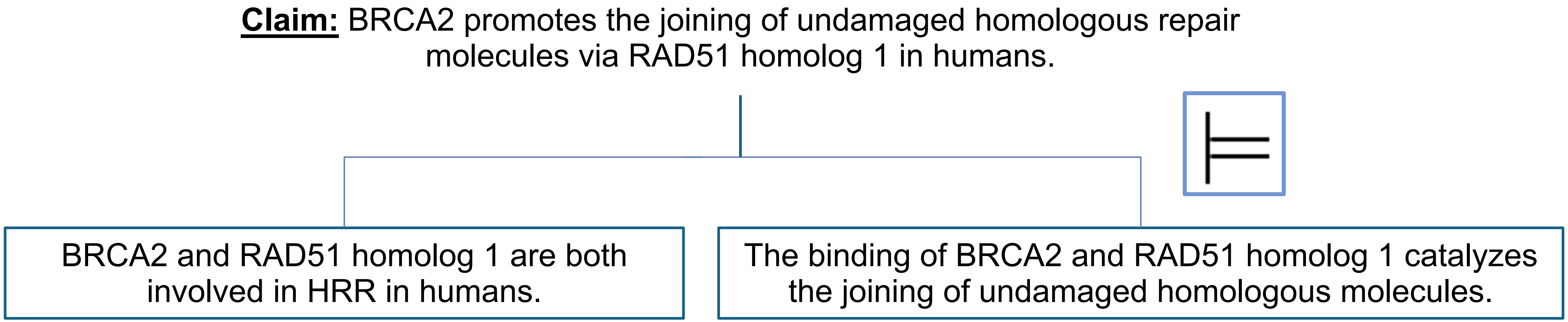
- Pregnant women

Expert-level scientific inference & explanation

Claim: BRCA2 promotes the joining of undamaged homologous repair molecules via RAD51 homolog 1 in humans.


BRCA2 and RAD51 homolog 1 are both involved in HRR in humans.
The binding of BRCA2 and RAD51 homolog 1 catalyzes the joining of undamaged homologous molecules.
RAD51 is a eukaryotic gene that encodes the RAD51 homolog gene.
BRCA2 promotes the assembly of RAD51 homolog 1 onto SS DNA in HRR.
BRCA2 is a human protein involved in DSB DNA break repair via HRR
BRCA2 is a human protein involved in HRR.
HRR is a DSB DNA repair process wherein damaged DNA is replaced by undamaged homologous molecules from sister chromatids or paternal/maternal copies of chromosomes.
BRCA2 is a human gene that encodes the BRCA2 protein.
BRCA2 protein is a tumour suppressor involved in HRR.
HRR repairs damage to DNA using information copied from a homologous undamaged molecule.
HRR is the primary process for repairing DNA double strand breaks.
Undamaged homologous molecules are provided by sister chromatids or paternal/maternal copies of chromosomes.

~1.000.000.000 facts



Claim: BRCA2 promotes the joining of undamaged homologous repair molecules via RAD51 homolog 1 in humans.

$\models$

BRCA2 and RAD51 homolog 1 are both involved in HRR in humans.

The binding of BRCA2 and RAD51 homolog 1 catalyzes the joining of undamaged homologous molecules.

Q: How do models represent sentences and their entailment relations?

Q: In which cases will inferences fail?

T|F?

Conclusion

Patients with loss of PALB2 may benefit from PARP1 inhibition due to synthetic lethality, causing cells to rely on a singular mechanism to repair cumulative damage to DNA.

$\vdash$

Intermediate Steps

24. Loss of PALB2 leads to a deficiency in HRR, causing the cells to rely on other DNA repair mechanisms. (Combination of premises 8, 15, 16, 21, 22)

25. Inhibiting PARP in cells lacking PALB2 results in the accumulation of DNA damage due to the reliance on a singular repair mechanism, leading to synthetic lethality. (Combination of premises 5, 9, 10, 24)

RAG

Premises

...

5- Inhibiting PARP results in accumulation of SS breaks.

6- NHEJ does not use a template to repair DSB and can cause increased genomic instability.

7- PARP1 synthesis PAR which recruits repair proteins to sites of DNA damage

8- In the absence of functional HRR genes, DNA repair defaults to NHEJ.

9- PARP1 synthesises PAR.

10- PAR recruits repair proteins to damaged DNA site.

...

15- PALB2 is required for the localization of BRCA2 to sites of DNA damage

16- PALB2 encodes a major BRCA2 binding partner that controls its intranuclear localization and stability.

17- RAD51 is a eukaryotic gene that encodes the RAD51 homolog gene.

18- BRCA2 promotes the assembly of RAD51 homolog 1 onto SS DNA in HRR.

19- BRCA2 is a human gene that encodes the BRCA2 protein.

20- BRCA2 protein is a tumour suppressor involved in HRR.

21- HRR is the primary process for repairing DNA double strand breaks.

22- HRR repairs damage to DNA using information copied from a homologous undamaged molecule.

23- Undamaged homologous molecules are provided by sister chromatids or paternal/maternal copies of chromosomes.

...

The Neuro-symbolic approach

The **Neuro**: Language Models (LMs) as the foundation for scaling-up language interpretation (content-based, flexible).

The **Symbolic**: LLMs alone do not deliver complex and controlled inference.

Epistemological foundations:

- Building on >2000 years foundations on epistemology & formal reasoning.
- Precisely defining formal and material inference.
- Integrating epistemological priors as controls within LMs.
- Evaluating on real-world inference conditions.

IN LOGIC WE TRUST

To summarise

Language understanding and inference implies:

- Representation of complex sentence structures.
- Interpretation of complex concepts.
- Interpretation of contextual differences.
- Step-wise, controlled inference.

…

Today

Methods for integrating the **flexibility of LMs** to the control of formal models (Neuro-symbolic NLP models).

The angle: less ‘task-oriented’.

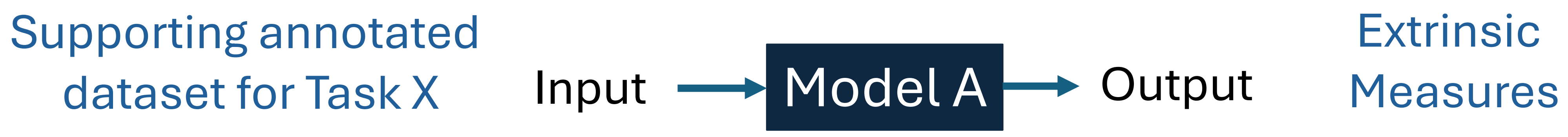
Zooming into the representation of well-defined linguistic objects (**sentences** and inference).

E.g.

- Sentences with complex structures.
- Sentences referring to conceptual representations
(e.g. definitions, explanations)
- Interface between content and structure.

Prevalent Paradigm (Extrinsic Evaluation)

Task X



Assumptions:

- Dataset is a proxy approximation for Task X.
- Dataset is roughly representative of the scope of Task X.
  - including the distribution of the ling./inf. phenomena associated with Task X.
- Out-of-Distribution (OOD) generalisation is defined in terms of other datasets.
- A characterisation of the ling./inf. phenomena associated with Task X are not at the centre.
- Aggregate extrinsic measures provide an absolute and comparative indicator of how Model A addresses Task X.

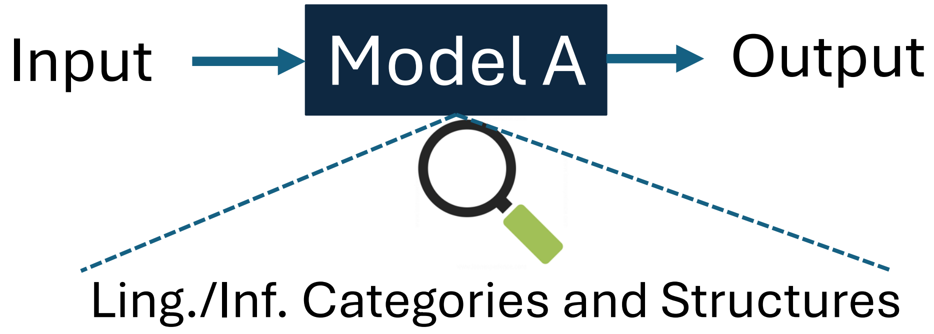
Overall nature of the empirical claims:

- Interventions behind Model A improves interpretation of Task X wrt to Datasets 1,2,3 …
- Interventions behind Model A improves interpretation of Task X as compared to Models B, C, D, …
- Without that intervention (ablated Model A'), *ceteris paribus*, we decrease of performance wrt A.

Representation/Interpretability-based Evaluation

Task X

Supporting annotated dataset for Task X



Extrinsic Measures

Intrinsic Measures

Assumptions:

- Interpreting Task X subsumes addressing ling./inf. categories α, β, γ. (common across other tasks).
- To address Task X it is desirable that the model induces a representation which reflects α, β, γ, ...
- A characterisation of the ling./inf. phenomena associated with Task X is not at the centre.
- Dataset covers α, β, γ, within a quantifiable distribution.
- Aggregate intrinsic measures provide an absolute and comparative indicator of how Model A addresses α, β, γ, ...

Overall nature of the empirical claims:

- Interventions behind Model A improves interpretation of α, β, γ as content-expressed in Datasets 1,2,3 ..
- Interventions behind Model A improves interpretation of α, β, γ as compared to Models B, C, D, ...
- Without that intervention (ablated Model A'), *ceteris paribus*, we decrease of performance wrt α, β, γ .

Value

- Promotes an evaluation perspective which is semantically granular.

- Allows a deeper understanding of the transferability of the results.
    - E.g. Target properties can be different across languages.

- Allows the design of models which are better linguistically grounded.

- Provides an alternative empirical pathway to do NLP beyond an extrinsic evaluation dogma (‘milking the F1-score cow’).

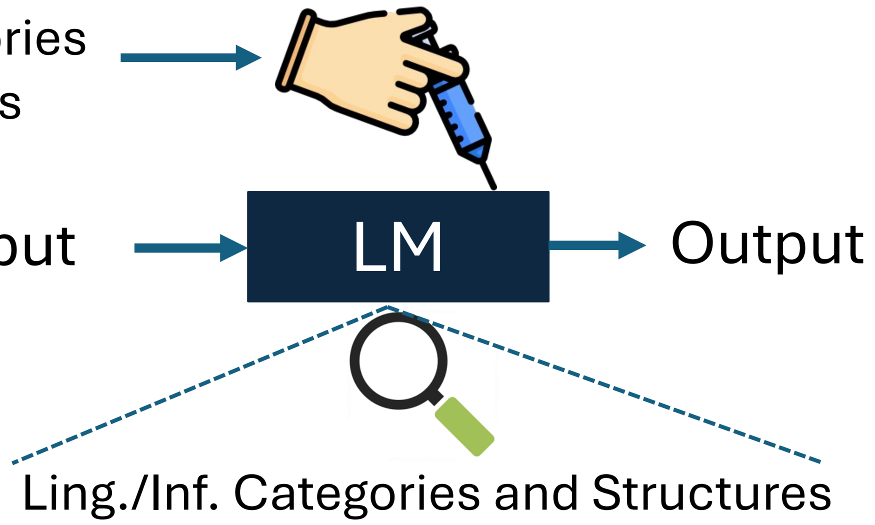
- Formal grounding as an enabler of safety mechanisms.

  (which types of inference are covered)

Formal intervention

Task X

Supporting annotated dataset for Task X


Linguistic Categories and Structures
Input
LM
Output
Ling./Inf. Categories and Structures

Extrinsic Measures

Intrinsic Measures

Representation & Reasoning

Premises
Facts
Background knowledge

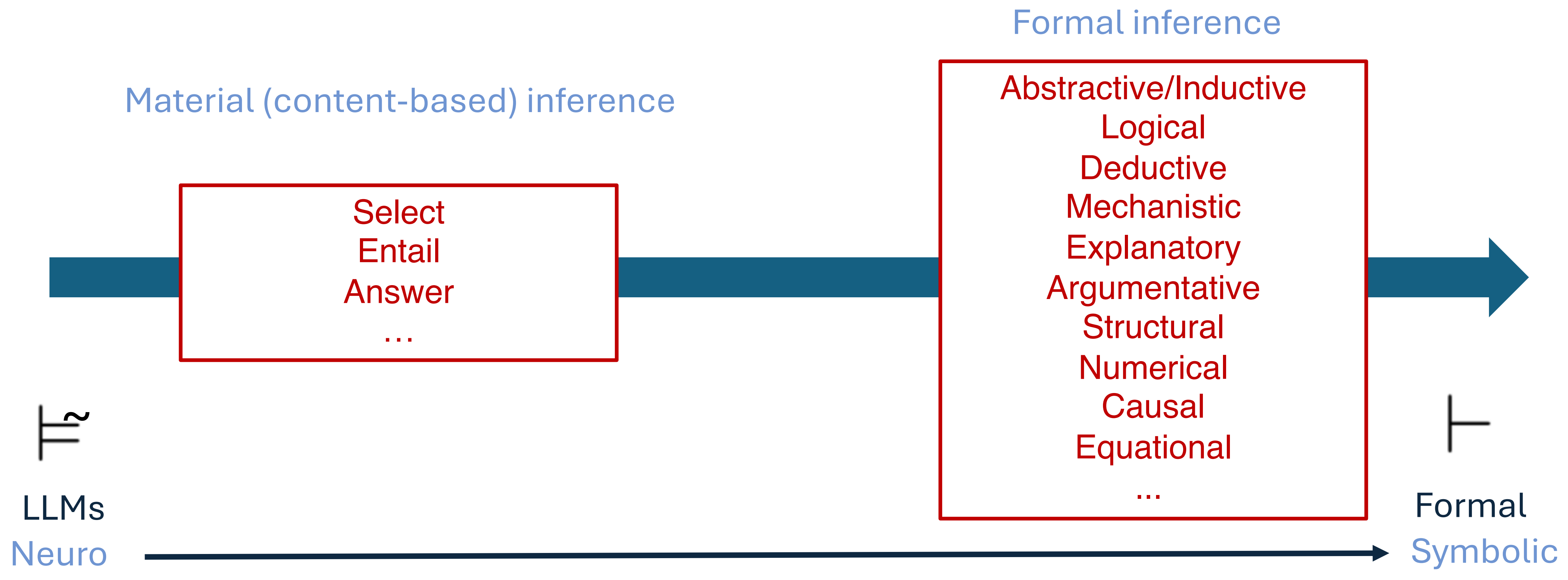
$\Gamma \models \Phi$ Conclusion

computes an answer

$\Gamma \vdash \Phi$

+ explains an answer

- step-wise explicit inference
- verifiability
- logical correctness
- completeness
- control



Outline for Today

Contrasting Formal vs Neural/Latent perspectives of semantics

Controlling Language Models (LMs)

Language Variational Autoencoders (VAEs)

Semantic Control via Conditional VAEs

Building & Probing Language VAEs (LangSpace & LangVAE)

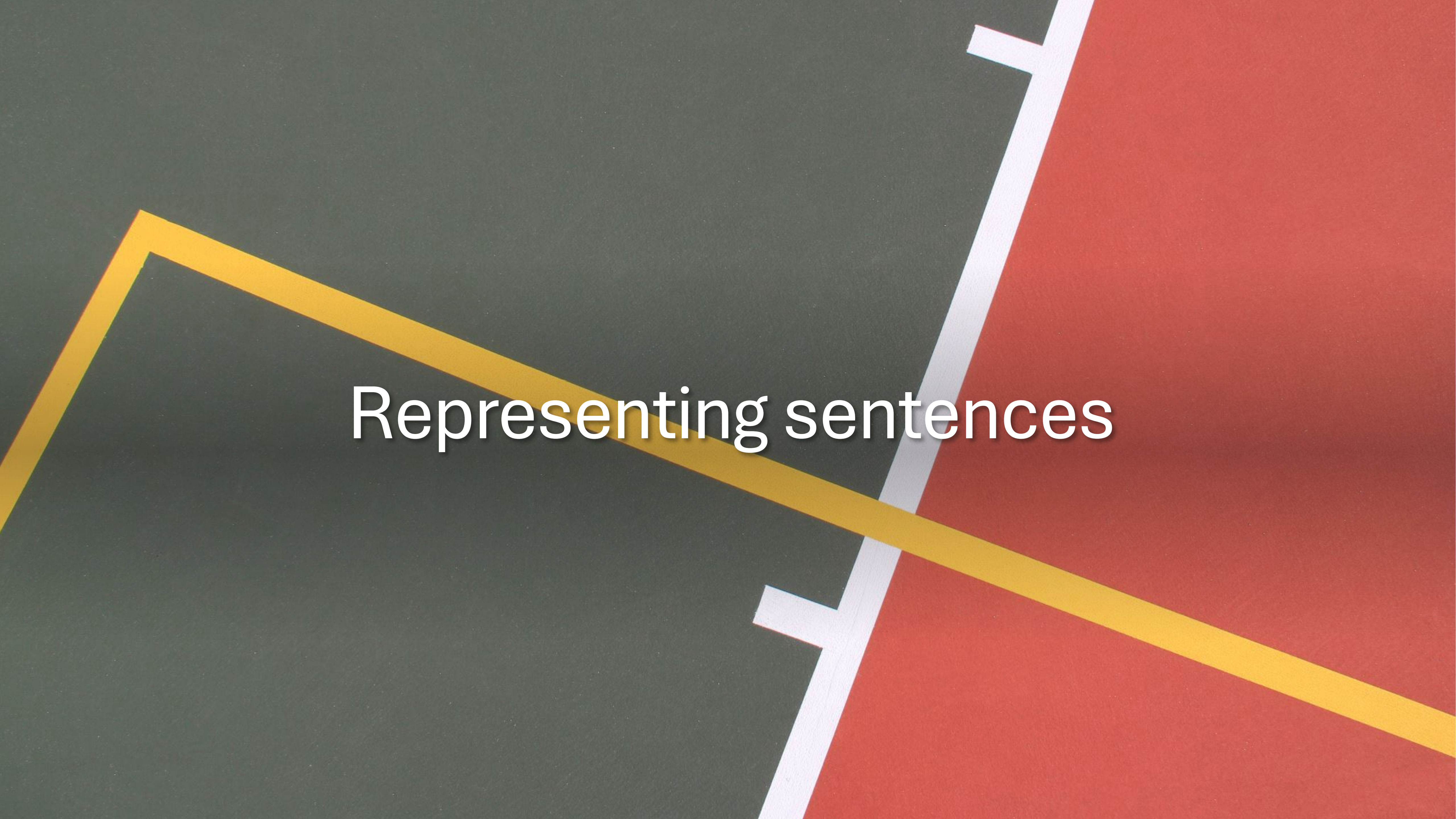
Improving Separability

Discretisation & Control

Syntactic & Structural Control

Trends

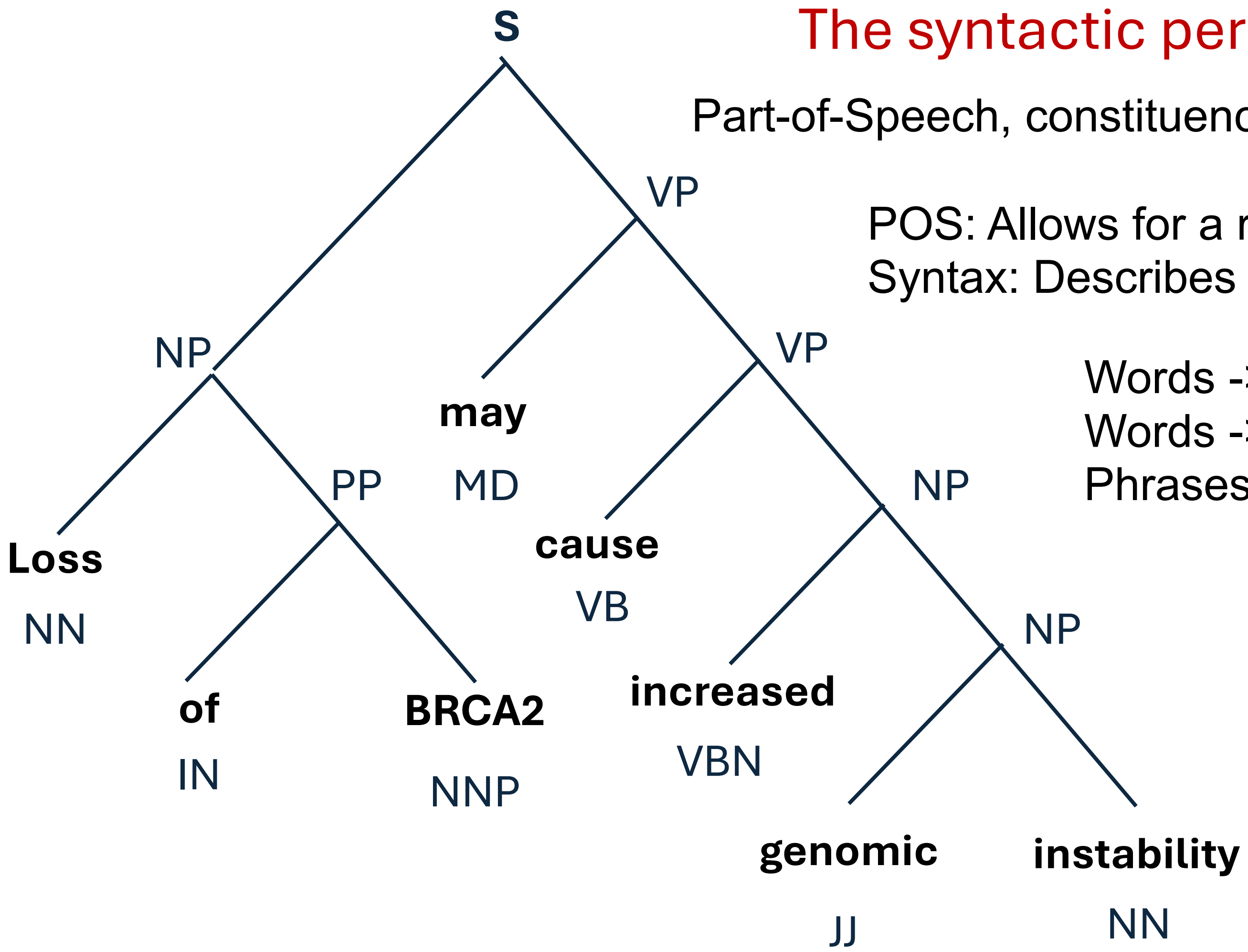
Neural vs Formal Representations



Representing sentences

Formal perspectives on sentence representation: Syntax

‘Loss of BRCA2 may cause increased genomic instability.’

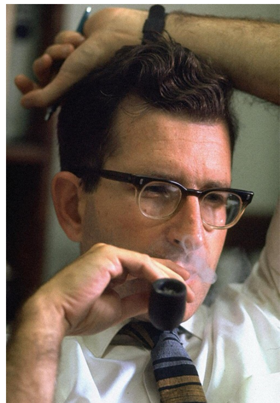


The syntactic perspective

Part-of-Speech, constituency, dependencies, ...

POS: Allows for a robust categorical system
Syntax: Describes how diferente types of word connect

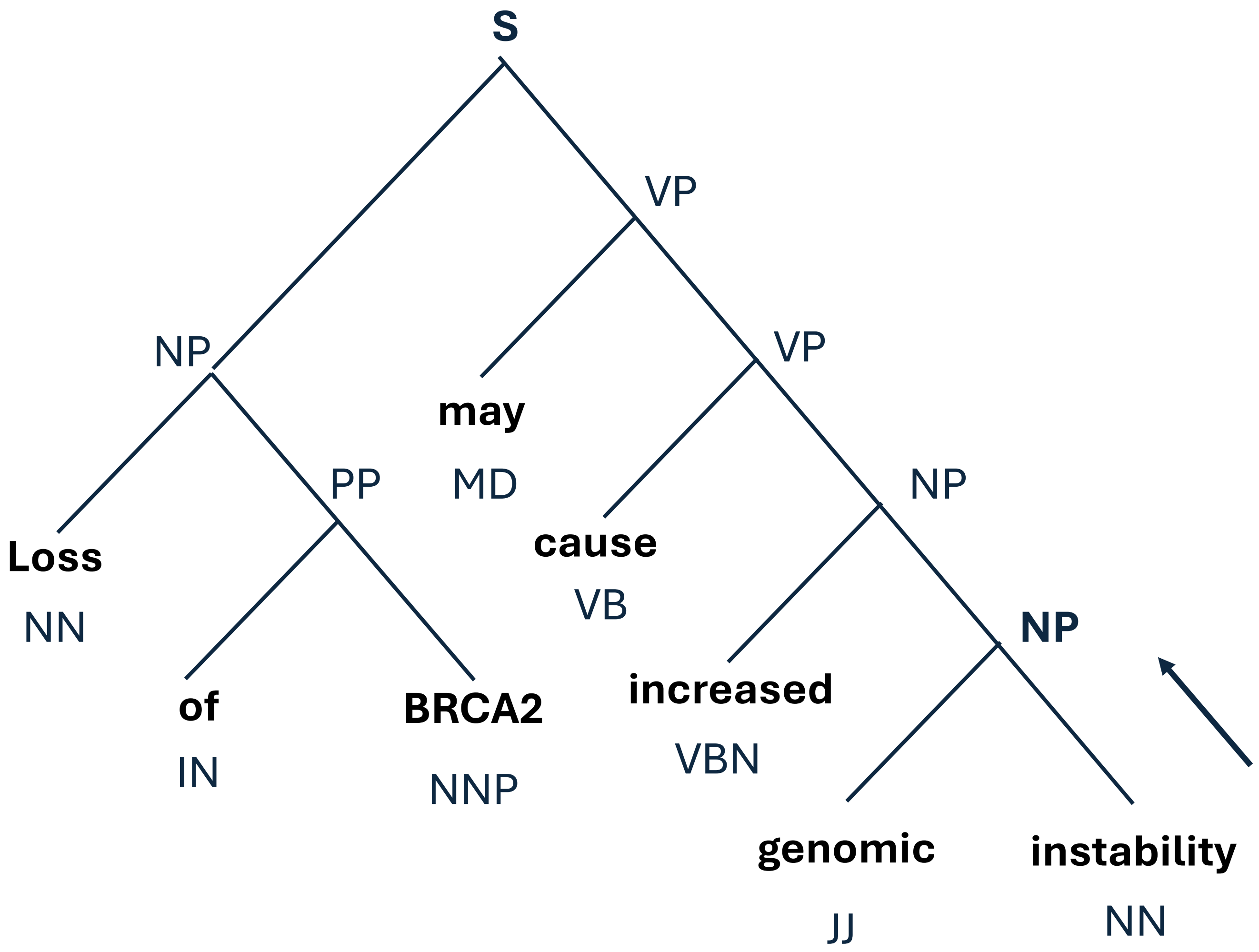
Words -> POS
Words -> Phrases
Phrases -> Syntactically correct sentence



Montague Semantics

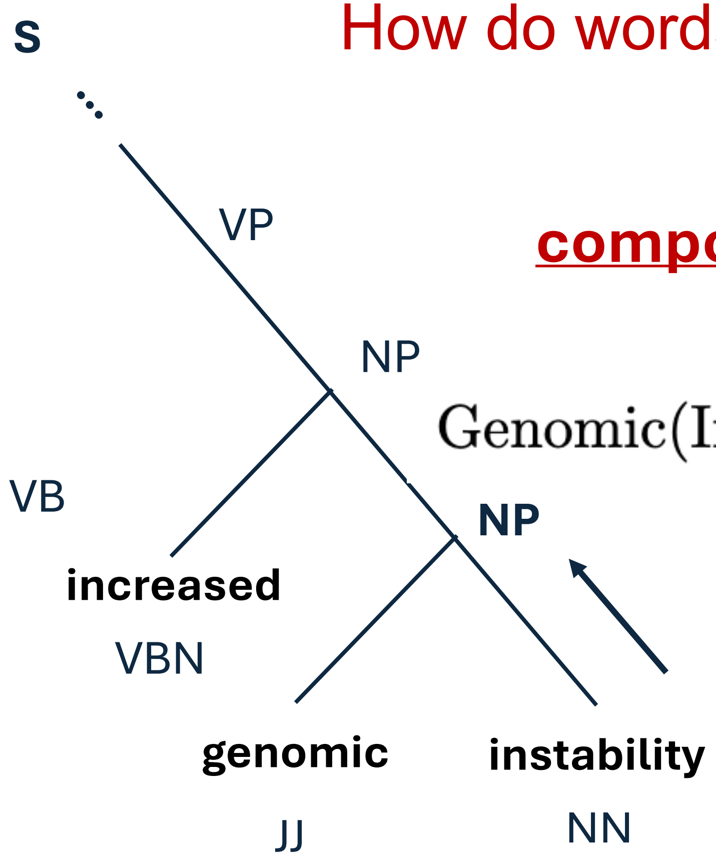
Adding the set-theoretical/functional perspective (Montague semantics)

‘Loss of BRCA2 may cause increased genomic instability.’



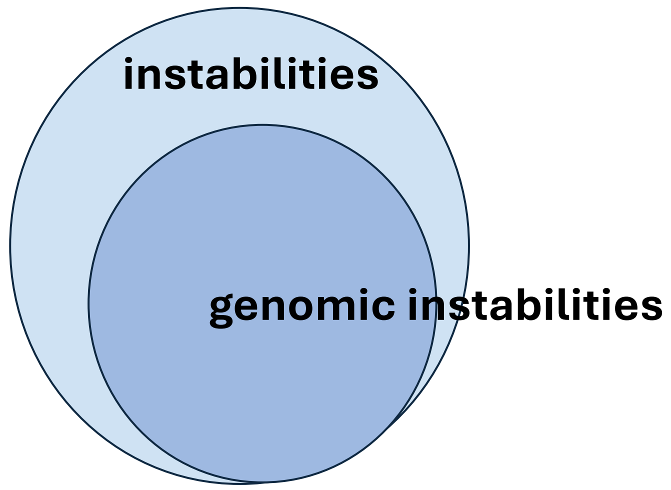
How do words, phrases functionally combine into a sentence?

compositionality



$$\text{Genomic}(\text{Instability}) = \lambda y.(\text{Instability}(y) \wedge \text{Genomic}(y))$$

$\lambda P.\lambda y.(P(y) \wedge \text{Genomic}(y))$ $\lambda y.\text{Instability}(y)$

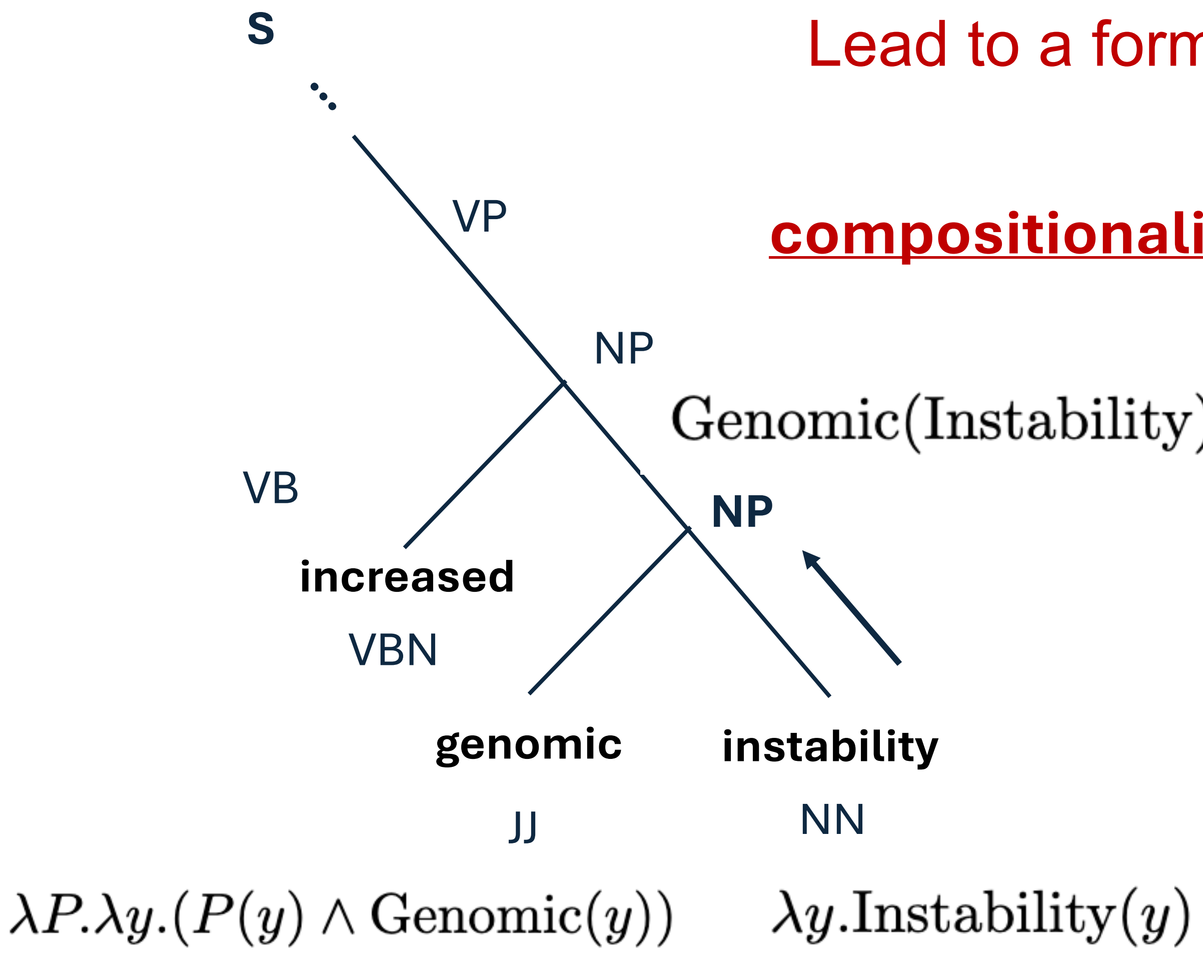


$$\exists x\,(\mathrm{Loss}(x) \wedge \mathrm{BRCA2}(x) \wedge \Diamond\,(\exists y\,(\mathrm{Increase}(y) \wedge \mathrm{Genomic}(y) \wedge \mathrm{Instability}(y)) \wedge \mathrm{Cause}(x,y)))$$

Lead to a formal meaning representation

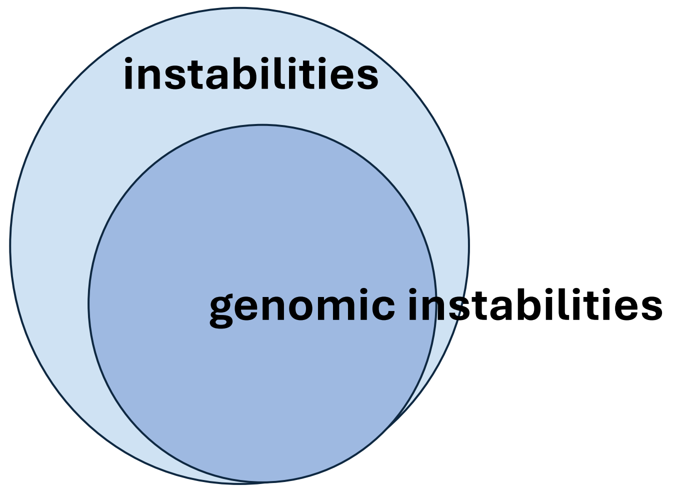
compositionality

Why formal?
- Well defined properties
- Computable expression



$$\mathrm{Genomic}(\mathrm{Instability}) = \lambda y.(\mathrm{Instability}(y) \wedge \mathrm{Genomic}(y))$$

$$\lambda P.\lambda y.(P(y) \wedge \mathrm{Genomic}(y)) \qquad \lambda y.\mathrm{Instability}(y)$$



Davidsonian Semantics

Event semantics perspective:

$$\exists e_1, e_2, e_3\ (\text{Loss}(e_1, \text{BRCA2}) \wedge \text{Cause}(e_2, e_1, \text{Increase}(e_3, \text{GenomicInstability}(e_3))) \wedge \text{Possible}(e_2))$$

1. e_1 is an event in which BRCA2 is lost.
2. e_2 is an event which is possibly caused by e_1 and results in e_3.
3. e_3 is an event of increasing genomic instability.

Neo-Davidsonian Semantics

The Neo-Davidsonian semantics separates the action or verb from its participants and properties, using distinct predicates to describe each aspect of an event.

$$\exists e_1, e_2, e_3 \left(\begin{array}{l} \text{Loss}(e_1) \land \text{Agent}(e_1, \text{BRCA2}) \land \\ \text{Cause}(e_2, e_1) \land \text{Possible}(e_2) \land \\ \text{Increase}(e_3) \land \text{Theme}(e_3, \text{GenomicInstability}) \land \\ \text{Result}(e_2, e_3) \end{array} \right)$$

1. e_1 is characterized by the predicate Loss and involves BRCA2 as an agent.
2. e_2 is a causative event possibly stemming from e_1 and results in e_3.
3. e_3 is characterized by the predicate Increase with GenomicInstability as its theme.

Abstract Meaning Representation (AMR)

‘Loss of BRCA2 may cause increased genomic instability.’

```
(cause-01
 :ARG0 (loss-01
      :ARG1 (gene
           :name (name :op1 "BRCA2")))
 :ARG1 (increase-01
      :ARG1 (instability-01
           :mod (genomic)))
 :mod (possible-01))
```

Semantic Role Labelling (Shallow semantics)

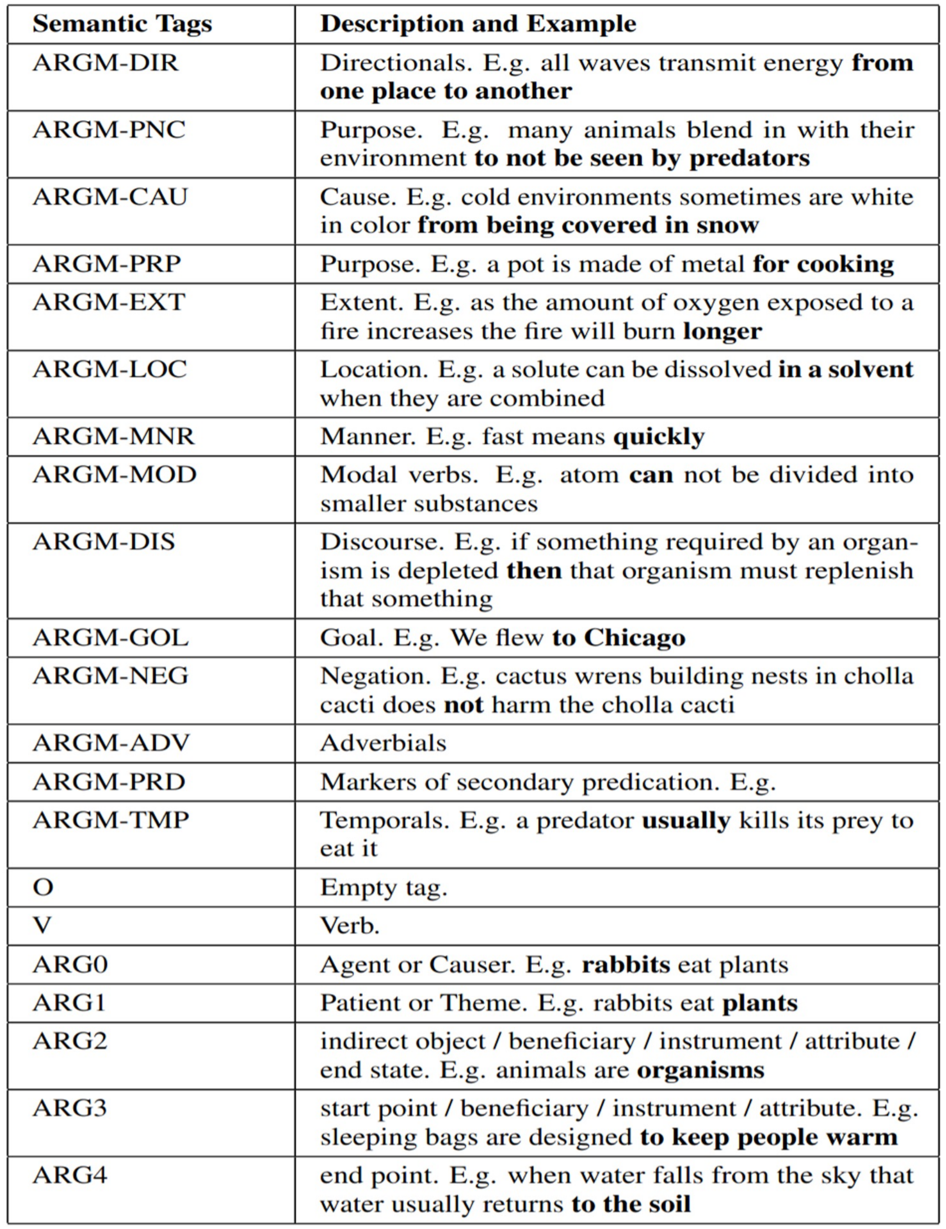
Argument Structure Theory (AST)

cause(Loss of BRCA2, increased genomic instability)

Agent **Effect**

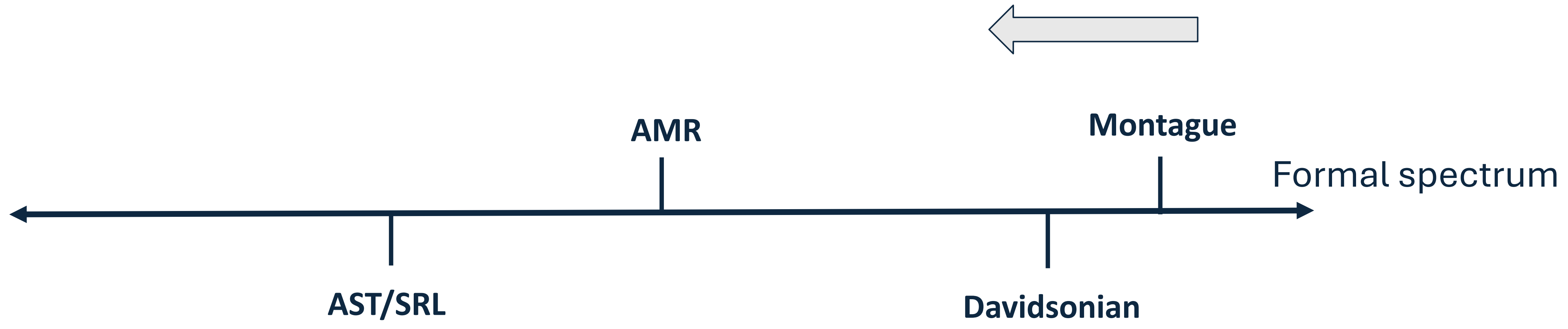
Predicate (V): "cause"
Agent (A0): "Loss of BRCA2"
Effect (A1): "increased genomic instability"

Thematic (θ) roles



Semantic Tags	**Description and Example**
ARGM-DIR	Directionals. E.g. all waves transmit energy **from one place to another**
ARGM-PNC	Purpose. E.g. many animals blend in with their environment **to not be seen by predators**
ARGM-CAU	Cause. E.g. cold environments sometimes are white in color **from being covered in snow**
ARGM-PRP	Purpose. E.g. a pot is made of metal **for cooking**
ARGM-EXT	Extent. E.g. as the amount of oxygen exposed to a fire increases the fire will burn **longer**
ARGM-LOC	Location. E.g. a solute can be dissolved **in a solvent** when they are combined
ARGM-MNR	Manner. E.g. fast means **quickly**
ARGM-MOD	Modal verbs. E.g. atom **can** not be divided into smaller substances
ARGM-DIS	Discourse. E.g. if something required by an organism is depleted **then** that organism must replenish that something
ARGM-GOL	Goal. E.g. We flew **to Chicago**
ARGM-NEG	Negation. E.g. cactus wrens building nests in cholla cacti does **not** harm the cholla cacti
ARGM-ADV	Adverbials
ARGM-PRD	Markers of secondary predication. E.g.
ARGM-TMP	Temporals. E.g. a predator **usually** kills its prey to eat it
O	Empty tag.
V	Verb.
ARG0	Agent or Causer. E.g. **rabbits** eat plants
ARG1	Patient or Theme. E.g. rabbits eat **plants**
ARG2	indirect object / beneficiary / instrument / attribute / end state. E.g. animals are **organisms**
ARG3	start point / beneficiary / instrument / attribute. E.g. sleeping bags are designed **to keep people warm**
ARG4	end point. E.g. when water falls from the sky that water usually returns **to the soil**

Formality spectrum



Less structure
Broader categories

More structure

+Robust parsing
+ Flexible
- Approximative
- Control

+ Expressive
+ Granular
+ Controlled
+ Well-defined properties
- More brittle

Representing complex sentences

A fluoroscopic study which is known as an upper gastrointesti-
nal series is typically the next step in management, although
if volvulus is suspected, caution with non water soluble con-
trast is mandatory as the usage of barium can impede surgi-
cal revision and lead to increased post operative complications.

Representing complex sentences

A fluoroscopic study which is known as an upper gastrointestinal series is typically the next step in management, although if volvulus is suspected, caution with non water soluble contrast is mandatory as the usage of barium can impede surgical revision and lead to increased post operative complications.

A fluoroscopic study is typically the next step in management.

Proposition 1

This fluoroscopic study is known as an upper gastrointestinal series.

Proposition 2

Volvulus is suspected.

Proposition 3

Caution with non water soluble contrast is mandatory.

Proposition 4

The usage of barium can impede surgical revision.

Proposition 5

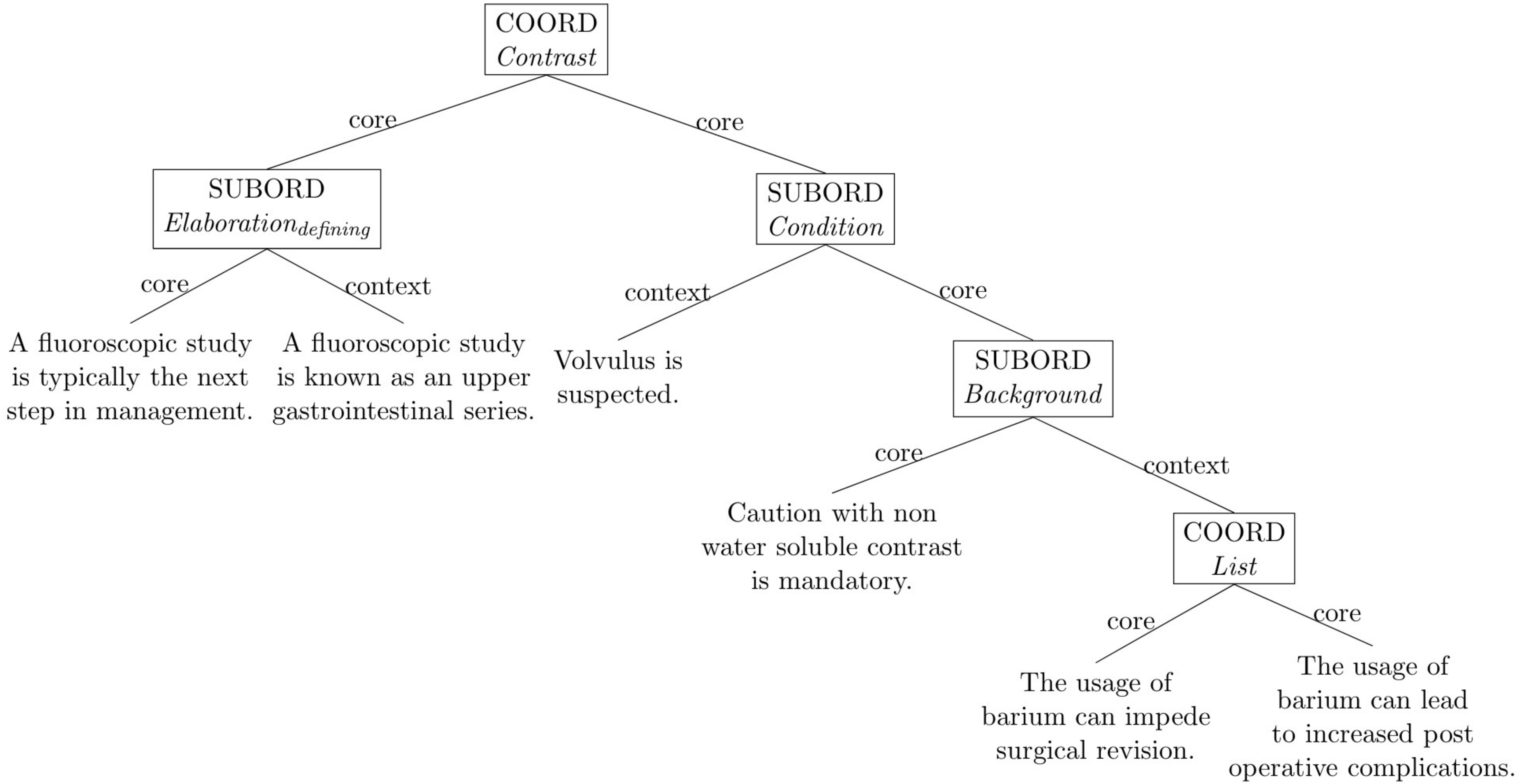
The usage of barium can lead to increased post operative complications.

Proposition 6

Complex Sentence Representation

	Clausal/Phrasal type	Hierarchy	# rules
	Clausal disembedding		
1	Coordinate clauses	coordinate	1
2	Adverbial clauses	subordinate	6
3a	Relative clauses (non-restrictive)	subordinate	5
3b	Relative clauses (restrictive)	subordinate	4
4	Reported speech	subordinate	4
	Phrasal disembedding		
5	Coordinate verb phrases	coordinate	1
6	Coordinate noun phrases	coordinate	2
6	Participial phrases	subordinate	4
8a	Appositions (non-restrictive)	subordinate	1
8b	Appositions (restrictive)	subordinate	1
9	Prepositional phrases	subordinate	3
10	Adjectival and adverbial phrases	subordinate	2
11	Lead NPs	subordinate	1
	Total		35

Niklaus, Cetto, Freitas, Handschuh ACL (2019)


COORD
Contrast
core
core
SUBORD
$Elaboration_{defining}$
SUBORD
Condition
core
context
context
core
A fluoroscopic study is typically the next step in management.
A fluoroscopic study is known as an upper gastrointestinal series.
Volvulus is suspected.
SUBORD
Background
core
context
Caution with non water soluble contrast is mandatory.
COORD
List
core
core
The usage of barium can impede surgical revision.
The usage of barium can lead to increased post operative complications.

Getting the concepts right: representing NL definitions

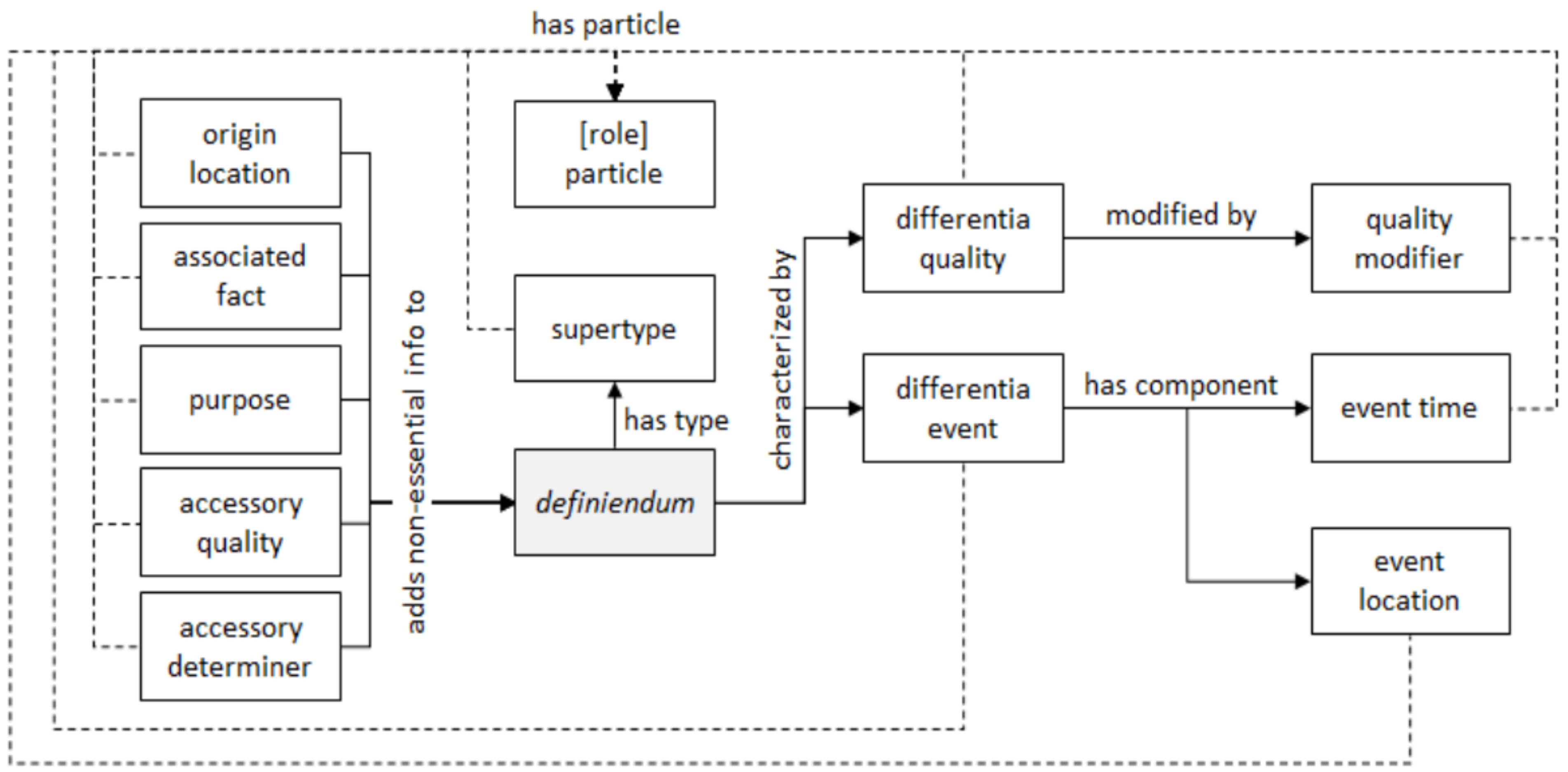
- Essential attributes of a conceptualisation.
- Abundance of NL definitions in discourse.
- **Definition semantic roles (DSR):** Decomposing conceptual components.

DEFINIENDUM **DIFFERENTIA QUALITY** **SUPERTYPE** **DIFFERENTIA-EVENT**

Homologous recombination repair is a DNA repair process that includes the invasion of an undamaged DNA molecule by a damaged molecule of identical or very similar sequence.

Santos, Freitas, Handschuh, AAAI (2018, 2019)

Representing definitional sentences



Q: Can these formal categories inform better conceptual representations?

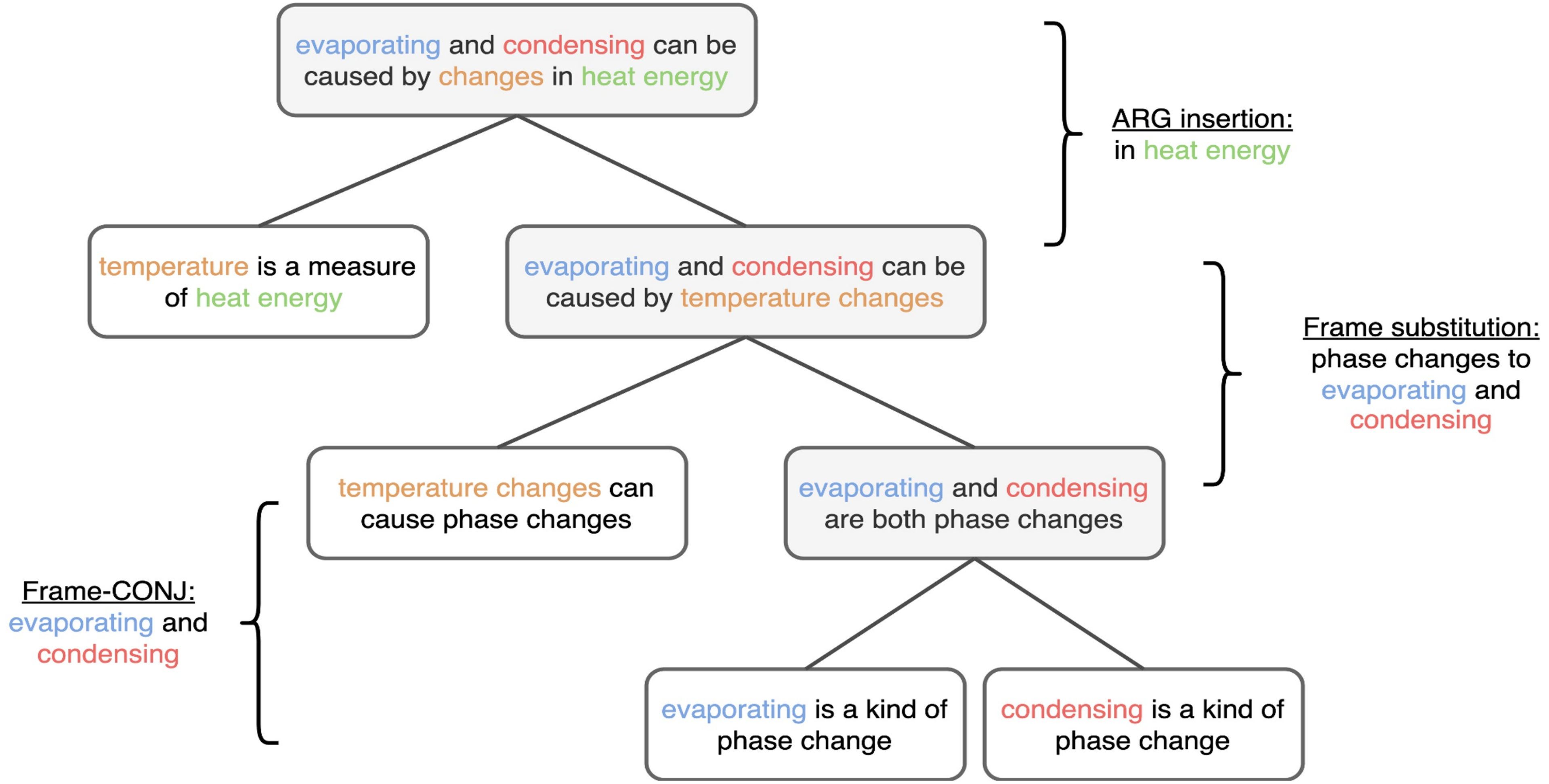
Santos, Freitas, Handschuh, CogAlex (2016)

Formal natural language inference

Natural Language Inference

E.g. EntailmentBank, each step shows distinct reasoning behaviour (i.e., substitution, conjunction, etc).

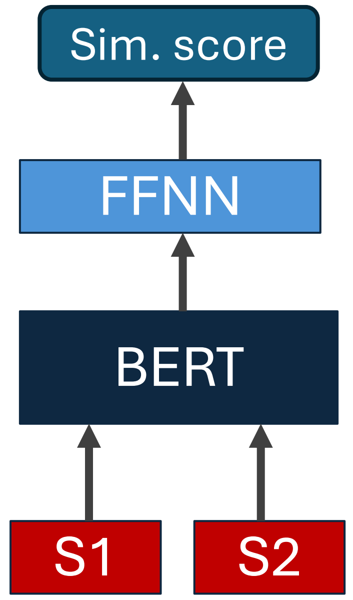
Question: in which way are evaporation and condensation are similar?
Answer: both are caused by phase changes in heat energy



evaporating and condensing can be caused by changes in heat energy
ARG insertion:
in heat energy
temperature is a measure of heat energy
evaporating and condensing can be caused by temperature changes
Frame substitution:
phase changes to evaporating and condensing
temperature changes can cause phase changes
evaporating and condensing are both phase changes
Frame-CONJ:
evaporating and condensing
evaporating is a kind of phase change
condensing is a kind of phase change

Contrasting to neural models

Cross-encoder model for sentence similarity



Scalability problem, pair-wise comparison

S1: Loss of BRCA2 may cause increased genomic instability.

S2: Genomic instability could increase as a result of BRCA2 loss.

S3: This is an unrelated sentence.

The SBERT Model

Reymers & Gurevych (EMNLP, 2019)

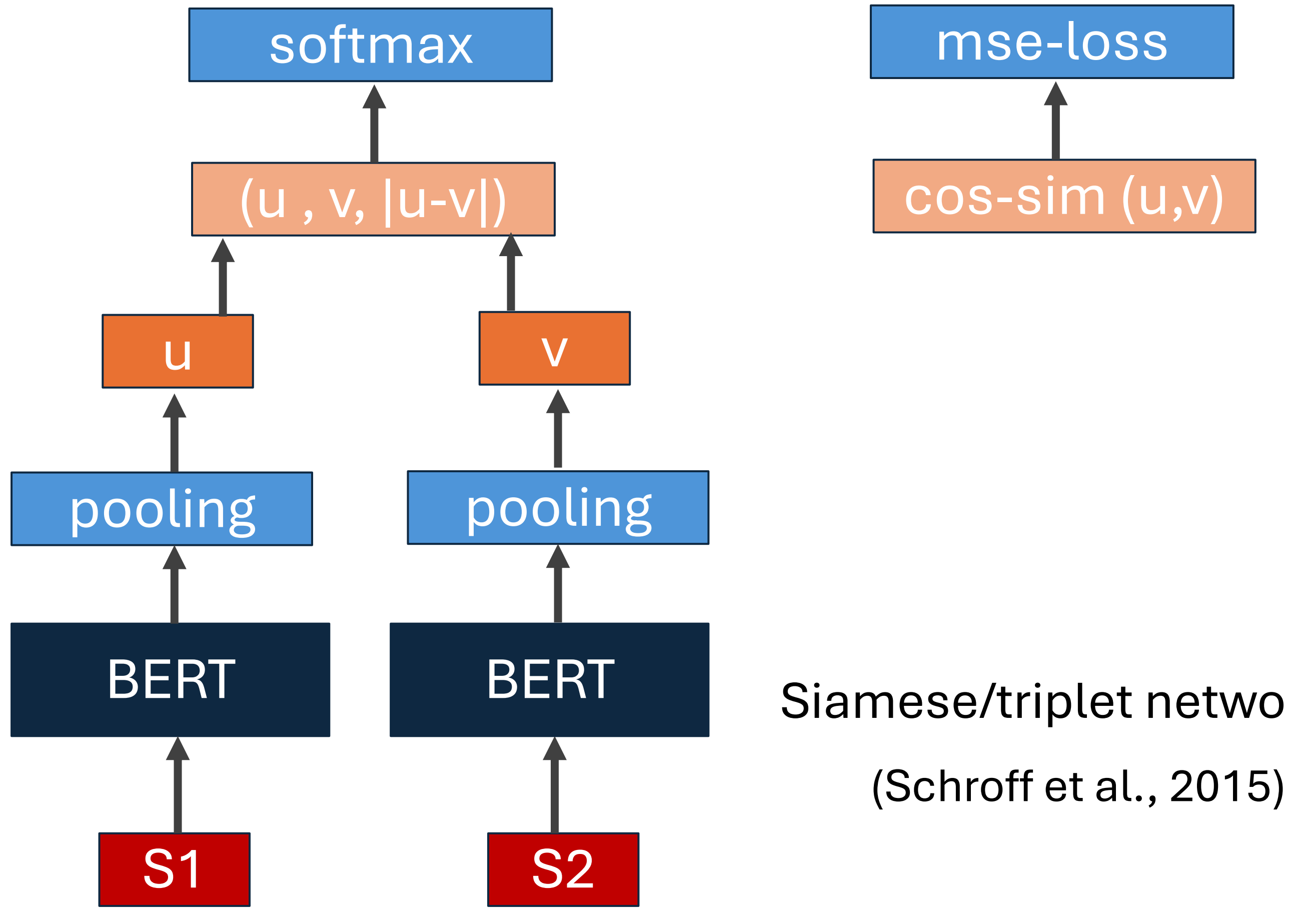
classification objective function

$$o = \mathrm{softmax}(W_t(u, v, |u - v|))$$

regression objective function

triplet objective function

$$max(||s_a - s_p|| - ||s_a - s_n|| + \epsilon, 0)$$



(Sa, Sp, Sn)

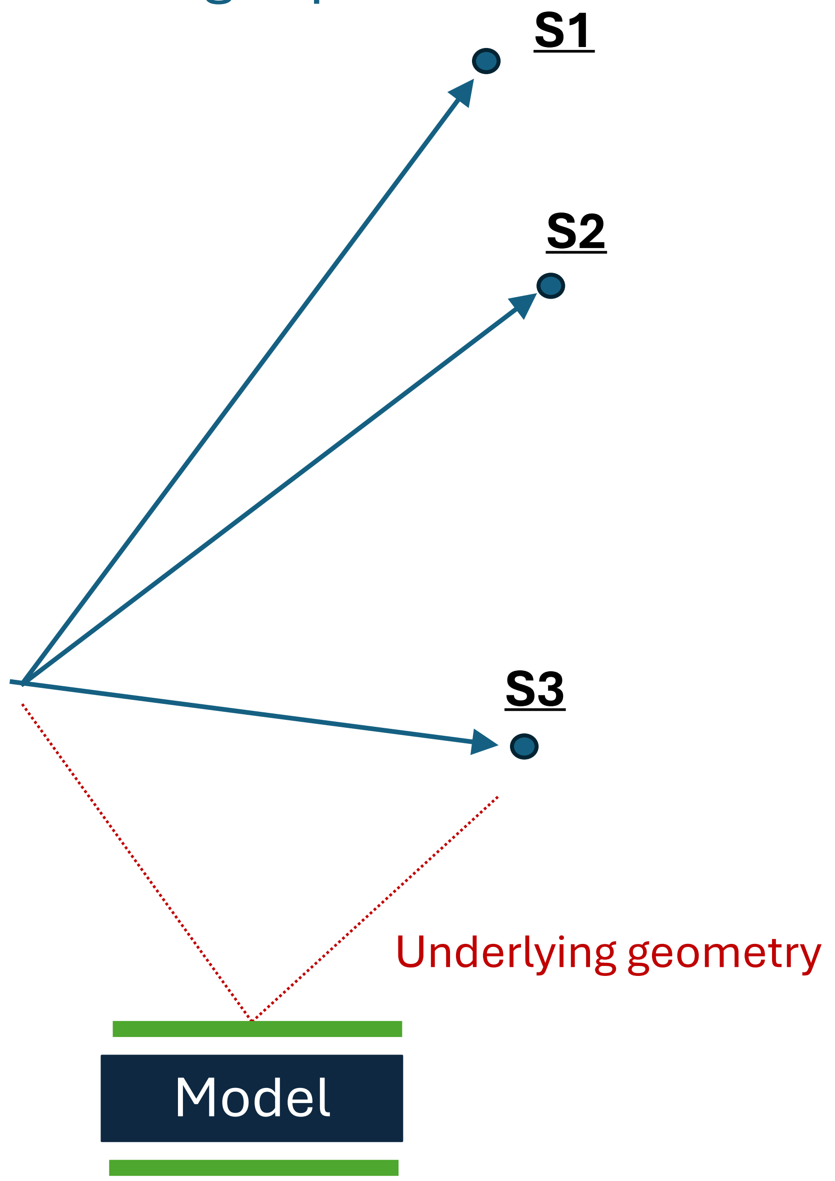
Siamese/triplet network structure

(Schroff et al., 2015)

SNLI (Bowman et al., 2015)
Multi-Genre NLI (Williams et al., 2018)

Sentence embeddings

Embeddings spaces



- Syntactic, semantic, compositional content, inference properties packaged as a vector
- Distributed

S1: Loss of BRCA2 may cause increased genomic instability.

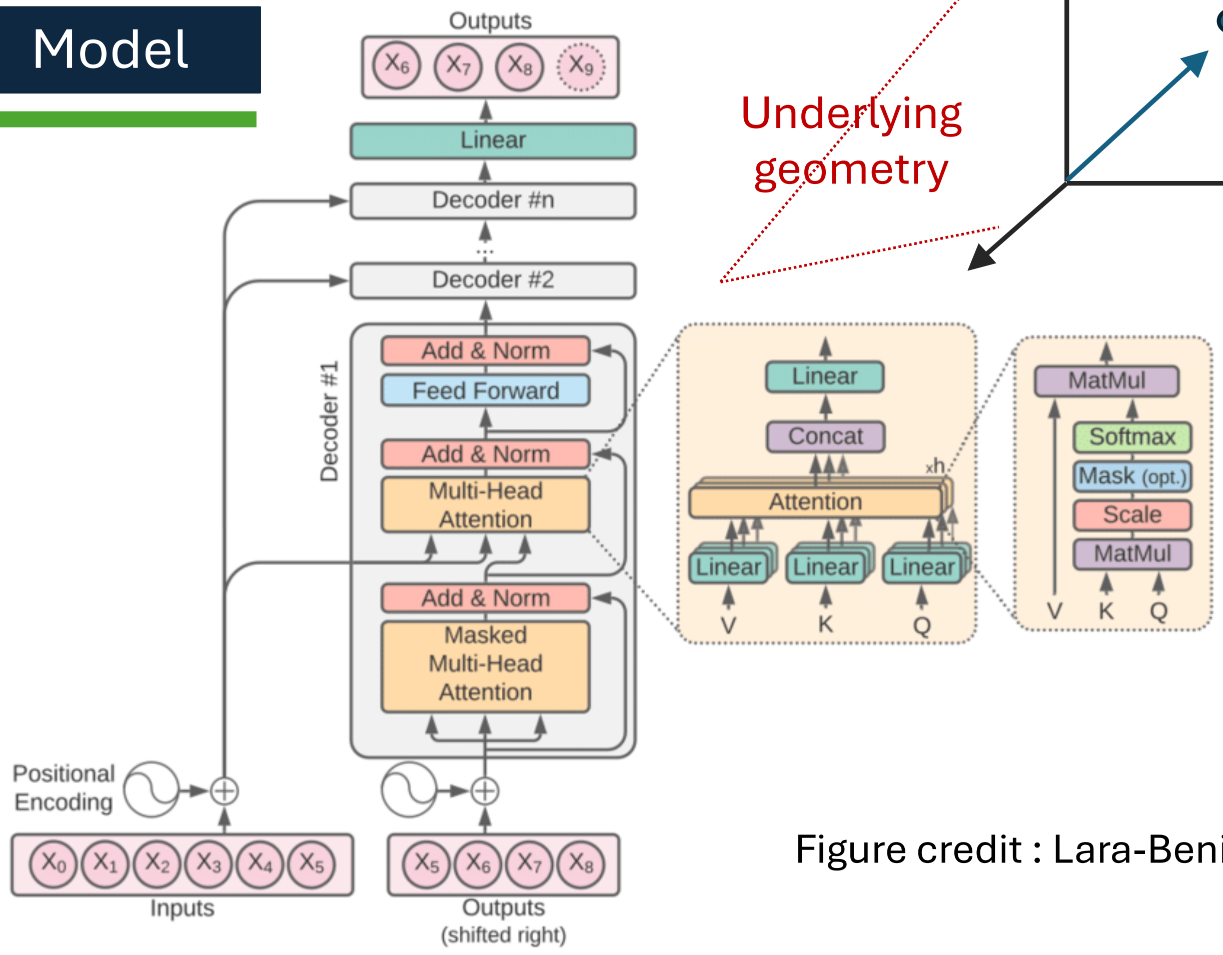
S2: Genomic instability could increase as a result of BRCA2 loss.

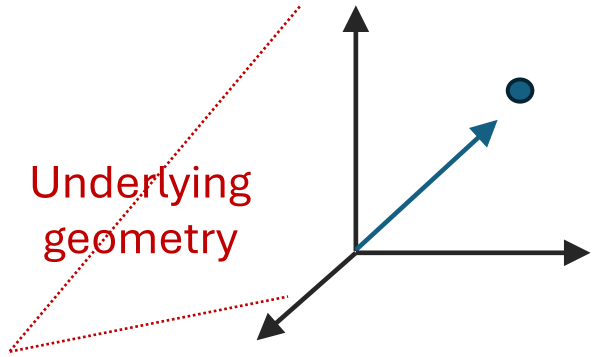
S3: This is an unrelated sentence.

Generative perspective

Model

... instability





Fn

- State space of latent semantic features
- Expressive latent semantics subspaces. (enabled by the multi-layer MHA, MLP, normalisation/residual components, etc)
- Not trivial to define a sentence representation

Figure credit : Lara-Benitez et al, 2020

Loss of BRCA2 may cause increased genomic ...

Contrasting Properties (Representation)

Neural	**Formal/Symbolic**
Approximative	Exact
High-dimensional vector space/geometrical	Set-based/logical
Similarity-based operations	Symbolic operations
Disambiguation ‘on-read’	Disambiguation ‘on-write’
Syntactic, semantic & content entanglement	Fully disentangled representation
Latent/Poorly interpretable ling. features	Explicit ling. features

Contrasting Properties (Inference)

Neural	Formal/Symbolic
Approximative inference	Exact inference
Content centered/Material inference	Syntax centered/Formal inference
Entangled inference relations	Well-defined inference relations
Low inference control	High inference control
Robust to incompleteness, variability	Requires completeness, brittle
Short-distance inference relations	Long-distance inference relations
Scalable	Not-scalable
Less interpretable	More interpretable

Neuro-symbolic NLP (objectives)

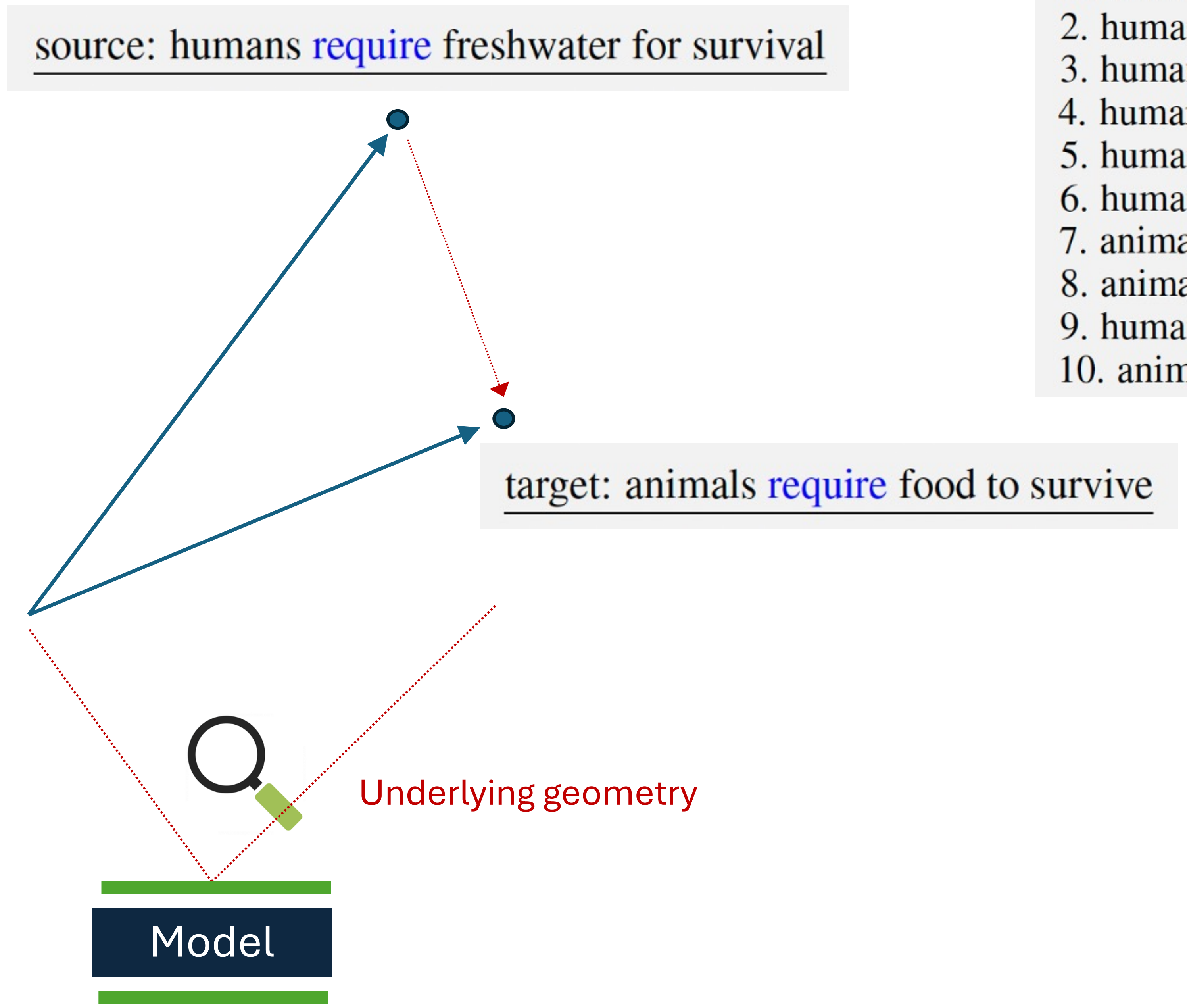
Produce representations of language which allows for the constructive integration of both perspectives.

(best of both worlds)

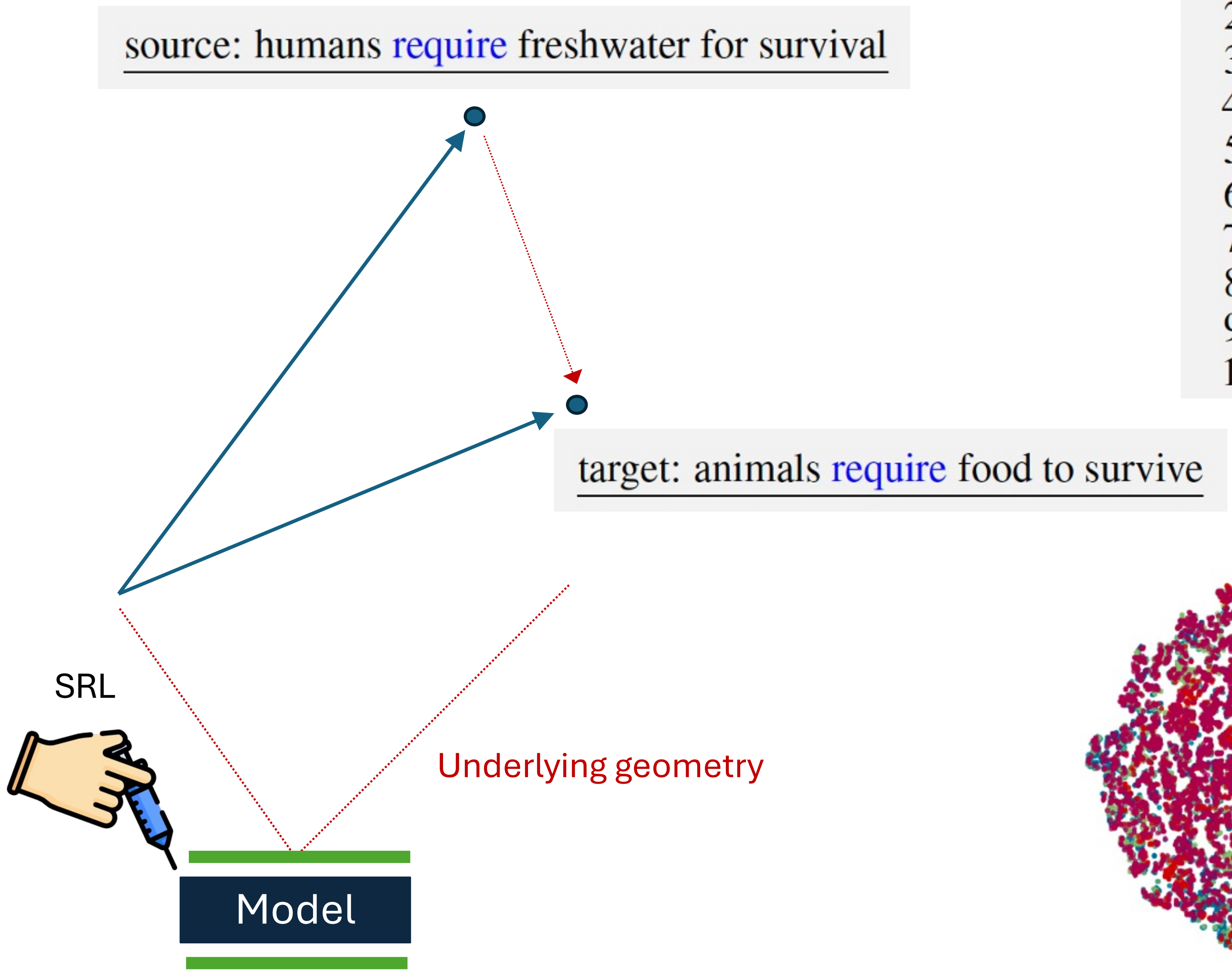
Embeddings spaces

Semantically inconsistent space

1. humans require water and food through fossil fuels
2. humans require water for survival
3. humans produce small amounts of consumer food
4. human has a positive impact on a plant's survival
5. humans convert food into animal prey
6. humans make food for themselves by eating
7. animals require food for survival
8. animals require nutrients from the air
9. humans eat plants for food
10. animals require food for survival

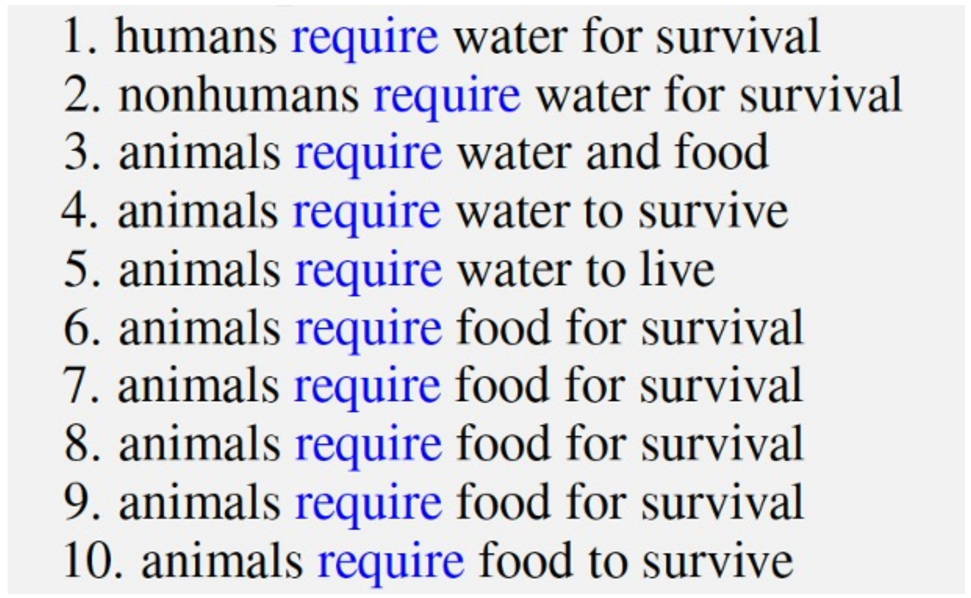

source: humans require freshwater for survival
target: animals require food to survive
Underlying geometry
Model

Embeddings spaces

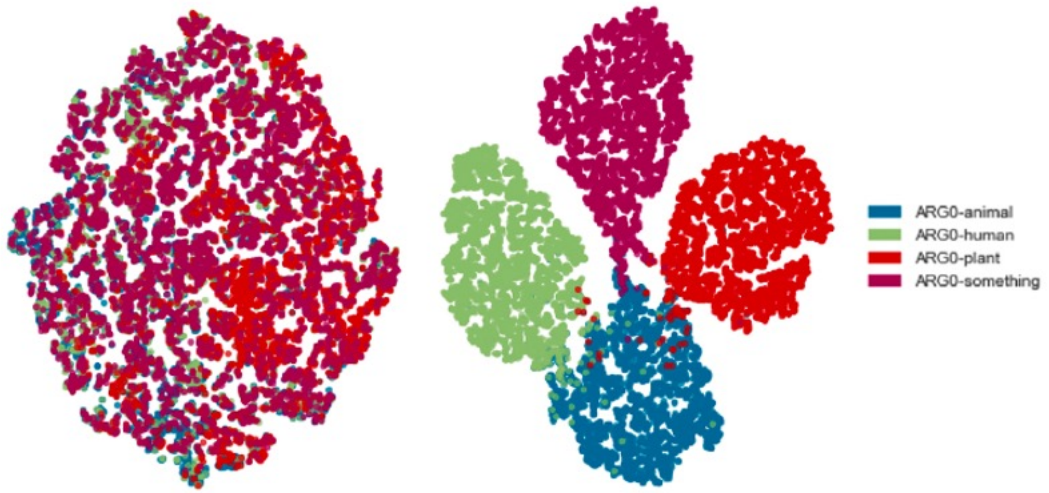

source: humans require freshwater for survival

target: animals require food to survive

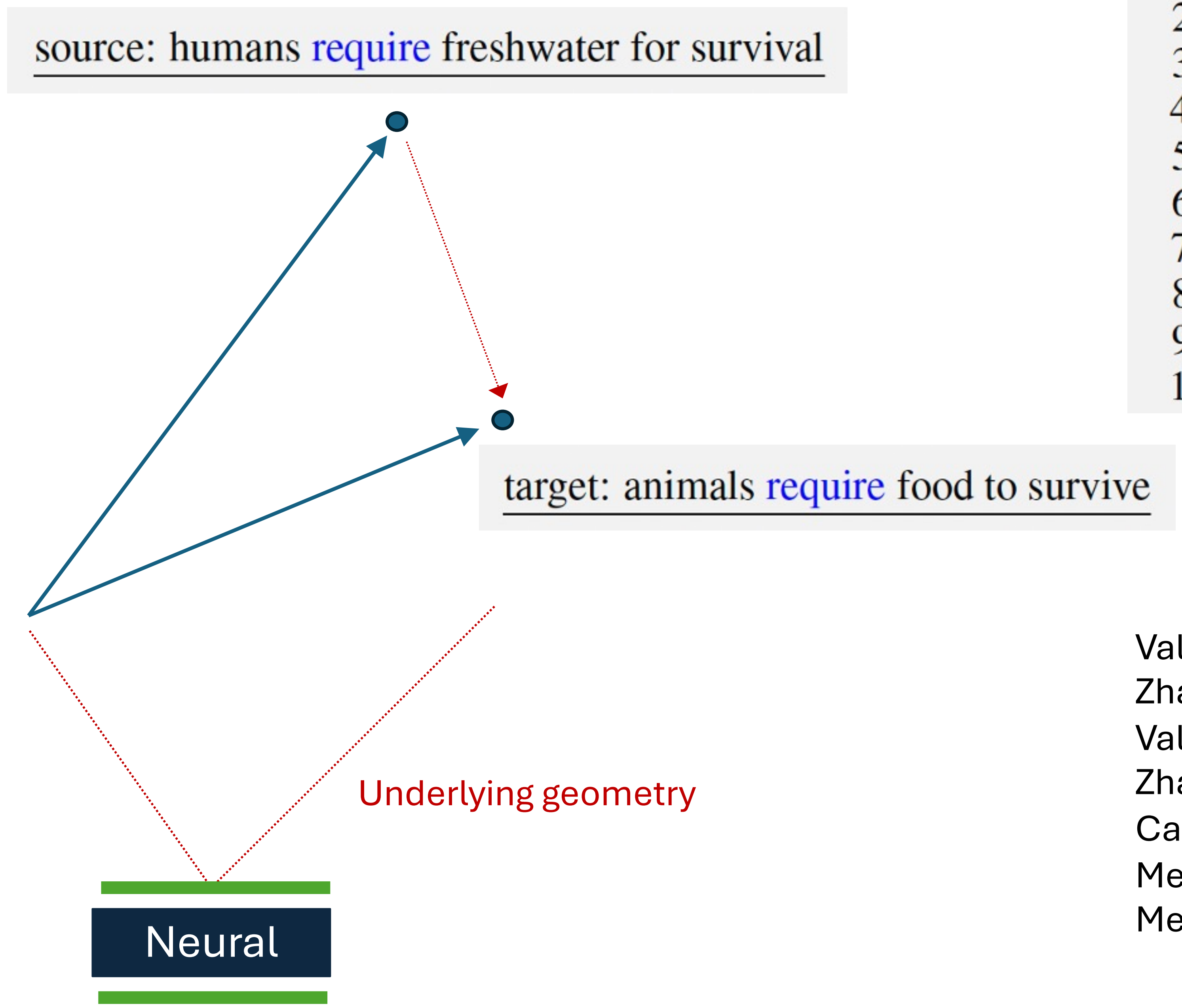
Improving semantic consistency


1. humans require water for survival
2. nonhumans require water for survival
3. animals require water and food
4. animals require water to survive
5. animals require water to live
6. animals require food for survival
7. animals require food for survival
8. animals require food for survival
9. animals require food for survival
10. animals require food to survive

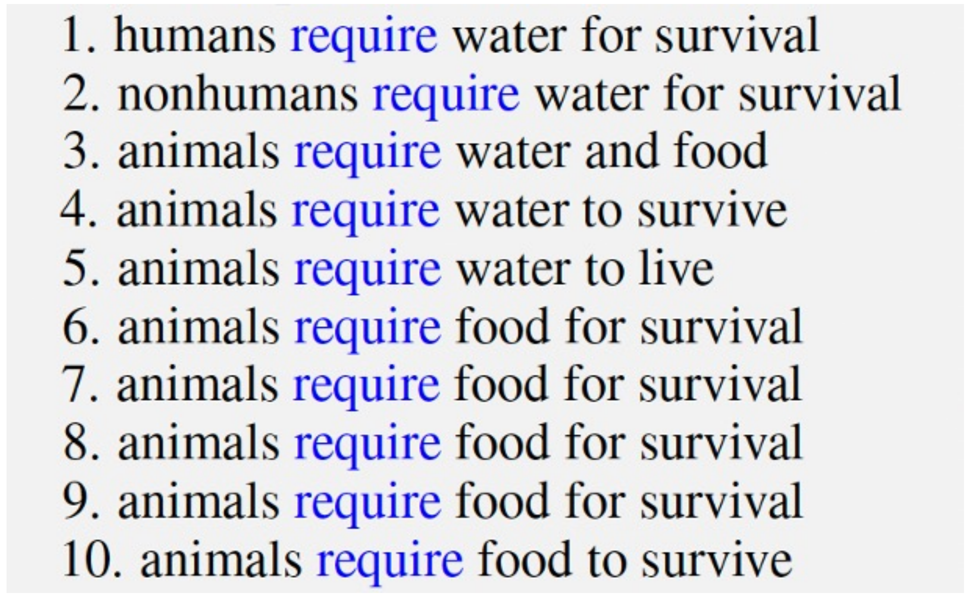
+ separation
+ disentanglement



Embeddings spaces


source: humans require freshwater for survival
target: animals require food to survive
Underlying geometry
Neural

Improving semantic consistency



1. humans require water for survival
2. nonhumans require water for survival
3. animals require water and food
4. animals require water to survive
5. animals require water to live
6. animals require food for survival
7. animals require food for survival
8. animals require food for survival
9. animals require food for survival
10. animals require food to survive

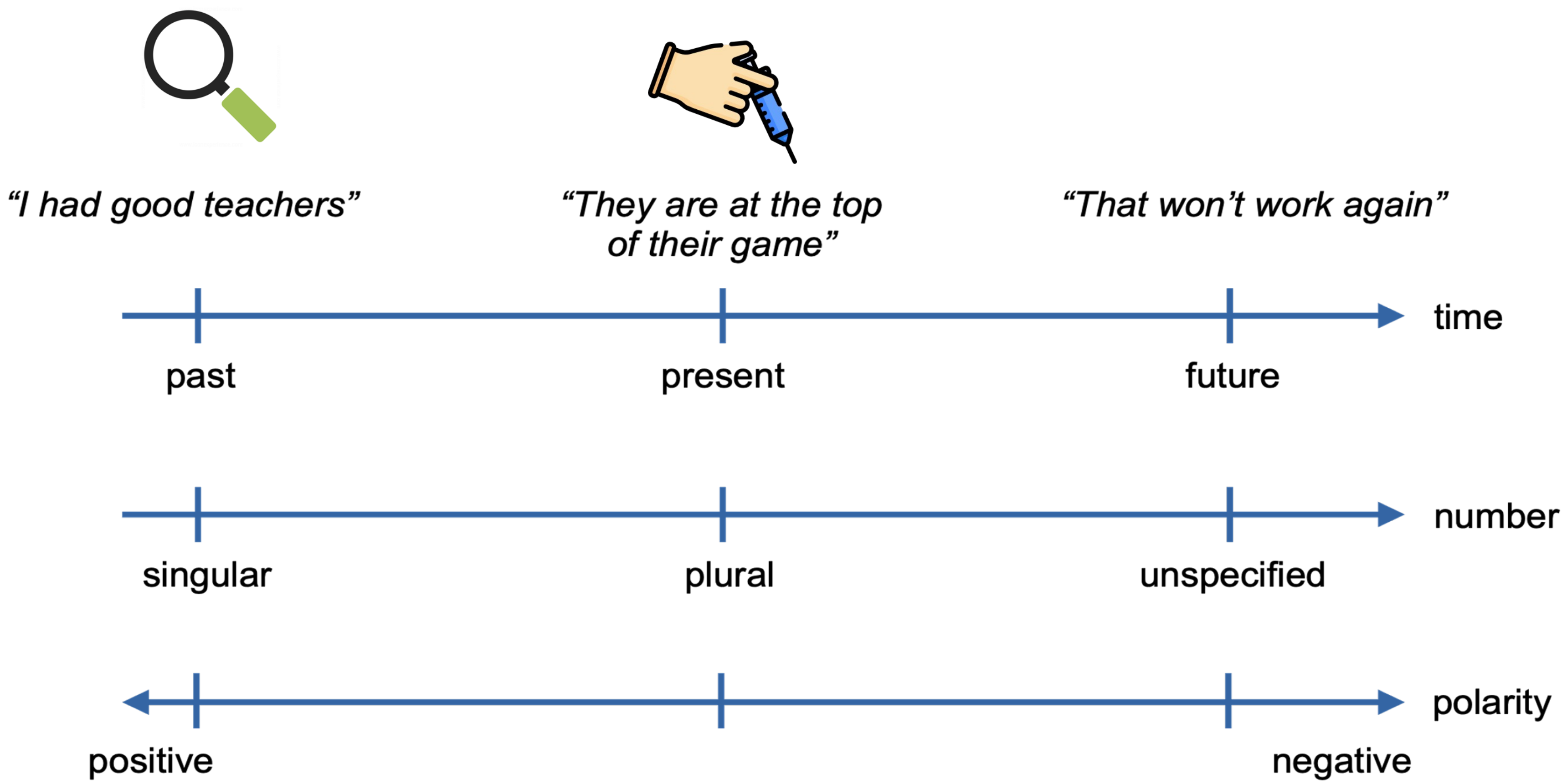
+ separation
+ disentanglement

Valentino et al, NAACL (2024)
Zhang et al, NAACL (2024)
Valentino et al, EACL (2024)
Zhang et al, EACL Findings (2024)
Carvalho et al, EACL Findings (2023)
Mercatali et al, NeurIPS (2022)
Mercatali & Freitas, EMNLP Findings (2021)

Language disentanglement

Separating the different dimensions of a model's latent space with specific linguistic feature (**descriptively** and **prescriptively**).

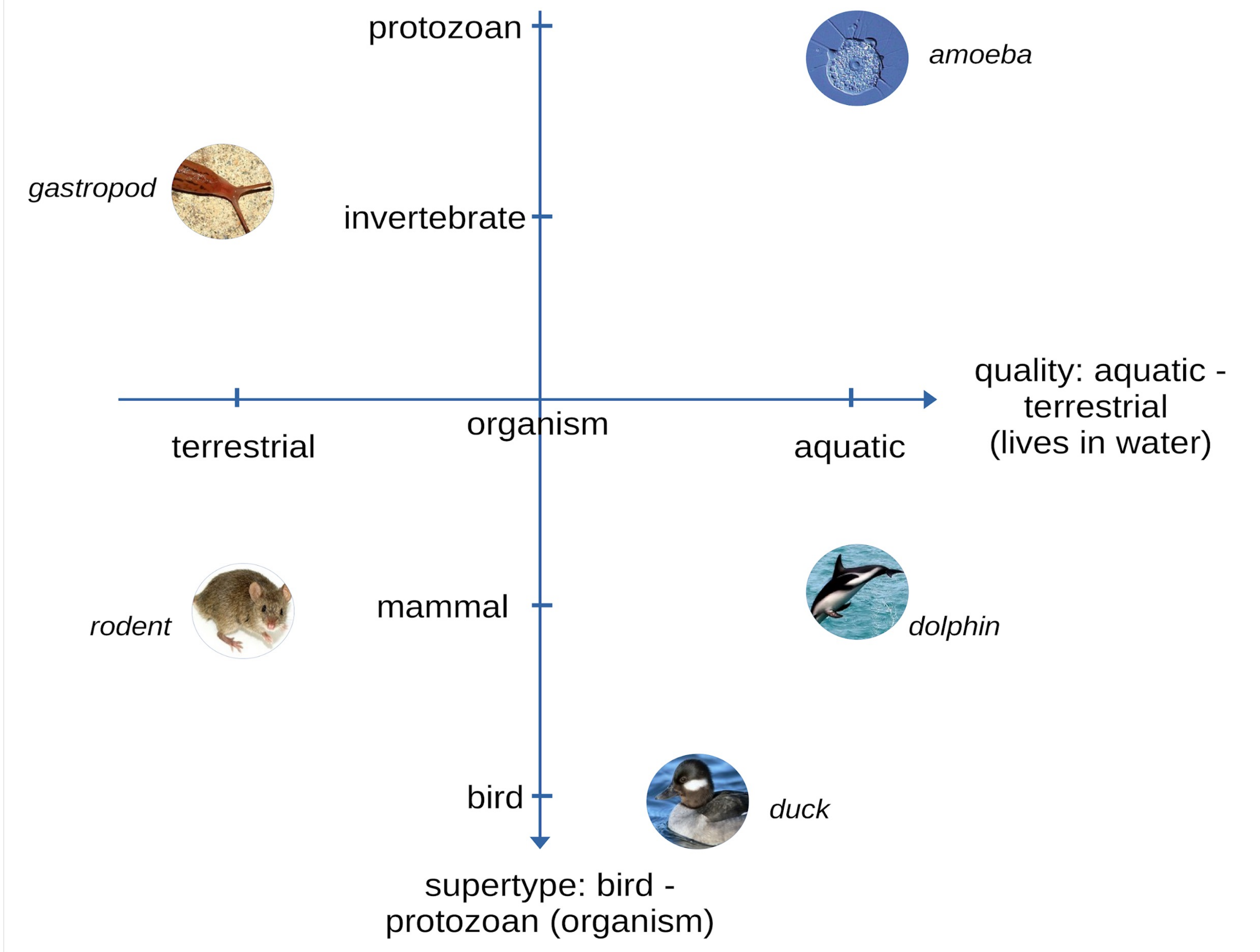
Linguistic factors



Language disentanglement

Abstract conceptual factors

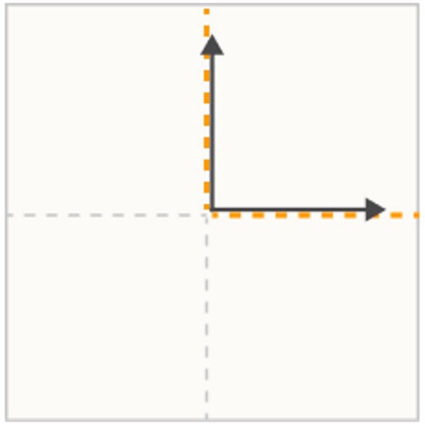
(more content-based)



Language disentanglement

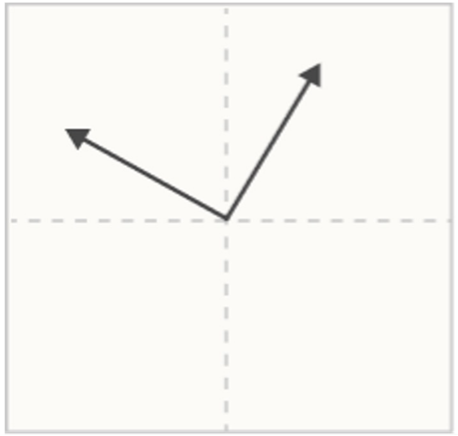
Disentanglement: features and dimensions alignment **(privileged)**. In facial images, for example, eyes, nose, mouth, etc., can be disentangled and localised in latent space.

“direction determinatew the features”



In a **privilged basis**, there is an incentive for features to align with basis dimensions. This doesn’t necessarily mean they will.

Examples: conv net neurons, transformer MLPs



In a **non-privileged basis**, features can be embedded in any direction. There is no reason to expect basis dimensions to be special.

Examples: word embeddings, transformer residual stream

source: https://transformer-circuits.pub/2022/solu/index.html#section-3-2

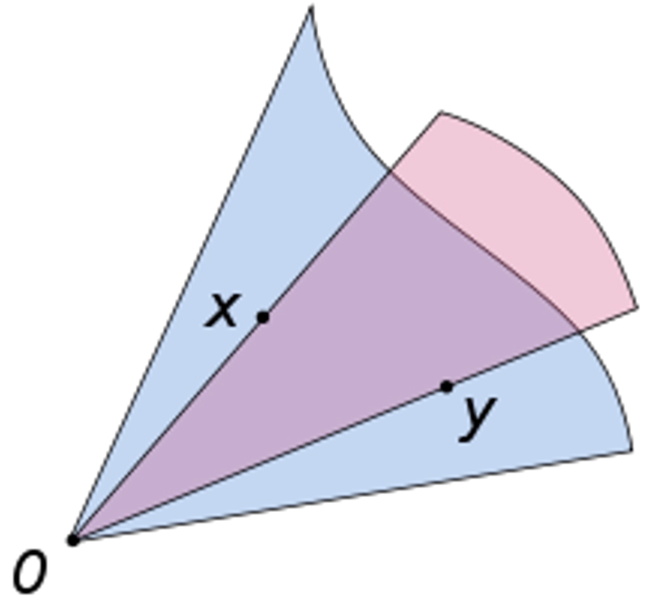
In transformers, however, the token embeddings, residual streams, and attention vectors are **non-privileged**, where more dimensions contribute to a feature.

***Q:** In sentence space, can sentence vectors with the same feature have similar directions in a **subspace**?*

Cone (as a semantic subspace)

Definition: In linear algebra, a **cone**, sometimes called a linear cone, is a **subset of a vector space** that is closed under positive scalar multiplication. that is, C is a cone if $x \in C$ implies $sx \in C$ for every positive scalar.

Convex cone: A cone C is a convex cone if $\alpha x + \beta y$ belongs to C, for any positive scalars α, β, and any x, y in C. A cone C is convex if and only if $C + C \subseteq C$.



Q: If x and y are sentence vectors, is there a convex cone available where all ***αx + βy*** *in this cone hold the same "feature" of those sentence vectors?*

source: from https://en.wikipedia.org/wiki/Convex_cone

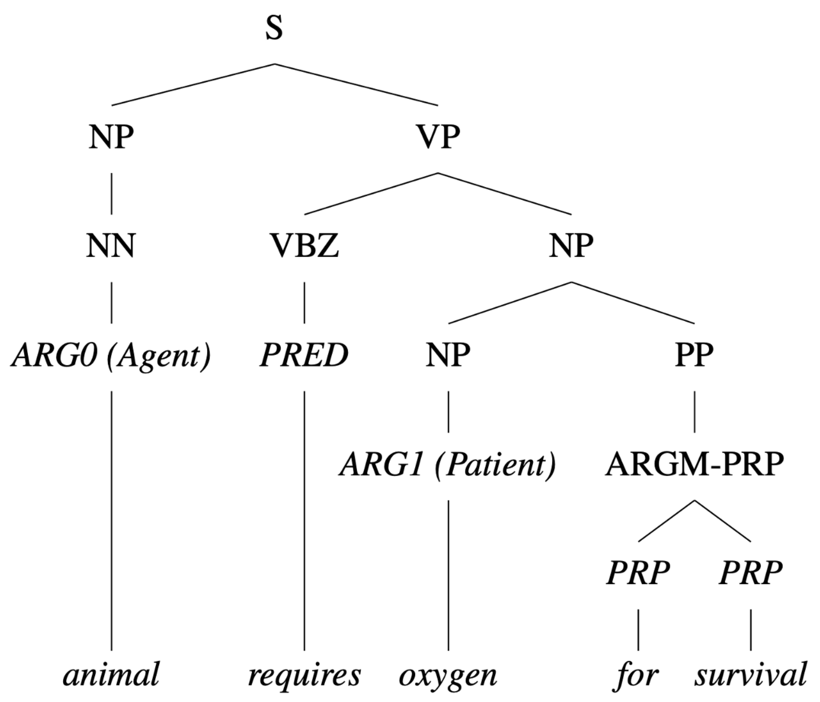
Disentangled sentence semantics

Sentence semantics: From *argument structure theory (AST)*, the sentence semantics is modelled by the relation between pred-arg structure, the associated semantic roles and distributional word content.

We simplify the **sentence semantics** as a composition of **role-content relations**:

$$sem(s) = \underbrace{t_1(c_1, r_1)}_{i.e., ARG0-animals} \oplus \cdots \oplus \underbrace{t_i(c_i, r_i)}_{PRP-survival}$$

$$\underbrace{animals}_{ARG0} \; \underbrace{require}_{PRED} \; \underbrace{oxygen}_{ARG1} \; \underbrace{for\ survival}_{ARGM-PRP}$$



Q: *Can we define separated convex role-content cones within the sentence space?*

[1] Ray S Jackendoff. 1992. Semantic structures, volume 18. MIT press.
[2] Beth Levin. 1993. English verb classes and alternations: A preliminary investigation. University of Chicago press.
[3] Malka Rappaport Hovav and Beth Levin. 2008. The english dative alternation: The case for verb sensitivityl. Journal of linguistics, 44(1):129–167.

Sentence semantic disentanglement

$$sem(s) = \underbrace{t_1(c_1, r_1)}_{i.e., ARG0-animals} \oplus \cdots \oplus \underbrace{t_i(c_i, r_i)}_{PRP-survival}$$

If the sentence semantics can be disentangled under $\bigoplus$, *sem(s)* can be decomposed into:

$$sem(s) = \{t_1(c_1, r_1)\} \oplus \cdots \oplus \{t_i(c_i, r_i)\}$$

where each set represents a specific role-content cluster resolved to a hypersolid over the latent space.

Given a set of *N* sentences with same *t(c,r)* but different *sem(s)*, the *t(c,r)* can be formed:

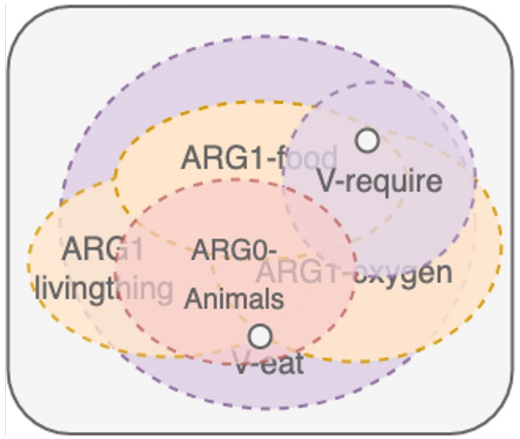
$$\{sem(s_1), ..., sem(s_N)\} = \{t(c, r)\}_{\times N} \oplus \{...\}$$

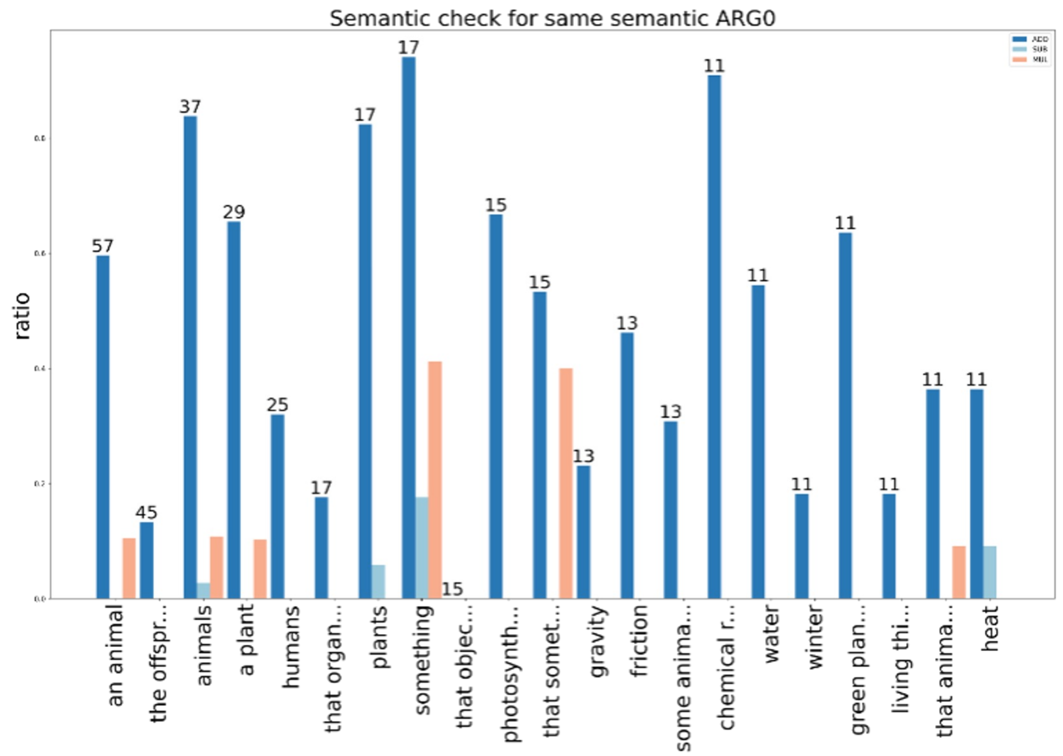
Therefore, we can evaluate the semantic disentanglement (i.e., **natural clustering property** [1]) by evaluating the density (recall) within same *t(c,r)* and separability (accuracy) between different *t(c,r)* via downstream classifier or linear interpolation [1].

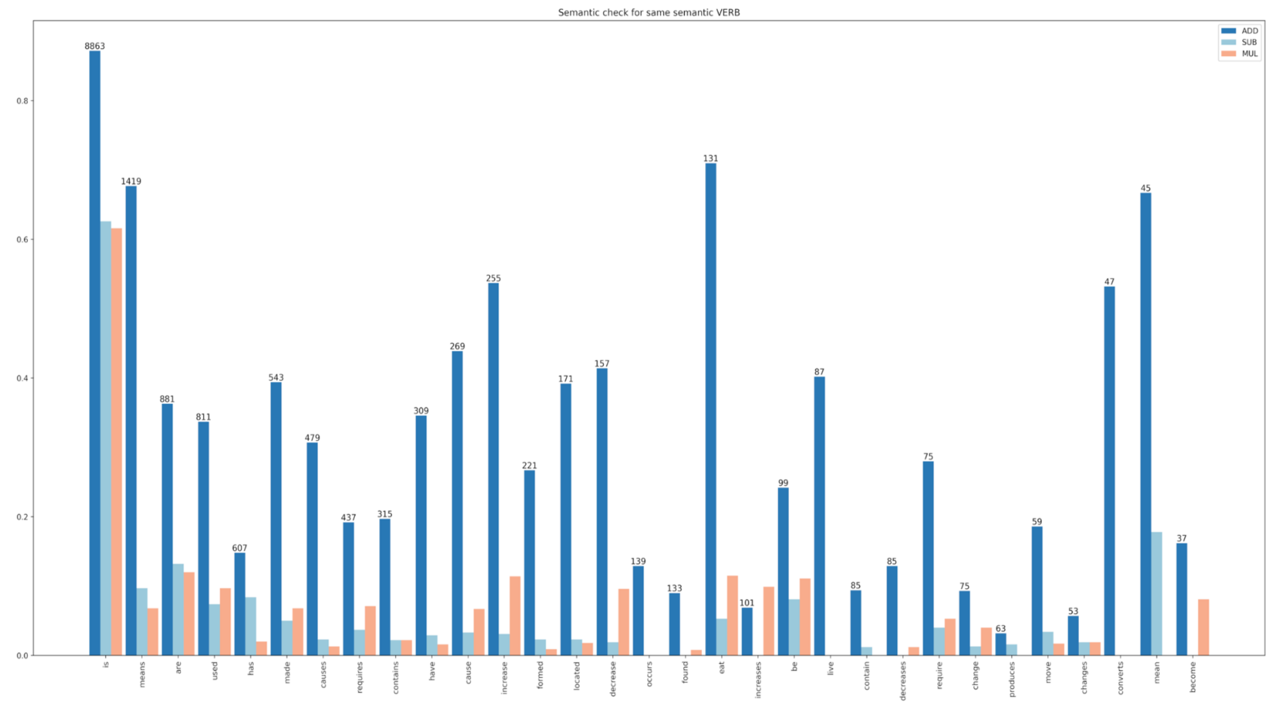
[1] Yoshua Bengio. 2013. Deep learning of representations: Looking forward. In *International conference on statistical language and speech processing*, pages 1–37. Springer.

Role-content cone

Observation: The addition operation ***αx + βy*** can hold the sentence semantic feature: role-content. We randomly sample the sentences with the same role-content and calculate the ratio of ADDed sentences with the same role-content (dark blue bar).





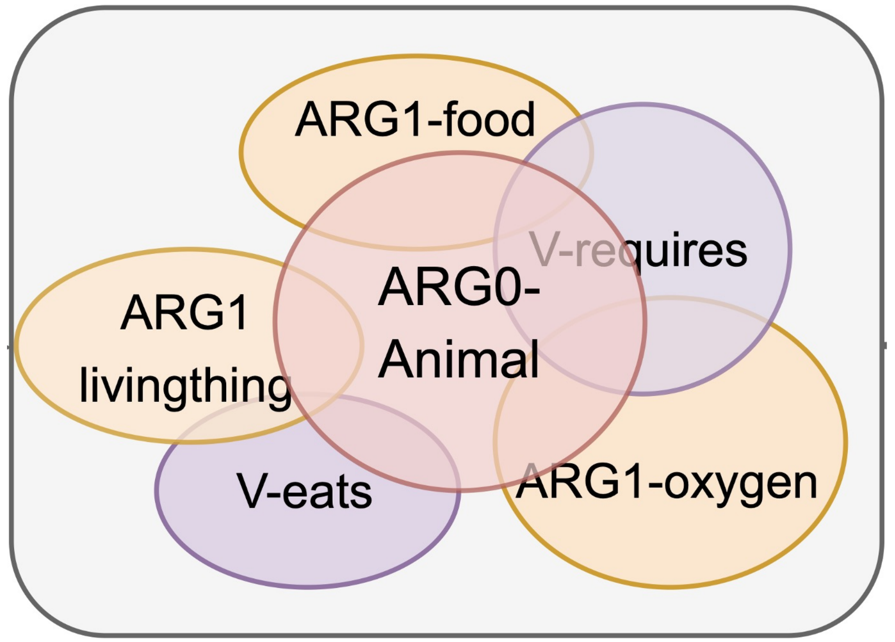


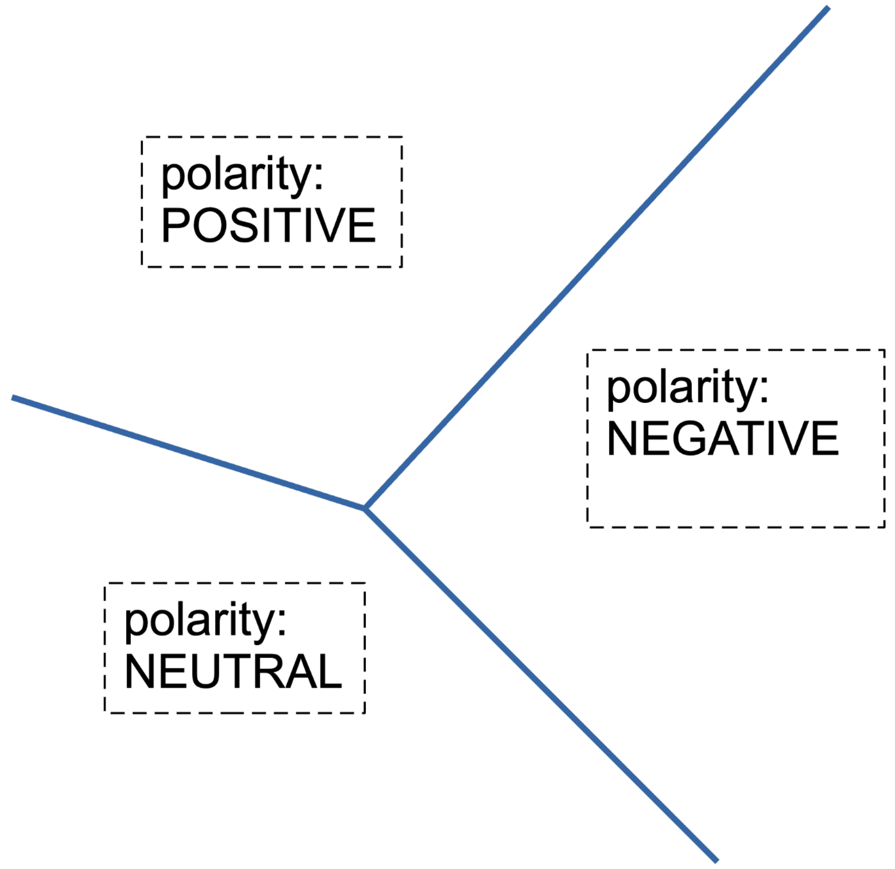
Problem: Different cones (i.e., role-contents) are still overlapped.

[1] Zhang, Y., Carvalho, D. S., Pratt-Hartmann, I., & Freitas, A. (2022). Quasi-symbolic explanatory nli via disentanglement: A geometrical examination. *arXiv preprint arXiv:2210.06230*.

Separability

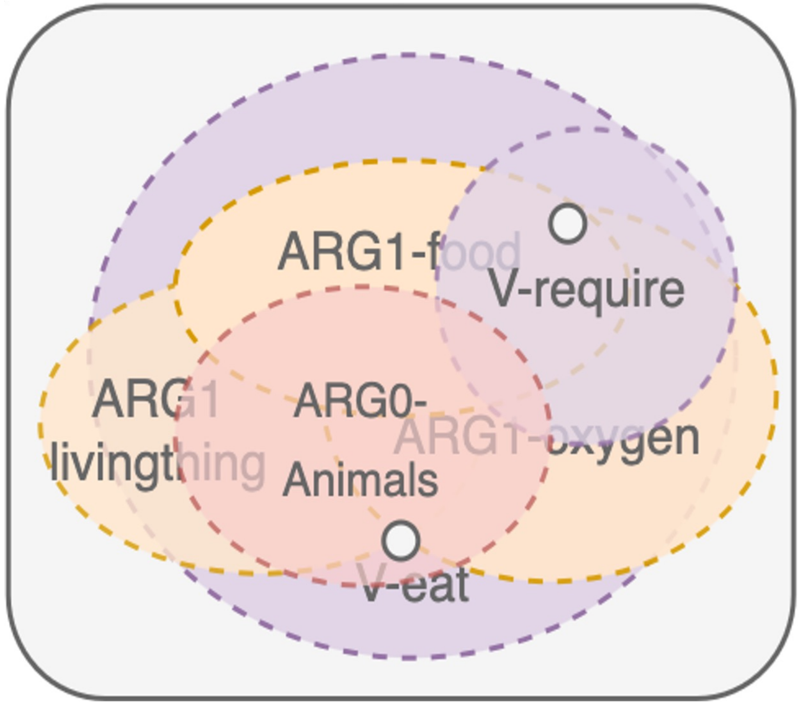
Separating semantic features into different regions (clusters) of a model's latent space:



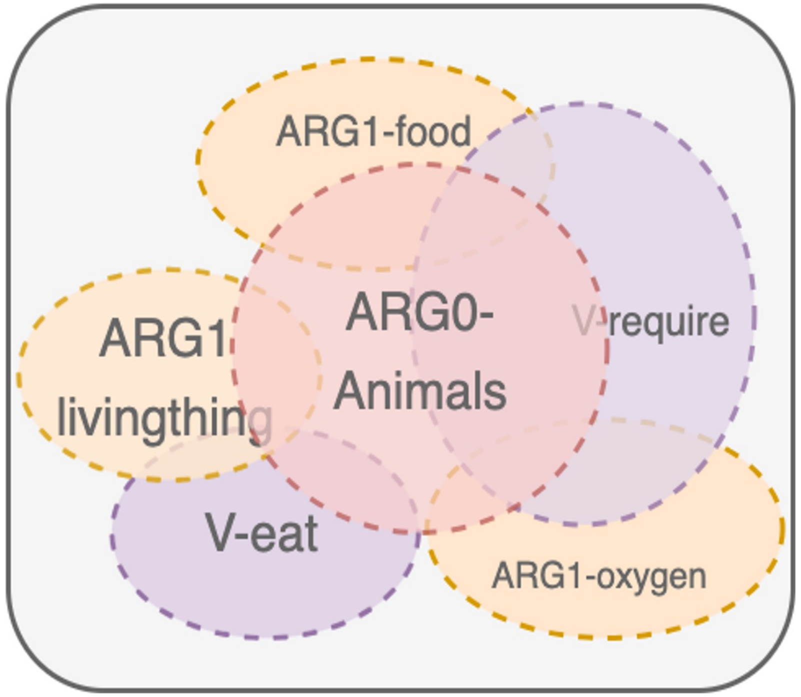


Separability

Can we offer geometric guarantees regarding the LM inference process?



VS



Controlling LMs

Style Transfer

An NLI task that consists in the separation between *style-content*.

[**style:** *active*]

The whole team helped pushing the rock

[**style:** *passive*]

The rock was pushed with help from the whole team

Style Transfer

Style transfer methods provide a foundation for improving control over generative models:

- Feature-oriented losses
- Disentanglement evaluation

However, further concepts are needed for control **beyond style-content separation**:

- Generative factors
- Feature localisation
- Input augmentation

Generative Factors

Independent underlying variables affecting the generation in a generative model.

This is manifested as a high value of:

$$|corr(Z_i, p(Y_j \in V))|$$

where Z_i is a single dimension in the model's latent space representation Z, Y_j (a generative factor) is a feature of the model's outputs Y, and V is a small subset of all possible values of Y_j.

Ideally, they can be mapped to interpretable linguistic features.

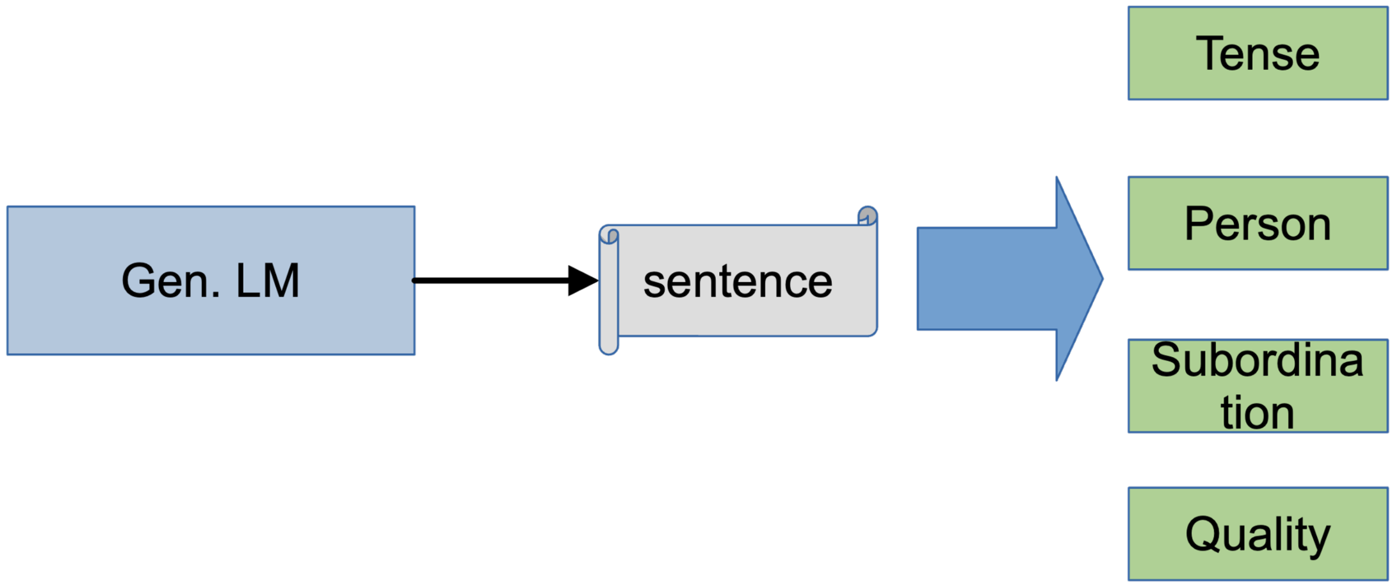
Generative Factors

Factors Y_j are often not explicit in the model’s outputs (e.g., tense, polarity of a sentence).

They can be observed through abstraction of the explicit feature space.

- An intended outcome of the training process.
- However often highly entangled (distributional prop.)

Generative Factors: Extraction



Extraction of such factors can be automated through specialised classifiers.

Generative Factors: Examples

Using linguistically grounded features:

- **Argument Structure Theory (AST):** categorising the semantic functions of arguments in relation to the verb (e.g. agent, patient, theme, instrument).

- **Definition Semantic Roles (DSR):** grouping the roles according to their contribution to either:

    - meaning (e.g., quality, location)
    - structure (e.g., main terms, modifiers)

Generative Factors: Examples

- Hu et al., 2017: sentiment, tense.
- Chen et. al., 2019: constituency parse, POS, paraphrase.
- Mercatali, Freitas., 2021: tense, subj-num, person-num, obj-num, gender, verb-obj, negation, verb-style, sent-type.
- Carvalho et. al., 2023: supertype, quality, location, modifier, statement, accessory, event.

Latent space (LS) manipulation

We can manipulate a latent space during training or fine-tuning, conforming it to a set of properties.

- Disentanglement of generative factors.
- Localisation of features for given factors.
- Linguistic consistency for linear operations.

LS manipulation: Bias induction

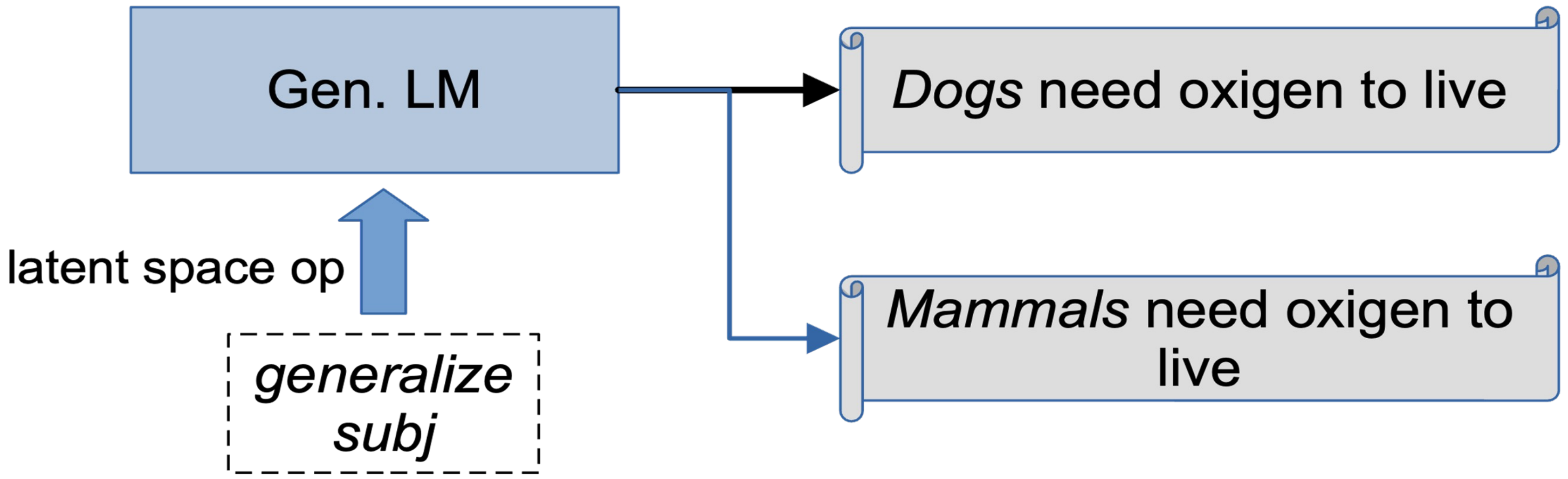
Inducing the necessary biases to the model can be typically achieved by:

- Augmenting the inputs with relevant features.
- Supervising the training / fine-tuning with the relevant features.
- Including generative factor losses to guide the training.

And their combination.

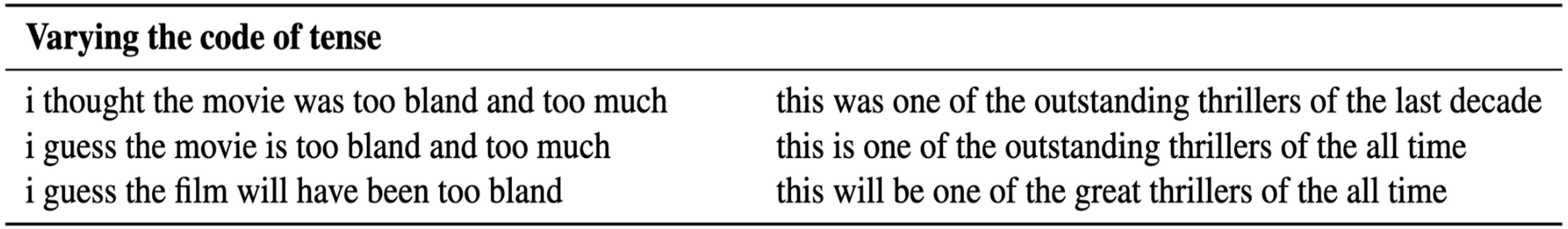
LS manipulation: Generation control

A disentangled, localized or linearly consistent latent space enables granular control over sentence generation.



Generation control: Examples

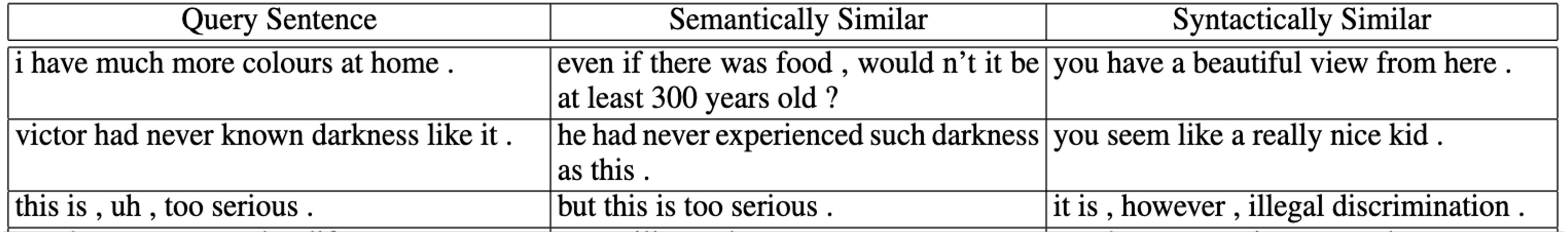
Hu et al., 2017: tense



Varying the code of tense	
i thought the movie was too bland and too much	this was one of the outstanding thrillers of the last decade
i guess the movie is too bland and too much	this is one of the outstanding thrillers of the all time
i guess the film will have been too bland	this will be one of the great thrillers of the all time

Generation control: Examples

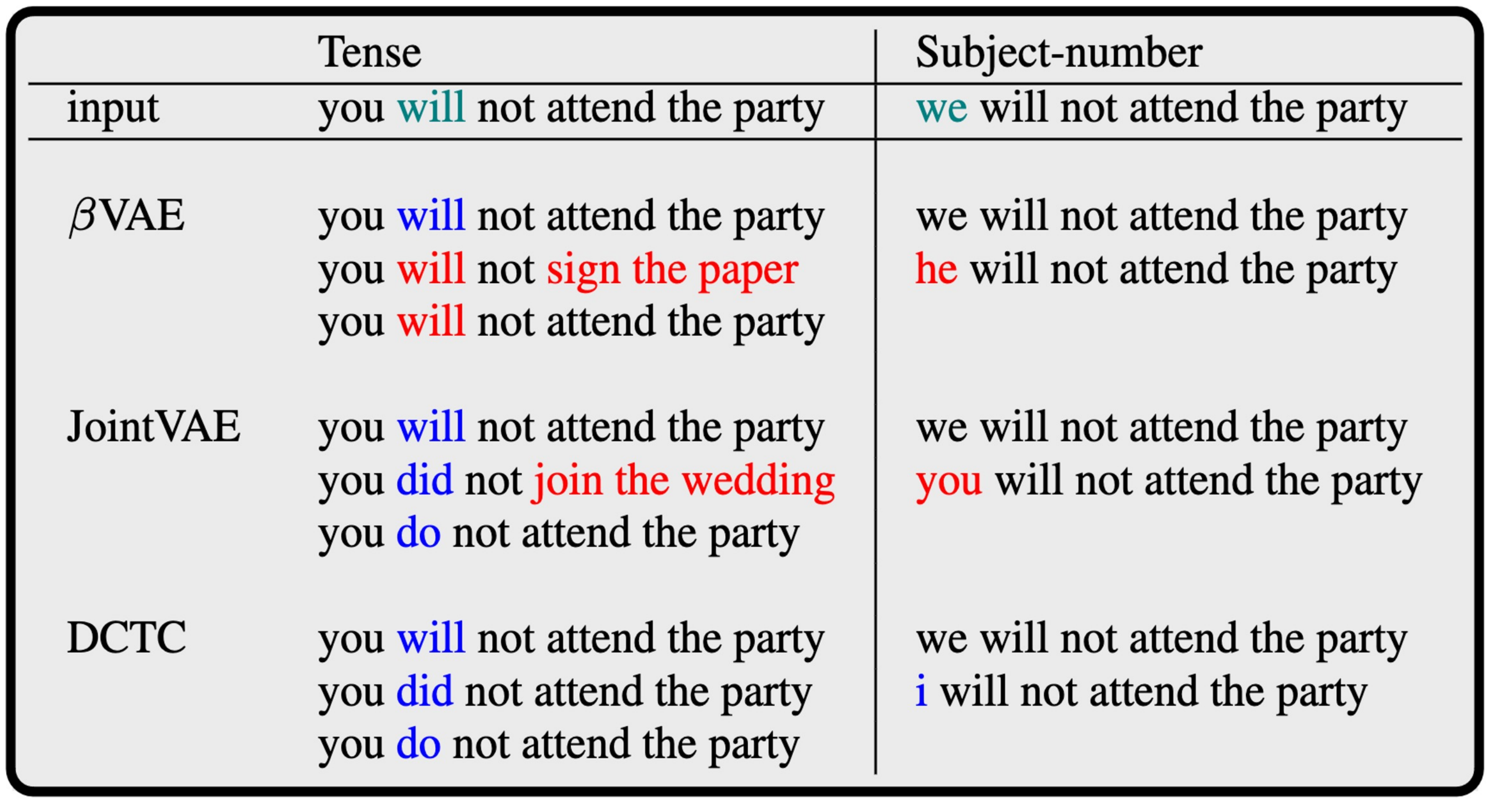
Chen et. al., 2019: (syntax-semantics)



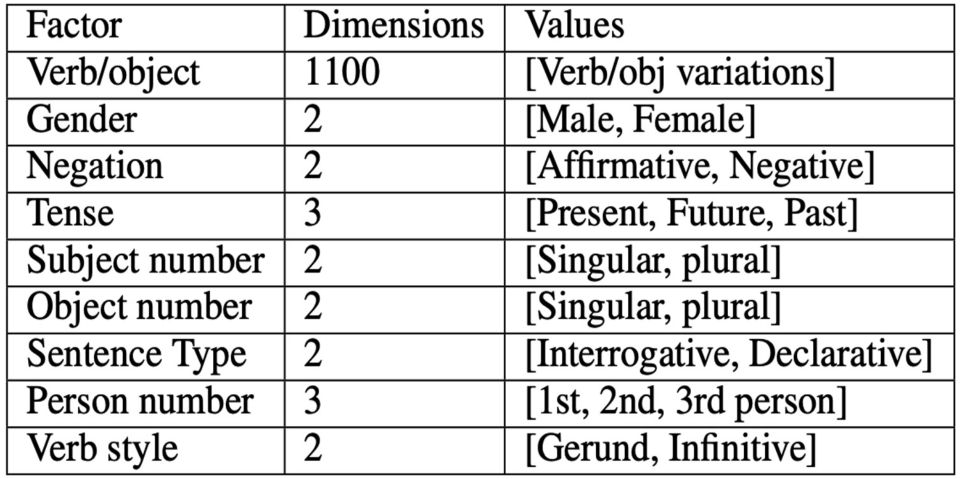
Query Sentence	Semantically Similar	Syntactically Similar
i have much more colours at home .	even if there was food , would n't it be at least 300 years old ?	you have a beautiful view from here .
victor had never known darkness like it .	he had never experienced such darkness as this .	you seem like a really nice kid .
this is , uh , too serious .	but this is too serious .	it is , however , illegal discrimination .

Generation control: Examples

Mercatali, Freitas., 2021: Syntactic factors



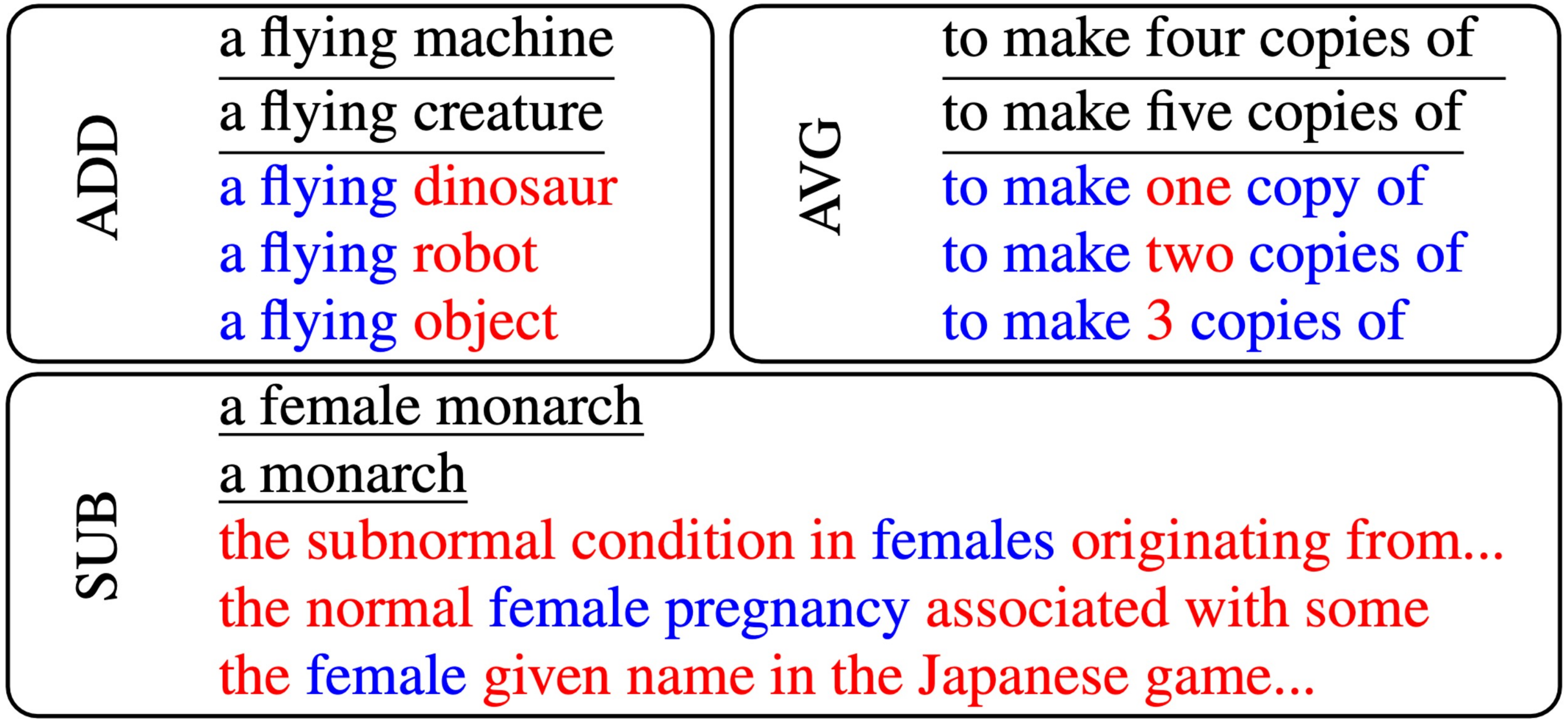
	Tense	Subject-number
input	you will not attend the party	we will not attend the party
βVAE	you will not attend the party	we will not attend the party
	you will not sign the paper	he will not attend the party
	you will not attend the party	
JointVAE	you will not attend the party	we will not attend the party
	you did not join the wedding	you will not attend the party
	you do not attend the party	
DCTC	you will not attend the party	we will not attend the party
	you did not attend the party	i will not attend the party
	you do not attend the party	



Factor	Dimensions	Values
Verb/object	1100	[Verb/obj variations]
Gender	2	[Male, Female]
Negation	2	[Affirmative, Negative]
Tense	3	[Present, Future, Past]
Subject number	2	[Singular, plural]
Object number	2	[Singular, plural]
Sentence Type	2	[Interrogative, Declarative]
Person number	3	[1st, 2nd, 3rd person]
Verb style	2	[Gerund, Infinitive]

Generation control: Examples

Carvalho et. al., 2023: supertype, quality (vector arithmetics)


ADD

a flying machine

a flying creature

a flying dinosaur
a flying robot
a flying object

AVG

to make four copies of

to make five copies of

to make one copy of
to make two copies of
to make 3 copies of

SUB

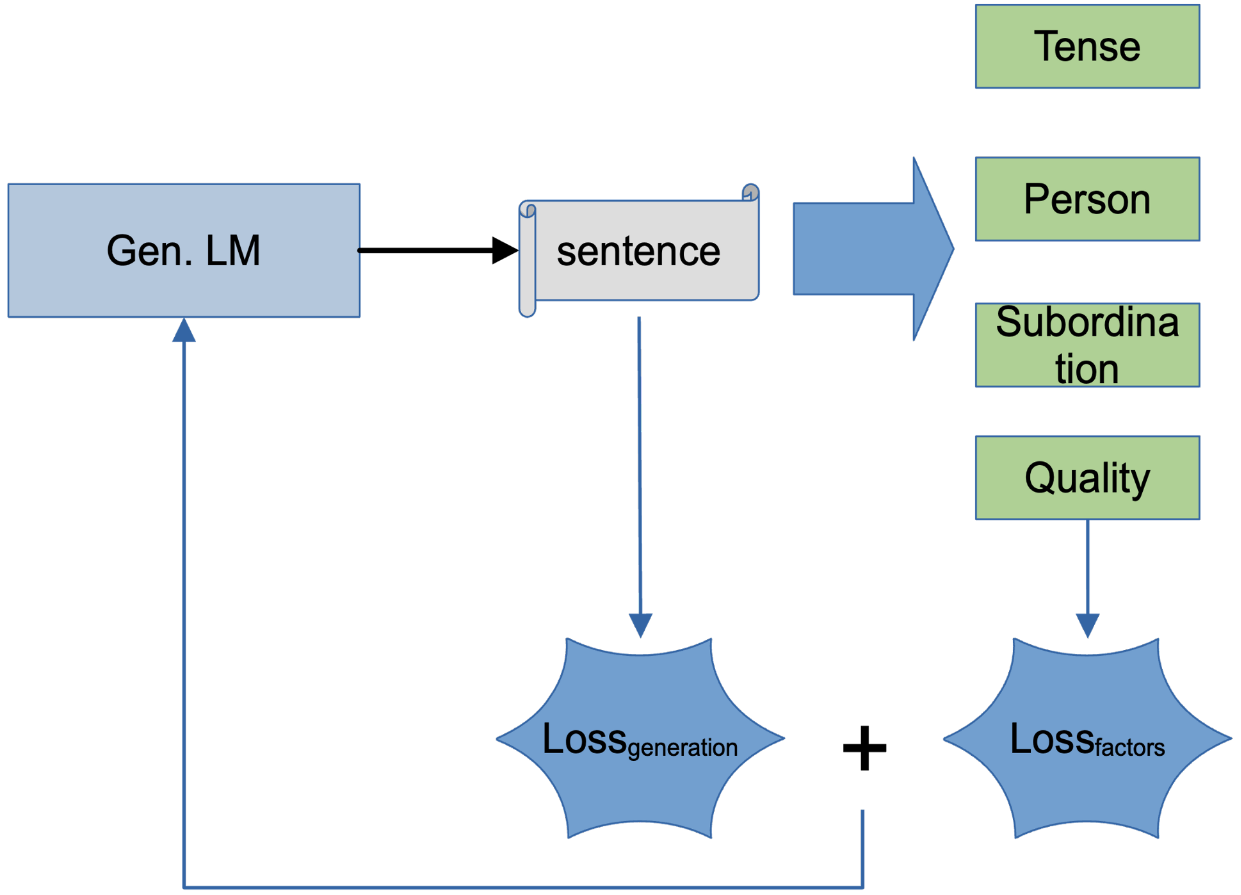
a female monarch

a monarch

the subnormal condition in females originating from...
the normal female pregnancy associated with some
the female given name in the Japanese game...

Linguistically-aware loss functions

Once linguistically grounded factors can be extracted from inputs and outputs, their expected labels can be used to calculate additional losses for training / fine tuning.


Tense
Person
Subordina tion
Quality
Gen. LM
sentence
$\mathrm{Loss}_{\mathrm{generation}}$
+
$\mathrm{Loss}_{\mathrm{factors}}$

Linguistically-aware loss functions: Examples

Hu et al., 2017: tense

- Discriminator probe

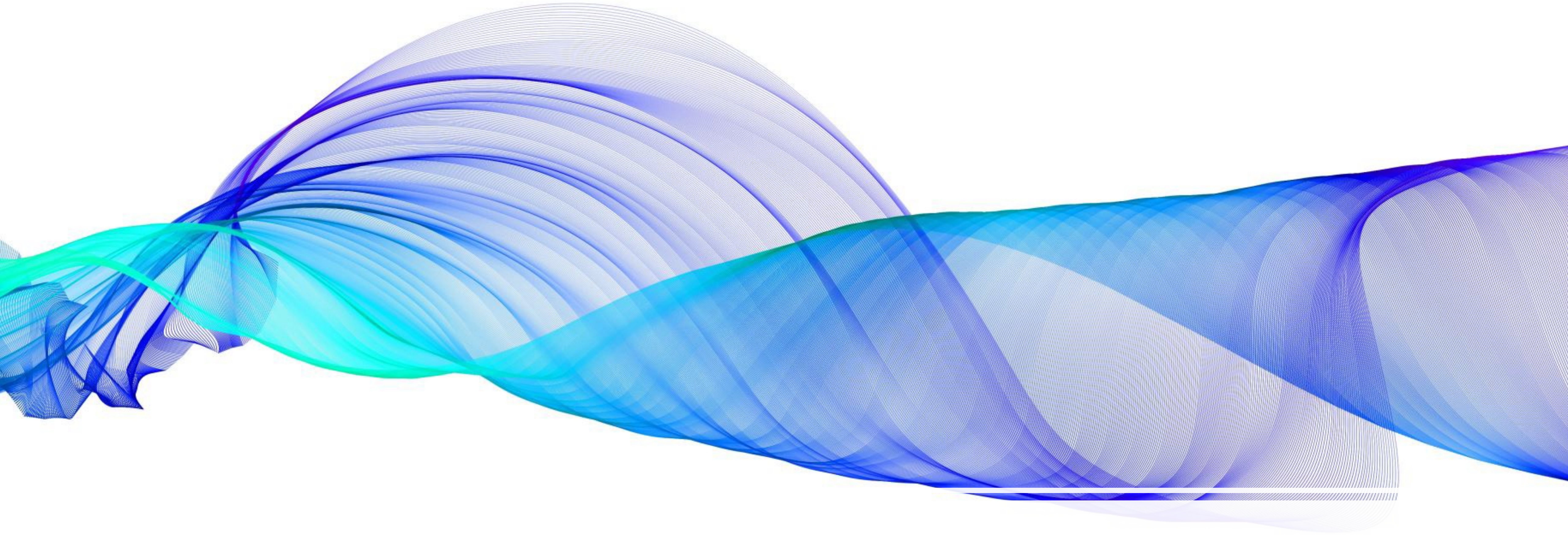
Chen et. al., 2019: word position, STS

- Paraphrase Reconstruction Loss
- Discriminative Paraphrase Loss (embeddings)
- Word Position Loss

Carvalho et. al., 2023: Definition Semantic Roles (DSR)

- DSR reconstruction loss (NLL)

Language Variational Autoencoders (VAEs)



What is a latent variable model?

Generative modelling task:

Assume:

- data samples $x1, x2, \ldots, xn$
- from a distribution of interest Q(x)
- unknown density

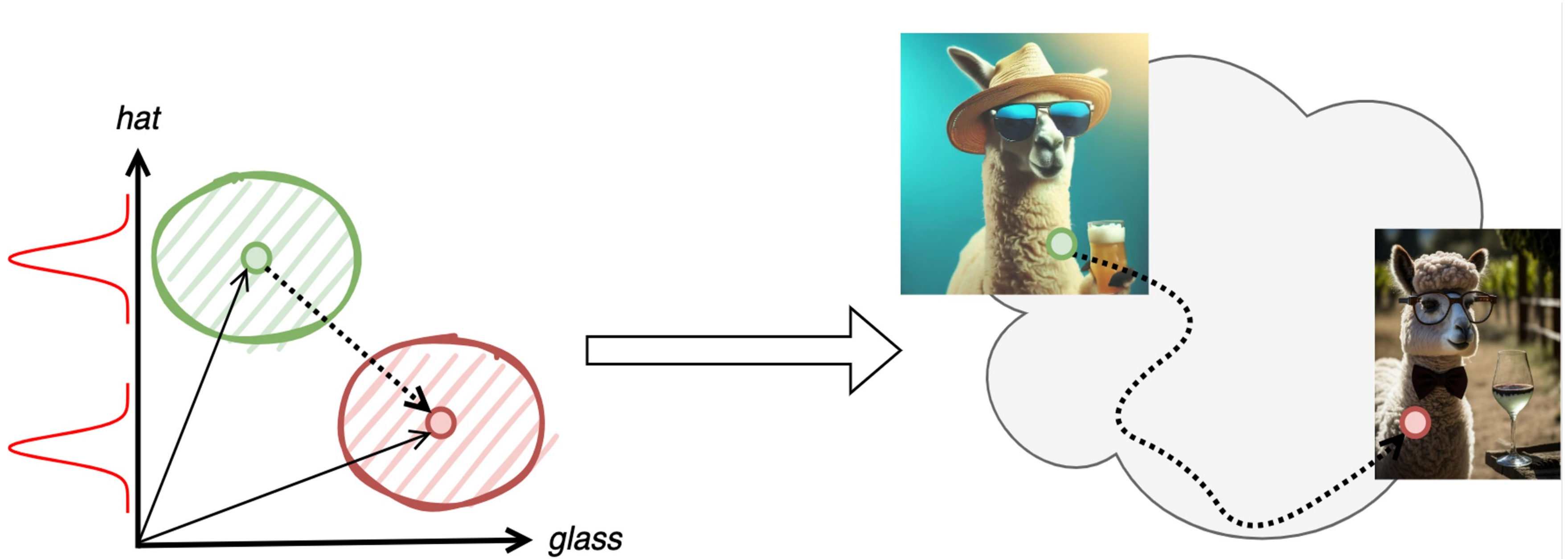
We're interested in using these samples to learn a probabilistic model approximating Q. In particular, we want efficient generation of new samples (approximately) distributed from Q.

Latent variable models: models the transformation from latent variable distribution (such as std Gaussian) to Q. They include variational autoencoders (VAE), generative adversarial networks (GAN), normalizing flow, diffusion, flow matching, etc.

Why we use latent variable model?

“What I cannot create, I do not understand.” - Richard P. Feynman

Latent variable model: provides a low-dimensional & smooth latent space (manifolds), which allow us to **“interpret”** and **“control”** data generation over complex unknown space.



Latent Space Geometry
(e.g., Euclidean space)

Observation Geometry
(e.g., non-euclidean space)

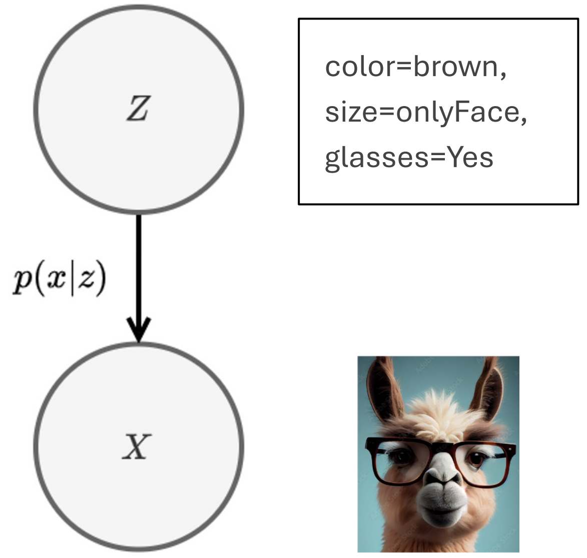
Overview

1. Variational AutoEncoder(VAE)	1. Latent variable model: p(x, z) 2. Variational inference: approximating true posterior 3. Evidence lower bound: Jensen's inequality 4. VAE architecture: fixed std Gaussian prior and posterior 5. Complex fixed prior and problem: vMF distribution and hole 6. Trainable prior: conditional VAE 7. Pytorch library: pythae
2. Language VAE	1. Transformer-based VAEs' architecture: Optimus 2. Objective function: negation of ELBO with KL cyclical and threshold tricks 3. Pytorch library: LangVAE
3. Latent semantic control methods	1. semantic geometry with normalizing flow: *"Learning Disentangled Semantic Space of Explanations via Invertible Neural Networks"* 1. syntax with graph neural network: *"graph-induced Semantic-Syntax Space in Transformer-based Variational AutoEncoder"* 1. discretization with vector quantization: *"Improving Semantic Control in Discrete Latent Spaces with Transformer Quantized Variational Autoencoders"* 1. label with conditional VAE: *"Learning disentangled representations for natural language definitions"* *"Toward Controllable Natural Language Inference through Lexical Inference Types"* *"LlaMaVAE: Guiding Large Language Model Generation via Continuous Latent Sentence Spaces"*

VAE: 1. Latent variable model

Latent variable model: models the joint distribution p(x, z) = p(x|z)p(z). For training stage, we can **only access to x.** Therefore, we marginalise out the latent variables *z*, the target distribution:

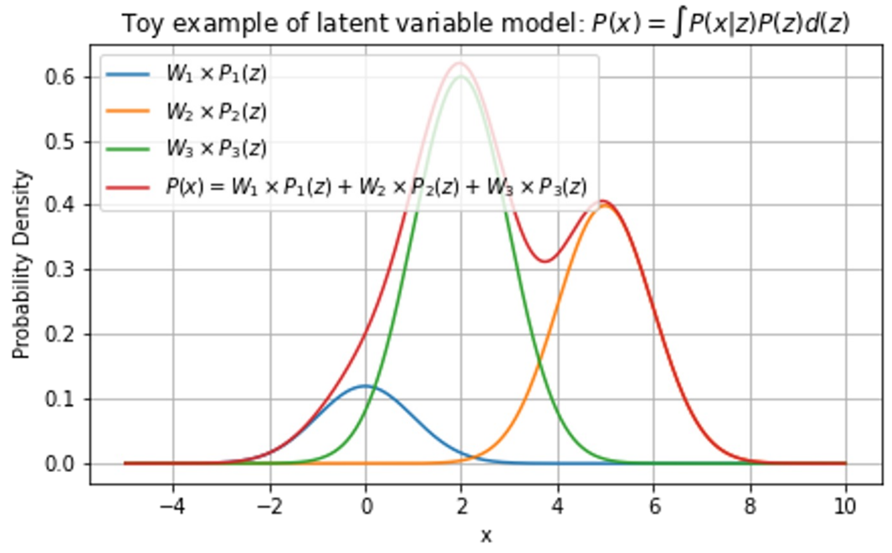
$$p(x) = \int p_\theta(x|z) \times p_\theta(z)d(z)$$



θ : represents the parameter we want to obtain.

$p_\theta(x|z)$: likelihood which represents the transformation from latent variables to observation.

$p_\theta(z)$: prior distribution of latent variables.



However, the integration is intractable!

VAE: 2. Variational inference

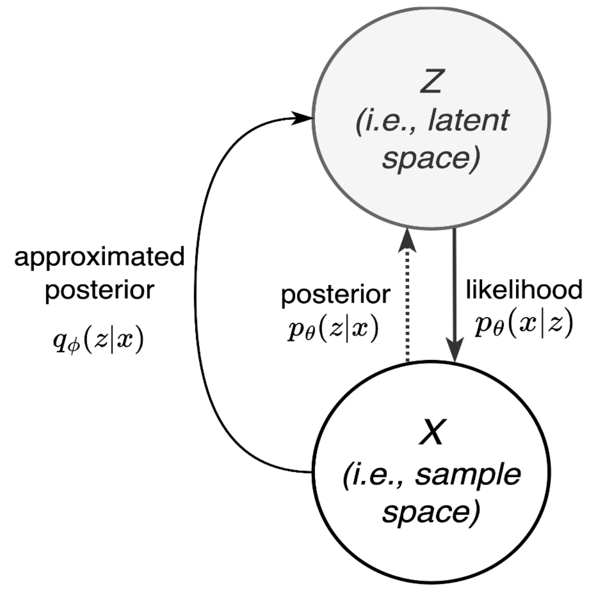
Variational Inference: To avoid integrating over the whole latent space, a natural question would be "*Can we infer any information about z after observing a sample?*" - true posterior: $p_\theta(\mathbf{z}|\mathbf{x})$

In VAEs, the idea from "*(amortised) variational inference*" is to approximate the true posterior $p_\theta(\mathbf{z}|\mathbf{x})$ with a network with parameter ϕ , denoted by $q_\phi(\mathbf{z}|\mathbf{x})$ (***approximate posterior***). We can use KL:

$$D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))$$

$$\boxed{\underbrace{\log p_\theta(\mathbf{x})}_{(1)} - \underbrace{D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}|\mathbf{x}))}_{(2)} = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})}\log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x})\|p_\theta(\mathbf{z}))}$$

We want to (1) maximize the probability of generating real data and (2) also minimize the difference between the true and estimated/aggregate/ approximate posteriors.



Variational inference

$$D_{KL}(q_\phi(z|x)||p_\theta(z|x))$$
$$= \int q_\phi(z|x)\log\frac{q_\phi(z|x)}{p_\theta(z|x)}d(z)$$
$$= \int q_\phi(z|x)\log\frac{q_\phi(z|x)p_\theta(x)}{p_\theta(z,x)}d(z)$$
$$= \int q_\phi(z|x)\left(\log p_\theta(x) + \log\frac{q_\phi(z|x)}{p_\theta(z,x)}\right)d(z)$$
$$= \int q_\phi(z|x)\log p_\theta(x)d(z) + \int q_\phi(z|x)\log\frac{q_\phi(z|x)}{p_\theta(z,x)}d(z)$$

Since $\int q_\phi(z|x)d(z) = 1$, we can get:

$$= \log p_\theta(x) + \int q_\phi(z|x)\log\frac{q_\phi(z|x)}{p_\theta(z,x)}d(z)$$
$$= \log p_\theta(x) + \int q_\phi(z|x)\log\frac{q_\phi(z|x)}{p_\theta(x|z)p_\theta(z)}d(z)$$
$$= \log p_\theta(x) + \mathbb{E}_{z\sim q_\phi(z|x)}\left[\log\frac{q_\phi(z|x)}{p_\theta(z)} - p_\theta(x|z)\right]$$
$$= \log p_\theta(x) + D_{KL}(q_\phi(z|x)||p_\theta(z)) - \mathbb{E}_{z\sim q_\phi(z|x)}p_\theta(x|z)$$

VAE: 3. Evidence lower bound (ELBO)

Evidence lower bound(ELBO): the right part is also named Evidence lower bound (ELBO): the lower bound of log likelihood of observation *x*.

$$
\begin{aligned}
\log p_\theta(x) &= \log \int_z f p_\theta(x,z) dz \\
&= \log \int_z p_\theta(x,z) \frac{q_\phi(z|x)}{q_\phi(z|x)} dz \\
&= \log \left(\mathbb{E}_{z\sim q_\phi(z|x)} \left[\frac{p_\theta(x,z)}{q_\phi(z|x)} \right] \right) \\
&\geq \mathbb{E}_{z\sim q_\phi(z|x)} \left[\log \frac{p_\theta(x,z)}{q_\phi(z|x)} \right] \text{, Jensen's inequality} \\
&= \mathbb{E}_z \left[\log p_\theta(x,z) \right] + \int_z q_\phi(z|x) \log \frac{1}{q_\phi(z|x)} dz \\
&= \int_z q_\phi(z|x) \log p_\theta(x,z) dz + \int_z q_\phi(z|x) \log \frac{1}{q_\phi(z|x)} dz \\
&= \int_z q_\phi(z|x) \left(\log p_\theta(x,z) - \log q_\phi(z|x) \right) dz \\
&= \underbrace{\int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz}_{(1)}
\end{aligned}
$$

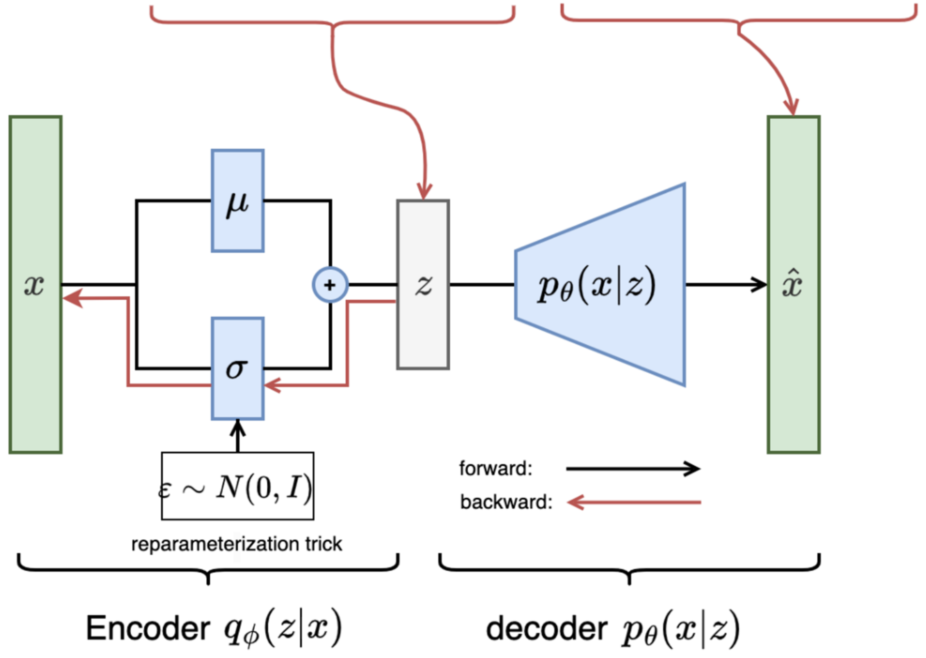
$$
\begin{aligned}
& D_{KL}(q_\phi(z|x) || p(z|x)) \\
&= \int_z q_\phi(z|x) \log \frac{q_\phi(z|x)}{p(z|x)} dz \\
&= - \int_z q_\phi(z|x) \log \frac{p_\theta(z|x)}{q_\phi(z|x)} dz \\
&= - \int_z q_\phi(z|x) \log \frac{p_\theta(z,x)}{q_\phi(z|x) p_\theta(x)} dz \\
&= - \left(\int_z q_\phi(z|x) \log \frac{p_\theta(z,x)}{q_\phi(z|x)} dz - \int_z q_\phi(z|x) \log p_\theta(x) dz \right) \\
&= \underbrace{- \int_z q_\phi(z|x) \log \frac{p_\theta(x,z)}{q_\phi(z|x)} dz}_{(1)} + \log p_\theta(x)
\end{aligned}
$$

VAE: 4. Gaussian prior and posterior

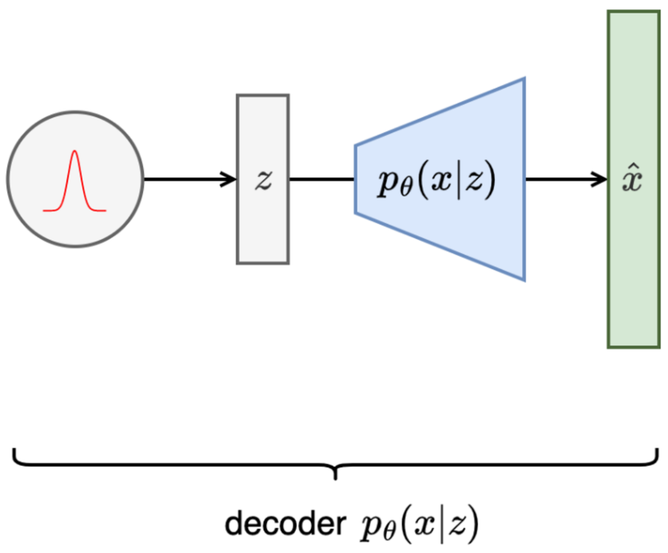
Architecture: When prior is a ***"fixed"*** std Gaussian distribution, the VAE training and inference can be visualised as:

Training:

$$\mathcal{L}_{\text{VAE}}(\theta,\phi) = \underbrace{D_{KL}(q_\phi(z|x)||p_\theta(z))}_{\textit{compress}} - \underbrace{E_{z\sim q_\phi(z|x)} \log p_\theta(x|z)}_{\textit{reconstruct}}$$



Inference:



Calculate KL between approxima
posterior and std Gaussian (dim=
the latent dimension, 0: batch size

```
KL = 0.5 * (mean.pow(2) + logvar.exp() - logvar - 1).sum(dim=1)
```

reparameterization trick: remove ***stochastic*** sampling process from ***deterministic*** backward propagation.

VAE: 4. Gaussian prior and posterior

$$p_1 = \mathcal{N}_1(\mu_1, \sigma_1), p_2 = \mathcal{N}_2(\mu_2, \sigma_2) \quad \mathcal{N}(\mu, \sigma) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x-\mu)^2}{2\sigma^2}}$$

$$\begin{aligned}
KL(p_1||p_2) &= \int_x p_1(x) log \frac{p_1(x)}{q_1(x)} dx \\
&= \int_x p_1(x) [log(\frac{1}{\sqrt{2\pi\sigma_1^2}} e^{-\frac{(x-\mu_1)^2}{2\sigma_1^2}}) - log(\frac{1}{\sqrt{2\pi\sigma_2^2}} e^{-\frac{(x-\mu_2)^2}{2\sigma_2^2}})] \\
&= \int_x p_1(x) [-\frac{1}{2} log 2\pi - log\sigma_1 - \frac{(x-\mu_1)^2}{2\sigma_1^2} + \frac{1}{2} log 2\pi + log\sigma_2 + \frac{(x-\mu_2)^2}{2\sigma_2^2}] \\
&= \int_x p_1(x) [log \frac{\sigma_2}{\sigma_1} + \frac{(x-\mu_2)^2}{2\sigma_2^2} - \frac{(x-\mu_1)^2}{2\sigma_1}] dx \\
&= A
\end{aligned}$$

$$\begin{aligned}
A &= \int_x p_1(x) [log \frac{\sigma_2}{\sigma_1} + \frac{(x-\mu_2)^2}{2\sigma_2^2} - \frac{(x-\mu_1)^2}{2\sigma_1}] dx \\
&= log \frac{\sigma_2}{\sigma_1} + \underbrace{\int_x p_1(x) \frac{(x-\mu_2)^2}{2\sigma_2^2} dx}_{\text{B}} - \frac{1}{2}
\end{aligned}$$

$$\begin{aligned}
B' &= \int_x p_1(x)(x-\mu_2)^2 dx \\
&= \int_x p_1(x)((x-\mu_1) + (\mu_1 - \mu_2))^2 dx \\
&= \int_x p_1(x)(x-\mu_1)^2 dx + 2(\mu_1 - \mu_2) \int_x p_1(x)(x-\mu_1) dx + (\mu_1 - \mu_2)^2 \\
&= \sigma_1^2 + 0 + (\mu_1 - \mu_2)^2 \\
&= \sigma_1^2 + (\mu_1 - \mu_2)^2
\end{aligned}$$

$$A = log \frac{\sigma_2}{\sigma_1} + \frac{1}{2\sigma_2^2}(\sigma_1^2 + (\mu_1 - \mu_2)^2) - \frac{1}{2} \qquad p_2 = \mathcal{N}_2(0, 1)$$

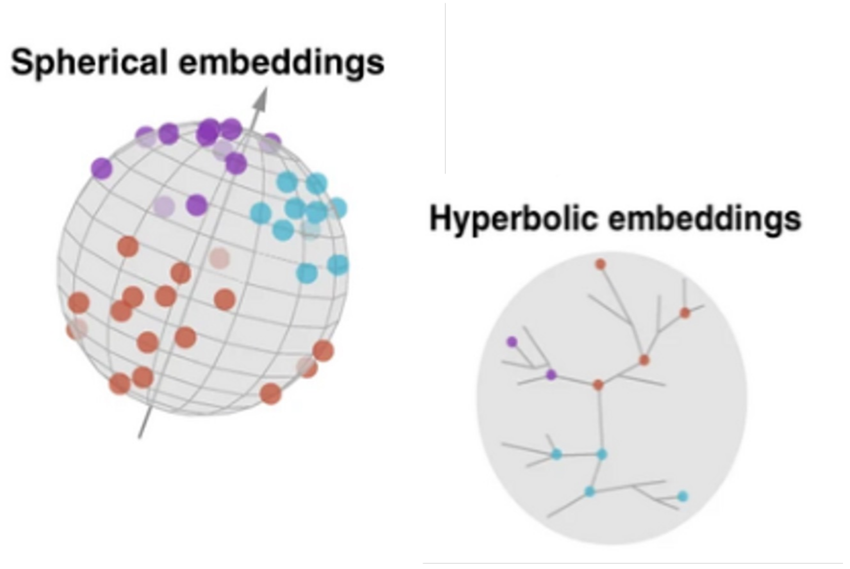
$$\boxed{KL(p_1||p_2) = -\frac{1}{2} \times [2 log\sigma_1 + 1 - \sigma_1^2 - \mu_1^2]}$$

```
KL = 0.5 * (mean.pow(2) + logvar.exp() - logvar - 1).sum(dim=1)
```

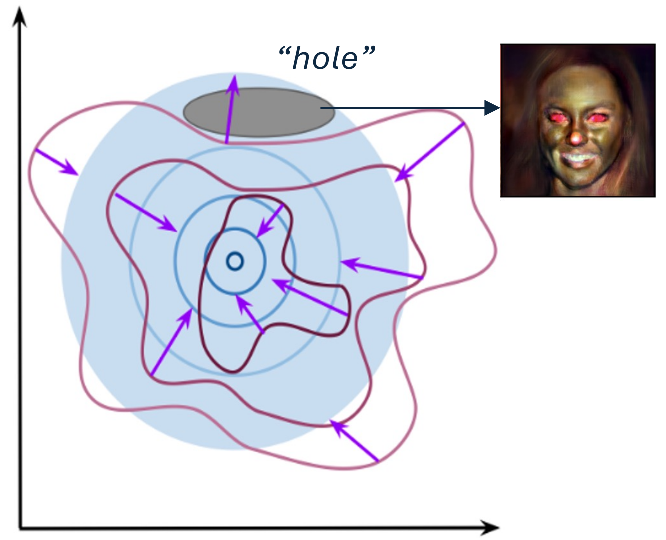
**The encoder output logvar rather than var^2 because the output of neural network might be < 0.*

VAE: 5. Problems with a complex fixed prior

Fixed prior: In addition to Gaussian distribution, there are more options to choose different prior and posterior distributions, such as *"von Mises-Fisher"* (i.e., hypersphere), etc. or more complex structure, such as hyperbolic [1], and hierarchical spaces.



Problem of fixed priors: due to the mismatch between prior and posterior during inference, the sampling from the area of prior, where the aggregated posterior assigns low probability while the prior assigns (relatively) high probability. This might lead to low quality generation. We refer it as *"hole"* problem [2].



Solution: To remedy this problem, we can use a ***trainable prior***.

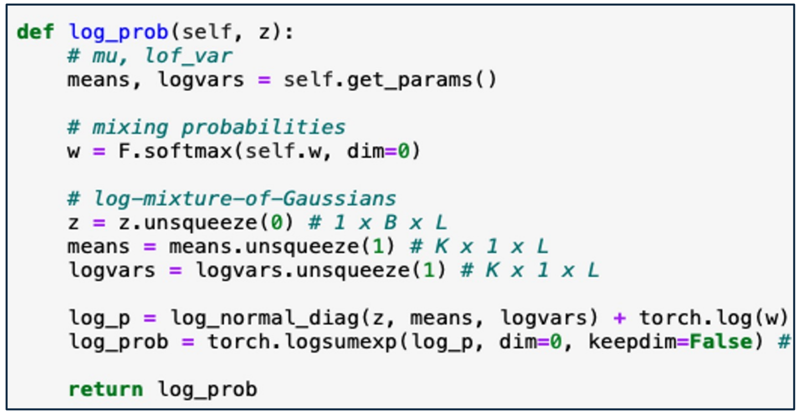
[1] Mathieu, E., Le Lan, C., Maddison, C. J., Tomioka, R., & Teh, Y. W. (2019). Continuous hierarchical representations with poincaré variational auto-encoders. *Advances in neural information processing systems*, *32*.

[2] Rezende, D. J., & Viola, F. (2018). Taming vaes. arXiv preprint arXiv:1810.00597.

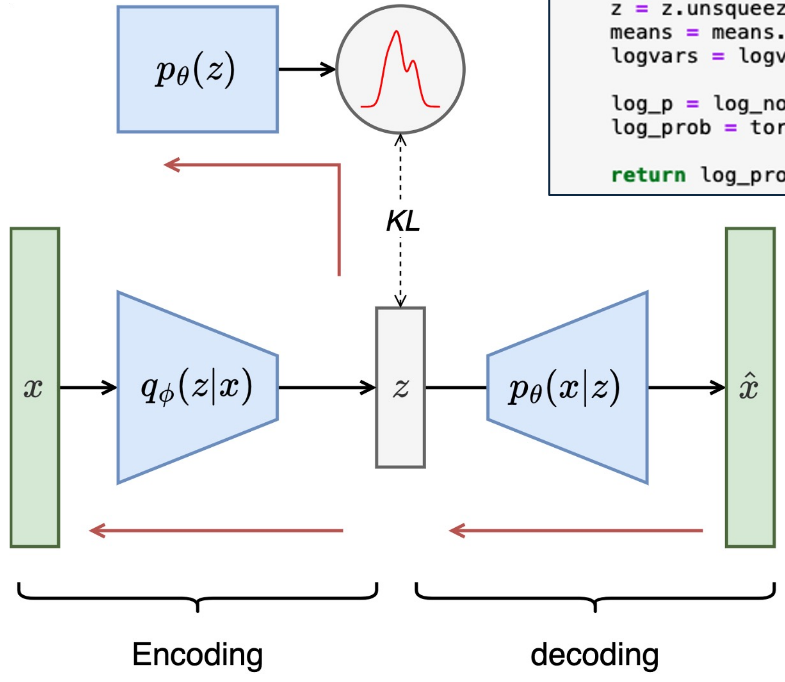
source from: https://jmtomczak.github.io/blog/7/7_priors.html#Introduction

VAE: 6. Trainable prior

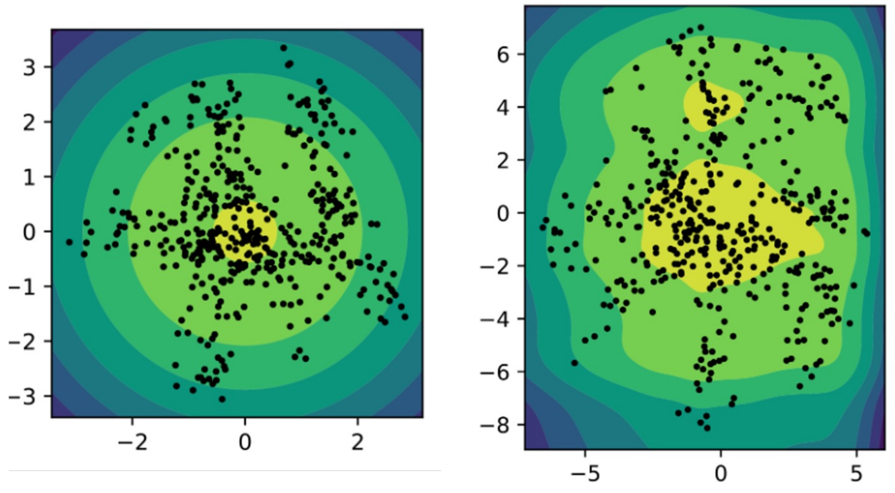
Trainable Prior: Since the fixed prior might be too rigid, it can cause the "hole" problem, we can design a learnable prior to induce the posterior and the prior try to match each other during training, such as Gaussian Mixture Prior, VAMP Prior, FlowPrior[1], conditional VAE (CVAE), etc.



```python
def log_prob(self, z):
    # mu, lof_var
    means, logvars = self.get_params()

    # mixing probabilities
    w = F.softmax(self.w, dim=0)

    # log-mixture-of-Gaussians
    z = z.unsqueeze(0) # 1 x B x L
    means = means.unsqueeze(1) # K x 1 x L
    logvars = logvars.unsqueeze(1) # K x 1 x L

    log_p = log_normal_diag(z, means, logvars) + torch.log(w)
    log_prob = torch.logsumexp(log_p, dim=0, keepdim=False) #

    return log_prob
```



contours represent prior where left: Gaussian, right: Gaussian mixture.



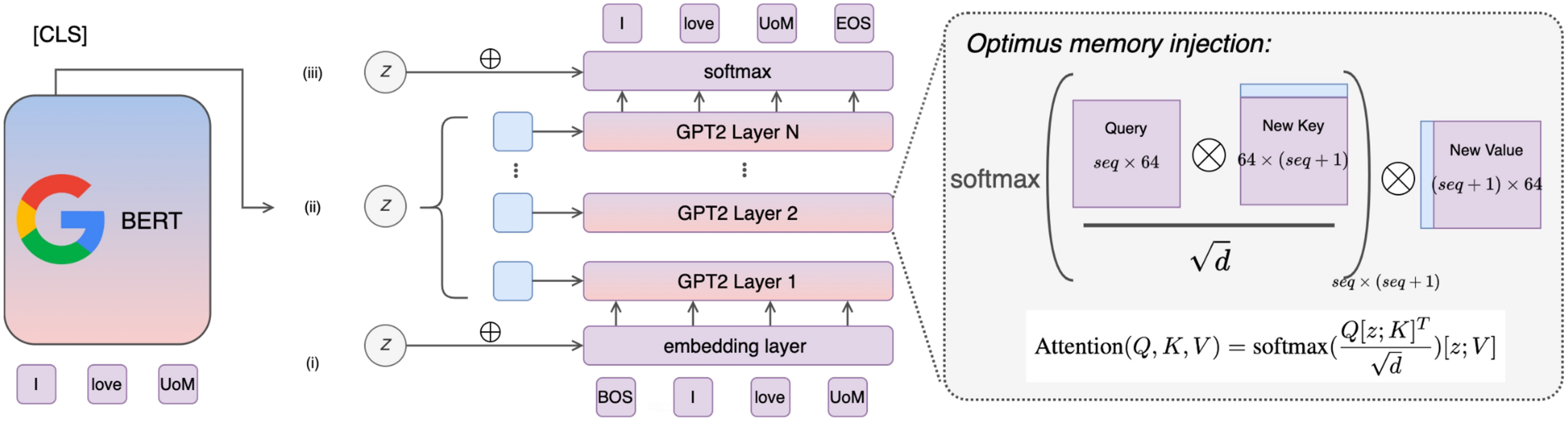
source from: https://jmtomczak.github.io/blog/7/7_priors.html#Introduction

[1]Xiaoan Ding and Kevin Gimpel. 2021. FlowPrior: Learning Expressive Priors for Latent Variable Sentence Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3242–3258, Online. Association for Computational Linguistics.

[2]Tomczak, J. M. (2022). Deep Generative Modeling. Springer Nature https://jmtomczak.github.io/blog/7/7_priors.html#Introduction

Language VAE: 1. Transformer-based VAEs

Optimus[1]: BERT-GPT2 architecture with Gaussian prior. The latent space is injected into the decoder with ***memory*** injection setup (ii), which operates over the low-rank attention weights (i.e, Key and Value) directly. This low-rank injection can avoid redundant information compared to (i) and (iii) [2].



$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q[z;K]^T}{\sqrt{d}})[z;V]$$

[1] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online. Association for Computational Linguistics.

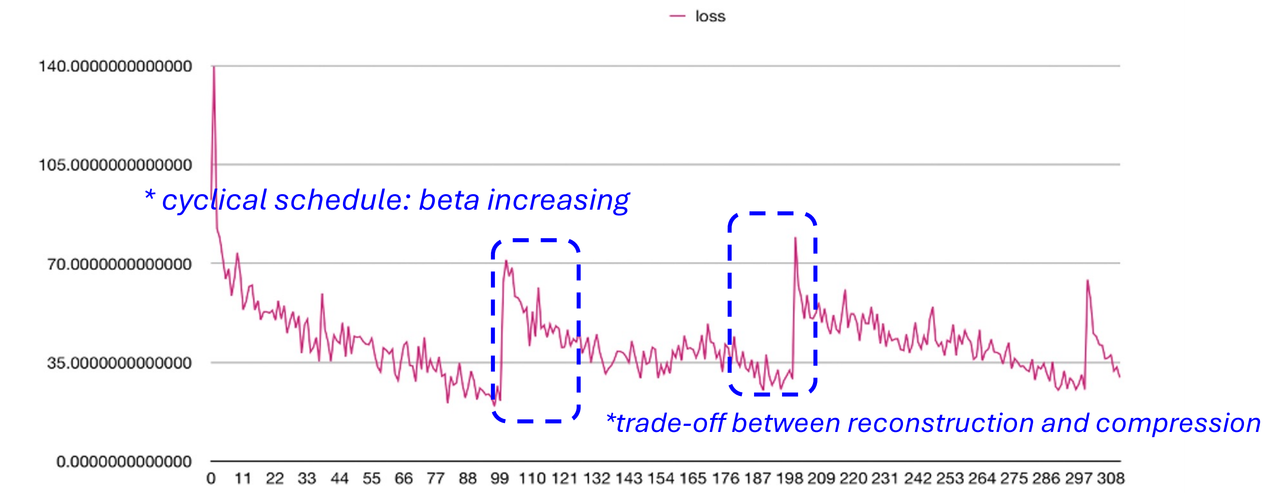
[2] Hu, J., Yi, X., Li, W., Sun, M., & Xie, X. (2022). Fuse it more deeply! a variational transformer with layer-wise latent variable inference for text generation. *arXiv preprint arXiv:2207.06130*.

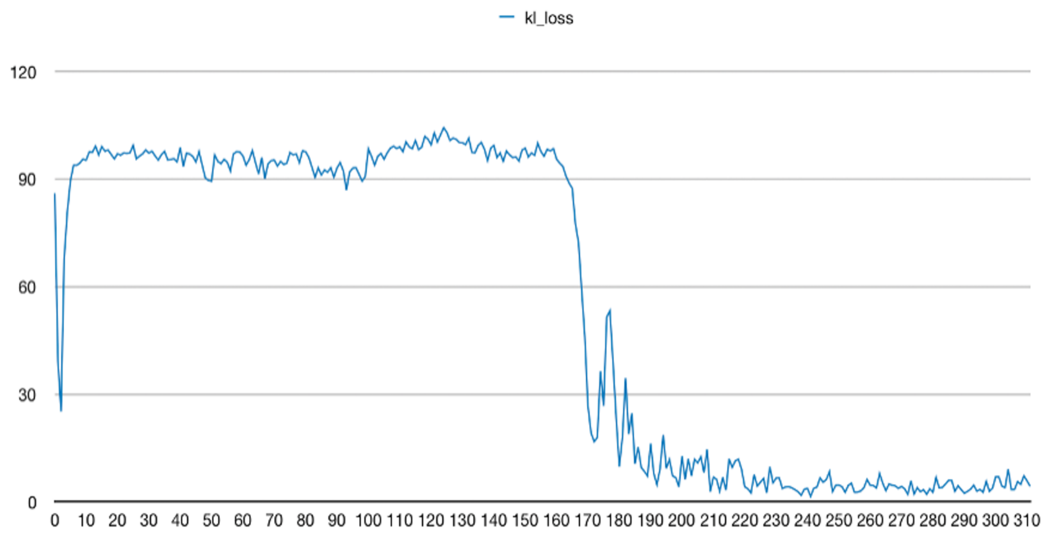
Language VAE: 2. Objective function

Objective function: the negation of ELBO, to avoid *KL vanishing (posterior collapse)*. Two tricks:

1. Cyclical schedule[1]: gradually and cyclically increase β from 0 to 1.
2. KL threshold scheme[2]: for each dimension, choose the max between threshold and KL.

$$\mathcal{L}_{\text{VAE}} = -\mathbb{E}_{q_\phi(z|x)} \log p_\theta(x|z) + \beta \sum_i \max\left[\lambda, \mathbf{KL} q_\phi(z_i|x) || p(z_i)\right]$$





[1] Hao Fu, Chunyuan Li, Xiaodong Liu, Jianfeng Gao, Asli Celikyilmaz, and Lawrence Carin. 2019. Cyclical annealing schedule: A simple approach to mitigating KL vanishing. In Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers), pages 240–250, Minneapolis, Minnesota. Association for Computational Linguistics.

[2] Bohan Li, Junxian He, Graham Neubig, Taylor BergKirkpatrick, and Yiming Yang. 2019. A surprisingly effective fix for deep latent variable modeling of text. In Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP), pages 3603– 3614, Hong Kong, China. Association for Computational Linguistics.

Language VAE: 3. Pytorch library

LangVAE: our demo can easily integrate different pretrained language models into VAE architecture.

Pretrained checkpoints:

https://huggingface.co/neuro-symbolic-ai

Train:

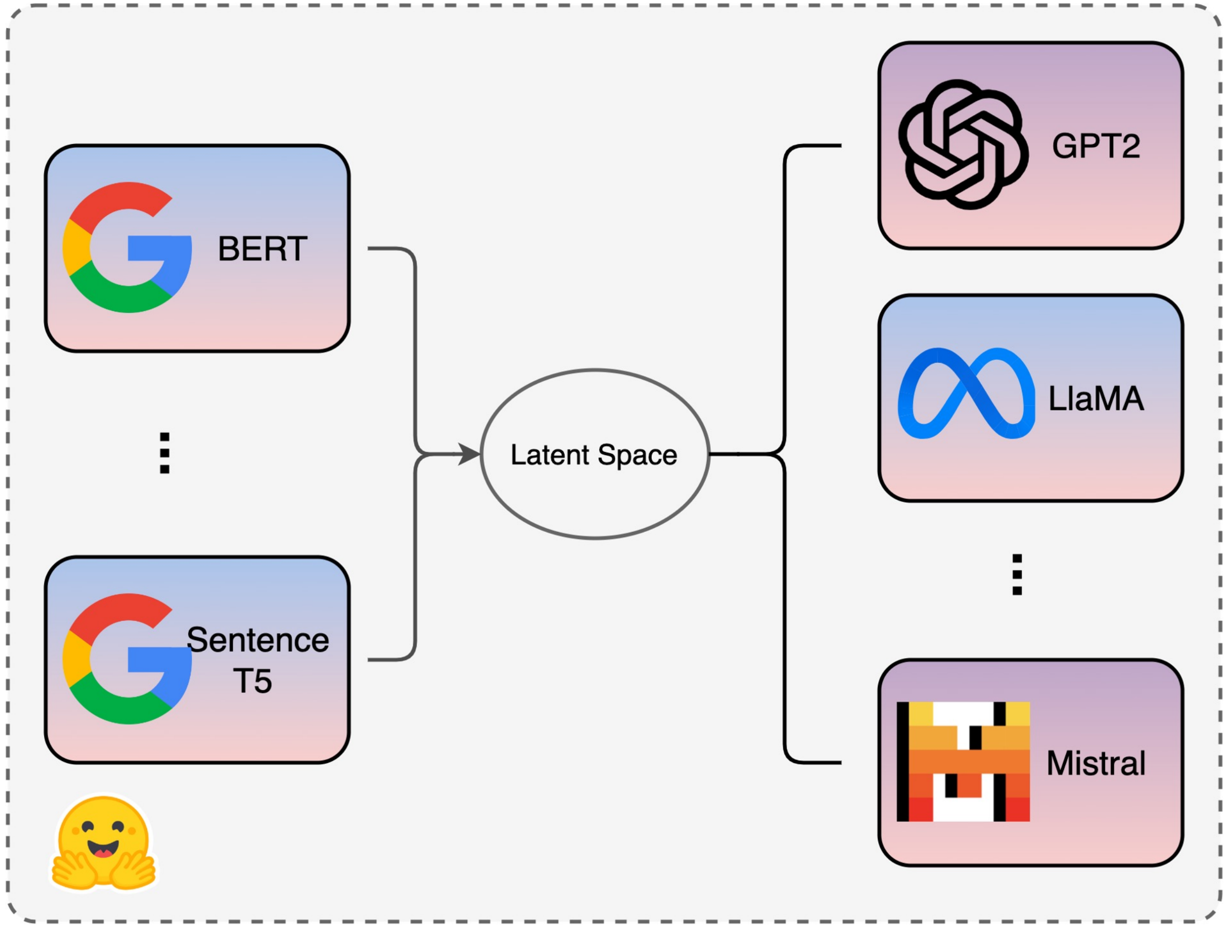
Only support Gaussian prior now.

https://github.com/neuro-symbolic-ai/LangVAE

Evaluation:

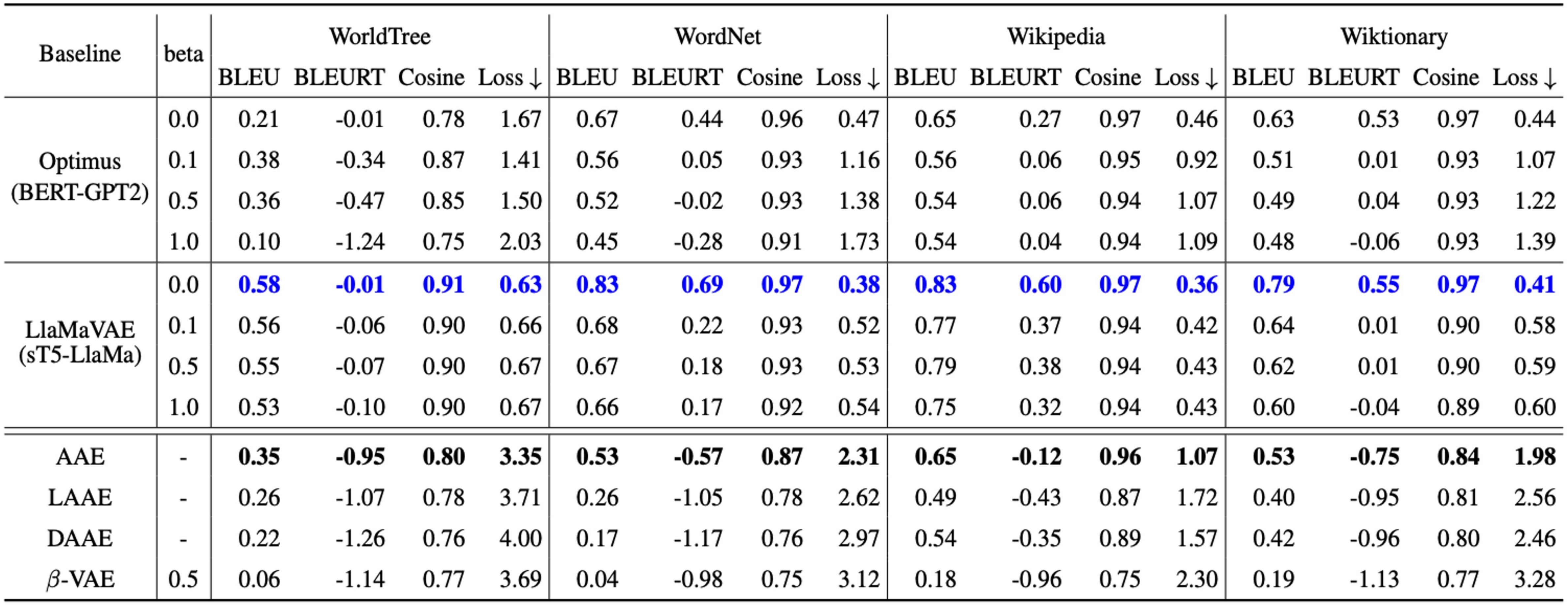
https://github.com/neuro-symbolic-ai/LangSpace

1. latent traversal;
2. interpolation;
3. arithmetic;
4. t-sne/UMAP/PCA;
5. disentanglement metrics.



Language VAE: 3. Train LangVAE

Train on the Language Modelling task: larger decoder (i.e., fixed LLaMA 1) leads to better performance[1].



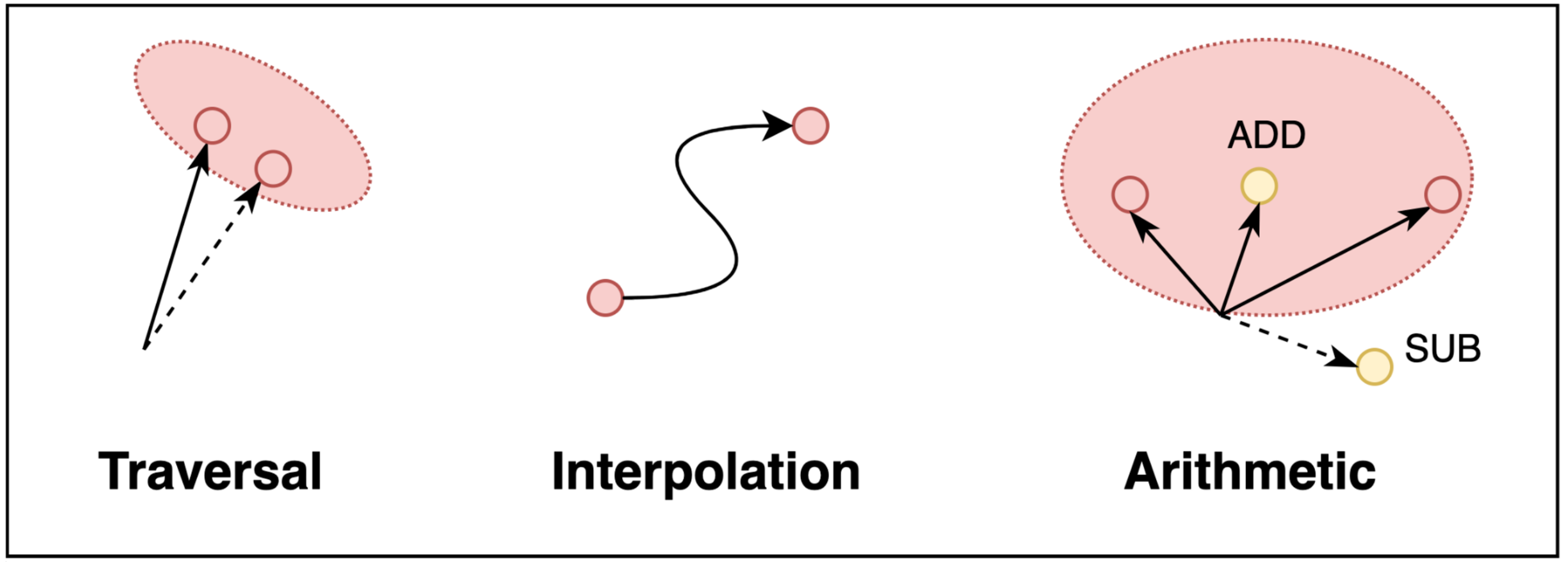
Baseline	beta	WorldTree				WordNet				Wikipedia				Wiktionary			
		BLEU	BLEURT	Cosine	Loss ↓	BLEU	BLEURT	Cosine	Loss ↓	BLEU	BLEURT	Cosine	Loss ↓	BLEU	BLEURT	Cosine	Loss ↓
Optimus (BERT-GPT2)	0.0	0.21	-0.01	0.78	1.67	0.67	0.44	0.96	0.47	0.65	0.27	0.97	0.46	0.63	0.53	0.97	0.44
	0.1	0.38	-0.34	0.87	1.41	0.56	0.05	0.93	1.16	0.56	0.06	0.95	0.92	0.51	0.01	0.93	1.07
	0.5	0.36	-0.47	0.85	1.50	0.52	-0.02	0.93	1.38	0.54	0.06	0.94	1.07	0.49	0.04	0.93	1.22
	1.0	0.10	-1.24	0.75	2.03	0.45	-0.28	0.91	1.73	0.54	0.04	0.94	1.09	0.48	-0.06	0.93	1.39
LlaMaVAE (sT5-LlaMa)	0.0	**0.58**	**-0.01**	**0.91**	**0.63**	**0.83**	**0.69**	**0.97**	**0.38**	**0.83**	**0.60**	**0.97**	**0.36**	**0.79**	**0.55**	**0.97**	**0.41**
	0.1	0.56	-0.06	0.90	0.66	0.68	0.22	0.93	0.52	0.77	0.37	0.94	0.42	0.64	0.01	0.90	0.58
	0.5	0.55	-0.07	0.90	0.67	0.67	0.18	0.93	0.53	0.79	0.38	0.94	0.43	0.62	0.01	0.90	0.59
	1.0	0.53	-0.10	0.90	0.67	0.66	0.17	0.92	0.54	0.75	0.32	0.94	0.43	0.60	-0.04	0.89	0.60
AAE	-	**0.35**	**-0.95**	**0.80**	**3.35**	**0.53**	**-0.57**	**0.87**	**2.31**	**0.65**	**-0.12**	**0.96**	**1.07**	**0.53**	**-0.75**	**0.84**	**1.98**
LAAE	-	0.26	-1.07	0.78	3.71	0.26	-1.05	0.78	2.62	0.49	-0.43	0.87	1.72	0.40	-0.95	0.81	2.56
DAAE	-	0.22	-1.26	0.76	4.00	0.17	-1.17	0.76	2.97	0.54	-0.35	0.89	1.57	0.42	-0.96	0.80	2.46
β-VAE	0.5	0.06	-1.14	0.77	3.69	0.04	-0.98	0.75	3.12	0.18	-0.96	0.75	2.30	0.19	-1.13	0.77	3.28

[1] Zhang, Y., Carvalho, D. S., Pratt-Hartmann, I., & Freitas, A. (2023). LlaMaVAE: Guiding Large Language Model Generation via Continuous Latent Sentence Spaces. *arXiv preprint arXiv:2312.13208*.

Language VAE: 3. Evaluate LangVAE

Evaluation: three semantic control operators to probe latent space geometry:

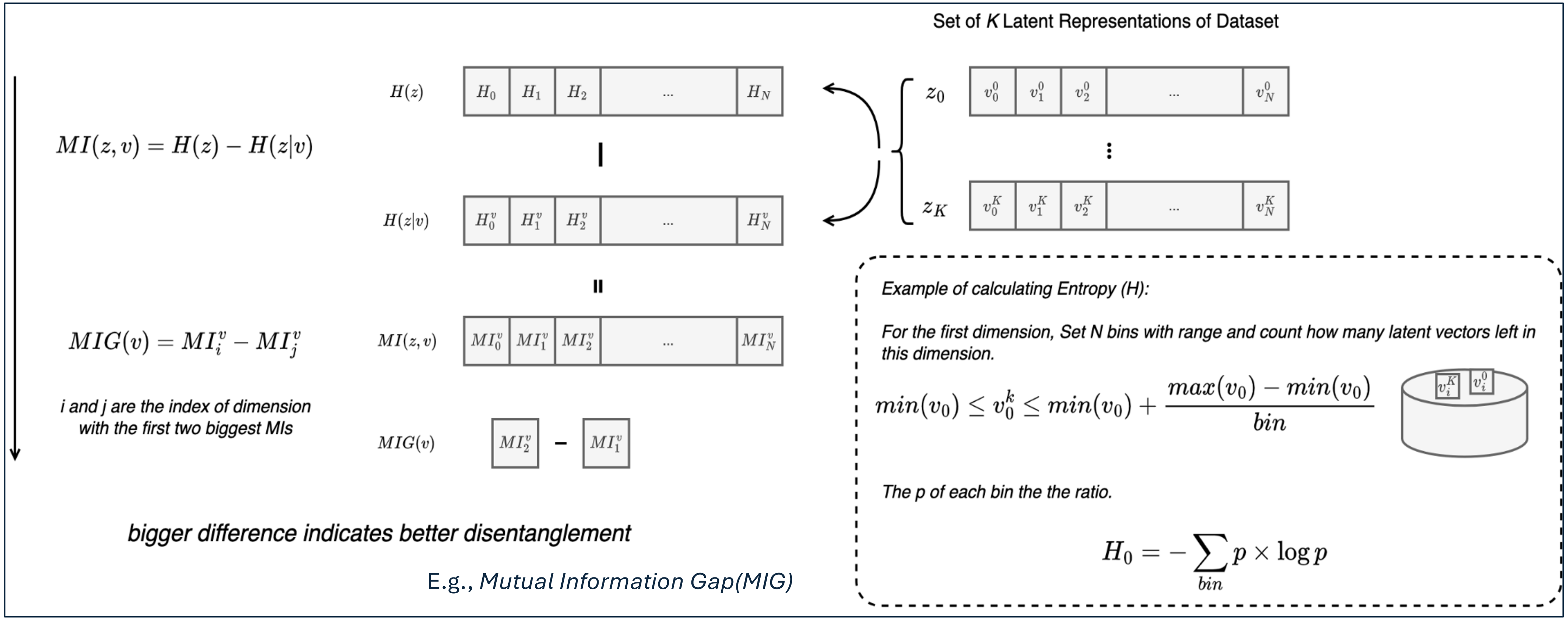
1. **Latent Traversal:** stochastic random walk over Gaussian space, such as *sampling each dimension, Brownian motion, Ornstein-Uhlenbeck*.
2. **Linear Interpolation:** generate a sequence of sentences following a spatial trajectory from source to target via latent arithmetics: $z_t = z_1 \cdot (1-t) + z_2 \cdot t$ with *t* increased from *0* to *1* by a step size of *0.1* where and represent latent vectors of source and target sentences, respectively.
3. **Latent Arithmetic:** Similar to word2vec, *king-man+woman=queen*, adding or subtracting latent sentence vectors.



Language VAE: 3. Evaluate LangVAE

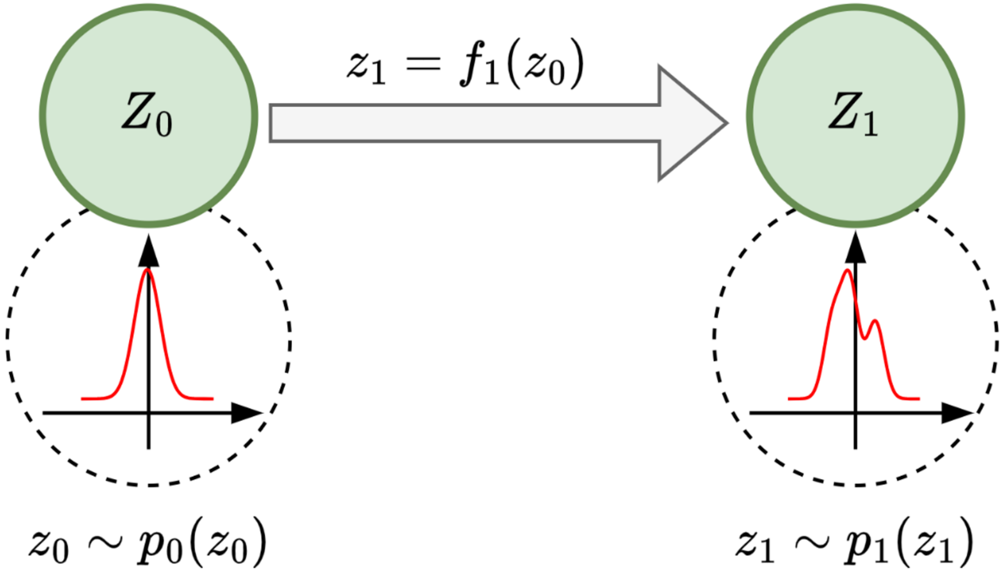
4. **Visualisation:** visualising semantic distribution/separation via t-SNE, UMAP, and PCA.

5. **Disentanglement metrics:** There are metrics widely applied in the Image domain to evaluate the disentanglement of latent spaces, including: 1. **mutual information gap (MIG)**, 2. **modularity**, 3. **disentanglement score**, 4. **completeness score**, 5. **informativeness score**, etc.


Set of K Latent Representations of Dataset
$MI(z,v) = H(z) - H(z|v)$
$H(z)$
H_0 H_1 H_2 ... H_N
$H(z|v)$
H_0^v H_1^v H_2^v ... H_N^v
$MI(z,v)$
MI_0^v MI_1^v MI_2^v ... MI_N^v
z_0 v_0^0 v_1^0 v_2^0 ... v_N^0
z_K v_0^K v_1^K v_2^K ... v_N^K
$MIG(v) = MI_i^v - MI_j^v$
i and j are the index of dimension with the first two biggest MIs
$MIG(v)$ MI_2^v – MI_1^v
bigger difference indicates better disentanglement
Example of calculating Entropy (H):
For the first dimension, Set N bins with range and count how many latent vectors left in this dimension.
$min(v_0) \leq v_0^k \leq min(v_0) + \frac{max(v_0) - min(v_0)}{bin}$
v_i^K v_i^0
The p of each bin the the ratio.
$H_0 = -\sum_{bin} p \times \log p$
E.g., *Mutual Information Gap(MIG)*

Normalising flow: 1. Change of variables

Change of variables formula: transformation from one distribution to another distribution.



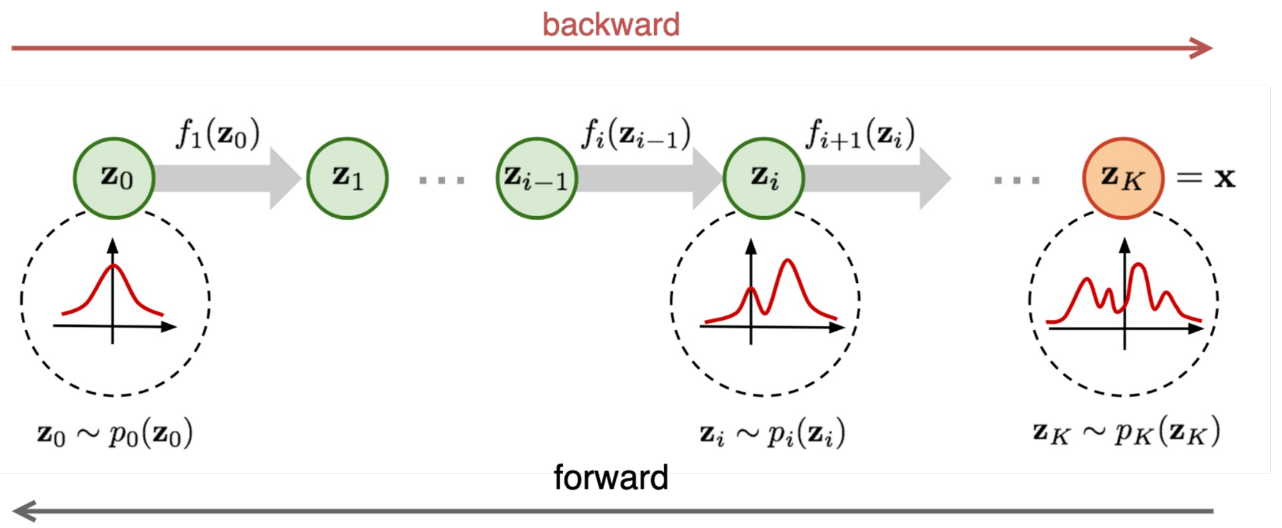
$$p_1(z_1) = p_0(z_0)\left|\frac{dz_0}{dz_1}\right|$$

$p_0(z_0)$	: a simple distribution
$p_1(z_1)$	: a complex distribution
f_1	: a neural network
$\left\lvert\frac{dz_o}{dz_1}\right\rvert$	: Jacobian determinant.

Normalise the probability density.

Normalising flow: 2. Objective function

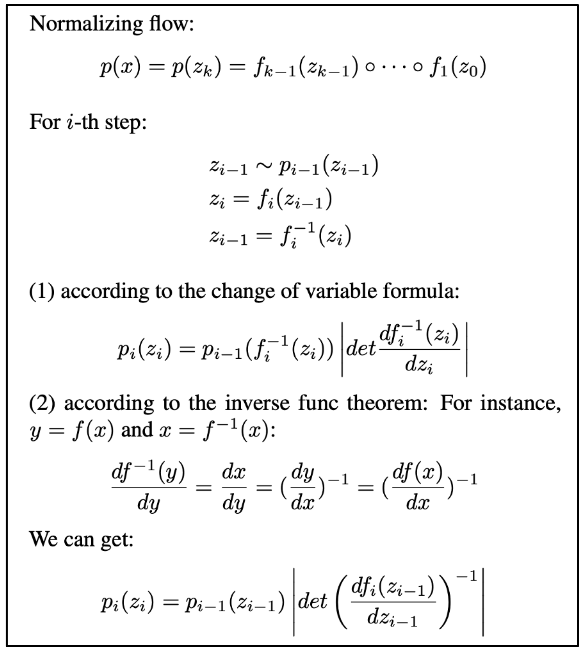
Normalising flow: a sequence of changes of variables.



source: https://lilianweng.github.io/posts/2018-10-13-flow-models/

Objective function: maximise the log-likelihood.

$$\log p(x) = \log p_0(z_0) - \sum_{i=1}^{K} \log \left| det \frac{df_i(z_{i-1})}{dz_{i-1}} \right|$$



Normalizing flow:

$$p(x) = p(z_k) = f_{k-1}(z_{k-1}) \circ \cdots \circ f_1(z_0)$$

For i-th step:

$$z_{i-1} \sim p_{i-1}(z_{i-1})$$
$$z_i = f_i(z_{i-1})$$
$$z_{i-1} = f_i^{-1}(z_i)$$

(1) according to the change of variable formula:

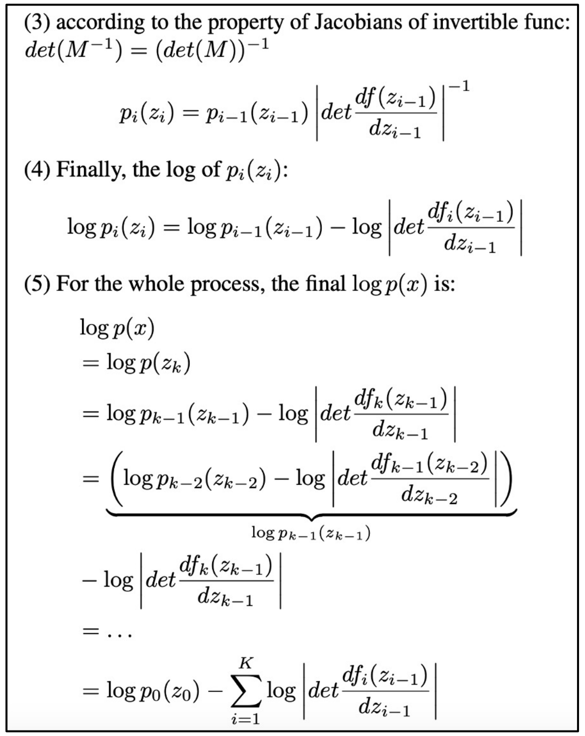
$$p_i(z_i) = p_{i-1}(f_i^{-1}(z_i)) \left| det \frac{df_i^{-1}(z_i)}{dz_i} \right|$$

(2) according to the inverse func theorem: For instance, $y = f(x)$ and $x = f^{-1}(x)$:

$$\frac{df^{-1}(y)}{dy} = \frac{dx}{dy} = (\frac{dy}{dx})^{-1} = (\frac{df(x)}{dx})^{-1}$$

We can get:

$$p_i(z_i) = p_{i-1}(z_{i-1}) \left| det \left(\frac{df_i(z_{i-1})}{dz_{i-1}} \right)^{-1} \right|$$



(3) according to the property of Jacobians of invertible func: $det(M^{-1}) = (det(M))^{-1}$

$$p_i(z_i) = p_{i-1}(z_{i-1}) \left| det \frac{df(z_{i-1})}{dz_{i-1}} \right|^{-1}$$

(4) Finally, the log of $p_i(z_i)$:

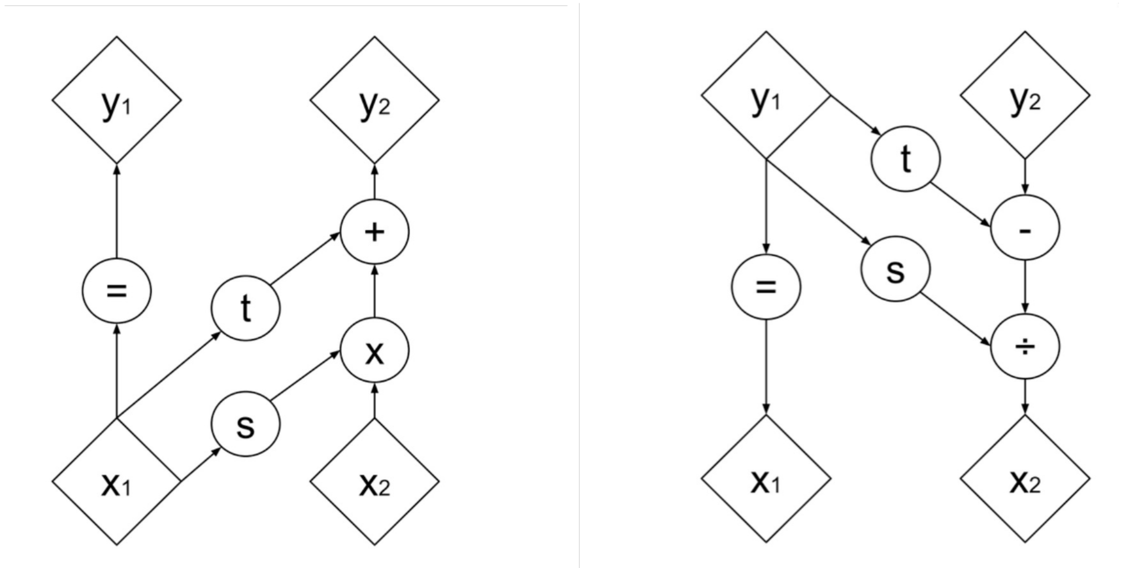
$$\log p_i(z_i) = \log p_{i-1}(z_{i-1}) - \log \left| det \frac{df_i(z_{i-1})}{dz_{i-1}} \right|$$

(5) For the whole process, the final $\log p(x)$ is:

$$\begin{aligned}
&\log p(x) \\
&= \log p(z_k) \\
&= \log p_{k-1}(z_{k-1}) - \log \left| det \frac{df_k(z_{k-1})}{dz_{k-1}} \right| \\
&= \underbrace{\left(\log p_{k-2}(z_{k-2}) - \log \left| det \frac{df_{k-1}(z_{k-2})}{dz_{k-2}} \right| \right)}_{\log p_{k-1}(z_{k-1})} \\
&- \log \left| det \frac{df_k(z_{k-1})}{dz_{k-1}} \right| \\
&= \ldots \\
&= \log p_0(z_0) - \sum_{i=1}^{K} \log \left| det \frac{df_i(z_{i-1})}{dz_{i-1}} \right|
\end{aligned}$$

Normalising flow: 3. Architecture

Architecture: each *f* is a neural network, such as affine coupling layer, which should satisfy two conditions:



(a) Forward propagation (b) Inverse propagation

1. get the inverse:

$$\boxed{x_{1:d}} = \boxed{y_{1:d}}$$
$$x_{d+1:D} = (y_{d+1:D} - t(\boxed{y_{1:d}})) \odot exp(-s(\boxed{y_{1:d}}))$$

the inputs of t and s do not change in both direction, therefore, they can be any kind of neural network.

2. easy to compute Jacobian:

$$\frac{\partial y}{\partial x^T} = \begin{bmatrix} I_d & 0 \\ \frac{\partial y_{d+1:D}}{\partial x^T_{1:d}} & diag(exp[s(x_{1:D})]) \end{bmatrix}$$

$$y_{1:d} = x_{1:d}$$
$$y_{d+1:D} = x_{d+1:D} \odot exp(s(x_{1:d})) + t(x_{1:d})$$

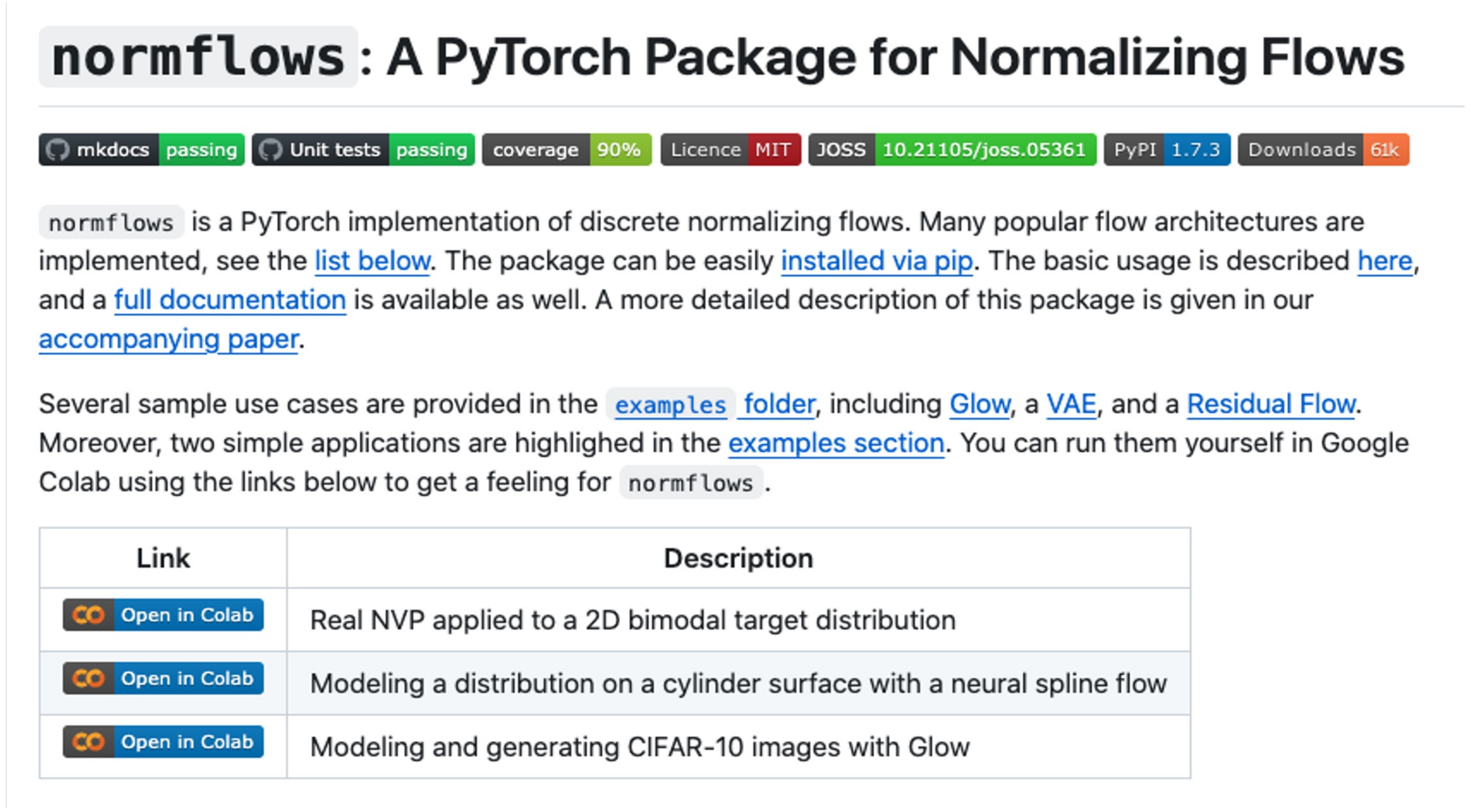
s and *t* can be arbitrary neural networks.

Normalising flow: Pytorch library

Pytorch framework for normalising flow:

FrEIA: https://vislearn.github.io/FrEIA/_build/html/tutorial/quickstart.html

`normflows`: https://github.com/VincentStimper/normalizing-flows



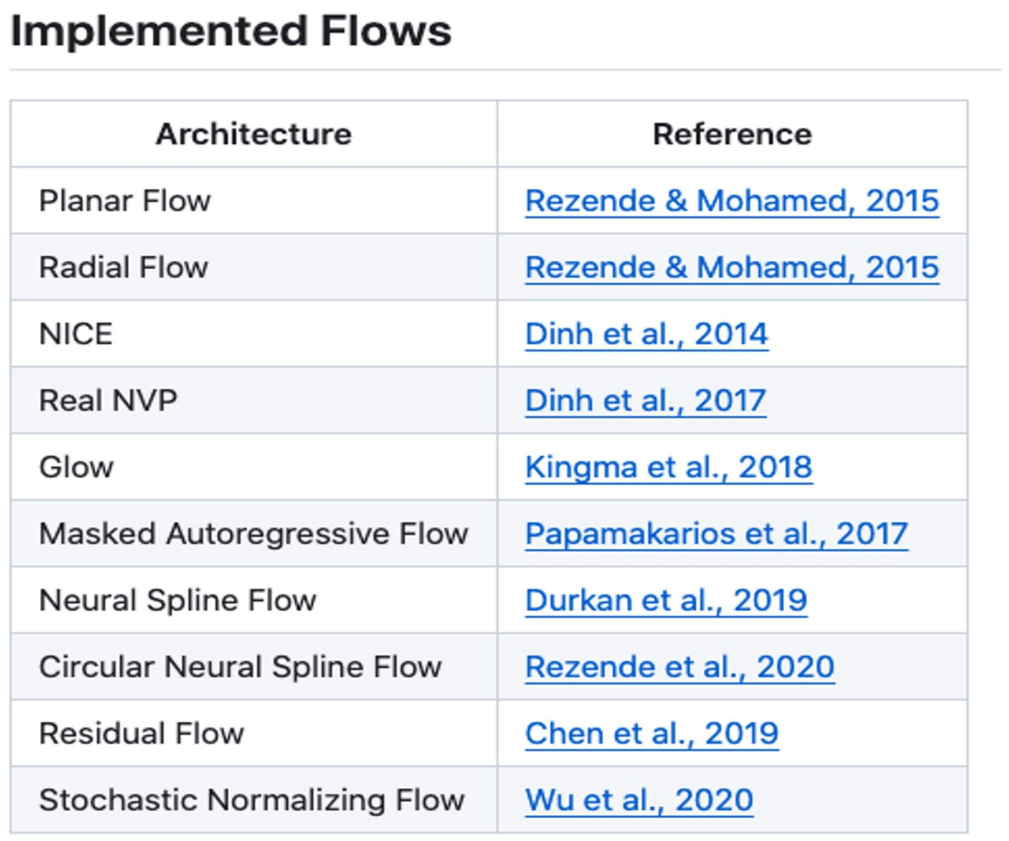
`normflows`: A PyTorch Package for Normalizing Flows

mkdocs passing | Unit tests passing | coverage 90% | Licence MIT | JOSS 10.21105/joss.05361 | PyPI 1.7.3 | Downloads 61k

`normflows` is a PyTorch implementation of discrete normalizing flows. Many popular flow architectures are implemented, see the list below. The package can be easily installed via pip. The basic usage is described here, and a full documentation is available as well. A more detailed description of this package is given in our accompanying paper.

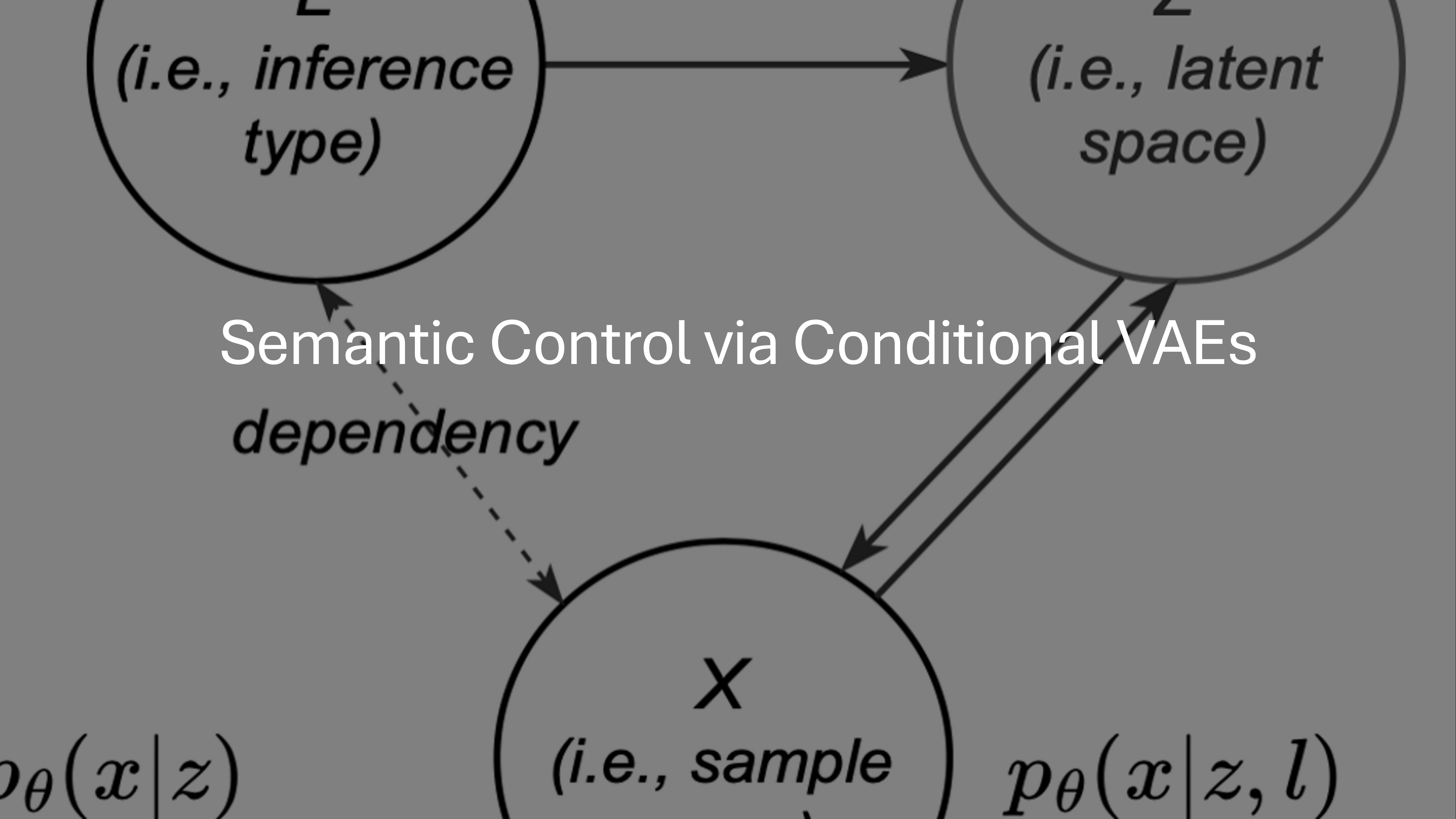
Several sample use cases are provided in the `examples` folder, including Glow, a VAE, and a Residual Flow. Moreover, two simple applications are highlighed in the examples section. You can run them yourself in Google Colab using the links below to get a feeling for `normflows`.

Link	Description
Open in Colab	Real NVP applied to a 2D bimodal target distribution
Open in Colab	Modeling a distribution on a cylinder surface with a neural spline flow
Open in Colab	Modeling and generating CIFAR-10 images with Glow



Implemented Flows

Architecture	Reference
Planar Flow	Rezende & Mohamed, 2015
Radial Flow	Rezende & Mohamed, 2015
NICE	Dinh et al., 2014
Real NVP	Dinh et al., 2017
Glow	Kingma et al., 2018
Masked Autoregressive Flow	Papamakarios et al., 2017
Neural Spline Flow	Durkan et al., 2019
Circular Neural Spline Flow	Rezende et al., 2020
Residual Flow	Chen et al., 2019
Stochastic Normalizing Flow	Wu et al., 2020



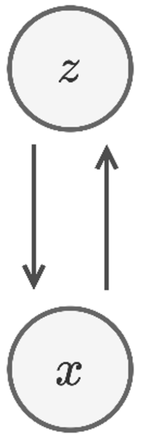
Semantic Control via Conditional VAEs

Conditional VAEs

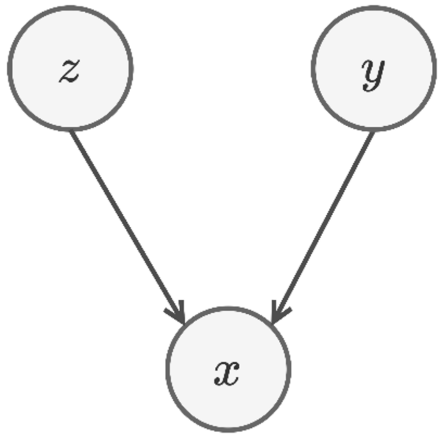
Recall: objective function of VAE (ELBO):

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x})} \log p_\theta(\mathbf{x}|\mathbf{z}) - D_{\mathrm{KL}}(q_\phi(\mathbf{z}|\mathbf{x}) \| p_\theta(\mathbf{z}))$$

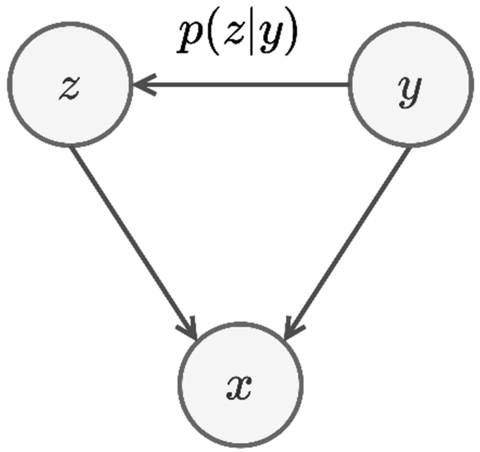
Two types of conditions: (1) z and y (i.e., label) are independent; (2) z and y are dependent.



$$p(x,z) = p(x|z)p(z)$$



$$p(x,y,z) = p(x|y,z) \times \underbrace{p(z)}_{prior} \times p(y)$$

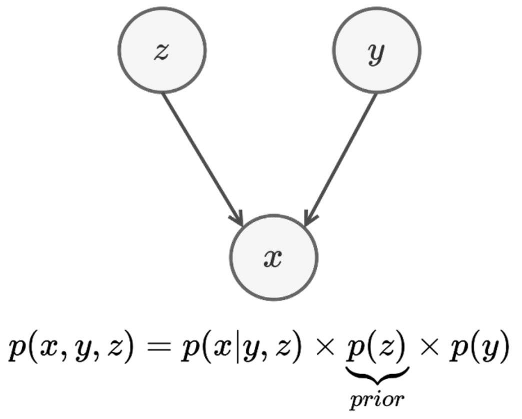


$$p(x,y,z) = p(x|y,z) \times \underbrace{p(z|y)}_{prior} \times p(y)$$

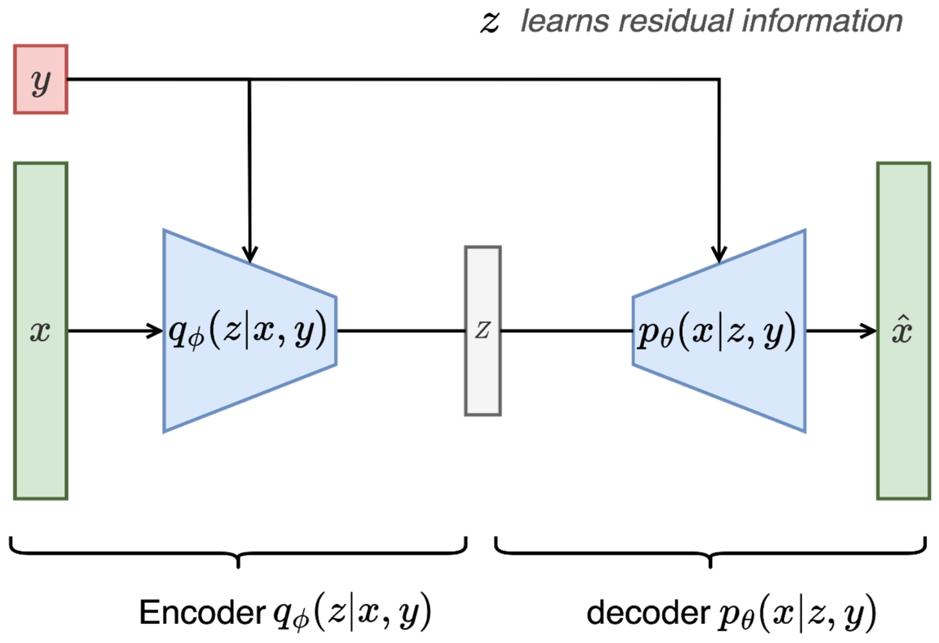
From introduction to CVAE: https://beckham.nz/2023/04/27/conditional-vaes.html

CVAE: when y and z are independent

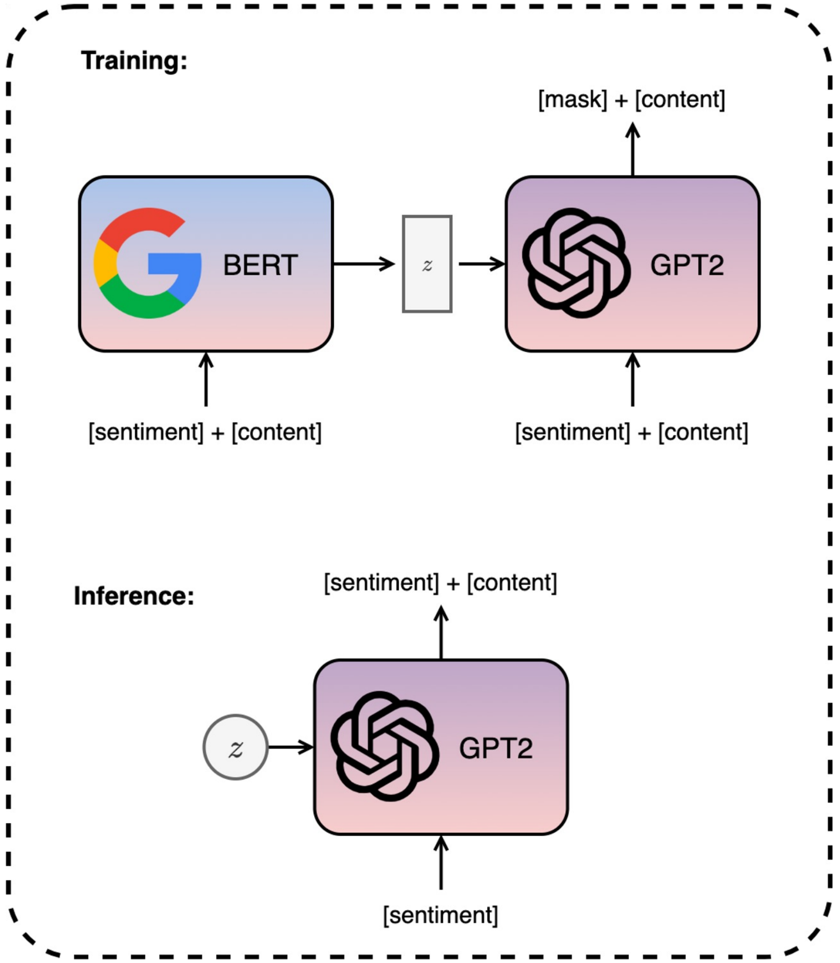
Independency: when y and *z* are independent, the label is injected into encoder and decoder during training. The prior is a fixed distribution.



$$p(x,y,z) = p(x|y,z) \times \underbrace{p(z)}_{prior} \times p(y)$$

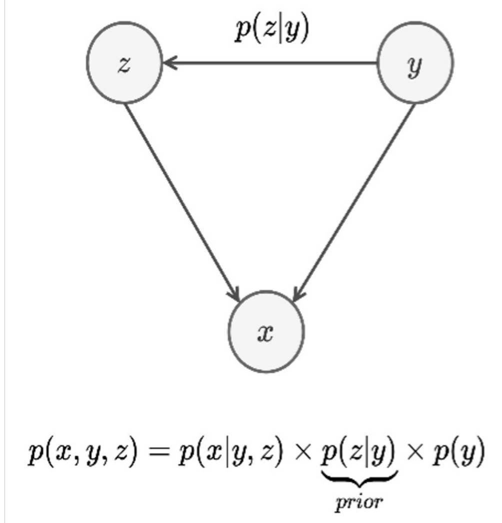


$$\mathbb{E}_{z\sim q_\phi(z|x,y)} \log p_\theta(x|z,y) - D_{KL}(q_\phi(z|x,y)||p_\theta(z))$$

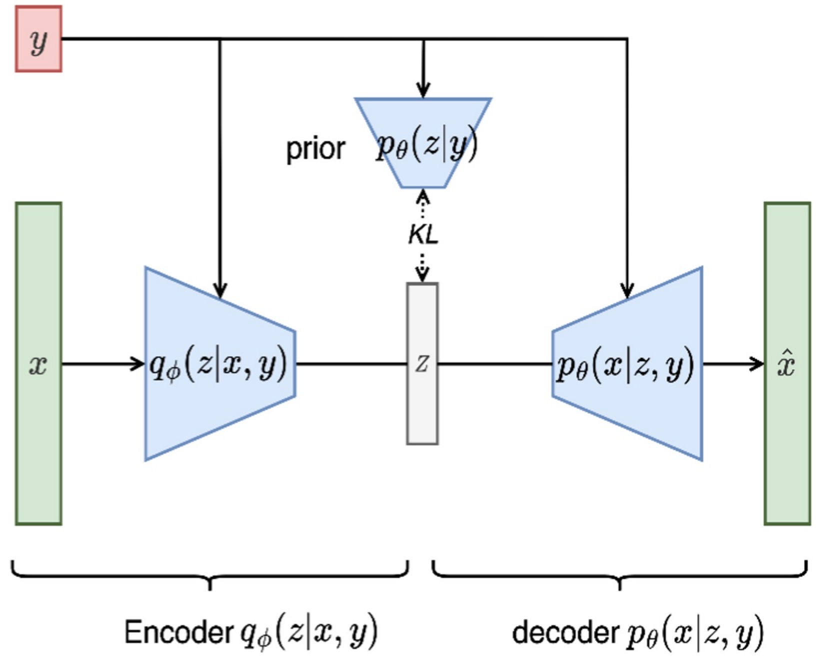


CVAE: when y and z are dependent

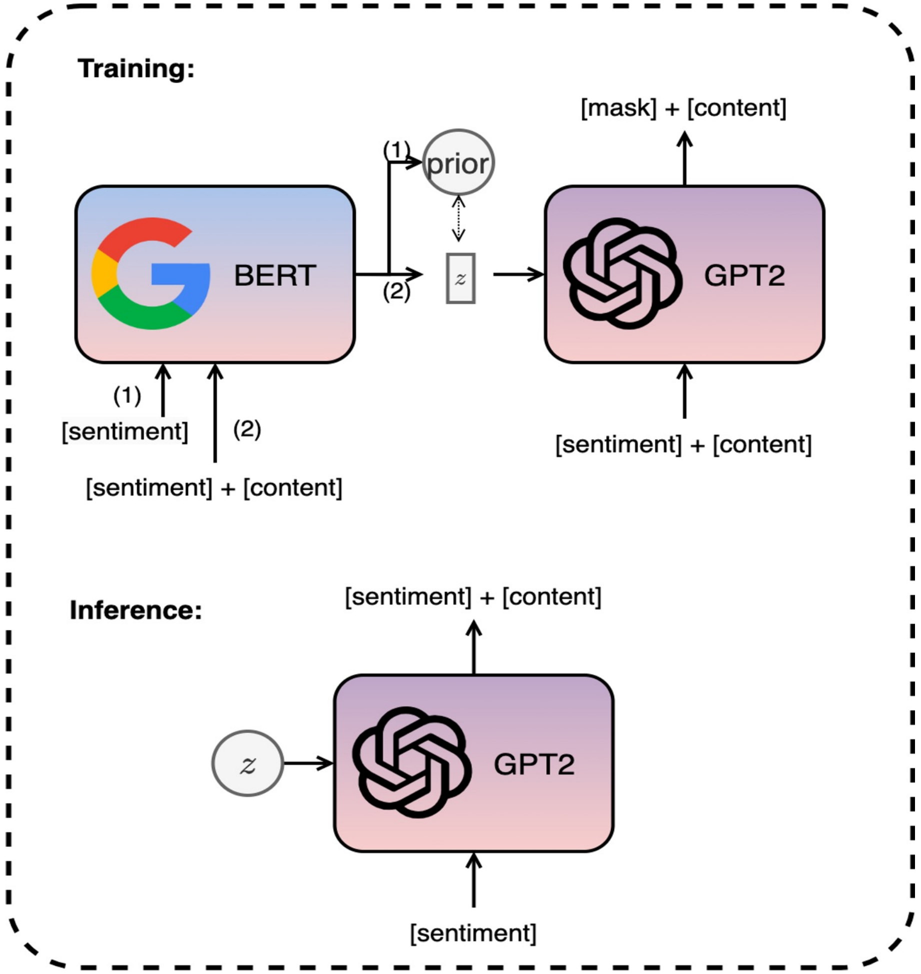
Dependency: when *y* and *z* are dependent, the prior can be a trainable encoder. The label is injected into encoder, decoder, and a "trainable" prior encoder.



$$p(x,y,z) = p(x|y,z) \times \underbrace{p(z|y)}_{prior} \times p(y)$$



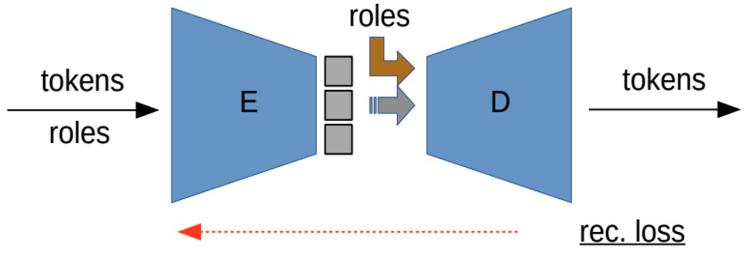
$$\mathbb{E}_{z\sim q_\phi(z|x,y)} \log p_\theta(x|z,y) - D_{KL}(q_\phi(z|x,y)||p_\theta(z|y))$$

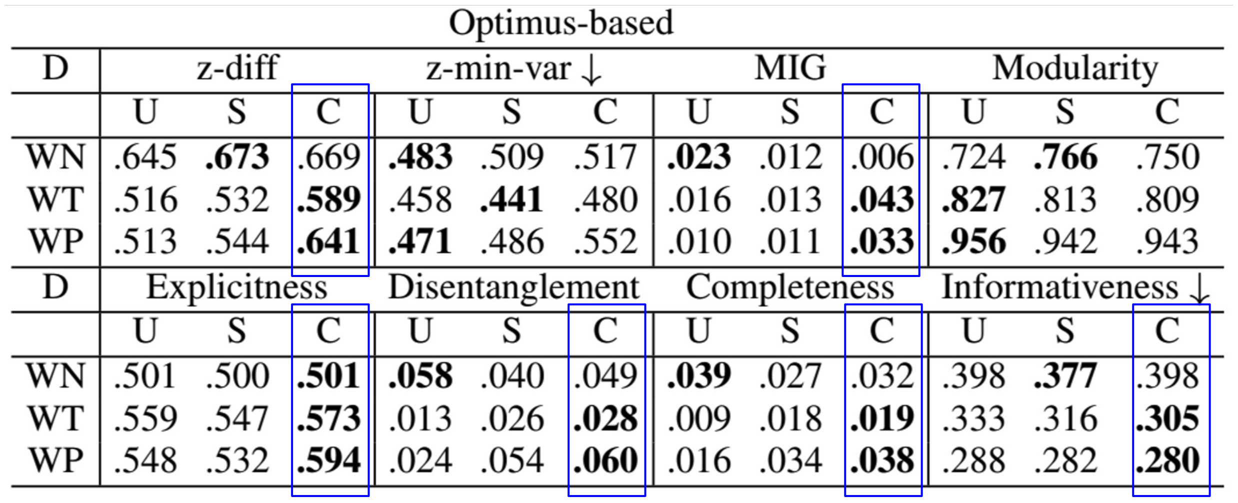


Carvalho, D. S., Mercatali, G., Zhang, Y., & Freitas, A. **Learning disentangled representations for natural language definitions.** EACL Findings (2023).

Background: investigating the disentanglement of semantic role label via CVAE when *y* and *z* are independent, denoted by C.

$$\mathbb{E}_{z\sim q_\phi(z|x,y)} \log p_\theta(x|z,y) - D_{KL}(q_\phi(z|x,y)||p_\theta(z))$$

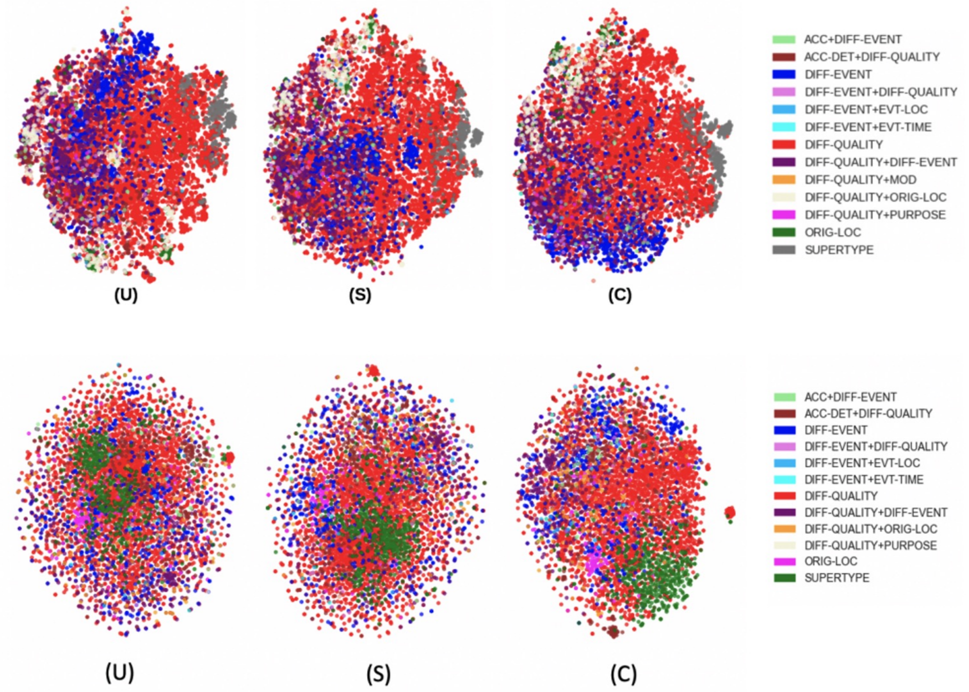




Optimus-based

D	z-diff U	z-diff S	z-diff C	z-min-var ↓ U	z-min-var ↓ S	z-min-var ↓ C	MIG U	MIG S	MIG C	Modularity U	Modularity S	Modularity C
WN	.645	**.673**	.669	**.483**	.509	.517	**.023**	.012	.006	.724	**.766**	.750
WT	.516	.532	**.589**	.458	**.441**	.480	.016	.013	**.043**	**.827**	.813	.809
WP	.513	.544	**.641**	**.471**	.486	.552	.010	.011	**.033**	**.956**	.942	.943

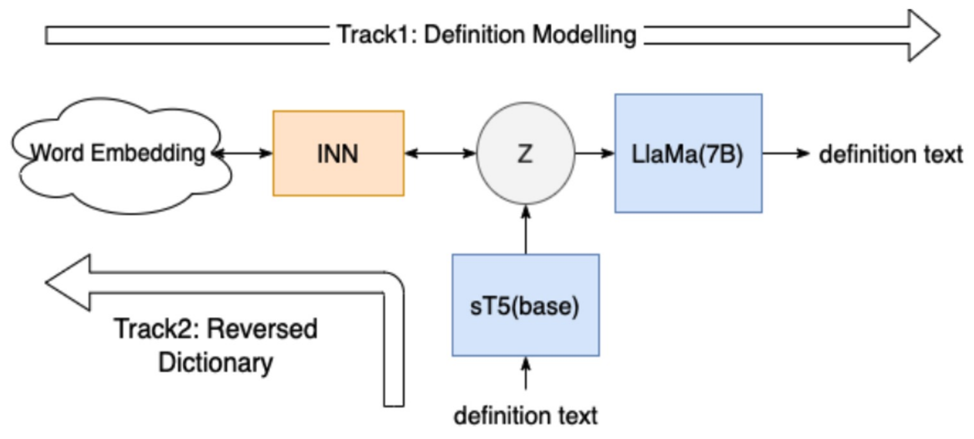
D	Explicitness U	Explicitness S	Explicitness C	Disentanglement U	Disentanglement S	Disentanglement C	Completeness U	Completeness S	Completeness C	Informativeness ↓ U	Informativeness ↓ S	Informativeness ↓ C
WN	.501	.500	**.501**	**.058**	.040	.049	**.039**	.027	.032	.398	**.377**	.398
WT	.559	.547	**.573**	.013	.026	**.028**	.009	.018	**.019**	.333	.316	**.305**
WP	.548	.532	**.594**	.024	.054	**.060**	.016	.034	**.038**	.288	.282	**.280**

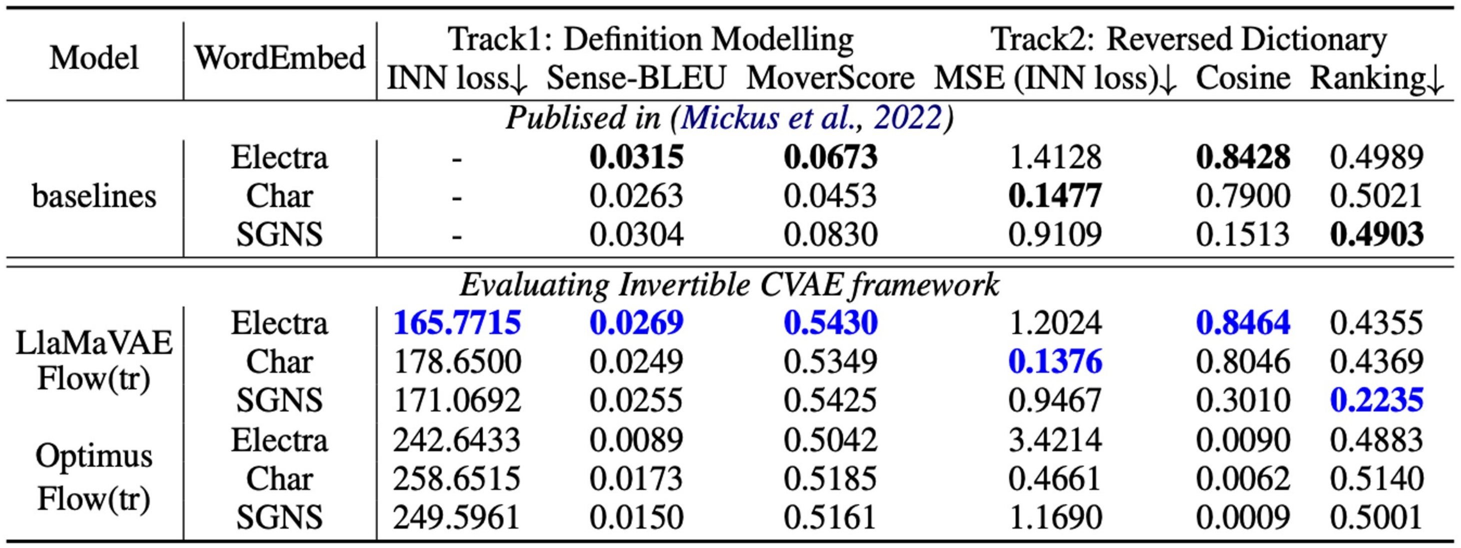


Observation: CVAE can improve semantic role disentanglement.

Zhang, Y., Carvalho, D. S., Pratt-Hartmann, I., & Freitas, A.
LlaMaVAE: Guiding Large Language Model Generation via Continuous Latent Sentence Spaces.
arXiv:2312.13208 (2023).

Background: investigating CVAE where the condition is word embedding, with the help of normalizing flow, we can now generate definition text condition on word embedding in definition modelling task[1].





Model	WordEmbed	Track1: Definition Modelling			Track2: Reversed Dictionary		
		INN loss↓	Sense-BLEU	MoverScore	MSE (INN loss)↓	Cosine	Ranking↓
Publised in (Mickus et al., 2022)							
baselines	Electra	-	**0.0315**	**0.0673**	1.4128	**0.8428**	0.4989
	Char	-	0.0263	0.0453	**0.1477**	0.7900	0.5021
	SGNS	-	0.0304	0.0830	0.9109	0.1513	**0.4903**
Evaluating Invertible CVAE framework							
LlaMaVAE Flow(tr)	Electra	**165.7715**	**0.0269**	**0.5430**	1.2024	**0.8464**	0.4355
	Char	178.6500	0.0249	0.5349	**0.1376**	0.8046	0.4369
	SGNS	171.0692	0.0255	0.5425	0.9467	0.3010	**0.2235**
Optimus Flow(tr)	Electra	242.6433	0.0089	0.5042	3.4214	0.0090	0.4883
	Char	258.6515	0.0173	0.5185	0.4661	0.0062	0.5140
	SGNS	249.5961	0.0150	0.5161	1.1690	0.0009	0.5001

Normalising flows can plug-in into pretrained VAEs to conditionally control text generation.

[1] Timothee Mickus, Kees Van Deemter, Mathieu Constant, and Denis Paperno. 2022. Semeval-2022 task 1: CODWOE – comparing dictionaries and word embeddings. In Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022), pages 1–14, Seattle, United States. Association for Computational Linguistics.

Zhang, Y., Carvalho, D. S., Pratt-Hartmann, I., & Freitas, A.
Towards controllable natural language inference through lexical inference types.
under review (2024).

Motivation & Question: Can natural language inference process be controlled via labels?

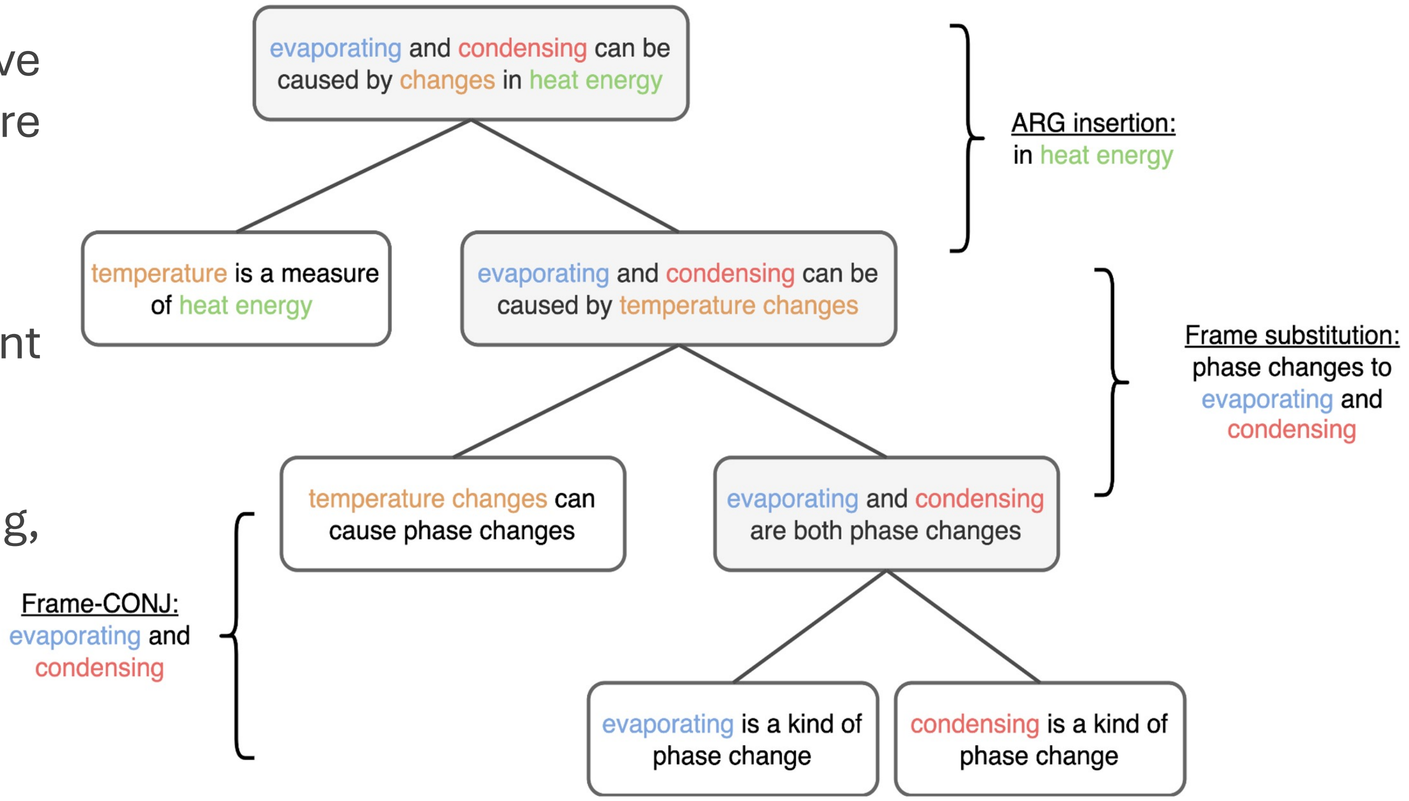
Target: we focus on syllogistic-style deductive inference (2 premises, 1 conclusion) to explore the controllability of explanatory NLI.

Contribution:

(1) Framing the expl. NLI model as a latent variable model.

(2) Ling./inf. priors can help model training, inference, and delivering inference control.

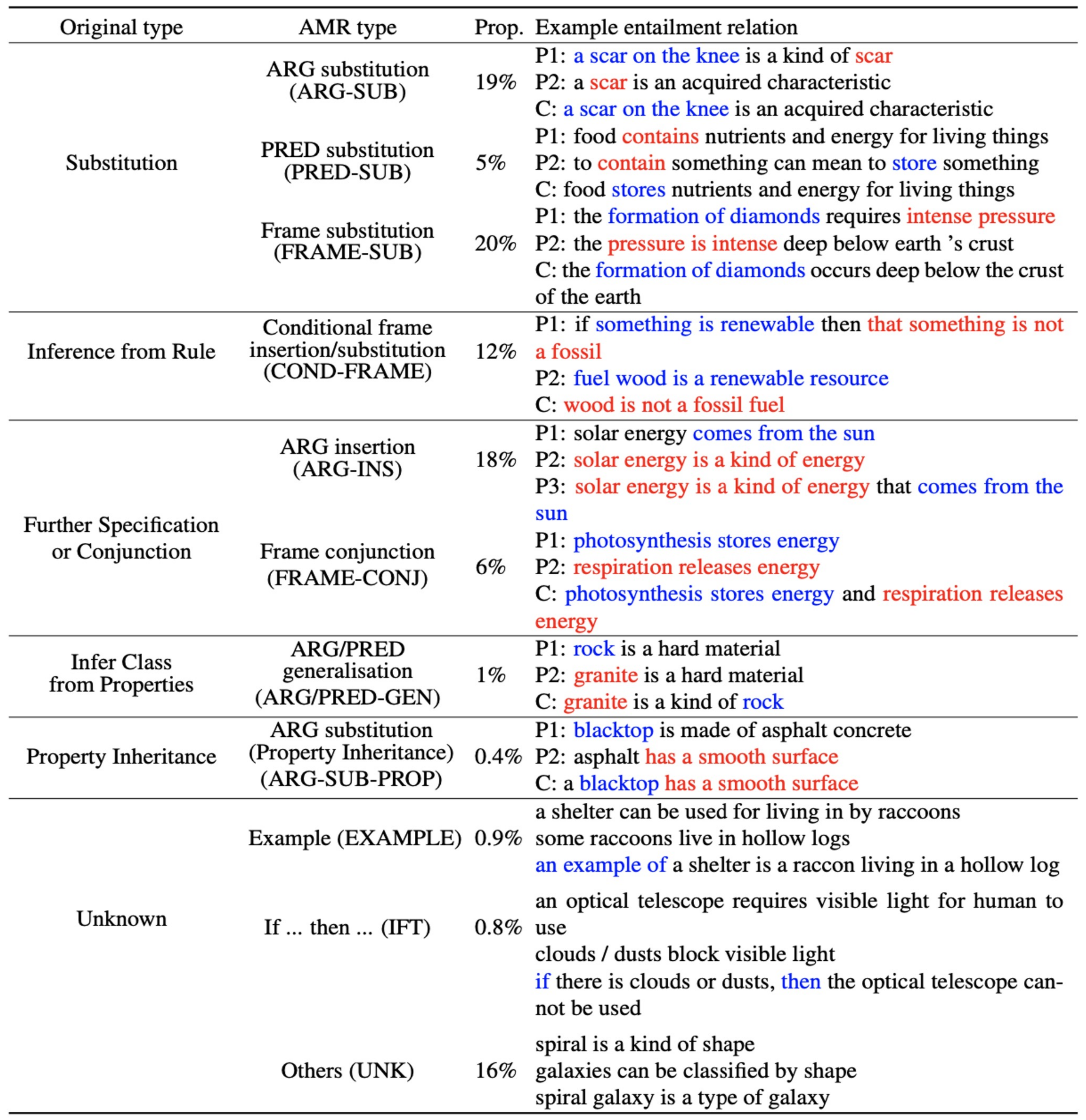
Question: in which way are evaporation and condensation are similar?
Answer: both are caused by phase changes in heat energy


evaporating and condensing can be caused by changes in heat energy
temperature is a measure of heat energy
evaporating and condensing can be caused by temperature changes
temperature changes can cause phase changes
evaporating and condensing are both phase changes
evaporating is a kind of phase change
condensing is a kind of phase change
ARG insertion: in heat energy
Frame substitution: phase changes to evaporating and condensing
Frame-CONJ: evaporating and condensing

[1] Dalvi, B., Jansen, P., Tafjord, O., Xie, Z., Smith, H., Pipatanangkura, L., & Clark, P. (2021). Explaining answers with entailment trees. *arXiv preprint arXiv:2104.08661*.

Methodology

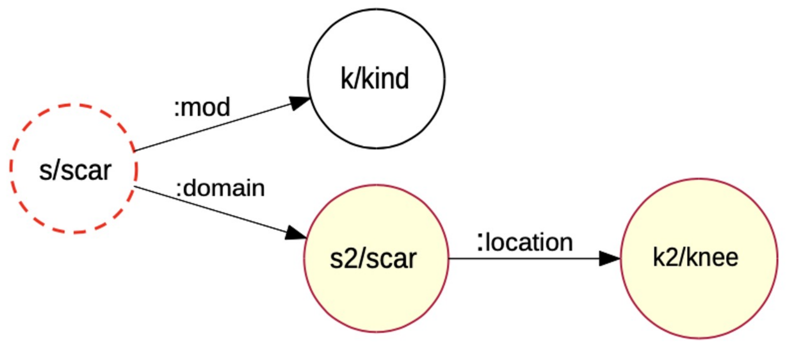
Annotation: For each inference pair in EntailmentBank, we annotate it via Abstract Meaning Representation (AMR) graph. The total number of annotation is around 5000.



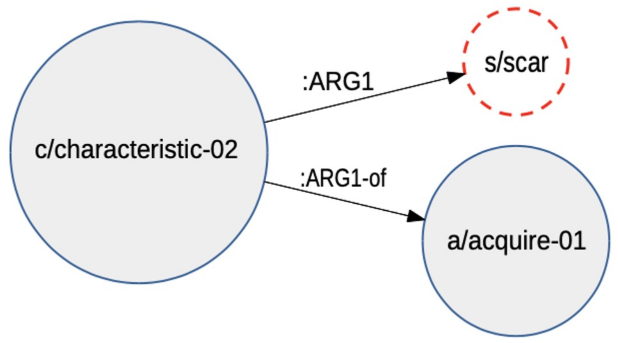
Original type	AMR type	Prop.	Example entailment relation
Substitution	ARG substitution (ARG-SUB)	19%	P1: a scar on the knee is a kind of scar
			P2: a scar is an acquired characteristic
			C: a scar on the knee is an acquired characteristic
	PRED substitution (PRED-SUB)	5%	P1: food contains nutrients and energy for living things
			P2: to contain something can mean to store something
			C: food stores nutrients and energy for living things
	Frame substitution (FRAME-SUB)	20%	P1: the formation of diamonds requires intense pressure
			P2: the pressure is intense deep below earth 's crust
			C: the formation of diamonds occurs deep below the crust of the earth
Inference from Rule	Conditional frame insertion/substitution (COND-FRAME)	12%	P1: if something is renewable then that something is not a fossil
			P2: fuel wood is a renewable resource
			C: wood is not a fossil fuel
Further Specification or Conjunction	ARG insertion (ARG-INS)	18%	P1: solar energy comes from the sun
			P2: solar energy is a kind of energy
			P3: solar energy is a kind of energy that comes from the sun
	Frame conjunction (FRAME-CONJ)	6%	P1: photosynthesis stores energy
			P2: respiration releases energy
			C: photosynthesis stores energy and respiration releases energy
Infer Class from Properties	ARG/PRED generalization (ARG/PRED-GEN)	1%	P1: rock is a hard material
			P2: granite is a hard material
			C: granite is a kind of rock
Property Inheritance	ARG substitution (Property Inheritance) (ARG-SUB-PROP)	0.4%	P1: blacktop is made of asphalt concrete
			P2: asphalt has a smooth surface
			C: a blacktop has a smooth surface
Unknown	Example (EXAMPLE)	0.9%	a shelter can be used for living in by raccoons
			some raccoons live in hollow logs
			an example of a shelter is a raccon living in a hollow log
	If ... then ... (IFT)	0.8%	an optical telescope requires visible light for human to use
			clouds / dusts block visible light
			if there is clouds or dusts, then the optical telescope cannot be used
	Others (UNK)	16%	spiral is a kind of shape
			galaxies can be classified by shape
			spiral galaxy is a type of galaxy

ARG-SUB

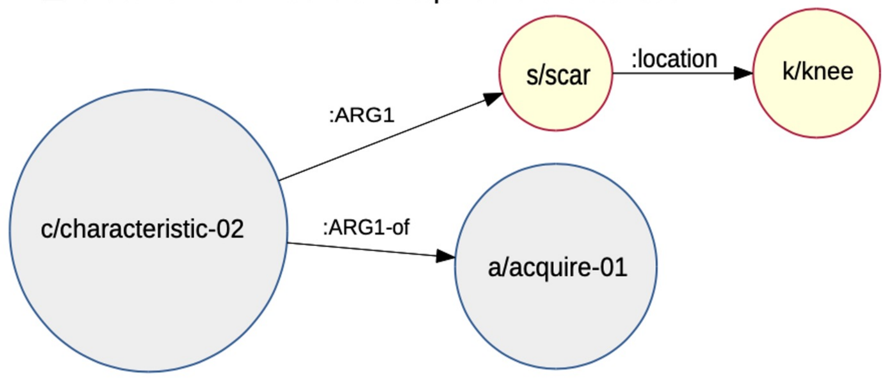
P1: a scar on the knee is a kind of scar



P2: a scar is an acquired characteristic

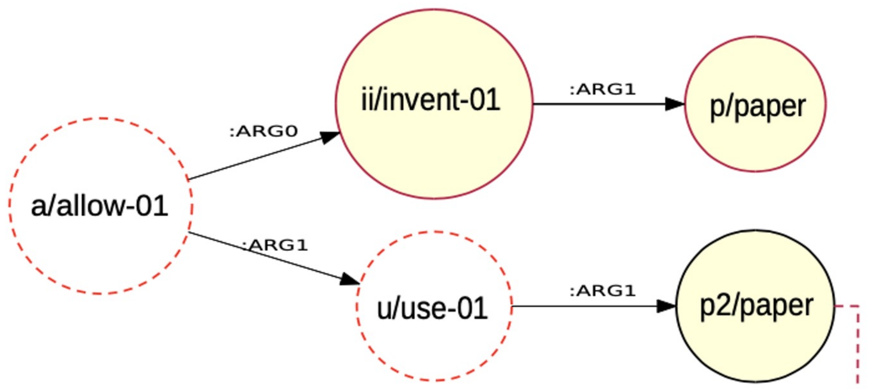


C: a scar on the knee is an acquired characteristic

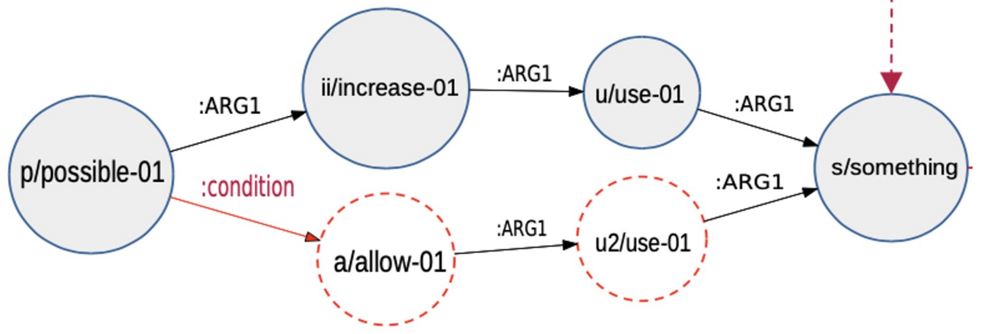


COND-FRAME

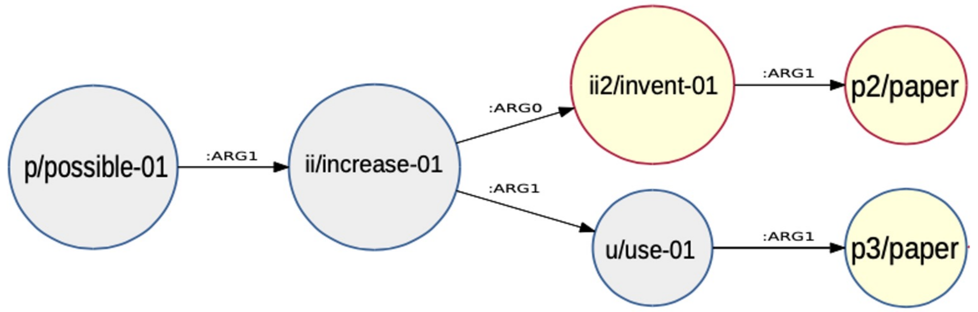
P1: inventing paper allows paper to be used



P2: if something is allowed to be used then the use of that something might increase

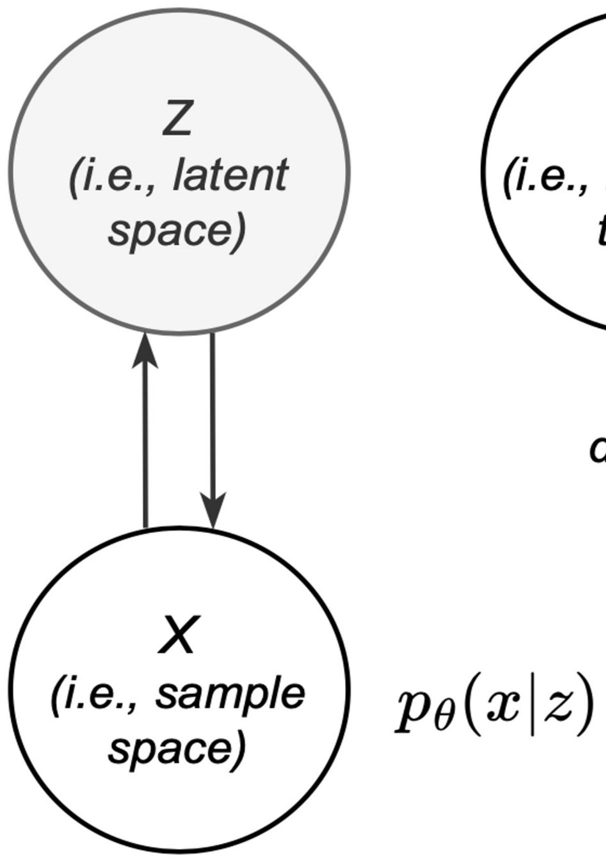


C: inventing paper might increase the use of paper

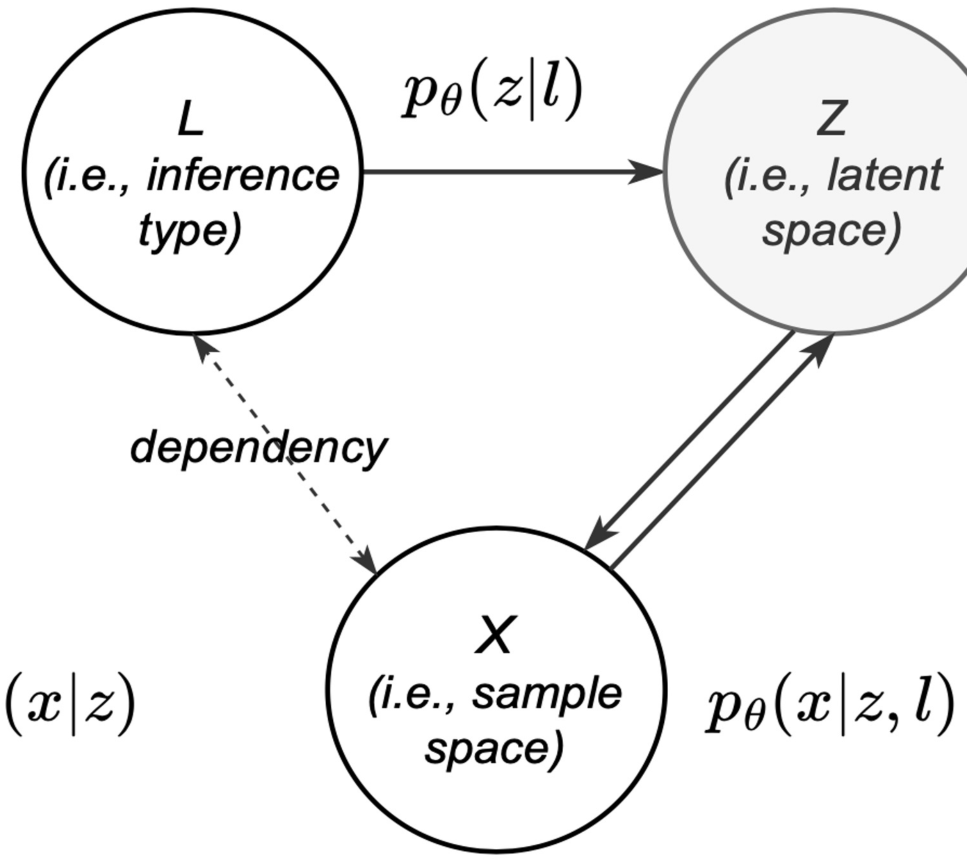


Methodology

Latent variable NLI model: (1) frame the NLI model, such as T5, as a latent variable model. For the conditional case, the label and *z* are dependent.



Latent NLI model

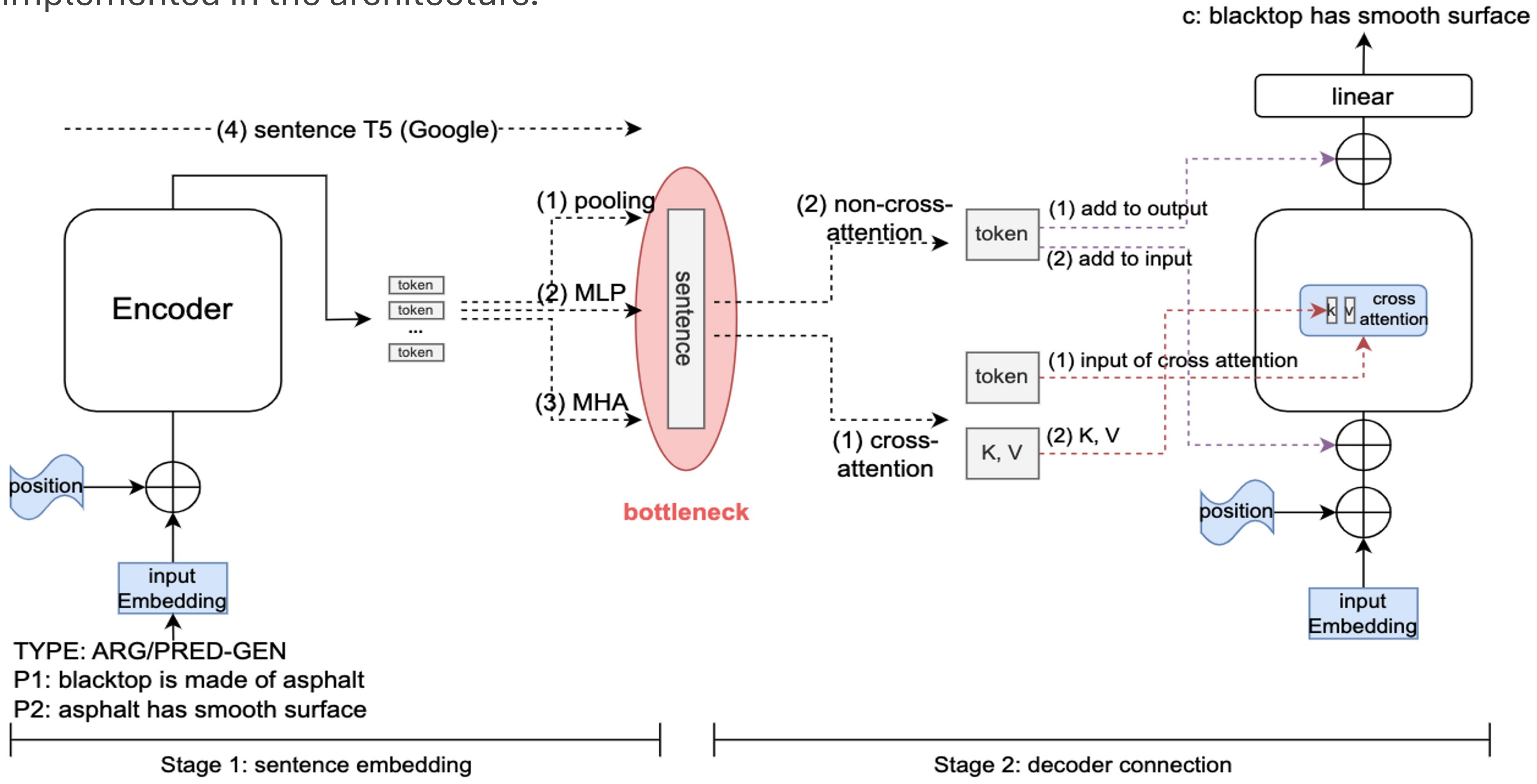


Conditional Latent NLI model

Methodology

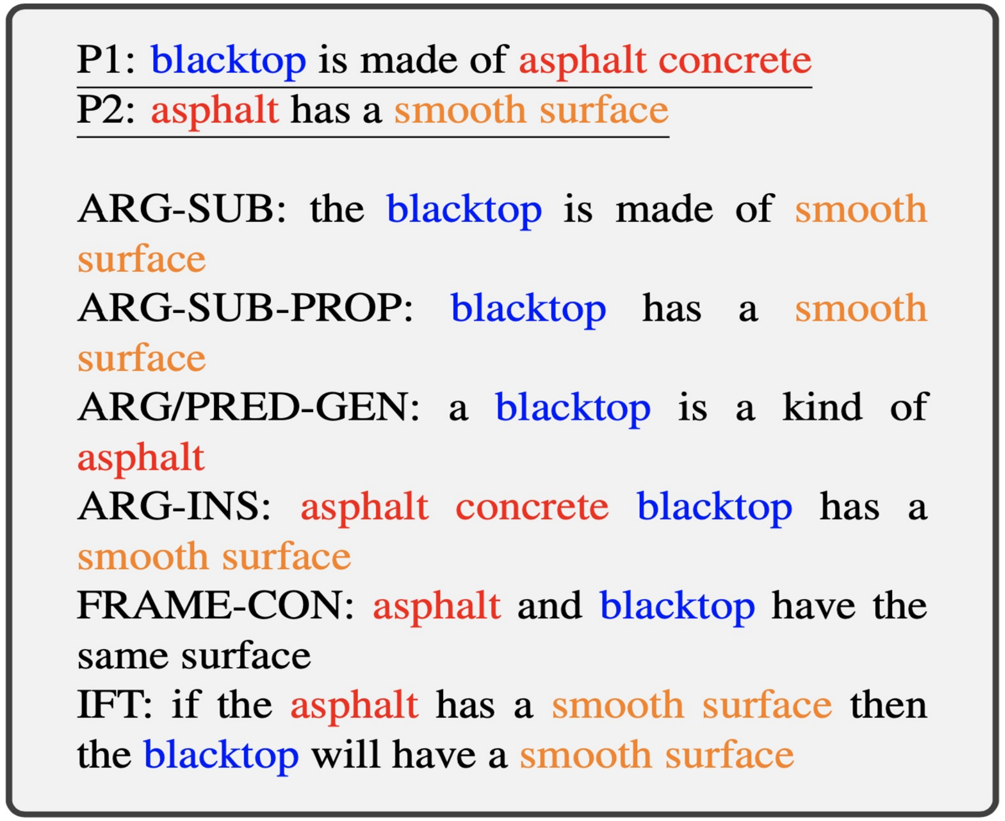
Latent variable NLI model: (1) frame the NLI model, such as T5, as a latent variable model.

As implemented in the architecture.



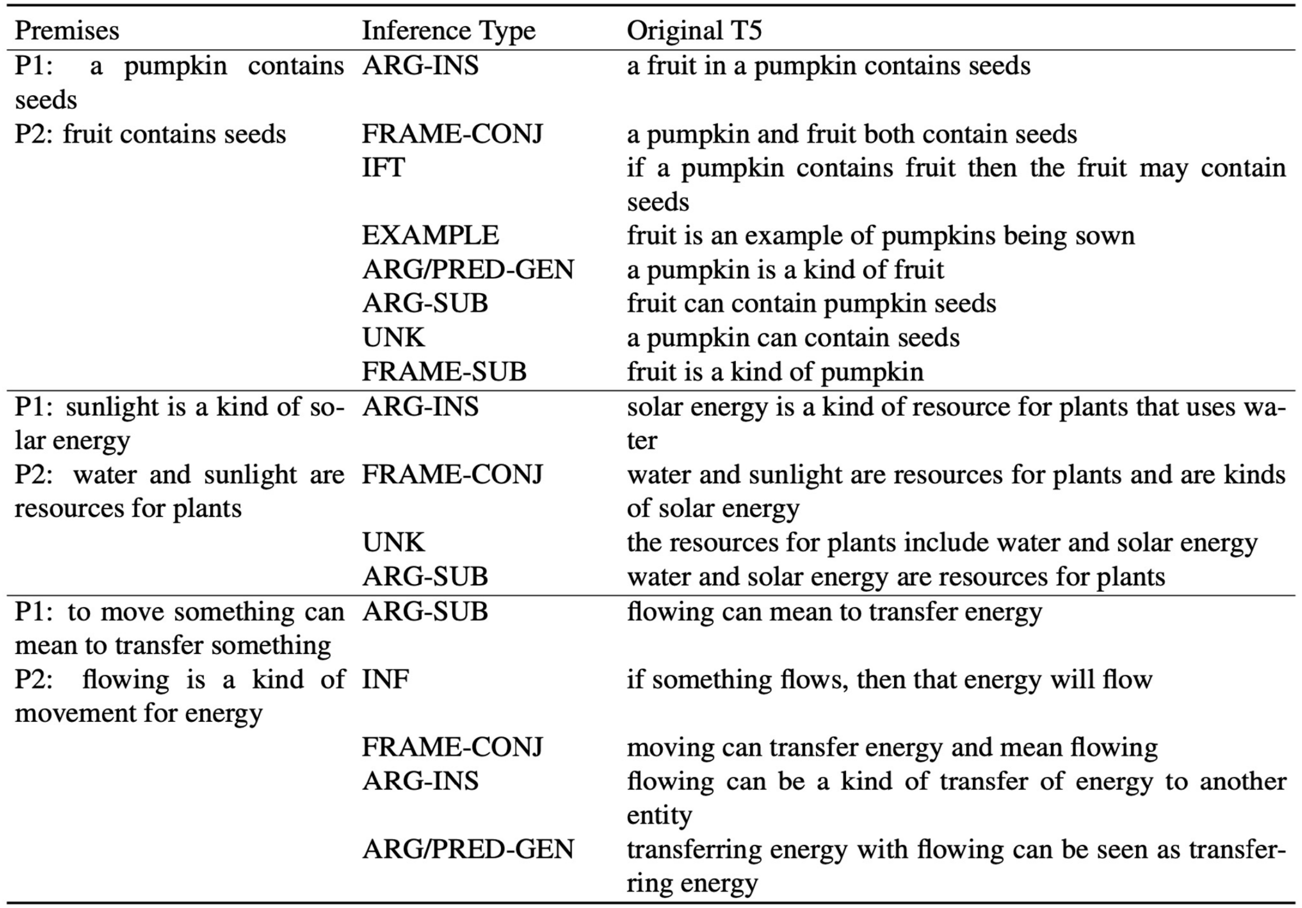
Empirical analysis

Can inf. types control inference behaviour? For encoder input, given premises, changing the [type].



P1: blacktop is made of asphalt concrete
P2: asphalt has a smooth surface

ARG-SUB: the blacktop is made of smooth surface
ARG-SUB-PROP: blacktop has a smooth surface
ARG/PRED-GEN: a blacktop is a kind of asphalt
ARG-INS: asphalt concrete blacktop has a smooth surface
FRAME-CON: asphalt and blacktop have the same surface
IFT: if the asphalt has a smooth surface then the blacktop will have a smooth surface



Premises	Inference Type	Original T5
P1: a pumpkin contains seeds	ARG-INS	a fruit in a pumpkin contains seeds
P2: fruit contains seeds	FRAME-CONJ	a pumpkin and fruit both contain seeds
	IFT	if a pumpkin contains fruit then the fruit may contain seeds
	EXAMPLE	fruit is an example of pumpkins being sown
	ARG/PRED-GEN	a pumpkin is a kind of fruit
	ARG-SUB	fruit can contain pumpkin seeds
	UNK	a pumpkin can contain seeds
	FRAME-SUB	fruit is a kind of pumpkin
P1: sunlight is a kind of solar energy	ARG-INS	solar energy is a kind of resource for plants that uses water
P2: water and sunlight are resources for plants	FRAME-CONJ	water and sunlight are resources for plants and are kinds of solar energy
	UNK	the resources for plants include water and solar energy
	ARG-SUB	water and solar energy are resources for plants
P1: to move something can mean to transfer something	ARG-SUB	flowing can mean to transfer energy
P2: flowing is a kind of movement for energy	INF	if something flows, then that energy will flow
	FRAME-CONJ	moving can transfer energy and mean flowing
	ARG-INS	flowing can be a kind of transfer of energy to another entity
	ARG/PRED-GEN	transferring energy with flowing can be seen as transferring energy

Inf. type can control the generation of conclusion,
indicating the inference behaviour is encoded in the label embedding.

Empirical analysis

Can annotation help model training and inference?

1.The inference type as the prefix for the premises at the Encoder (**E**ncoder **P**refix):

the inference type is [type] </s> p1 </s> p2

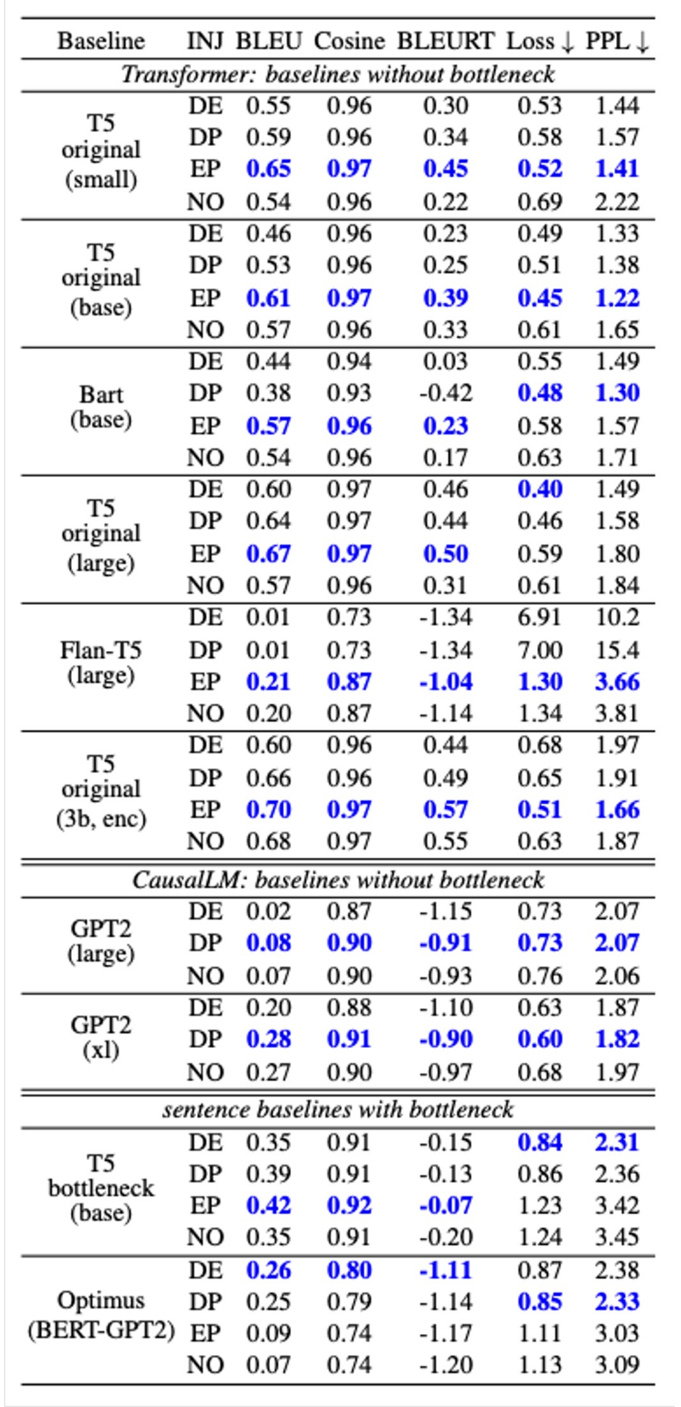
2.The inference type as the prefix for the conclusion in the Decoder (**D**ecoder **P**refix):

</s> the inference type is [type]. con

3.The inference type at the end of the conclusion in the Decoder (**D**ecoder **E**nd):

</s> con. the inference type is [type]

The annotations can support model training.



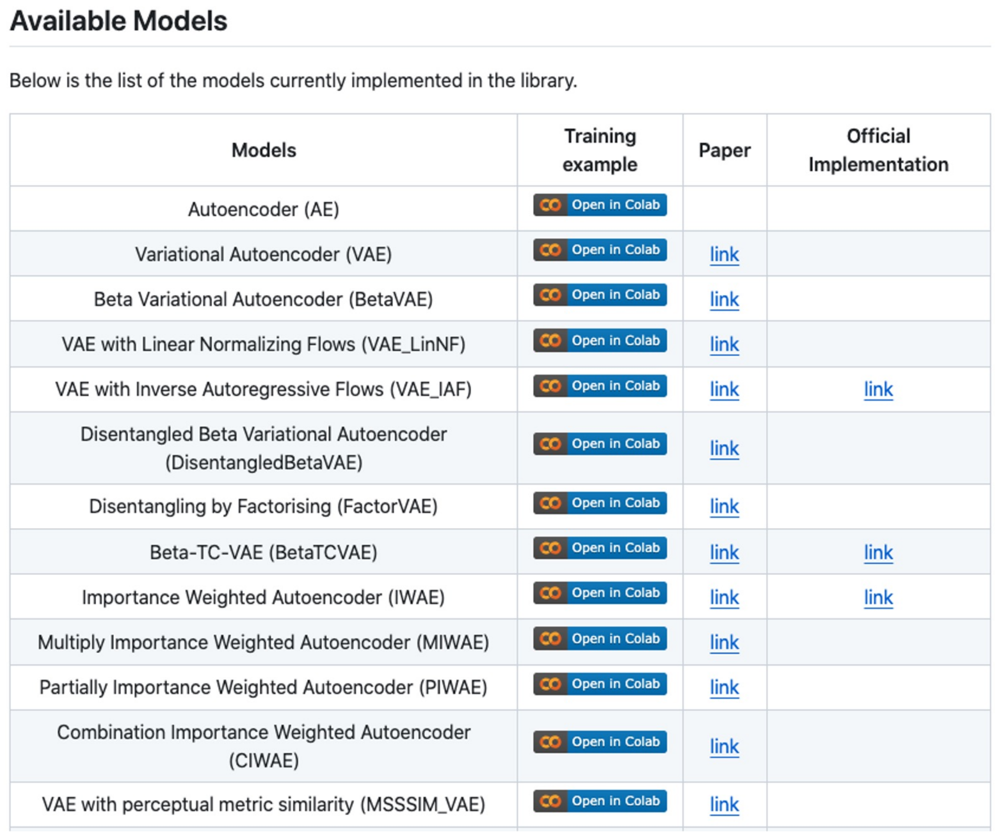
Baseline	INJ	BLEU	Cosine	BLEURT	Loss ↓	PPL ↓
Transformer: baselines without bottleneck						
T5 original (small)	DE	0.55	0.96	0.30	0.53	1.44
	DP	0.59	0.96	0.34	0.58	1.57
	EP	**0.65**	**0.97**	**0.45**	**0.52**	**1.41**
	NO	0.54	0.96	0.22	0.69	2.22
T5 original (base)	DE	0.46	0.96	0.23	0.49	1.33
	DP	0.53	0.96	0.25	0.51	1.38
	EP	**0.61**	**0.97**	**0.39**	**0.45**	**1.22**
	NO	0.57	0.96	0.33	0.61	1.65
Bart (base)	DE	0.44	0.94	0.03	0.55	1.49
	DP	0.38	0.93	-0.42	**0.48**	**1.30**
	EP	**0.57**	**0.96**	**0.23**	0.58	1.57
	NO	0.54	0.96	0.17	0.63	1.71
T5 original (large)	DE	0.60	0.97	0.46	**0.40**	1.49
	DP	0.64	0.97	0.44	0.46	1.58
	EP	**0.67**	**0.97**	**0.50**	0.59	1.80
	NO	0.57	0.96	0.31	0.61	1.84
Flan-T5 (large)	DE	0.01	0.73	-1.34	6.91	10.2
	DP	0.01	0.73	-1.34	7.00	15.4
	EP	**0.21**	**0.87**	**-1.04**	**1.30**	**3.66**
	NO	0.20	0.87	-1.14	1.34	3.81
T5 original (3b, enc)	DE	0.60	0.96	0.44	0.68	1.97
	DP	0.66	0.96	0.49	0.65	1.91
	EP	**0.70**	**0.97**	**0.57**	**0.51**	**1.66**
	NO	0.68	0.97	0.55	0.63	1.87
CausalLM: baselines without bottleneck						
GPT2 (large)	DE	0.02	0.87	-1.15	0.73	2.07
	DP	**0.08**	**0.90**	**-0.91**	**0.73**	**2.07**
	NO	0.07	0.90	-0.93	0.76	2.06
GPT2 (xl)	DE	0.20	0.88	-1.10	0.63	1.87
	DP	**0.28**	**0.91**	**-0.90**	**0.60**	**1.82**
	NO	0.27	0.90	-0.97	0.68	1.97
sentence baselines with bottleneck						
T5 bottleneck (base)	DE	0.35	0.91	-0.15	**0.84**	**2.31**
	DP	0.39	0.91	-0.13	0.86	2.36
	EP	**0.42**	**0.92**	**-0.07**	1.23	3.42
	NO	0.35	0.91	-0.20	1.24	3.45
Optimus (BERT-GPT2)	DE	**0.26**	**0.80**	**-1.11**	0.87	2.38
	DP	0.25	0.79	-1.14	**0.85**	**2.33**
	EP	0.09	0.74	-1.17	1.11	3.03
	NO	0.07	0.74	-1.20	1.13	3.09

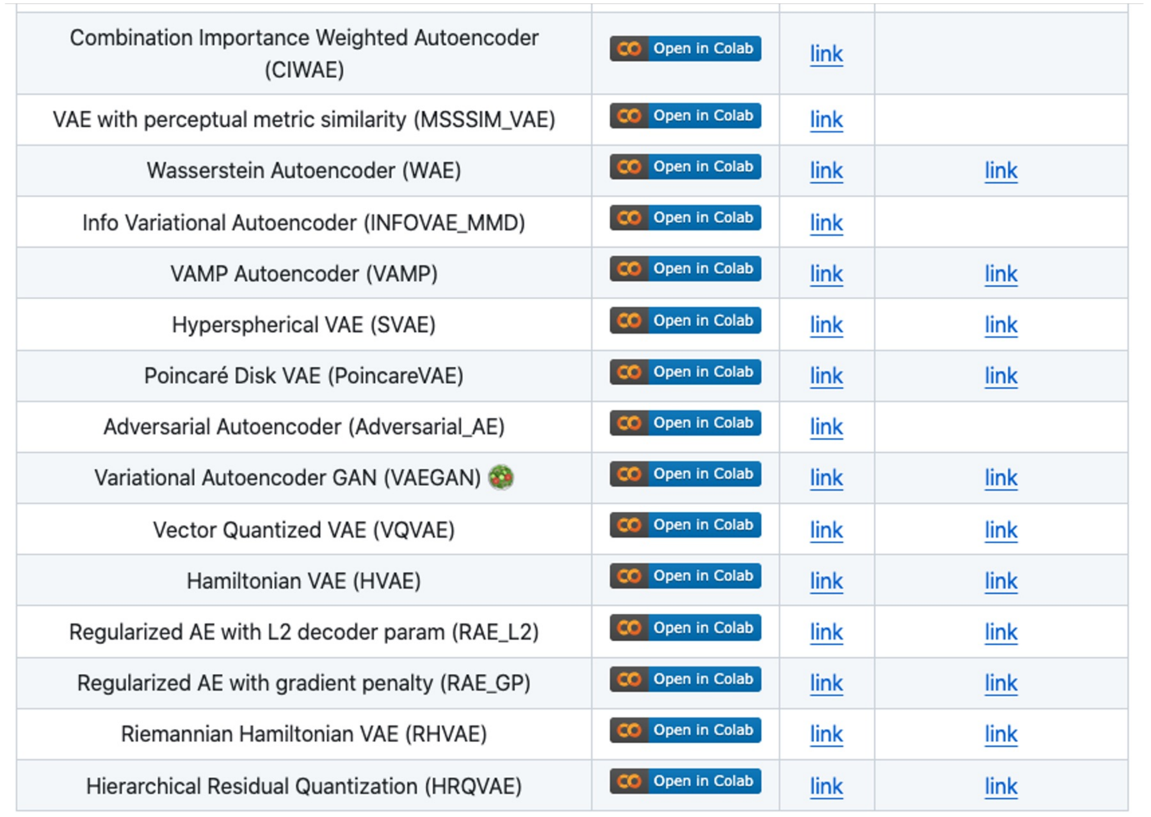
Building & Probing Language VAEs (LangSpace & LangVAE)

Pytorch library

Pythae: https://github.com/clementchadebec/benchmark_VAE

Deep Generative Modelling: https://github.com/jmtomczak/intro_dgm





Available Models

Below is the list of the models currently implemented in the library.

Models	Training example	Paper	Official Implementation
Autoencoder (AE)	Open in Colab		
Variational Autoencoder (VAE)	Open in Colab	link	
Beta Variational Autoencoder (BetaVAE)	Open in Colab	link	
VAE with Linear Normalizing Flows (VAE_LinNF)	Open in Colab	link	
VAE with Inverse Autoregressive Flows (VAE_IAF)	Open in Colab	link	link
Disentangled Beta Variational Autoencoder (DisentangledBetaVAE)	Open in Colab	link	
Disentangling by Factorising (FactorVAE)	Open in Colab	link	
Beta-TC-VAE (BetaTCVAE)	Open in Colab	link	link
Importance Weighted Autoencoder (IWAE)	Open in Colab	link	link
Multiply Importance Weighted Autoencoder (MIWAE)	Open in Colab	link	
Partially Importance Weighted Autoencoder (PIWAE)	Open in Colab	link	
Combination Importance Weighted Autoencoder (CIWAE)	Open in Colab	link	
VAE with perceptual metric similarity (MSSSIM_VAE)	Open in Colab	link	
Combination Importance Weighted Autoencoder (CIWAE)	Open in Colab	link	
VAE with perceptual metric similarity (MSSSIM_VAE)	Open in Colab	link	
Wasserstein Autoencoder (WAE)	Open in Colab	link	link
Info Variational Autoencoder (INFOVAE_MMD)	Open in Colab	link	
VAMP Autoencoder (VAMP)	Open in Colab	link	link
Hyperspherical VAE (SVAE)	Open in Colab	link	link
Poincaré Disk VAE (PoincareVAE)	Open in Colab	link	link
Adversarial Autoencoder (Adversarial_AE)	Open in Colab	link	
Variational Autoencoder GAN (VAEGAN)	Open in Colab	link	link
Vector Quantized VAE (VQVAE)	Open in Colab	link	link
Hamiltonian VAE (HVAE)	Open in Colab	link	link
Regularized AE with L2 decoder param (RAE_L2)	Open in Colab	link	link
Regularized AE with gradient penalty (RAE_GP)	Open in Colab	link	link
Riemannian Hamiltonian VAE (RHVAE)	Open in Colab	link	link
Hierarchical Residual Quantization (HRQVAE)	Open in Colab	link	link

Language VAE: Pytorch library

LangVAE: our demo can easily integrate different pretrained language models into VAE architecture.

Pretrained checkpoints:

https://huggingface.co/neuro-symbolic-ai

Train:

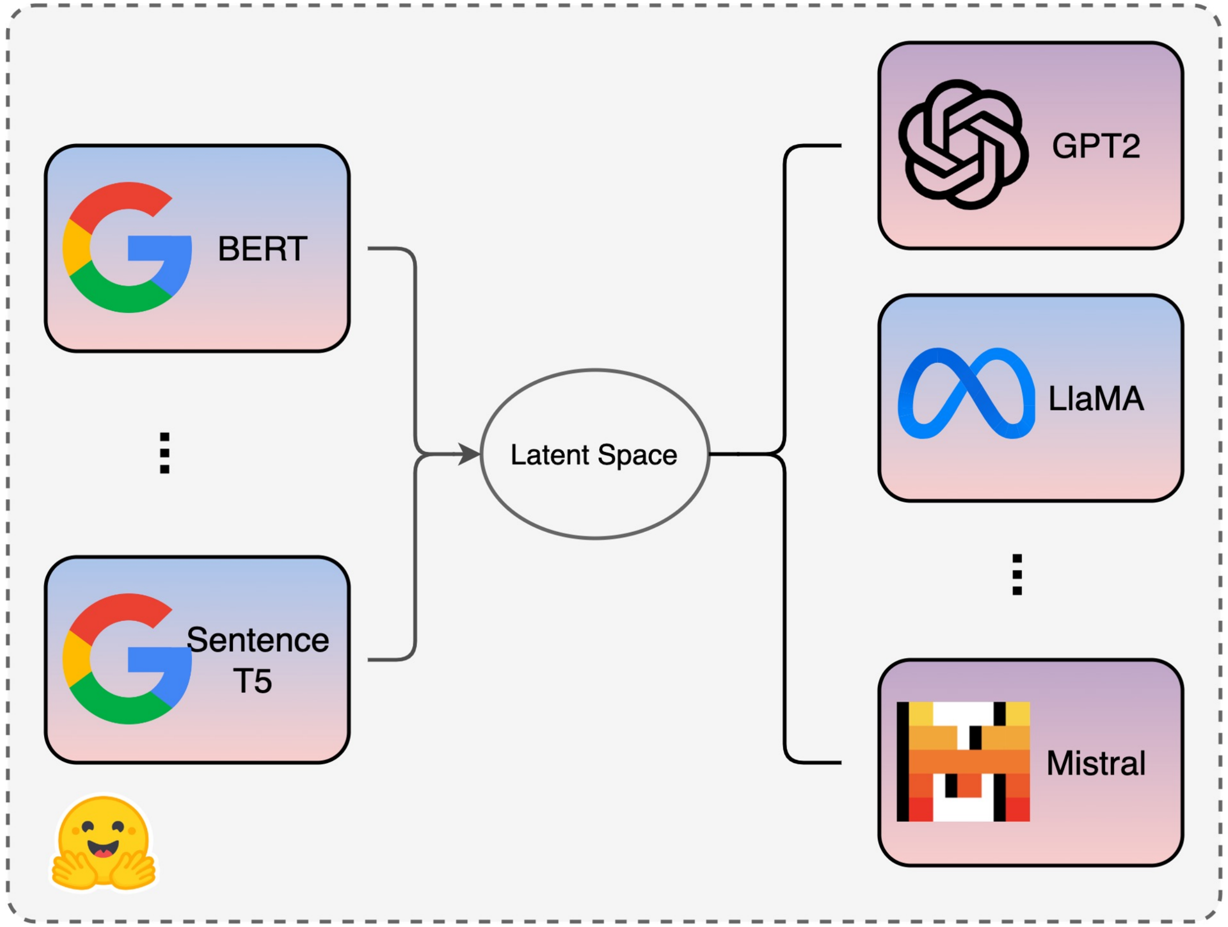
Only support Gaussian prior now.

https://github.com/neuro-symbolic-ai/LangVAE

Evaluation:

https://github.com/neuro-symbolic-ai/LangSpace

1. latent traversal;
2. interpolation;
3. arithmetic;
4. t-sne/UMAP/PCA;
5. disentanglement metrics.



LangVAE: Easy to train Language VAEs

LangVAE is a python library for agile experimentation with language VAEs.

Featuring:

Easy integration of encoder and decoder models available from HuggingFace.

Tokenisation facility for any model combination.

Modular architecture, facilitating customisation.

Easy upload of trained models to HuggingFace.

LangVAE: Easy to train Language VAEs

Basic training script: BERT-GPT2

```
dataset = [sent for sent in EntailmentBankDataSet()
                if (sent.annotations["type"] == "answer" or
                    sent.annotations["type"].startswith("context"))]
eval_size = int(0.1 * len(dataset))

decoder = SentenceDecoder("gpt2", LATENT_SIZE, MAX_SENT_LEN)
encoder = SentenceEncoder("bert-base-cased", LATENT_SIZE, decoder.tokenizer)
train_dataset = TokenizedDataSet(dataset[:-eval_size], decoder.tokenizer, decoder.max_len)
eval_dataset = TokenizedDataSet(dataset[-eval_size:], decoder.tokenizer, decoder.max_len)

model_config = VAEConfig(...)
model = LangVAE(model_config, encoder, decoder)

training_config = CyclicalScheduleKLThresholdTrainerConfig(...)
pipeline = LanguageTrainingPipeline(training_config=training_config, model=model)
pipeline(train_data=train_dataset, eval_data=eval_dataset)
```

LangVAE: Easy to train Language VAEs

SentenceDecoder: Encapsulates decoder model and latent injection strategies (memory, embeddings).

Defines the tokenizer model for inputs

```
decoder = SentenceDecoder(model_path, latent_size, max_sent_len)
```

- model_path: the name/path of the HuggingFace model to be used. It will be automatically loaded using the transformers library (e.g., “gpt2”).
- latent_size: dimension of the VAE latent space (e.g., 64).
- max_sent_len: maximum sentence length in tokens.

LangVAE: Easy to train Language VAEs

SentenceEncoder: Encapsulates encoder model and converts input tokens from the decoder tokenizer, so only the decoder tokens are needed.

```
encoder = SentenceEncoder(model_path, latent_size, decoder.tokenizer)
```

- model_path: same as SentenceDecoder, but with an encoder model (e.g., “bert-base-cased”).
- latent_size: same as SentenceDecoder
- decoder.tokenizer: tokenizer model from a SentenceDecoder instance.

LangVAE: Easy to train Language VAEs

TokenizedDatasets: tokenizes and batches input sentences, using an interface derived from pytorch datasets.

Accepts two formats:

- Simple list of strings.
- Instance of *SentenceDataset* from the saf_datasets library.

Provides one-hot encoded sentence tensors L×V, where L is the sentence length and V is the decoder vocabulary size.

```
from saf_datasets import WordNetFilteredDataSet

dataset = WordNetFilteredDataSet()
decoder = SentenceDecoder("gpt2", 32, 64)
tok_dataset = TokenizedDataSet(dataset, decoder.tokenizer, decoder.max_len)
```

LangVAE: Easy to train Language VAEs

Configuration and pipeline setup

```
model_config = VAEConfig(
   input_dim=(dataset[0]["data"].shape[-2], dataset[0]["data"].shape[-1]),
   latent_dim=32
)

model = LangVAE(model_config, encoder, decoder)
```

LangVAE: Easy to train Language VAEs

Configuration and pipeline setup

```
training_config = CyclicalScheduleKLThresholdTrainerConfig(
    output_dir='def_expl_vae',
    num_epochs=5,
    learning_rate=1e-4,
    per_device_train_batch_size=50,
    per_device_eval_batch_size=50,
    steps_saving=1,
    optimizer_cls="AdamW",
    scheduler_cls="ReduceLROnPlateau",
    scheduler_params={"patience": 5, "factor": 0.5},
    max_beta=1.0,
    n_cycles=40,
    target_kl=2.0
)

pipeline = LanguageTrainingPipeline(training_config=training_config, model=model)
```

LangVAE: Easy to train Language VAEs

Starting the training process

```
pipeline(
   train_data=train_dataset,
   eval_data=eval_dataset
)
```

LangVAE: Easy to train Language VAEs

Examples:

https://colab.research.google.com/drive/1CCFvPWsQU2VX41guHGT2-uFgHogAejDv

Code:

https://github.com/neuro-symbolic-ai/LangVAE

LangSpace: Easy to probe Language VAEs

LangSpace is a python library for quick testing and probing of language VAEs.

It features:

- A collection of probing methods, adapted for language VAE models.
- A modular architecture, for implementation of flexible and reusable probes.
- Extensible reporting methods.

LangSpace: Easy to probe Language VAEs

Loading models

```
from langvae import LangVAE

model = LangVAE.load_from_hf_hub(models.OPTIMUS_ENTAILMENTBANK, allow_pickle=True)
```

LangSpace: Easy to probe Language VAEs

Loading datasets

```
from saf_datasets import EntailmentBankDataSet

eb_dataset = [sent for sent in EntailmentBankDataSet.from_resource("pos+lemma+ctag+dep+srl#noproof")
                    if (sent.annotations["type"] == "answer" or sent.annotations["type"].startswith("context"))]
```

LangSpace: Easy to probe Language VAEs

Quantitative probes: Interpolation

```
from langspace.probe import InterpolationProbe
from langspace.metrics.interpolation import InterpolationMetric as InterpMetric

eval_metrics = [InterpMetric.QUALITY, InterpMetric.SMOOTHNESS]
interp_report = InterpolationProbe(model, dataset, eval=eval_metrics).report()
print(interp_report)
interp_report.to_csv("interpolation.csv")
```

LangSpace: Easy to probe Language VAEs

Quantitative probes: Interpolation

source	**target**	**distance**	**generate**
humans require freshwater for survival	animals require food to survive	1.000	humans require water for survival … animals require food for survival … animals require food to survive
the sun is in the northern hemisphere	food is a source of energy for animals / plants	0.380	the sun is in in solar hemisphere … the sun is a source energy for called plants … food is a source of energy for animals / plants

LangSpace: Easy to probe Language VAEs

Quantitative probes: Disentanglement metrics

```
from langspace.probe import DisentanglementProbe

gen_factors = {
   "direction": ["ARGM-DIR"],
   "cause": ["ARGM-CAU"],
   "purpose": ["ARGM-PRP","ARGM-PNC", "ARGM-GOL"],
   "more": ["ARGM-EXT"],
   "location": ["ARGM-LOC"],
   …
}

disentang_probe = DisentanglementProbe(model, dataset, sample_size=1000,
metrics=["z-diff", "z-min-var", "Disentanglement", "Modularity"], gen_factors=gen_factors)
disentang_report = disentang_probe.report()
print(interp_report)
interp_report.to_csv(”disentanglement.csv")
```

LangSpace: Easy to probe Language VAEs

Quantitative probes: Disentanglement metrics

z-diff	**z-min-var**	**MIG**	**Completeness**	**Informativeness**
0.05 (±0.00)	0.25 (±0.00)	0.02 (±0.02)	1.00 (±0.00)	0.58 (±0.29)

LangSpace: Easy to probe Language VAEs

Qualitative probes: Traversal

```
from langspace.probe import TraversalProbe

trav_report = TraversalProbe(model, dataset, sample_size=10, dims=list(range(32))).report()
print(trav_report)
trav_report.to_csv("traversal.csv")
```

LangSpace: Easy to probe Language VAEs

Qualitative probes: Traversal

seeds	**dim**	**distance**	**generate**
Earth revolves around the sun.	0	0.079735	light revolves around the sun.
Earth revolves around the sun.	0	0.249271	light revolves around the sun.
Earth revolves around the sun.	0	0.457066	light revolves around the sun.
…	…	…	…
leo is a kind of constellation	31	1.574725	leo is a kind of constellation
leo is a kind of constellation	31	3.739711	smo is a kind of constellation
leo is a kind of constellation	31	3.886802	chloro is a kind of cell

LangSpace: Easy to probe Language VAEs

Qualitative probes: Vector arithmetic

```python
from langspace.probe import ArithmeticProbe
from langspace.ops.arithmetic import ArithmeticOps

arith_report = ArithmeticProbe(model, dataset, ops=list(ArithmeticOps)).report()
print(arith_report)
arith_report.to_csv("arithm.csv")
```

LangSpace: Easy to probe Language VAEs

Qualitative probes: Vector arithmetic

source	**target**	**op**	**generate**
animals require food for survival	animals require warmth for survival	sum	animals require food for survival
water vapor is invisible	the water is warm	sum	the water is invisible
animals require food for survival	animals require warmth for survival	sub	cal 5 chain carbohydrate makes a kind of food
water vapor is invisible	the water is warm	sub	igneous is formed chemically in crystallizing
animals require food for survival	animals require warmth for survival	avg	animals require food for survival
water vapor is invisible	the water is warm	avg	the water is invisible

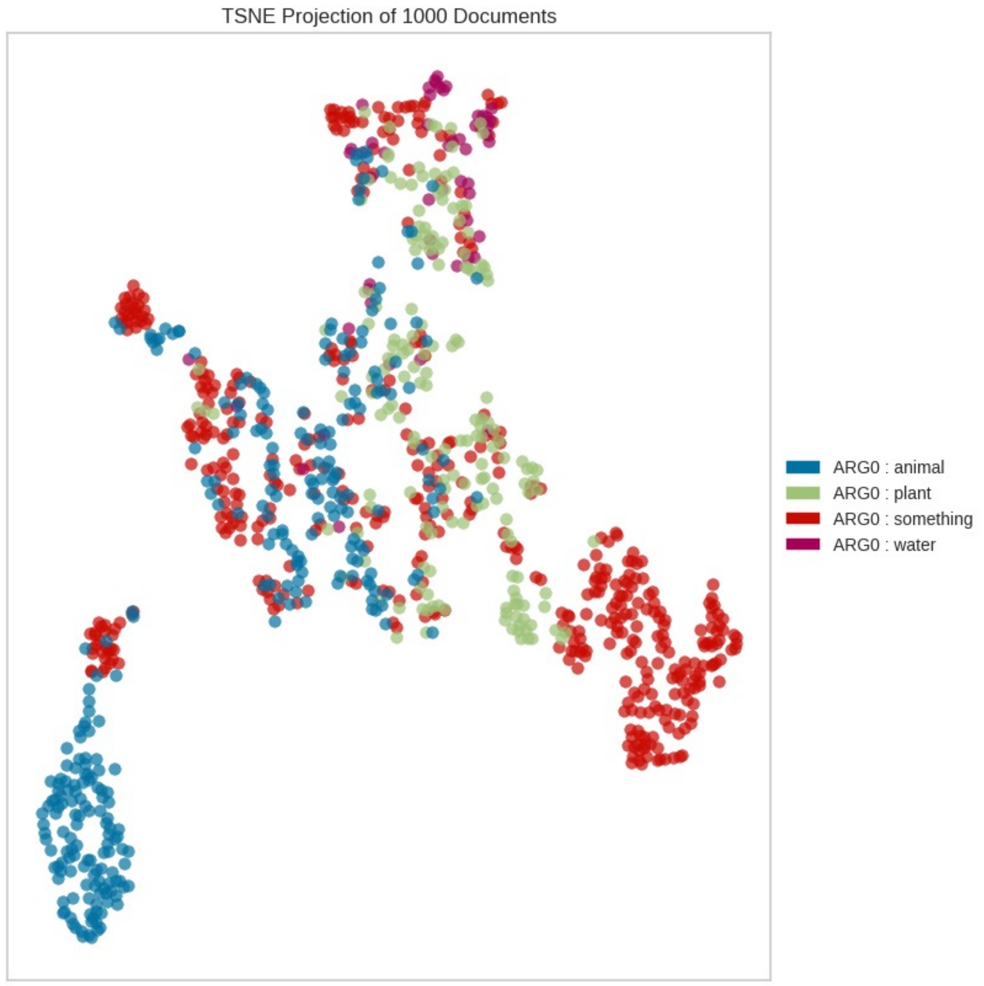
LangSpace: Easy to probe Language VAEs

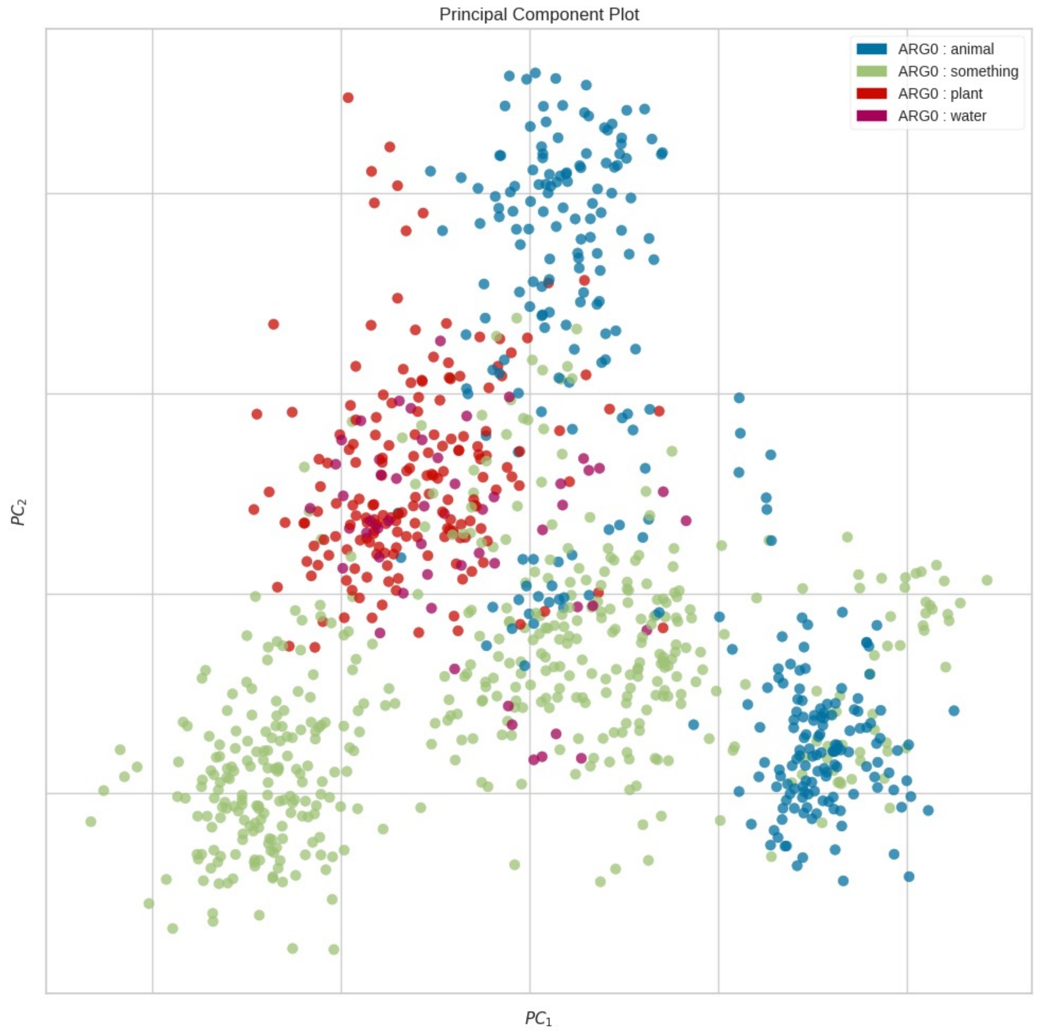
Qualitative probes: Cluster visualisation

```
viz_list = [[" ".join([tok.surface for tok in sent.tokens]),
         " ".join([tok.annotations["srl_0"] for tok in sent.tokens])]
        for sent in eb_dataset]

target_role = ['ARG0 : animal', 'ARG0 : water', 'ARG0 : plant', 'ARG0 : something']
target_viz_list = ClusterVisualizationProbe.role_content_viz(viz_list, target_role, sample_size=1000, TopK=5)
cluster_viz_report = ClusterVisualizationProbe(model, target_viz_list, sample_size=sample_size,
methods=[CvM.TSNE]).report()
```

LangSpace: Easy to probe Language VAEs

Qualitative probes: Cluster visualisation





LangSpace: Easy to probe Language VAEs

Examples:

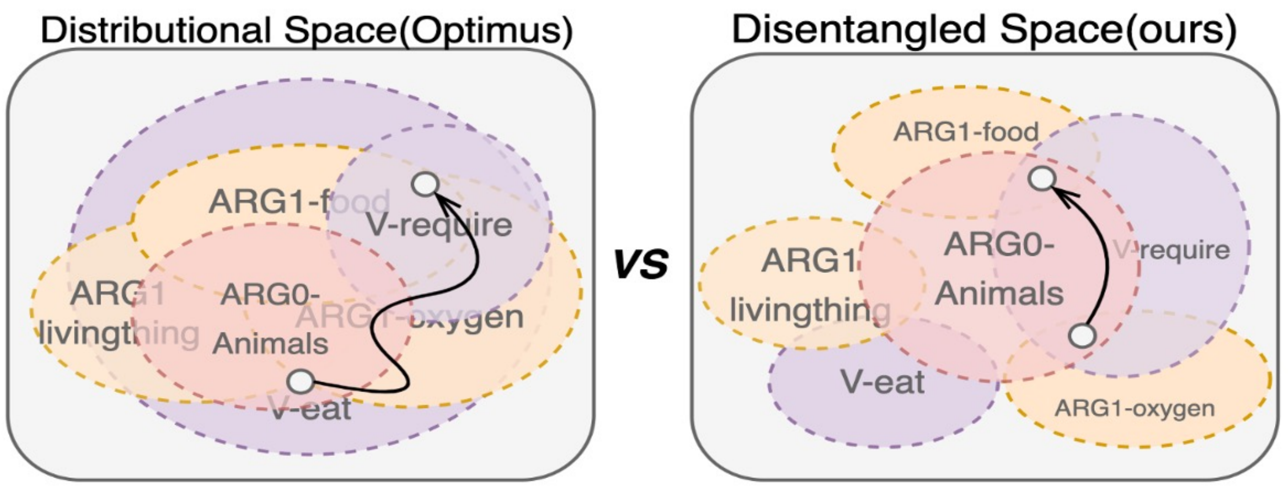
https://colab.research.google.com/drive/18Jath7q3_hn2uWyait9p3hOperphSo4S

Code: https://github.com/neuro-symbolic-ai/LangSpace

Improving separability

Zhang, Y., Carvalho, D. S., Freitas, A. **Learning disentangled semantic spaces of explanations via invertible neural networks**. *ACL 2024*.

Instead: ***General*** semantic control and improve the ***localisation*** of latent sentence spaces, **which can shorten the gap between deep latent semantics and formal linguistic representations**.



localised/formal semantic control (Optimus)

Interpolation path:
animals require oxygen for survival
1. animals require oxygen to survival
2. producer lives in an environment
3. human needs water and oxygen
...
9. animals eat food for survival
animals require food for survival

localised/formal semantic control (Ours)

Interpolation path:
animals require oxygen for survival
1. animals require oxygen to survival
2. animals require water
3. animals require water and oxygen
...
9. animals require food for survival
animals require food for survival

our objective: Granular semantic sentence control and manipulation

Contributions:

1. New notions on sentence semantic disentanglement from the perspective of "*argument structure theory (AST)*".

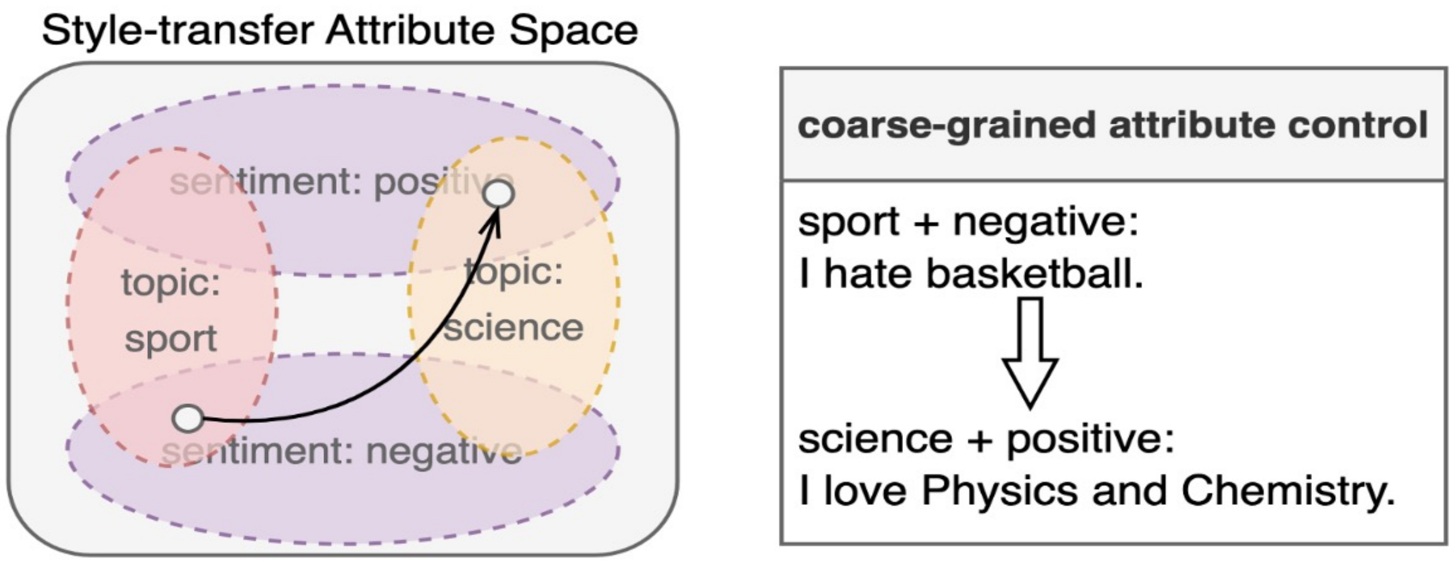
2. Flow-based INN into AutoEncoder to control sentence generation.

3. Supervised approach to flow-based INN to learn a higher separation and disentanglement of semantic features.

4. Geometrical data augmentation.

Zhang, Y., Carvalho, D. S., Freitas, A. **Learning disentangled semantic spaces of explanations via invertible neural networks**. *ACL 2024*.

Overview: Most previous work have concentrated on disentangling *"task-specific"* generative factors, such as sentiment, within the context of style transfer.



sport + negative:
I hate basketball.

science + positive:
I love Physics and Chemistry.

their objective: sentence control for sentiment/topic transfer *(Liu et al., 2023)*

Instead: ***general*** semantic control and improve the ***localisation*** of latent sentence spaces, **which can shorten the gap between deep latent semantics and formal linguistic representations**.

Methodology

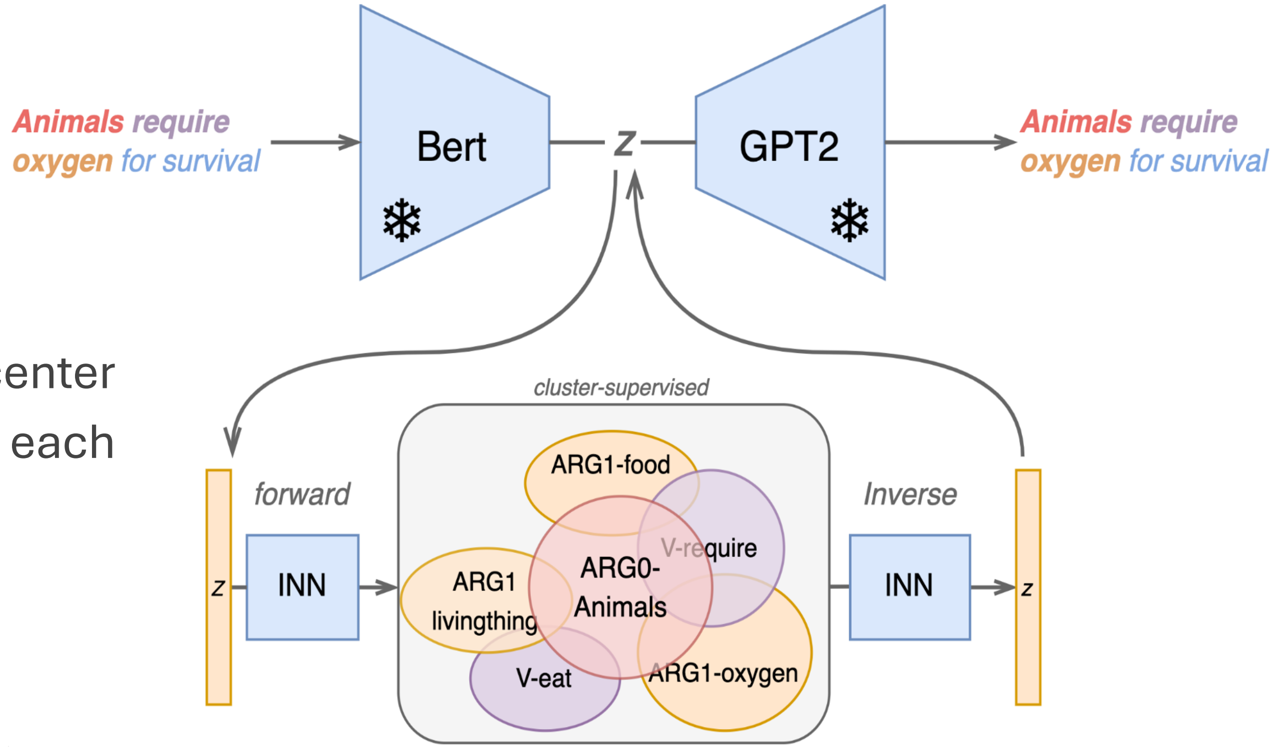
Overview: We first encode each sentence with pretrained AutoEncoder. Then, train the flow-based INN to learn a latent space with better semantic disentanglement (i.e., role-content separation).

Unsupervised: Maximize the exact log-likelihood:

$$\mathcal{L}_{\text{unsup}} = -\mathbb{E}_{x\sim p(x)}\Big[T(E(x))\Big]^2 + \log\left|T'(E(x))\right|$$

Supervised: for each role-content cluster, given the center embedding and a variance < 1, the points around each center will be more densely distributed.

$$\mathcal{L}_{\text{sup}} = -\,\mathbb{E}_{x\sim p_{cluster}(x)}\frac{\Big[T(E(x)) - \mu_{cluster}\Big]^2}{1-\sigma^2} + \log\Big|T'(E(x))\Big|$$



Methodology

Data augmentation: Usinng the arithmetic and traversal operators to support data augmentation for each role-content cluster, described as follows:

(1) given two sentence embeddings with same role-content, calculate their average:

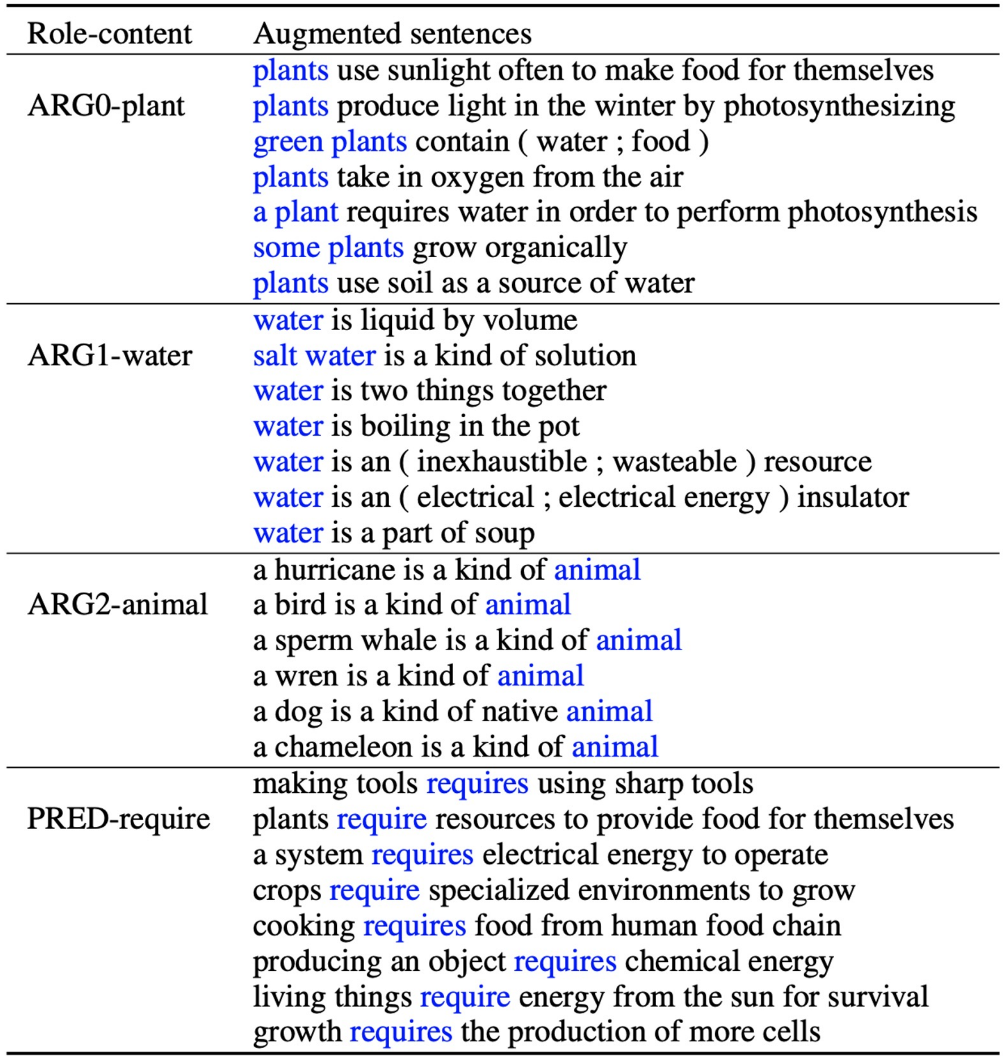
$$(1) \quad \mathrm{v} = average(E'(x_i), E'(x_j))$$

(2) re-sample each dimension of resulting vector (traversing its neighbours).

$$(2) \quad \mathrm{v}_{neighbour} = \mathrm{v}[i] \sim N(0,1)_{\forall i \in \{0,..,size(\mathrm{v})\}}$$

(3) decode it and keep the sentence holding the same role-content.

$$(3) \quad x_{new} = D'(\mathrm{v}_{neighbour})$$



Role-content	Augmented sentences
ARG0-plant	plants use sunlight often to make food for themselves
	plants produce light in the winter by photosynthesizing
	green plants contain (water ; food)
	plants take in oxygen from the air
	a plant requires water in order to perform photosynthesis
	some plants grow organically
	plants use soil as a source of water
ARG1-water	water is liquid by volume
	salt water is a kind of solution
	water is two things together
	water is boiling in the pot
	water is an (inexhaustible ; wasteable) resource
	water is an (electrical ; electrical energy) insulator
	water is a part of soup
ARG2-animal	a hurricane is a kind of animal
	a bird is a kind of animal
	a sperm whale is a kind of animal
	a wren is a kind of animal
	a dog is a kind of native animal
	a chameleon is a kind of animal
PRED-require	making tools requires using sharp tools
	plants require resources to provide food for themselves
	a system requires electrical energy to operate
	crops require specialized environments to grow
	cooking requires food from human food chain
	producing an object requires chemical energy
	living things require energy from the sun for survival
	growth requires the production of more cells

Empirical analysis

Visualisation: evaluating semantic separability via t-SNE and PCA visualisers.

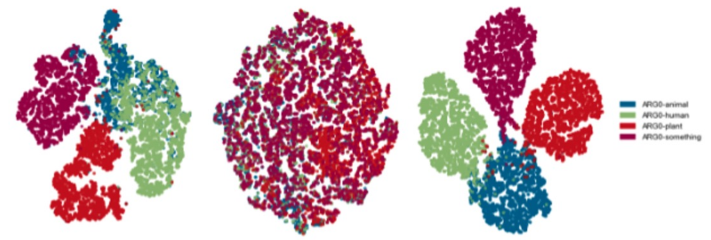


Figure 3: ARG0: t-SNE plot, different colour represents different content regions (blue: animal, green: human, red: plant, purple: something) (left: Optimus, middle:

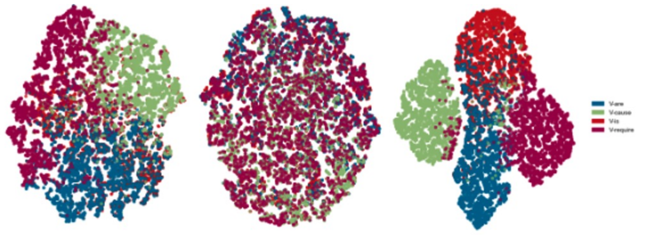


Figure 4: PRED: t-SNE plot (blue: are, green: cause, red: is, purple: require). PCA plot is in Figure 13.

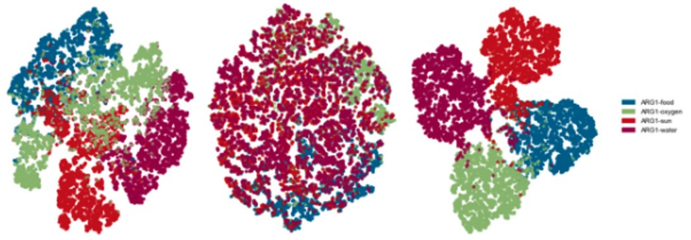


Figure 10: ARG1: t-SNE plot (blue: *food*, green: *oxygen*, red: *sun*, purple: *water*). Supervision (right) induces separability comparable with ARG0. PCA plot is provided in Figure 12.

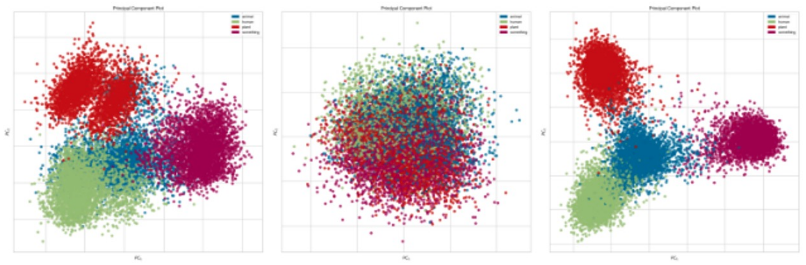


Figure 11: PCA visualization for *ARG0*.

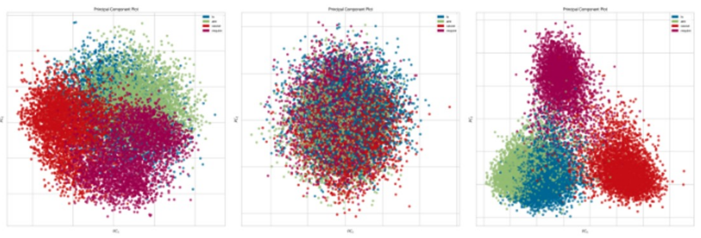


Figure 13: PCA visualization for *PRED*.

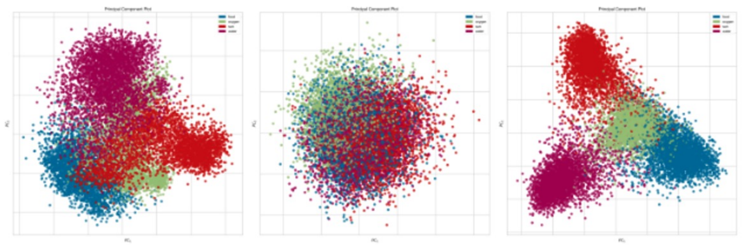


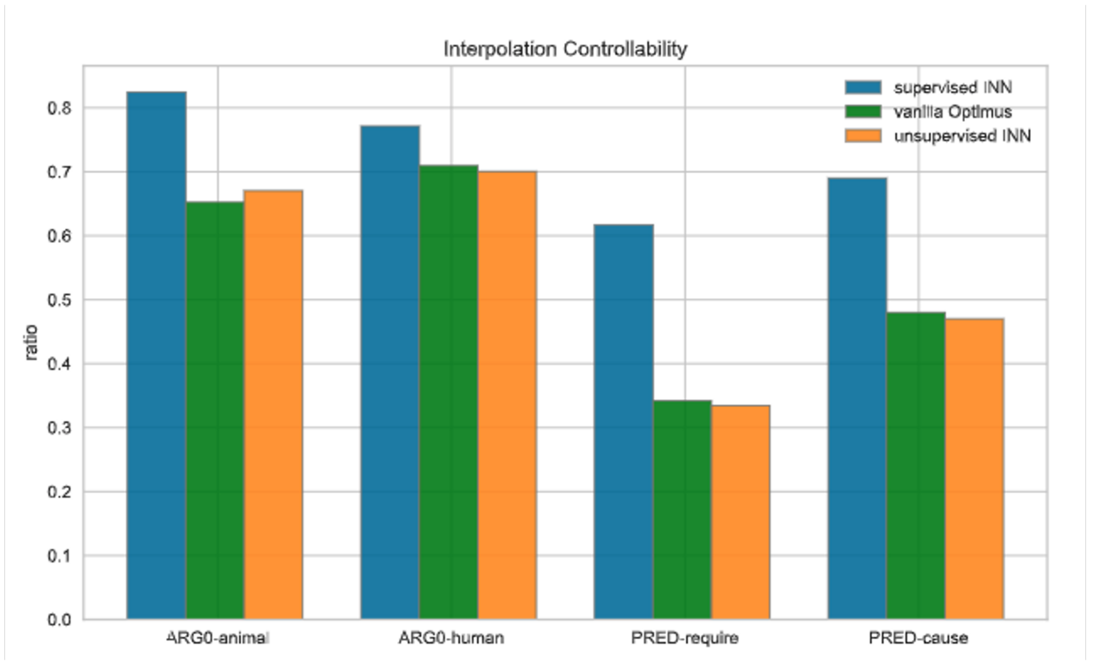
Figure 12: PCA visualization for *ARG1*.

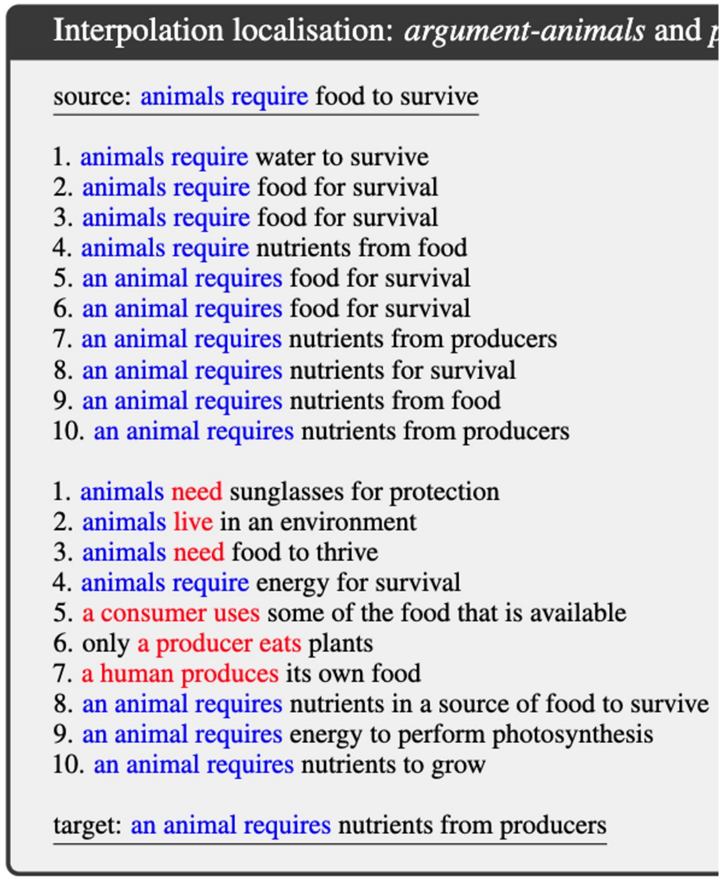
(Optimus) **(Unsupervised)** **(Supervised)**

Supervised (right) leads to better semantic separation than Optimus(left) and un-supervision (middle).

Empirical analysis

Interpolation localisation: Evaluate the disentanglement via linear interpolation. Given two sentences with same semantic feature, a disentangled space should hold the same feature during interpolation.



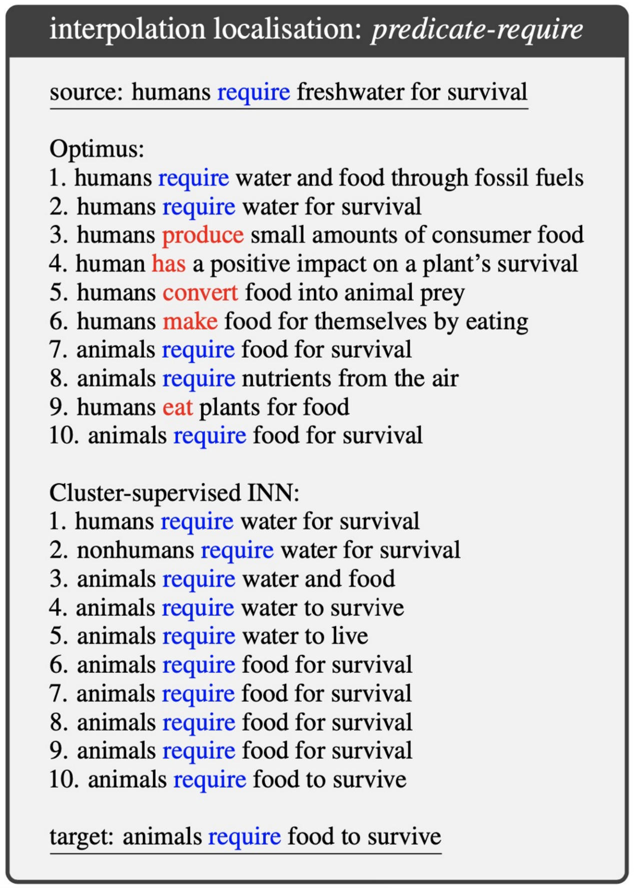

Interpolation localisation: *argument-animals* and *p*

source: animals require food to survive

1. animals require water to survive
2. animals require food for survival
3. animals require food for survival
4. animals require nutrients from food
5. an animal requires food for survival
6. an animal requires food for survival
7. an animal requires nutrients from producers
8. an animal requires nutrients for survival
9. an animal requires nutrients from food
10. an animal requires nutrients from producers

1. animals need sunglasses for protection
2. animals live in an environment
3. animals need food to thrive
4. animals require energy for survival
5. a consumer uses some of the food that is available
6. only a producer eats plants
7. a human produces its own food
8. an animal requires nutrients in a source of food to survive
9. an animal requires energy to perform photosynthesis
10. an animal requires nutrients to grow

target: an animal requires nutrients from producers


interpolation localisation: *predicate-require*

source: humans require freshwater for survival

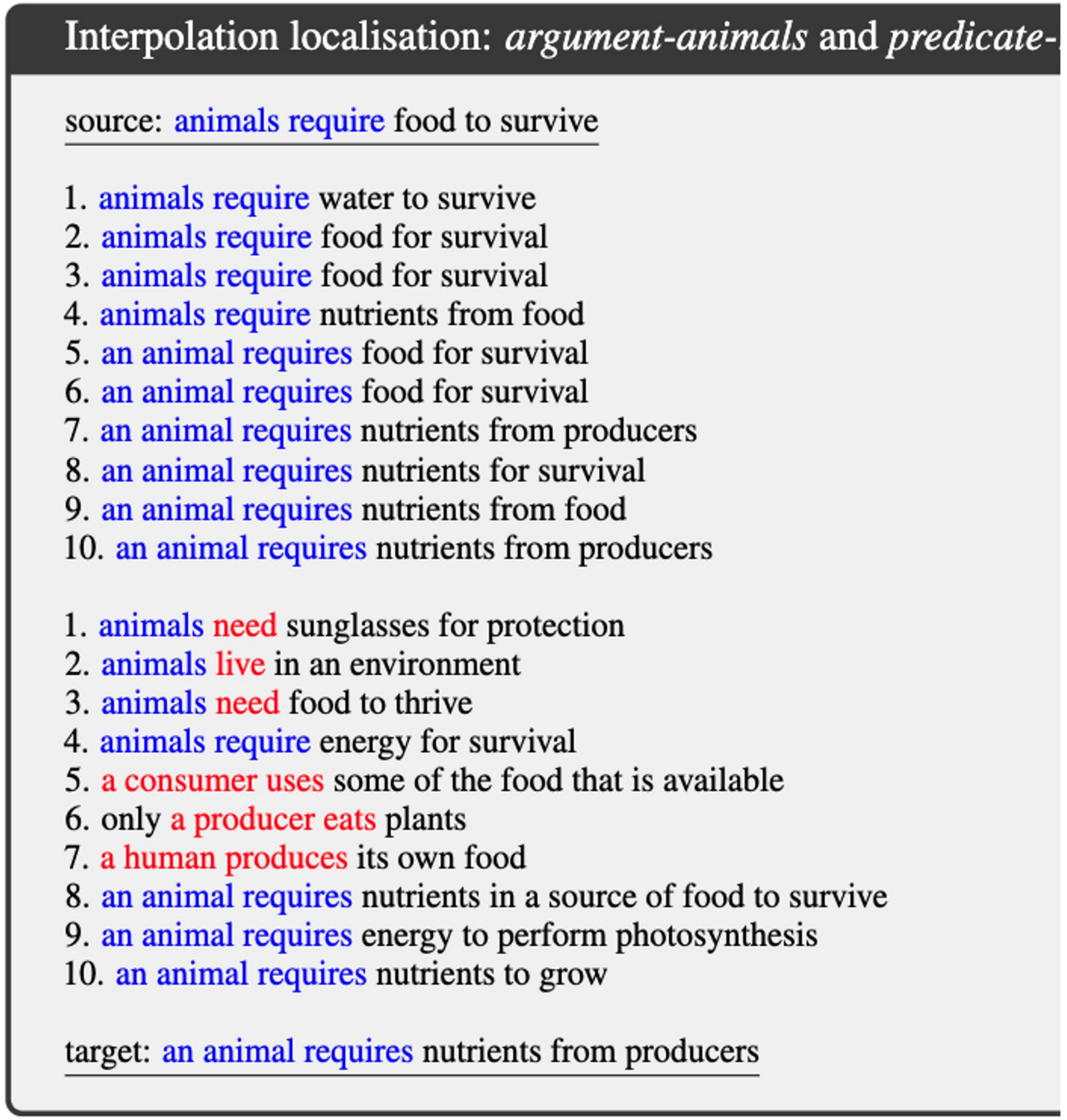
Optimus:
1. humans require water and food through fossil fuels
2. humans require water for survival
3. humans produce small amounts of consumer food
4. human has a positive impact on a plant's survival
5. humans convert food into animal prey
6. humans make food for themselves by eating
7. animals require food for survival
8. animals require nutrients from the air
9. humans eat plants for food
10. animals require food for survival

Cluster-supervised INN:
1. humans require water for survival
2. nonhumans require water for survival
3. animals require water and food
4. animals require water to survive
5. animals require water to live
6. animals require food for survival
7. animals require food for survival
8. animals require food for survival
9. animals require food for survival
10. animals require food to survive

target: animals require food to survive

Observation: Supervised INN outperforms both in quantitative and qualitative evaluations.

Empirical analysis



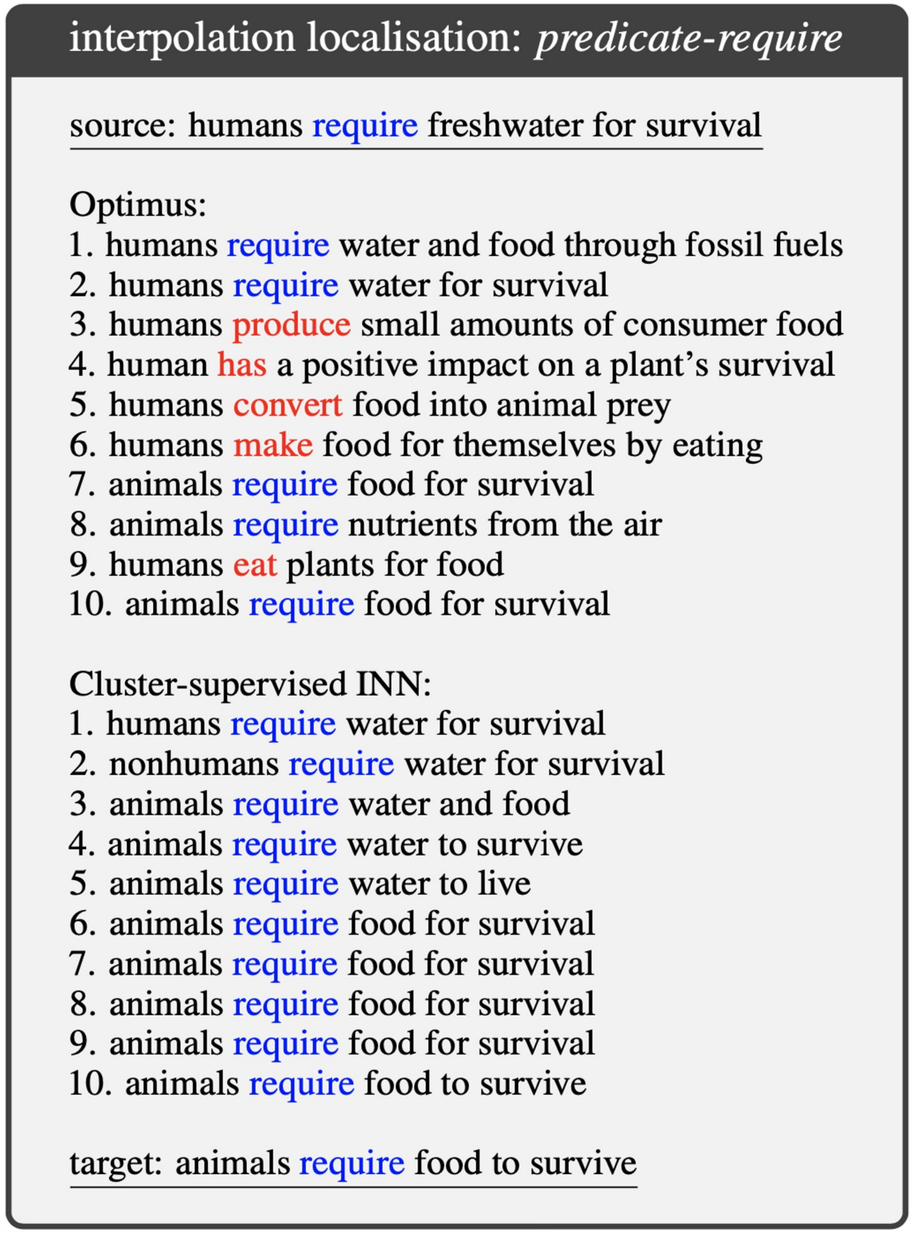
Interpolation localisation: *argument-animals* and *predicate-*

source: animals require food to survive

1. animals require water to survive
2. animals require food for survival
3. animals require food for survival
4. animals require nutrients from food
5. an animal requires food for survival
6. an animal requires food for survival
7. an animal requires nutrients from producers
8. an animal requires nutrients for survival
9. an animal requires nutrients from food
10. an animal requires nutrients from producers

1. animals need sunglasses for protection
2. animals live in an environment
3. animals need food to thrive
4. animals require energy for survival
5. a consumer uses some of the food that is available
6. only a producer eats plants
7. a human produces its own food
8. an animal requires nutrients in a source of food to survive
9. an animal requires energy to perform photosynthesis
10. an animal requires nutrients to grow

target: an animal requires nutrients from producers



interpolation localisation: *predicate-require*

source: humans require freshwater for survival

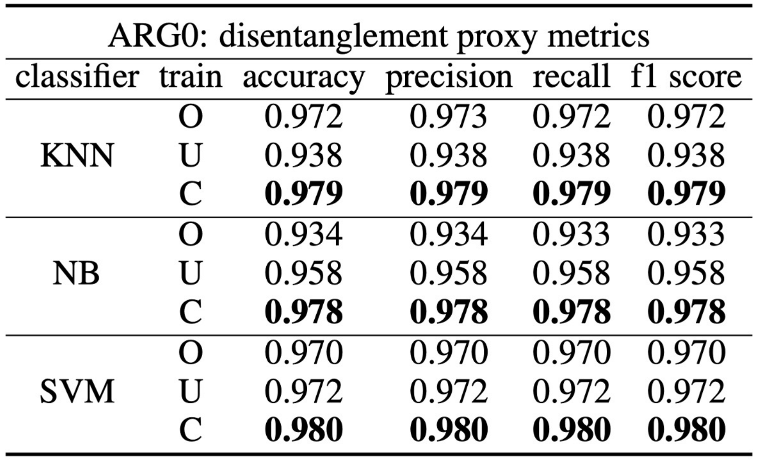
Optimus:
1. humans require water and food through fossil fuels
2. humans require water for survival
3. humans produce small amounts of consumer food
4. human has a positive impact on a plant’s survival
5. humans convert food into animal prey
6. humans make food for themselves by eating
7. animals require food for survival
8. animals require nutrients from the air
9. humans eat plants for food
10. animals require food for survival

Cluster-supervised INN:
1. humans require water for survival
2. nonhumans require water for survival
3. animals require water and food
4. animals require water to survive
5. animals require water to live
6. animals require food for survival
7. animals require food for survival
8. animals require food for survival
9. animals require food for survival
10. animals require food to survive

target: animals require food to survive

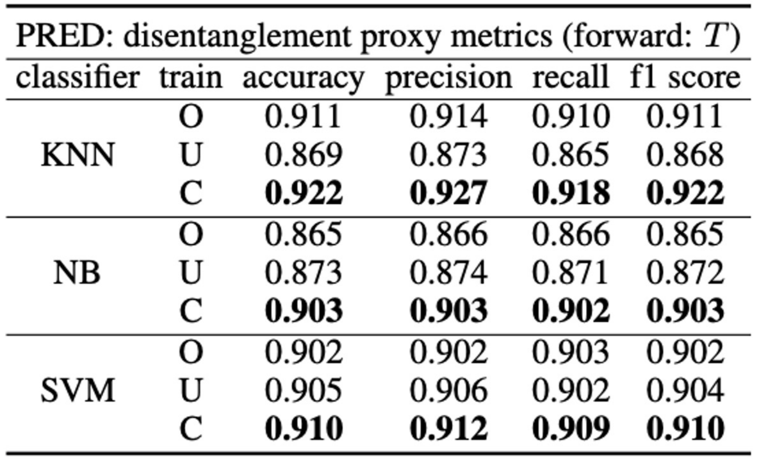
Empirical analysis

Downstream classifiers: evaluate the role-content separation via non-parametric classifier: K-neighbours (KNN) and parametric classifiers: Naive Bayes (NB) and Support Vector Machine (SVM).



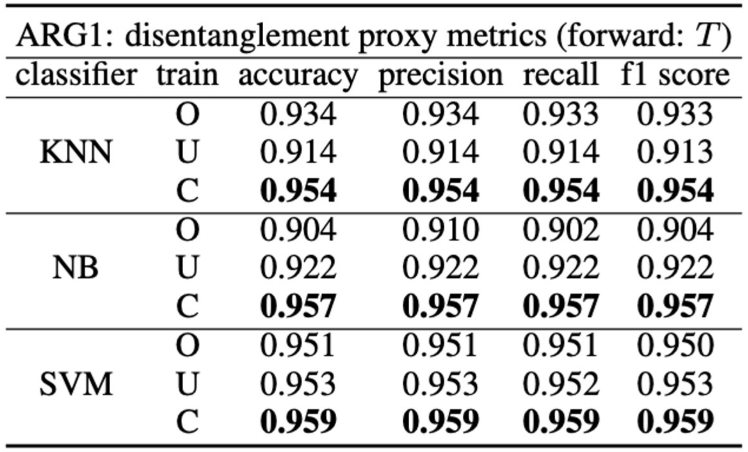
ARG0: disentanglement proxy metrics

classifier	train	accuracy	precision	recall	f1 score
KNN	O	0.972	0.973	0.972	0.972
	U	0.938	0.938	0.938	0.938
	C	**0.979**	**0.979**	**0.979**	**0.979**
NB	O	0.934	0.934	0.933	0.933
	U	0.958	0.958	0.958	0.958
	C	**0.978**	**0.978**	**0.978**	**0.978**
SVM	O	0.970	0.970	0.970	0.970
	U	0.972	0.972	0.972	0.972
	C	**0.980**	**0.980**	**0.980**	**0.980**



PRED: disentanglement proxy metrics (forward: T)

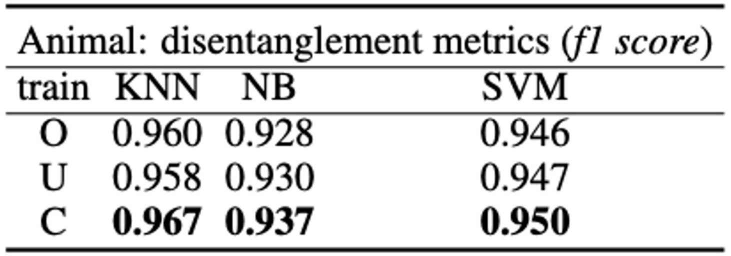
classifier	train	accuracy	precision	recall	f1 score
KNN	O	0.911	0.914	0.910	0.911
	U	0.869	0.873	0.865	0.868
	C	**0.922**	**0.927**	**0.918**	**0.922**
NB	O	0.865	0.866	0.866	0.865
	U	0.873	0.874	0.871	0.872
	C	**0.903**	**0.903**	**0.902**	**0.903**
SVM	O	0.902	0.902	0.903	0.902
	U	0.905	0.906	0.902	0.904
	C	**0.910**	**0.912**	**0.909**	**0.910**



ARG1: disentanglement proxy metrics (forward: T)

classifier	train	accuracy	precision	recall	f1 score
KNN	O	0.934	0.934	0.933	0.933
	U	0.914	0.914	0.914	0.913
	C	**0.954**	**0.954**	**0.954**	**0.954**
NB	O	0.904	0.910	0.902	0.904
	U	0.922	0.922	0.922	0.922
	C	**0.957**	**0.957**	**0.957**	**0.957**
SVM	O	0.951	0.951	0.951	0.950
	U	0.953	0.953	0.952	0.953
	C	**0.959**	**0.959**	**0.959**	**0.959**

Observation:
(1) supervised (C) outperforms both unsupervised(U) and Optimus(O).
(2) (U) outperforms (O) in NB and SVM (encoder + flow can improve the representation capabilities of approximated posterior).



Animal: disentanglement metrics (*f1 score*)

train	KNN	NB	SVM
O	0.960	0.928	0.946
U	0.958	0.930	0.947
C	**0.967**	**0.937**	**0.950**

Discretised spaces and control

Discretisation: 1. Vector Quantisation

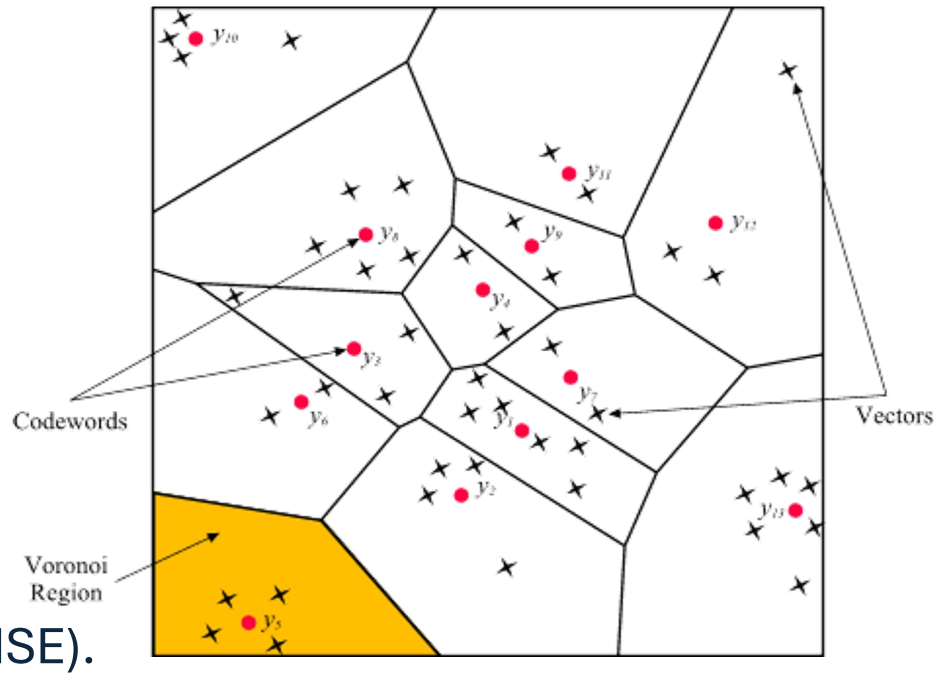
Vector quantisation(VQ): vector quantisation aims to maps *k-dimensional* input vectors *X* in the vector space R^k into a finite set of vectors $Y = \{y_i: i = 1, 2, \ldots, N\}$. Each vector y_i *is* called a **code vector** and the set of all the code vectors is called a **codebook**.

To select y_i from codebook to represent *xi*, we can use *L2* distance (like k-mean).

Codebook initialisation: it can be randomly initialised from a distribution (Normal, uniform). More initialisations:

https://www.mqasem.net/vectorquantization/vq.html

```
self._embedding = nn.Embedding(self._num_embeddings, self._embedding_dim)
self._embedding.weight.data.normal_()
```



Measurement the performance of VQ: using mean square error (MSE).

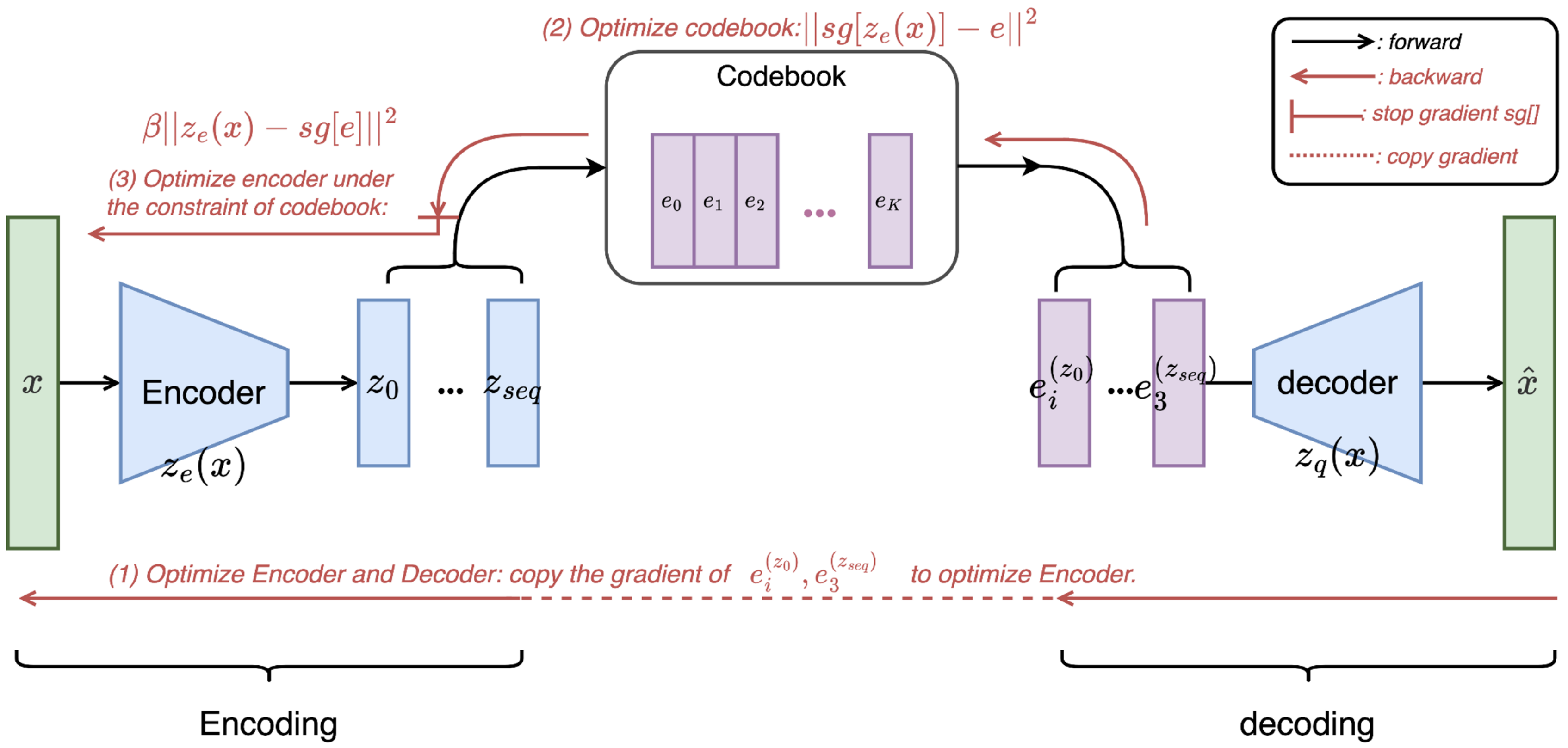
$$MSE = \sum_i (x_i - y^{(x_i)})^2$$

source: https://www.mqasem.net/vectorquantization/vq.html

voronoi region: https://en.wikipedia.org/wiki/Voronoi_diagram

Discretisation: 2. VQ-VAE

VQ-VAE: it [1] first encode a text into token embeddings. Then, selecting the nearest codebook vector as the input of decoder.



(2) Optimize codebook: $||sg[z_e(x)] - e||^2$
Codebook
e_0 e_1 e_2 ... e_K
$\beta||z_e(x) - sg[e]||^2$
(3) Optimize encoder under the constraint of codebook:
: forward
: backward
: stop gradient sg[]
: copy gradient
x
Encoder
$z_e(x)$
z_0 ... z_{seq}
$e_i^{(z_0)}$ ... $e_3^{(z_{seq})}$
decoder
$z_q(x)$
$\hat{x}$
(1) Optimize Encoder and Decoder: copy the gradient of $e_i^{(z_0)}, e_3^{(z_{seq})}$ to optimize Encoder.
Encoding
decoding

$$L = \log p(x|z_q(x)) + \|\text{sg}[z_e(x)] - e\|_2^2 + \beta\|z_e(x) - \text{sg}[e]\|_2^2$$

[1] Aaron Van Den Oord, Oriol Vinyals, et al. 2017. Neural discrete representation learning. Advances in neural information processing systems, 30.

Zhang, Y., Carvalho, D. S., Valentino, M., Pratt-Hartmann, I., & Freitas, A. **Improving Semantic Control in Discrete Latent Spaces with Transformer Quantized Variational Autoencoders**. *EACL Findings 2024*.

Overview: integrating T5 with vector quantisation, named T5VQVAE, to alleviate information bottleneck of posterior for enhancing semantic control.

Language modelling & Inference tasks:

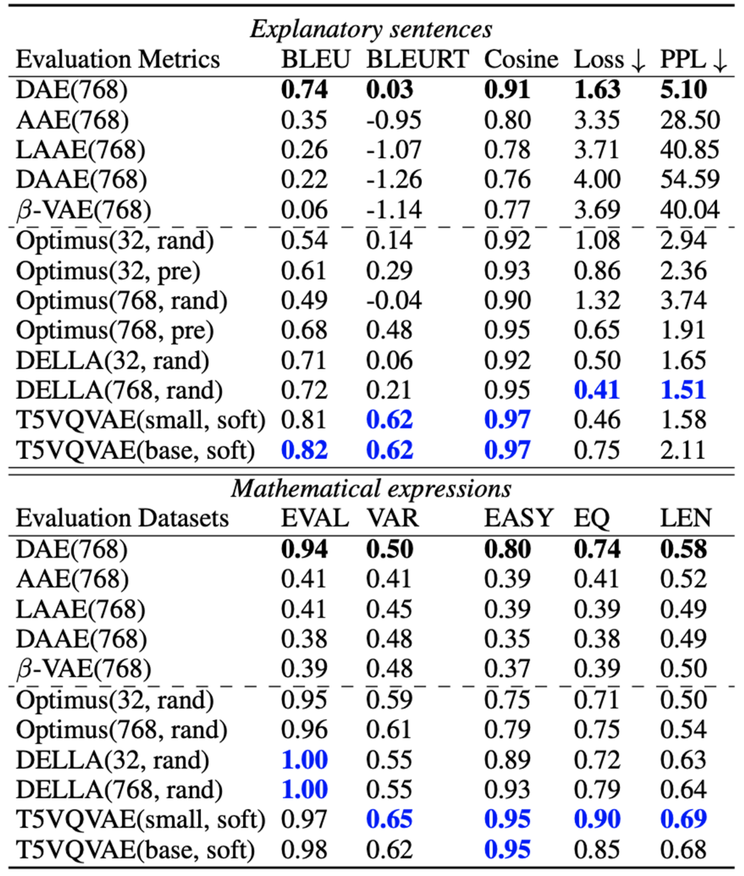
Data:

- Explanations and mathematical expressions.

Evaluation:

- BLEU for math modelling and inference with four OOD testsets.
- BLEU, BLEURT, Cosine, Loss, PPL for explanations.

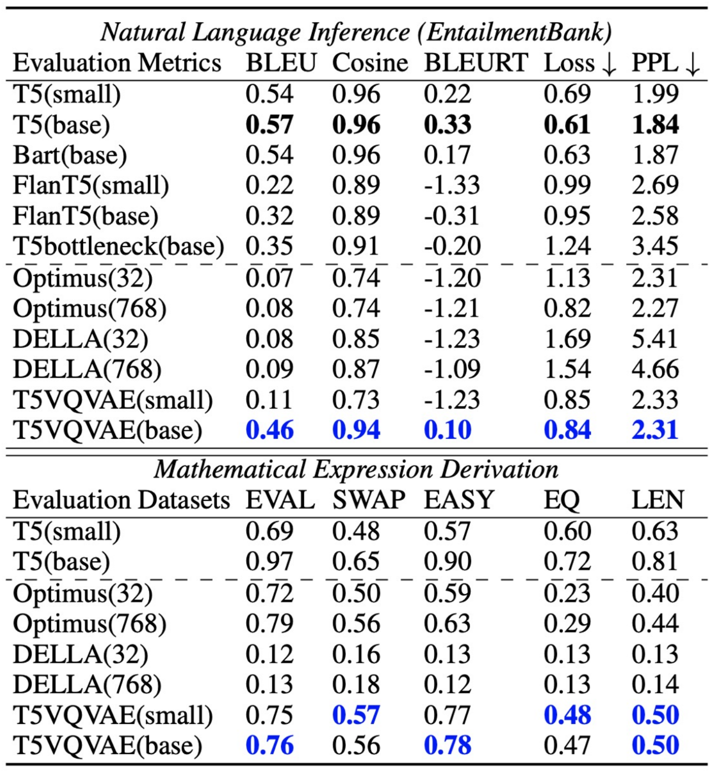
T5VQVAE outperforms Optimus on both tasks.



	Explanatory sentences				
Evaluation Metrics	BLEU	BLEURT	Cosine	Loss ↓	PPL ↓
DAE(768)	**0.74**	**0.03**	**0.91**	**1.63**	**5.10**
AAE(768)	0.35	-0.95	0.80	3.35	28.50
LAAE(768)	0.26	-1.07	0.78	3.71	40.85
DAAE(768)	0.22	-1.26	0.76	4.00	54.59
β-VAE(768)	0.06	-1.14	0.77	3.69	40.04
Optimus(32, rand)	0.54	0.14	0.92	1.08	2.94
Optimus(32, pre)	0.61	0.29	0.93	0.86	2.36
Optimus(768, rand)	0.49	-0.04	0.90	1.32	3.74
Optimus(768, pre)	0.68	0.48	0.95	0.65	1.91
DELLA(32, rand)	0.71	0.06	0.92	0.50	1.65
DELLA(768, rand)	0.72	0.21	0.95	**0.41**	**1.51**
T5VQVAE(small, soft)	0.81	**0.62**	**0.97**	0.46	1.58
T5VQVAE(base, soft)	**0.82**	**0.62**	**0.97**	0.75	2.11

	Mathematical expressions				
Evaluation Datasets	EVAL	VAR	EASY	EQ	LEN
DAE(768)	**0.94**	**0.50**	**0.80**	**0.74**	**0.58**
AAE(768)	0.41	0.41	0.39	0.41	0.52
LAAE(768)	0.41	0.45	0.39	0.39	0.49
DAAE(768)	0.38	0.48	0.35	0.38	0.49
β-VAE(768)	0.39	0.48	0.37	0.39	0.50
Optimus(32, rand)	0.95	0.59	0.75	0.71	0.50
Optimus(768, rand)	0.96	0.61	0.79	0.75	0.54
DELLA(32, rand)	**1.00**	0.55	0.89	0.72	0.63
DELLA(768, rand)	**1.00**	0.55	0.93	0.79	0.64
T5VQVAE(small, soft)	0.97	**0.65**	**0.95**	**0.90**	**0.69**
T5VQVAE(base, soft)	0.98	0.62	**0.95**	0.85	0.68

Table 1: AutoEncoding task evaluation on the test set



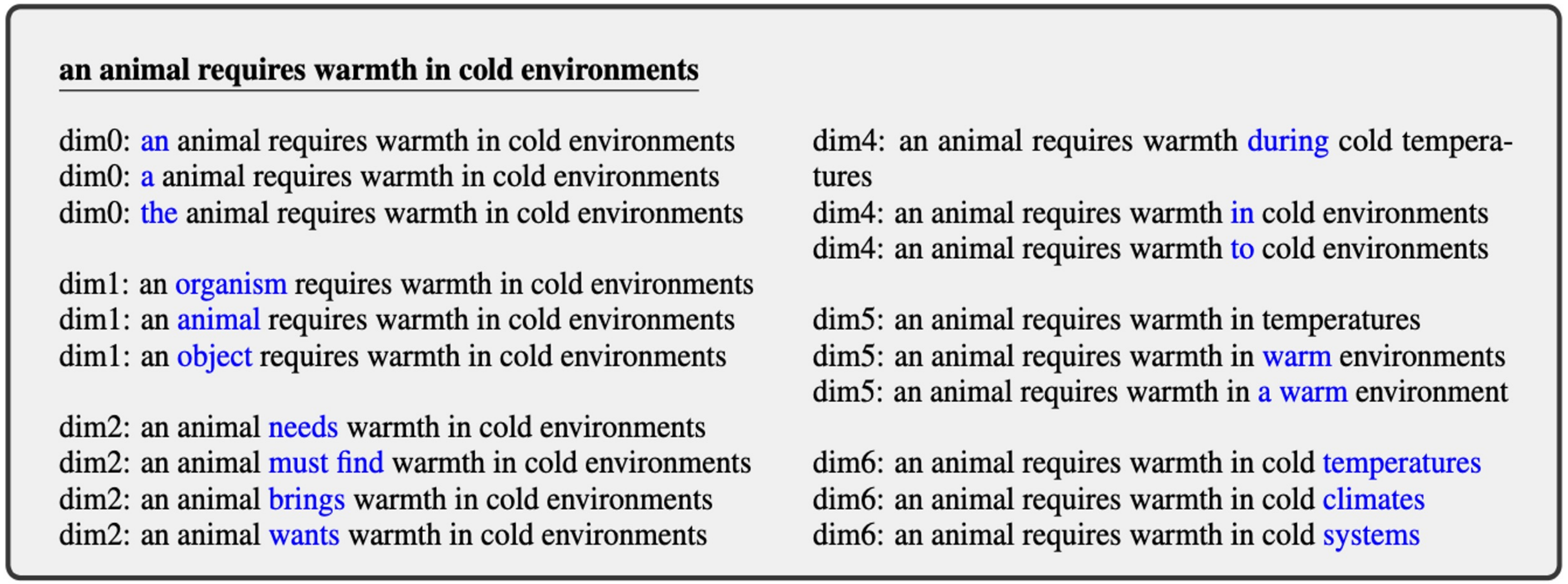
	Natural Language Inference (EntailmentBank)				
Evaluation Metrics	BLEU	Cosine	BLEURT	Loss ↓	PPL ↓
T5(small)	0.54	0.96	0.22	0.69	1.99
T5(base)	**0.57**	**0.96**	**0.33**	**0.61**	**1.84**
Bart(base)	0.54	0.96	0.17	0.63	1.87
FlanT5(small)	0.22	0.89	-1.33	0.99	2.69
FlanT5(base)	0.32	0.89	-0.31	0.95	2.58
T5bottleneck(base)	0.35	0.91	-0.20	1.24	3.45
Optimus(32)	0.07	0.74	-1.20	1.13	2.31
Optimus(768)	0.08	0.74	-1.21	0.82	2.27
DELLA(32)	0.08	0.85	-1.23	1.69	5.41
DELLA(768)	0.09	0.87	-1.09	1.54	4.66
T5VQVAE(small)	0.11	0.73	-1.23	0.85	2.33
T5VQVAE(base)	**0.46**	**0.94**	**0.10**	**0.84**	**2.31**

	Mathematical Expression Derivation				
Evaluation Datasets	EVAL	SWAP	EASY	EQ	LEN
T5(small)	0.69	0.48	0.57	0.60	0.63
T5(base)	0.97	0.65	0.90	0.72	0.81
Optimus(32)	0.72	0.50	0.59	0.23	0.40
Optimus(768)	0.79	0.56	0.63	0.29	0.44
DELLA(32)	0.12	0.16	0.13	0.13	0.13
DELLA(768)	0.13	0.18	0.12	0.13	0.14
T5VQVAE(small)	0.75	**0.57**	0.77	**0.48**	**0.50**
T5VQVAE(base)	**0.76**	0.56	**0.78**	0.47	**0.50**

Empirical analysis

Geometrical evaluation: evaluate controllability of latent space via Traversal, arithmetic, and interpolation.

Traversal: given an input, re-sampling each dimension.



an animal requires warmth in cold environments

dim0: an animal requires warmth in cold environments
dim0: a animal requires warmth in cold environments
dim0: the animal requires warmth in cold environments

dim1: an organism requires warmth in cold environments
dim1: an animal requires warmth in cold environments
dim1: an object requires warmth in cold environments

dim2: an animal needs warmth in cold environments
dim2: an animal must find warmth in cold environments
dim2: an animal brings warmth in cold environments
dim2: an animal wants warmth in cold environments

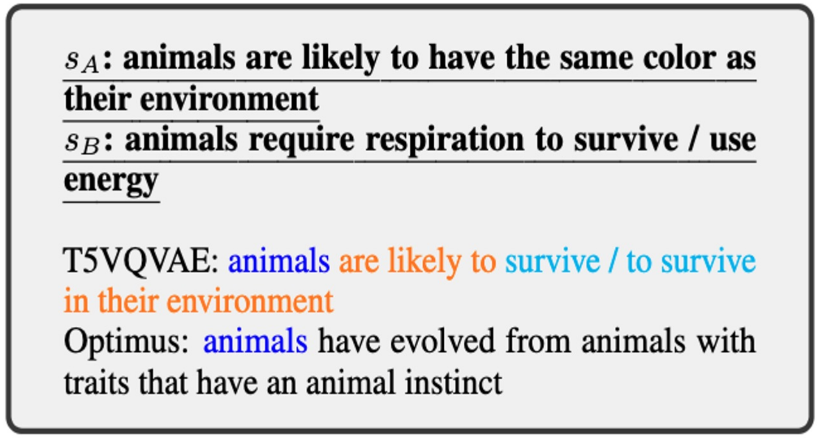
dim4: an animal requires warmth during cold temperatures
dim4: an animal requires warmth in cold environments
dim4: an animal requires warmth to cold environments

dim5: an animal requires warmth in temperatures
dim5: an animal requires warmth in warm environments
dim5: an animal requires warmth in a warm environment

dim6: an animal requires warmth in cold temperatures
dim6: an animal requires warmth in cold climates
dim6: an animal requires warmth in cold systems

Table 3: T5VQVAE(base): traversals showing controlled semantic concepts in explanations. We also provide the traversal of Optimus latent space for comparison in Table 13.

Arithmetic:



s_A: animals are likely to have the same color as their environment
s_B: animals require respiration to survive / use energy

T5VQVAE: animals are likely to survive / to survive in their environment
Optimus: animals have evolved from animals with traits that have an animal instinct

Table 6: Latent arithmetic $s_A + s_B$ for T5VQVAE(base) and Optimus(32). blue, orange, and shallow blue indicate the semantic information from both s_A and s_B, from s_A only, from s_B only, respectively.

Empirical analysis

Interpolation: interpolating over discrete space (i.e., codebook).

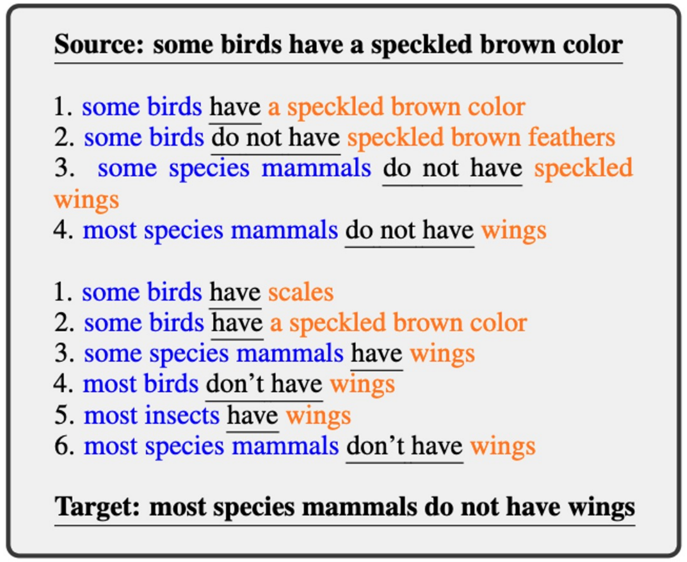
For each token, calculate the weighted minimal intermediate token between its preceding token and the target token.

$$z_1^{w_i} = e^{k_1}, z_2^{w_i} = e^{k_2}, \text{where } \; i = [1, ..., L]$$

$$z_t^{w_i} = z^k, \text{where}$$

$$k = \text{argmin}_j \; (1-t) \times \left\| z_{t-0.1}^{w_i} - z^j \right\|_2 + t \times \left\| z_2^{w_i} - z^j \right\|_2$$

$$s_t = [z_t^{w_1}; \dots; z_t^{w_L}]$$



Source: some birds have a speckled brown color

1. some birds have a speckled brown color
2. some birds do not have speckled brown feathers
3. some species mammals do not have speckled wings
4. most species mammals do not have wings

1. some birds have scales
2. some birds have a speckled brown color
3. some species mammals have wings
4. most birds don't have wings
5. most insects have wings
6. most species mammals don't have wings

Target: most species mammals do not have wings

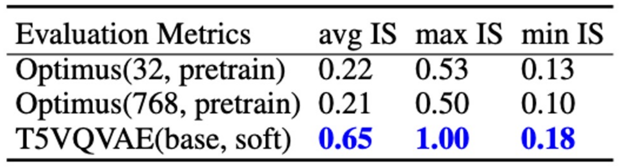
Table 4: Interpolation for T5VQVAE (top) and Optimus (bottom) where blue, underline, and orange represent subject, verb, and object, respectively. Only unique sentences are shown.

Interpolation smoothness: calculating the ratio between ideal semantic distance (i.e., aligned distance between source and target) and actual distance (i.e., sum of aligned semantic distances between each pair of adjacent sentences in the path).

$$\text{IS} = \mathbb{E}_{(s_0,\dots,s_T)\sim P} \frac{\delta(\text{align}(s_0, s_T))}{\sum_{t=0}^{T} \delta(\text{align}(s_t, s_{t+0.1}))}$$

δ : sentence semantic distance

align : sentence feature alignment

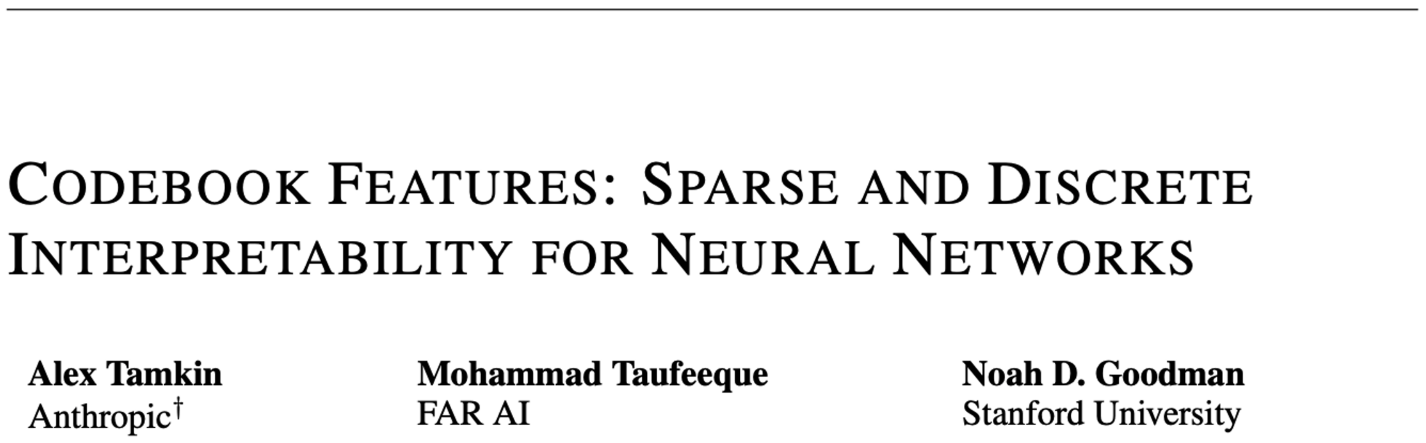


Evaluation Metrics	avg IS	max IS	min IS
Optimus(32, pretrain)	0.22	0.53	0.13
Optimus(768, pretrain)	0.21	0.50	0.10
T5VQVAE(base, soft)	**0.65**	**1.00**	**0.18**

Table 5: Interpolation smoothness.

Observation: T5VQVAE leads to smoother interpolation path.

Related work

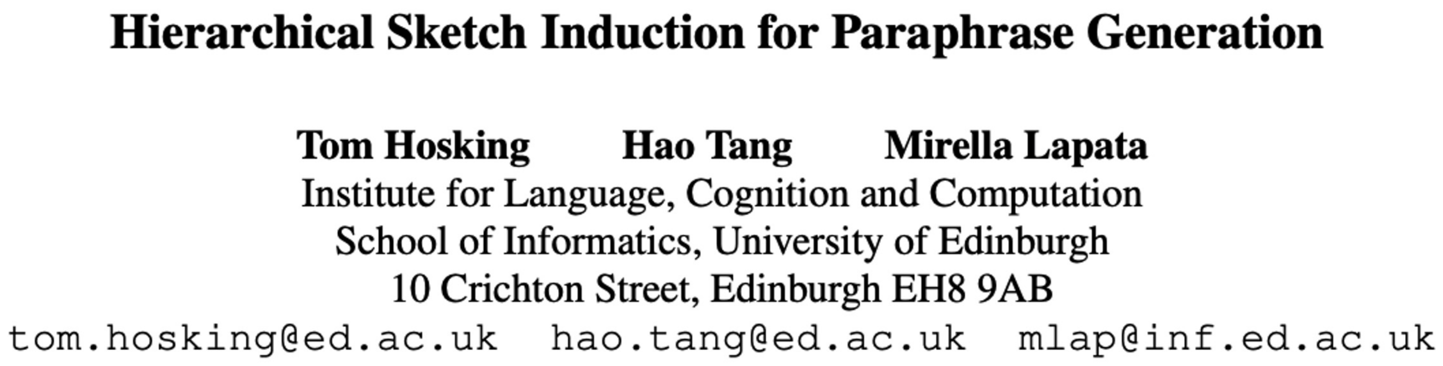


CODEBOOK FEATURES: SPARSE AND DISCRETE INTERPRETABILITY FOR NEURAL NETWORKS

Alex Tamkin
Anthropic[†]

Mohammad Taufeeque
FAR AI

Noah D. Goodman
Stanford University



Hierarchical Sketch Induction for Paraphrase Generation

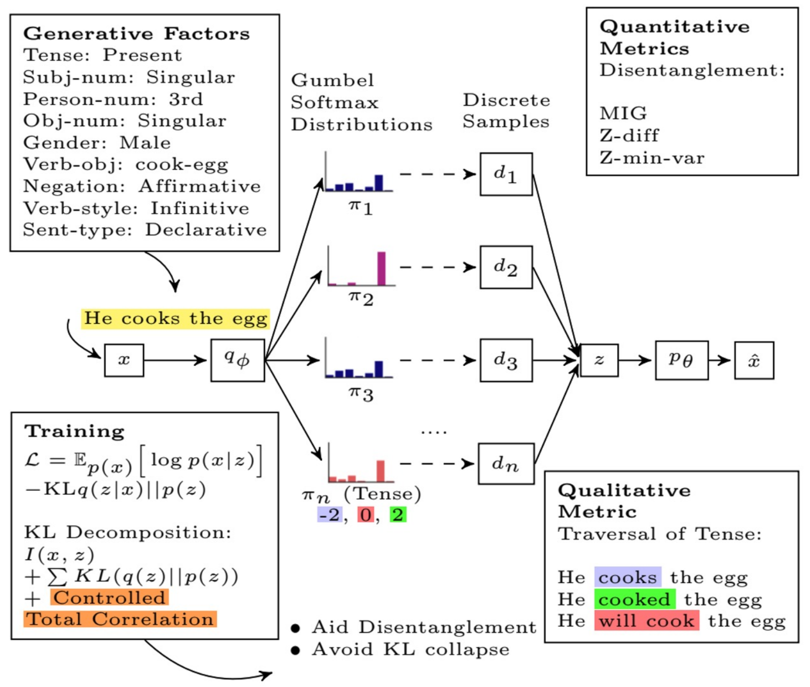
Tom Hosking **Hao Tang** **Mirella Lapata**
Institute for Language, Cognition and Computation
School of Informatics, University of Edinburgh
10 Crichton Street, Edinburgh EH8 9AB
tom.hosking@ed.ac.uk hao.tang@ed.ac.uk mlap@inf.ed.ac.uk

Related work

Disentangling Generative Factors in Natural Language with Discrete Variational Autoencoders

Giangiacomo Mercatali [*]
University of Manchester

André Freitas [†]
Idiap Research Institute
University of Manchester



Factor	Dimensions	Values
Verb/object	1100	[Verb/obj variations]
Gender	2	[Male, Female]
Negation	2	[Affirmative, Negative]
Tense	3	[Present, Future, Past]
Subject number	2	[Singular, plural]
Object number	2	[Singular, plural]
Sentence Type	2	[Interrogative, Declarative]
Person number	3	[1st, 2nd, 3rd person]
Verb style	2	[Gerund, Infinitive]

Syntactic & structural controls

Graph Neural Networks

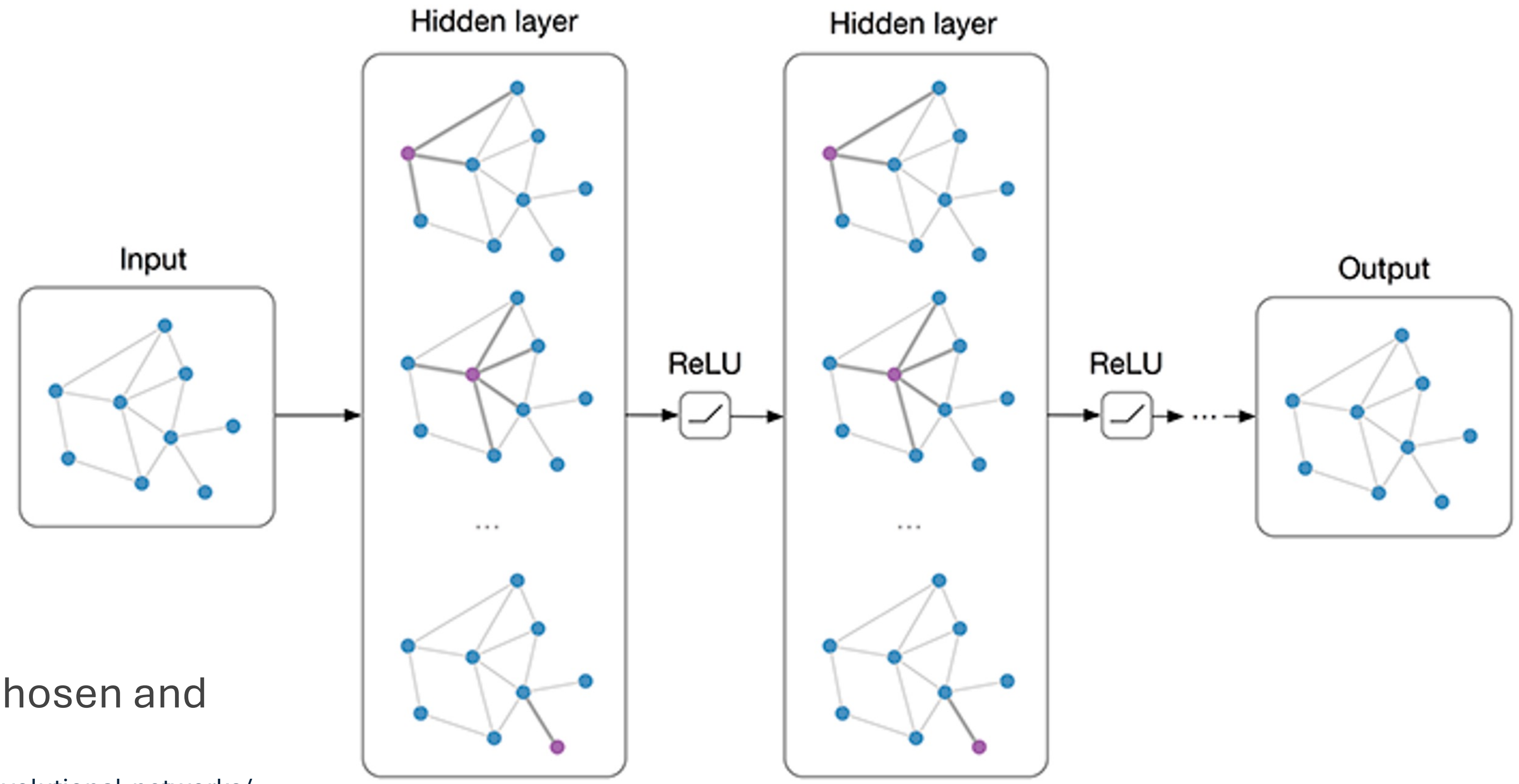
Graph neural network: learns a function of signals/features on a graph G=(V,E) which takes as input: (1) **node embedding** *(i.e., V)* and (2) **adjacency matrix** *(i.e., E)*.

E.g., given a GNN with L layers, the *l-th* layer can then be written as:

$$H^{(l+1)} = f^l(H^l, A)$$

H^0 : initial node embeddings.

A : adjacency matrix.



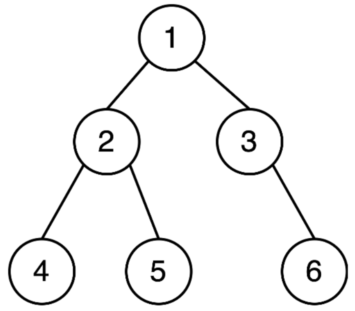
Distinct models differ only in how $f(\cdot,\cdot)$ is chosen and parameterised.

source: https://tkipf.github.io/graph-convolutional-networks/

Graph Neural Networks

Graph Convolutional Network:

$$H^{(l+1)} = f^{l}(H^{l}, A) = \sigma(\boxed{AH^{l}}W^{l})$$



graph

$$\begin{bmatrix} 0 & 1 & 1 & 0 & 0 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

Adjacency

$$\begin{bmatrix} 2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 3 & 0 & 0 & 0 & 0 \\ 0 & 0 & 2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Degree

Two limitations:

1. Multiplication with A means that, for every node, we **sum up all the feature vectors of all neighboring nodes** but not the node itself. We can "fix" this by enforcing self-loops in the graph: simply add the identity matrix to A. or D - A (L = D - A, L is *Combinatorial Laplacian*).

2. The second major limitation is that A is typically not normalised and therefore the multiplication with A will completely change the scale of the feature vectors. E.g, some nodes have more connections. We can solve it by multiplying D^{-1} where D is the diagonal node degree matrix.

$$H^{(l+1)} = f^{l}(H^{l}, A) = \sigma(\underbrace{D^{-\frac{1}{2}}(D - A)D^{-\frac{1}{2}}}_{L^{sym}}H^{l}W^{l})$$

multiplying the left and right by the square root of the degrees of nodes i and j respectively is to consider the degrees of the points on both sides of an edge.

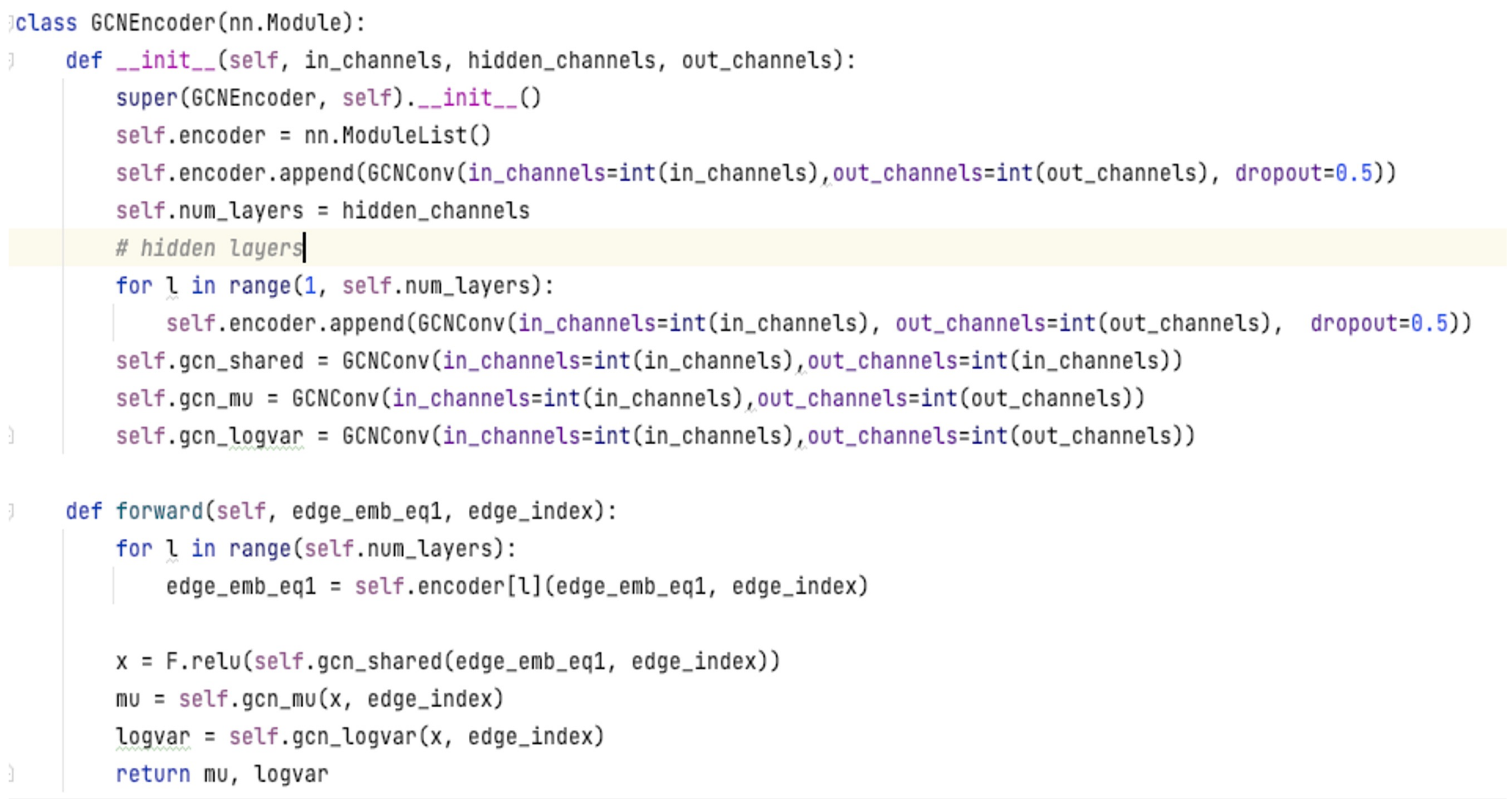
Symmetric normalised Laplacian

$$D^{-\frac{1}{2}}D = I$$

Graph Neural Networks

Pytorch framework:

PyTorch Geometric(PyG): https://pytorch-geometric.readthedocs.io/en/latest/



```python
class GCNEncoder(nn.Module):
    def __init__(self, in_channels, hidden_channels, out_channels):
        super(GCNEncoder, self).__init__()
        self.encoder = nn.ModuleList()
        self.encoder.append(GCNConv(in_channels=int(in_channels),out_channels=int(out_channels), dropout=0.5))
        self.num_layers = hidden_channels
        # hidden layers
        for l in range(1, self.num_layers):
            self.encoder.append(GCNConv(in_channels=int(in_channels), out_channels=int(out_channels),  dropout=0.5))
        self.gcn_shared = GCNConv(in_channels=int(in_channels),out_channels=int(in_channels))
        self.gcn_mu = GCNConv(in_channels=int(in_channels),out_channels=int(out_channels))
        self.gcn_logvar = GCNConv(in_channels=int(in_channels),out_channels=int(out_channels))

    def forward(self, edge_emb_eq1, edge_index):
        for l in range(self.num_layers):
            edge_emb_eq1 = self.encoder[l](edge_emb_eq1, edge_index)

        x = F.relu(self.gcn_shared(edge_emb_eq1, edge_index))
        mu = self.gcn_mu(x, edge_index)
        logvar = self.gcn_logvar(x, edge_index)
        return mu, logvar
```

Zhang, Y., Valentino, M., Carvalho, D. S., Pratt-Hartmann, I., & Freitas, A.
Graph-Induced Syntactic-Semantic Spaces in Transformer-Based Variational AutoEncoders.
NAACL Findings 2024.

Motivation: Syntactic injection of language models.

Syntactic injection of language models via low-dimensional latent Gaussian space with graph neural networks.

What's the relation between syntax and semantics in this work? semantics: **word content + order** (i.e, word order typology); syntax: **constituency tree - word content**.

How to get the syntactic tree? constituency tree parser.

[1] Laurent Sartran, Samuel Barrett, Adhiguna Kuncoro, Miloš Stanojević, Phil Blunsom, and Chris Dyer. 2022. Transformer Grammars: Augmenting Transformer Language Models with Syntactic Inductive Biases at Scale. *Transactions of the Association for Computational Linguistics*, 10:1423–1439.

[2] Xiang Hu, Qingyang Zhu, Kewei Tu, Wei Wu, "Augmenting transformers with recursively composed multi-grained representations". In *the Twelfth International Conference on Learning Representations (ICLR 2024)*, Vienna, Austria, May 7-11, 2024.

Methodology:
Q1. How to efficiently encode syntax in latent spaces?

Encoding syntax in latent space: we first propose four encoding strategies to evaluate their capabilities to represent syntactic information.

Single encoder with multi-task learning:

(1) LSTM: jointly train with LSTM decoder.

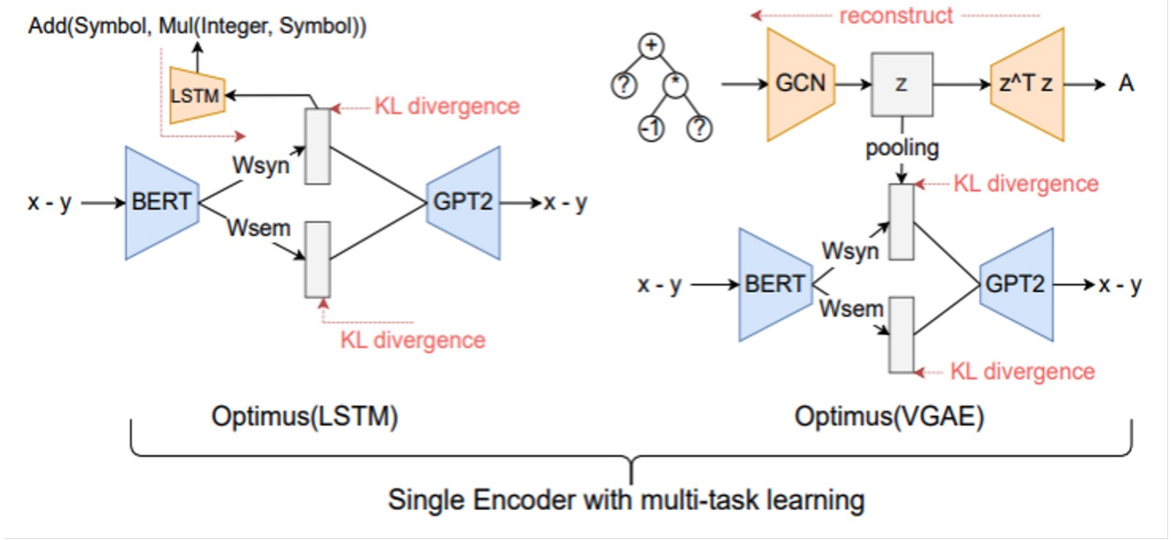
(2) VGAE: jointly train with Graph VAE.

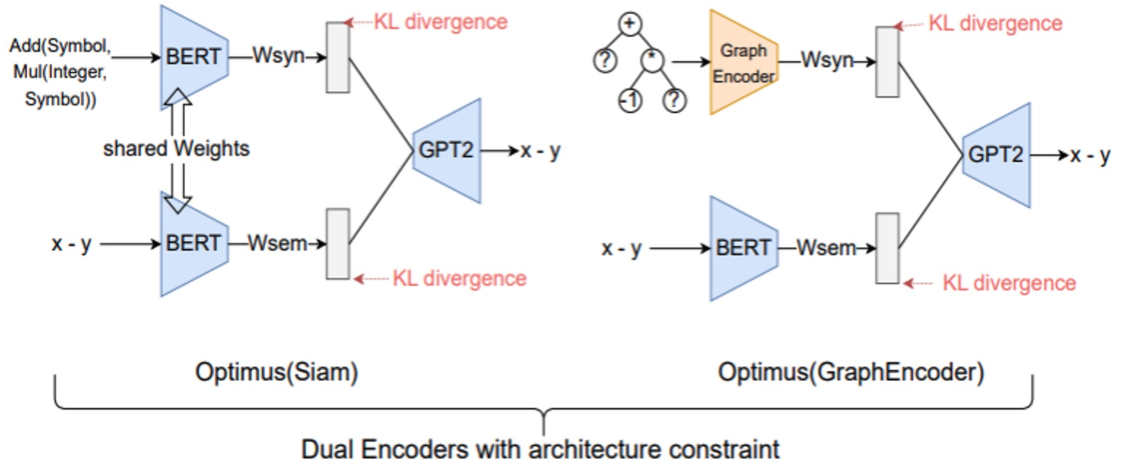
Dual encoders with architectural constraints:

(3) Siam: two bert encoders, one with flatten syntax.

(4) GraphEncoder: graph and language encoders.

Targeted injected space: Optimus.

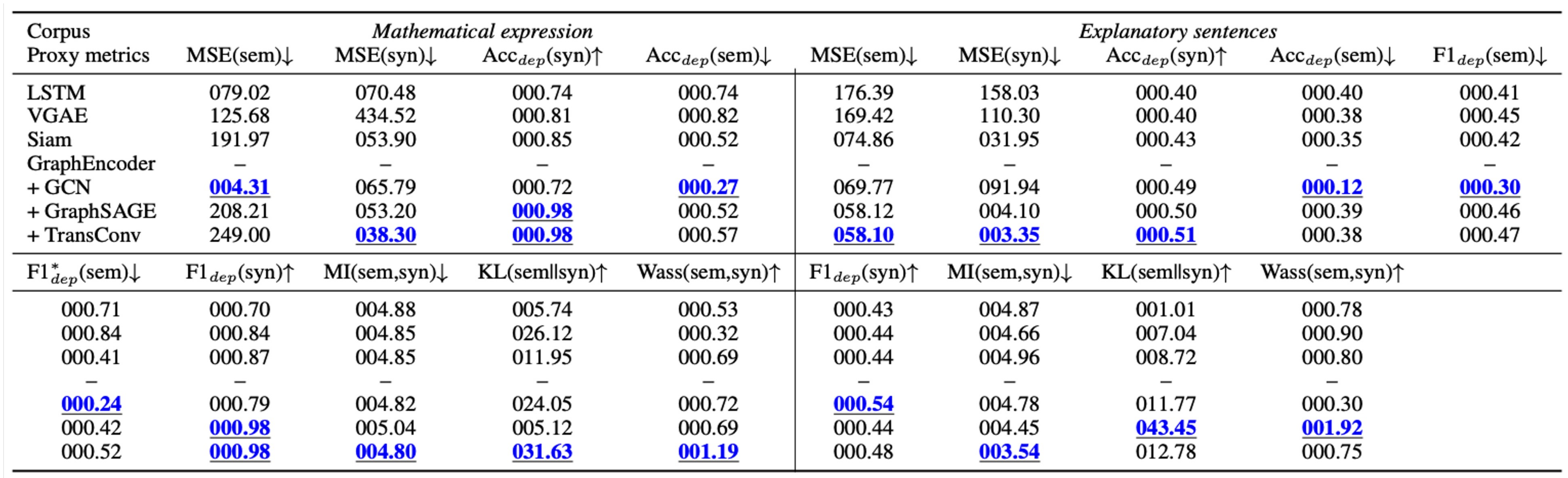




Empirical analysis

Syntactic representation evaluation: quantitatively evaluating syntax space, including:

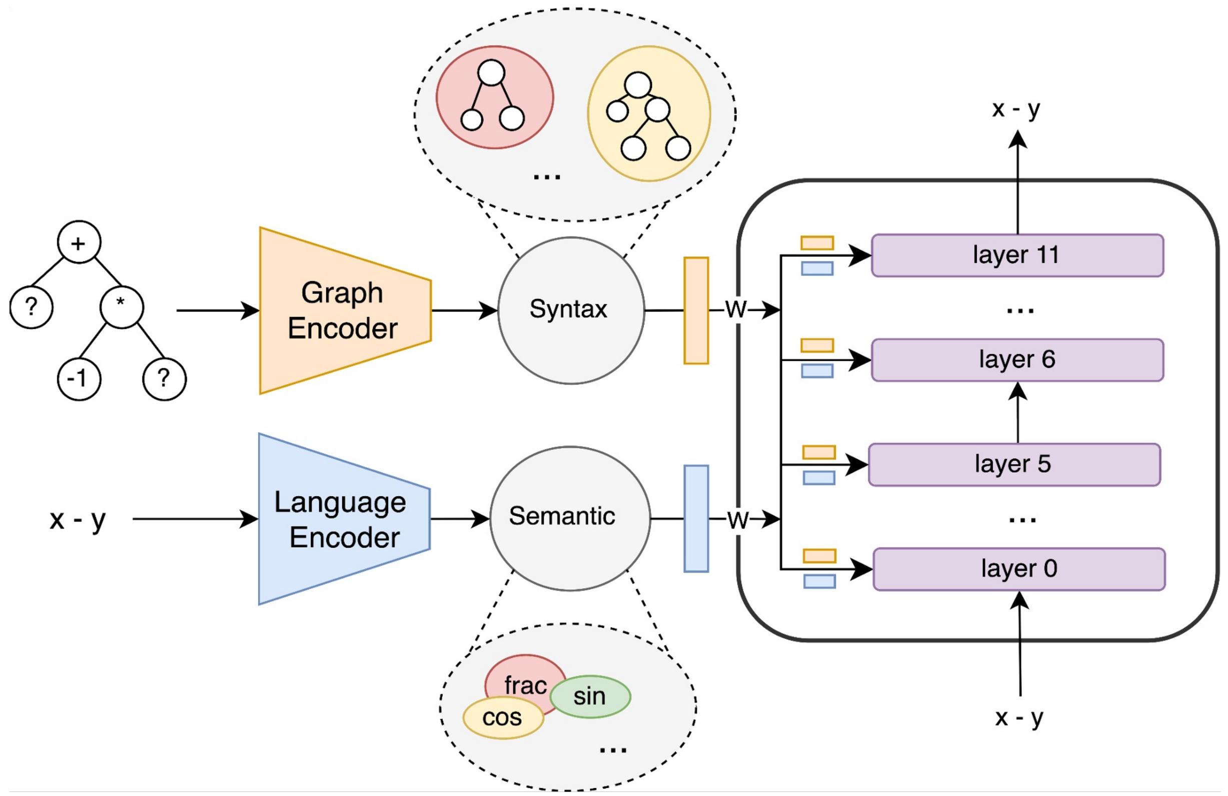
(1) latent space geometry: sentences with the same/different features are clustered/separated in the latent space. In this case, we can evaluate the organisation of the latent space via MSE of k-mean, denoted by *MSE(sem/syn)*.
(2) tree depth: we train a linear classifier to predict tree depth.
(3) semantic-syntax separation: Mutual Information, KL divergence, and Wasserstein distance.

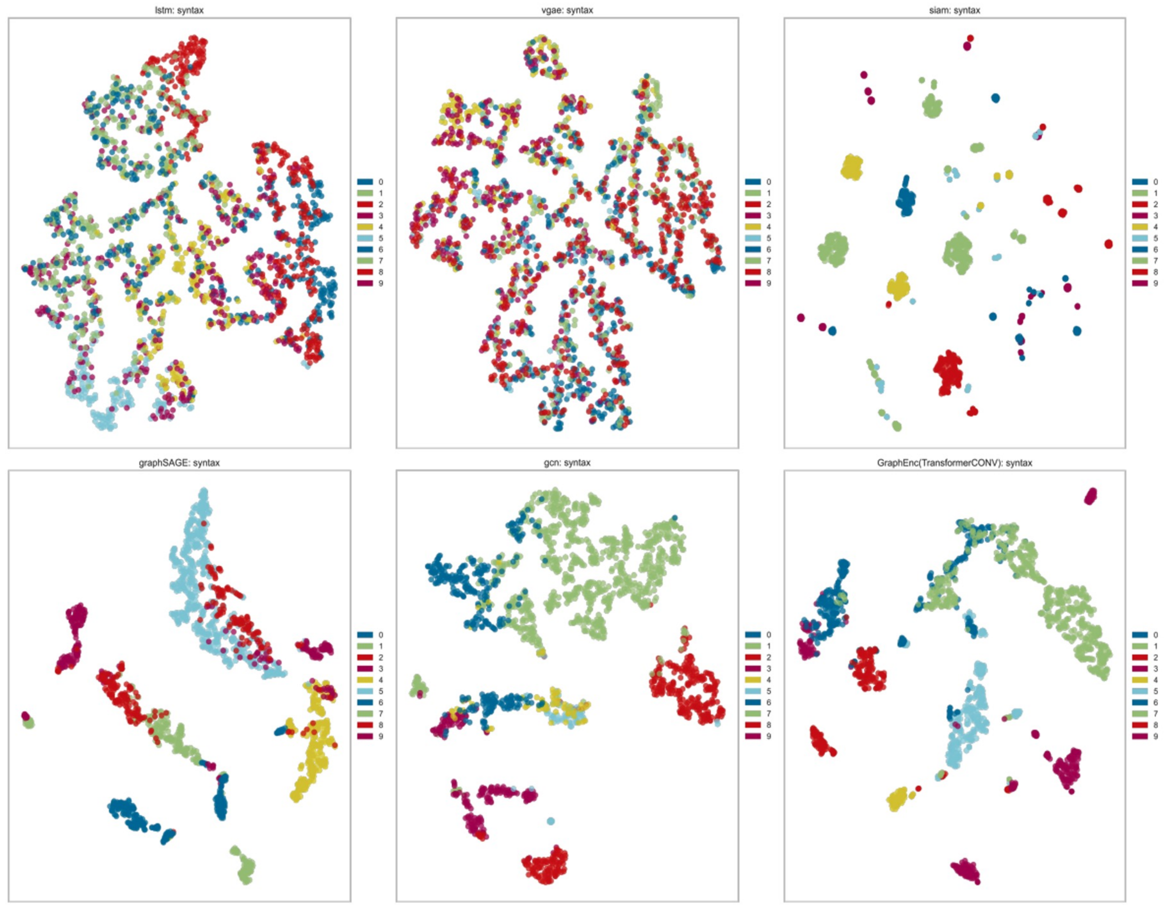


Corpus	*Mathematical expression*									*Explanatory sentences*								
Proxy metrics	MSE(sem)↓	MSE(syn)↓	Acc_{dep}(syn)↑	Acc_{dep}(sem)↓	F1^{*}_{dep}(sem)↓	F1_{dep}(syn)↑	MI(sem,syn)↓	KL(sem‖syn)↑	Wass(sem,syn)↑	MSE(sem)↓	MSE(syn)↓	Acc_{dep}(syn)↑	Acc_{dep}(sem)↓	F1_{dep}(sem)↓	F1_{dep}(syn)↑	MI(sem,syn)↓	KL(sem‖syn)↑	Wass(sem,syn)↑
LSTM	079.02	070.48	000.74	000.74	000.71	000.70	004.88	005.74	000.53	176.39	158.03	000.40	000.40	000.41	000.43	004.87	001.01	000.78
VGAE	125.68	434.52	000.81	000.82	000.84	000.84	004.85	026.12	000.32	169.42	110.30	000.40	000.38	000.45	000.44	004.66	007.04	000.90
Siam	191.97	053.90	000.85	000.52	000.41	000.87	004.85	011.95	000.69	074.86	031.95	000.43	000.35	000.42	000.44	004.96	008.72	000.80
GraphEncoder	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–	–
+ GCN	**004.31**	065.79	000.72	**000.27**	**000.24**	000.79	004.82	024.05	000.72	069.77	091.94	000.49	**000.12**	**000.30**	**000.54**	004.78	011.77	000.30
+ GraphSAGE	208.21	053.20	**000.98**	000.52	000.42	**000.98**	005.04	005.12	000.69	058.12	004.10	000.50	000.39	000.46	000.44	004.45	**043.45**	**001.92**
+ TransConv	249.00	**038.30**	**000.98**	000.57	000.52	**000.98**	**004.80**	**031.63**	**001.19**	**058.10**	**003.35**	**000.51**	000.38	000.47	000.48	**003.54**	012.78	000.75

Empirical analysis

Visualisation of syntax space: evaluating cluster and separation of syntax space via t-SNE. If the latent space can encode the clear syntax feature, we should see clear syntax cluster and separation.



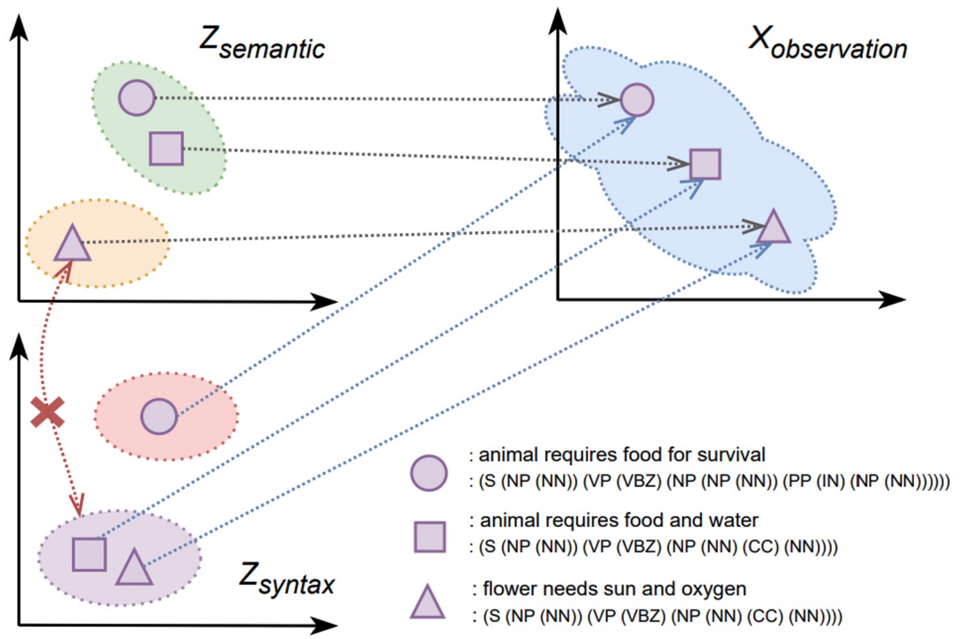


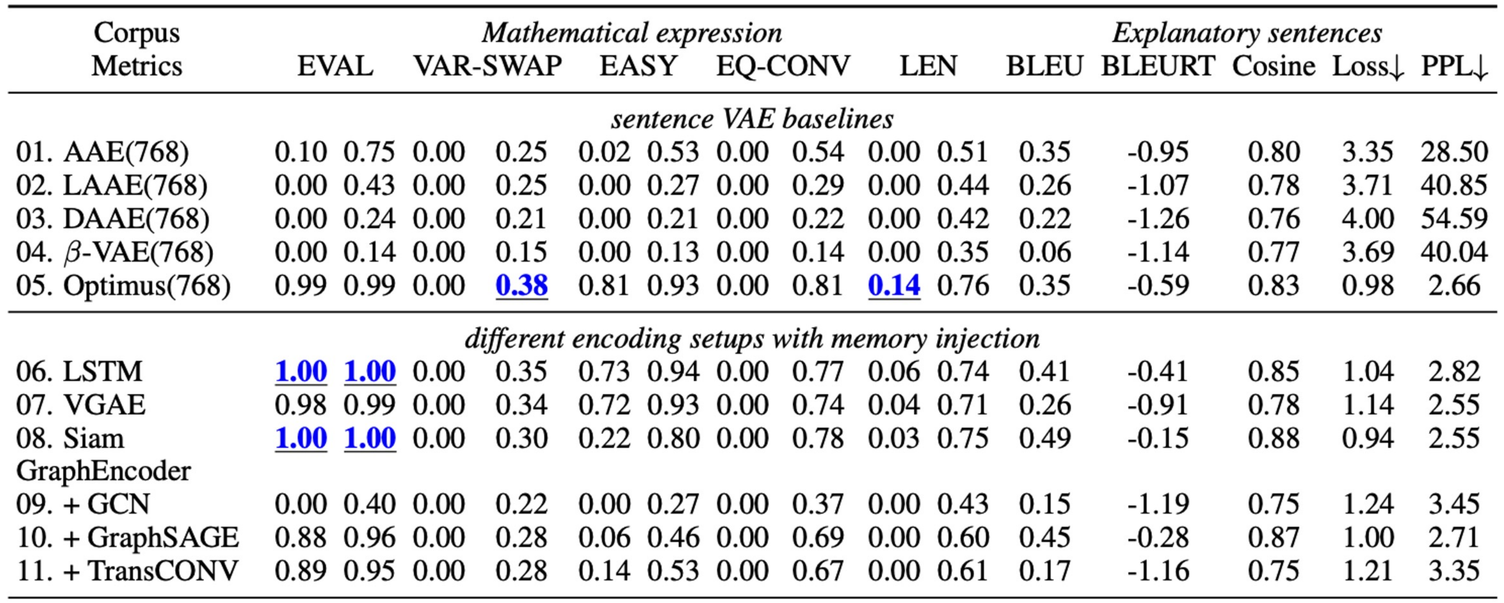
Observation: graph-language encoders can better represent syntax information and semantic-syntax separation.

(top: LSTM, VGAE, Siam, **bottom: graph encoders with GraphSAGE, GCN, TransformerCONV**).

Empirical analysis

Decoding problem: decoding under heterogeneous spaces (graph-language encoders) leads to ***worse language modelling performance*** (lines *05 vs 09-11*) because of distinct latent space geometries from syntax and semantic spaces.


$Z_{semantic}$
$X_{observation}$
Z_{syntax}
: animal requires food for survival
: (S (NP (NN)) (VP (VBZ) (NP (NP (NN)) (PP (IN) (NP (NN))))))
: animal requires food and water
: (S (NP (NN)) (VP (VBZ) (NP (NN) (CC) (NN))))
: flower needs sun and oxygen
: (S (NP (NN)) (VP (VBZ) (NP (NN) (CC) (NN))))



Corpus	*Mathematical expression*										*Explanatory sentences*				
Metrics	EVAL		VAR-SWAP		EASY		EQ-CONV		LEN		BLEU	BLEURT	Cosine	Loss↓	PPL↓
sentence VAE baselines															
01. AAE(768)	0.10	0.75	0.00	0.25	0.02	0.53	0.00	0.54	0.00	0.51	0.35	-0.95	0.80	3.35	28.50
02. LAAE(768)	0.00	0.43	0.00	0.25	0.00	0.27	0.00	0.29	0.00	0.44	0.26	-1.07	0.78	3.71	40.85
03. DAAE(768)	0.00	0.24	0.00	0.21	0.00	0.21	0.00	0.22	0.00	0.42	0.22	-1.26	0.76	4.00	54.59
04. β-VAE(768)	0.00	0.14	0.00	0.15	0.00	0.13	0.00	0.14	0.00	0.35	0.06	-1.14	0.77	3.69	40.04
05. Optimus(768)	0.99	0.99	0.00	**0.38**	0.81	0.93	0.00	0.81	**0.14**	0.76	0.35	-0.59	0.83	0.98	2.66
different encoding setups with memory injection															
06. LSTM	**1.00**	**1.00**	0.00	0.35	0.73	0.94	0.00	0.77	0.06	0.74	0.41	-0.41	0.85	1.04	2.82
07. VGAE	0.98	0.99	0.00	0.34	0.72	0.93	0.00	0.74	0.04	0.71	0.26	-0.91	0.78	1.14	2.55
08. Siam	**1.00**	**1.00**	0.00	0.30	0.22	0.80	0.00	0.78	0.03	0.75	0.49	-0.15	0.88	0.94	2.55
GraphEncoder															
09. + GCN	0.00	0.40	0.00	0.22	0.00	0.27	0.00	0.37	0.00	0.43	0.15	-1.19	0.75	1.24	3.45
10. + GraphSAGE	0.88	0.96	0.00	0.28	0.06	0.46	0.00	0.69	0.00	0.60	0.45	-0.28	0.87	1.00	2.71
11. + TransCONV	0.89	0.95	0.00	0.28	0.14	0.53	0.00	0.67	0.00	0.61	0.17	-1.16	0.75	1.21	3.35

** As for math expression, we evaluate it with BLEU on four Out-Of-Distribution test sets.*

Methodology:
Q2. How to decode over heterogeneous spaces?

Decoding heterogeneous space: we inject distinct spaces into different spaces of decoder.

Optimus(mem): the latent space is injected into K and V.

$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{Q[z; K]^T}{\sqrt{d}})[z; V]$$

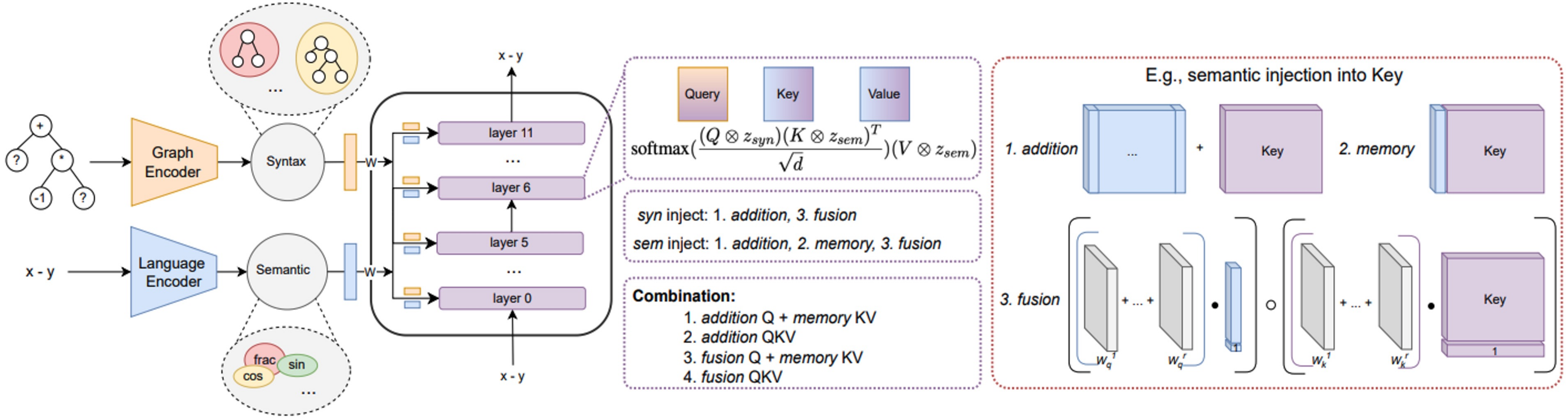
Ours: injecting semantic-syntax spaces into different decoder's space. That is, injecting syntax into Q and semantic into K and V.

$$\text{softmax}(\frac{(Q \otimes z_{syn})(K \otimes z_{sem})^T}{\sqrt{d}})(V \otimes z_{sem})$$

Methodology: Q2. how to decode over heterogeneous space?

Three injection operations: (1) *addition*, (2) *mem*, (3) *tensor fusion*[1]. For syntax injection: (1) and (3). For semantic injection: (1), (2), and (3).

Finally, four combinations: *addition Q + mem KV; addition QKV; fusion Q+mem KV; fusion QKV*


x - y
Graph Encoder
Syntax
Language Encoder
Semantic
W
layer 11
layer 6
layer 5
layer 0
Query
Key
Value
$\mathrm{softmax}(\frac{(Q \otimes z_{syn})(K \otimes z_{sem})^T}{\sqrt{d}})(V \otimes z_{sem})$
syn inject: 1. *addition*, 3. *fusion*
sem inject: 1. *addition*, 2. *memory*, 3. *fusion*
Combination:
1. *addition* Q + *memory* KV
2. *addition* QKV
3. *fusion* Q + *memory* KV
4. *fusion* QKV
E.g., semantic injection into Key
1. addition
2. memory
3. fusion
frac
sin
cos

[1] Zhun Liu, Ying Shen, Varun Bharadhwaj Lakshminarasimhan, Paul Pu Liang, AmirAli Bagher Zadeh, and Louis-Philippe Morency. 2018. Efficient lowrank multimodal fusion with modality-specific factors. In Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 2247–2256, Melbourne, Australia. Association for Computational Linguistics.

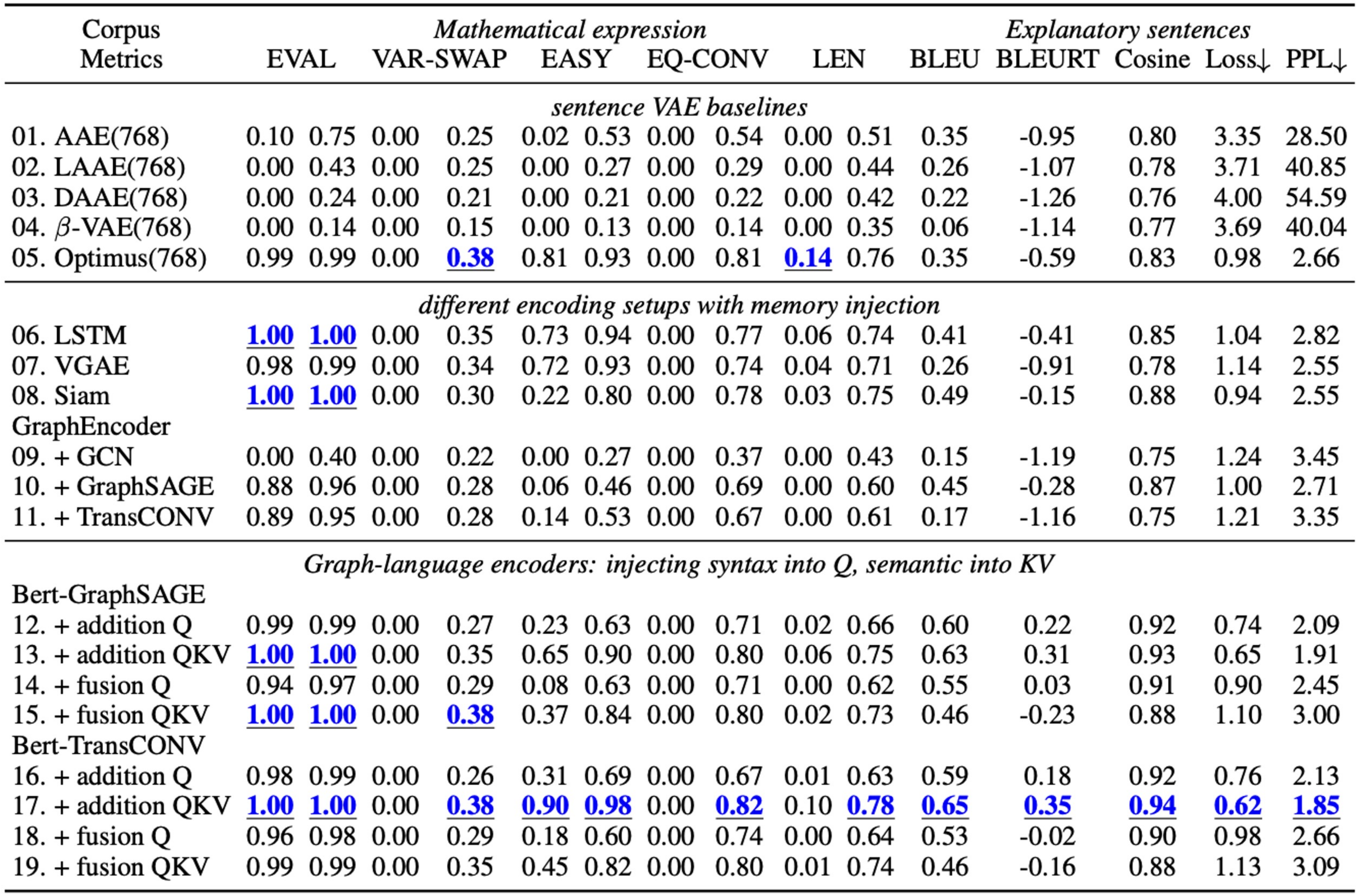
Empirical analysis

Language modelling task:

1. injecting only syntax in Q can improve LM performances on explanatory sentences. (05 vs 12,14,16,18).

2. injecting semantic and syntax spaces into different attention components can additionally improve model performance. (lines 9-11 vs 12, 14, 16, 18)

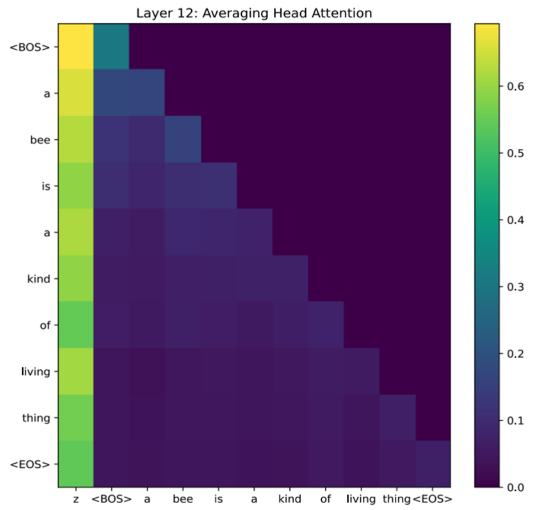
3. addition injection with Bert - TransCONV (line 17) can achieve the best overall results.



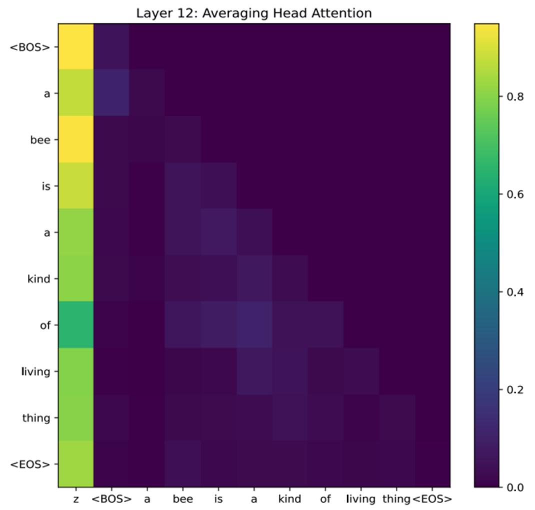
Corpus	*Mathematical expression*										*Explanatory sentences*				
Metrics	EVAL		VAR-SWAP		EASY		EQ-CONV		LEN		BLEU	BLEURT	Cosine	Loss↓	PPL↓
sentence VAE baselines															
01. AAE(768)	0.10	0.75	0.00	0.25	0.02	0.53	0.00	0.54	0.00	0.51	0.35	-0.95	0.80	3.35	28.50
02. LAAE(768)	0.00	0.43	0.00	0.25	0.00	0.27	0.00	0.29	0.00	0.44	0.26	-1.07	0.78	3.71	40.85
03. DAAE(768)	0.00	0.24	0.00	0.21	0.00	0.21	0.00	0.22	0.00	0.42	0.22	-1.26	0.76	4.00	54.59
04. β-VAE(768)	0.00	0.14	0.00	0.15	0.00	0.13	0.00	0.14	0.00	0.35	0.06	-1.14	0.77	3.69	40.04
05. Optimus(768)	0.99	0.99	0.00	**0.38**	0.81	0.93	0.00	0.81	**0.14**	0.76	0.35	-0.59	0.83	0.98	2.66
different encoding setups with memory injection															
06. LSTM	**1.00**	**1.00**	0.00	0.35	0.73	0.94	0.00	0.77	0.06	0.74	0.41	-0.41	0.85	1.04	2.82
07. VGAE	0.98	0.99	0.00	0.34	0.72	0.93	0.00	0.74	0.04	0.71	0.26	-0.91	0.78	1.14	2.55
08. Siam	**1.00**	**1.00**	0.00	0.30	0.22	0.80	0.00	0.78	0.03	0.75	0.49	-0.15	0.88	0.94	2.55
GraphEncoder															
09. + GCN	0.00	0.40	0.00	0.22	0.00	0.27	0.00	0.37	0.00	0.43	0.15	-1.19	0.75	1.24	3.45
10. + GraphSAGE	0.88	0.96	0.00	0.28	0.06	0.46	0.00	0.69	0.00	0.60	0.45	-0.28	0.87	1.00	2.71
11. + TransCONV	0.89	0.95	0.00	0.28	0.14	0.53	0.00	0.67	0.00	0.61	0.17	-1.16	0.75	1.21	3.35
Graph-language encoders: injecting syntax into Q, semantic into KV															
Bert-GraphSAGE															
12. + addition Q	0.99	0.99	0.00	0.27	0.23	0.63	0.00	0.71	0.02	0.66	0.60	0.22	0.92	0.74	2.09
13. + addition QKV	**1.00**	**1.00**	0.00	0.35	0.65	0.90	0.00	0.80	0.06	0.75	0.63	0.31	0.93	0.65	1.91
14. + fusion Q	0.94	0.97	0.00	0.29	0.08	0.63	0.00	0.71	0.00	0.62	0.55	0.03	0.91	0.90	2.45
15. + fusion QKV	**1.00**	**1.00**	0.00	**0.38**	0.37	0.84	0.00	0.80	0.02	0.73	0.46	-0.23	0.88	1.10	3.00
Bert-TransCONV															
16. + addition Q	0.98	0.99	0.00	0.26	0.31	0.69	0.00	0.67	0.01	0.63	0.59	0.18	0.92	0.76	2.13
17. + addition QKV	**1.00**	**1.00**	0.00	**0.38**	**0.90**	**0.98**	0.00	**0.82**	0.10	**0.78**	**0.65**	**0.35**	**0.94**	**0.62**	**1.85**
18. + fusion Q	0.96	0.98	0.00	0.29	0.18	0.60	0.00	0.74	0.00	0.64	0.53	-0.02	0.90	0.98	2.66
19. + fusion QKV	0.99	0.99	0.00	0.35	0.45	0.82	0.00	0.80	0.01	0.74	0.46	-0.16	0.88	1.13	3.09

Empirical analysis

Question: Why graph-language encoders can improve language modelling performance?



$$\text{Attention}(Q,K,V) = \text{softmax}(\frac{Q[z;K]^T}{\sqrt{d}})[z;V]$$



$$\text{softmax}(\frac{(Q \otimes z_{syn})[z_{sem};K]^T}{\sqrt{d}})[z_{sem};V]$$

Gold explanations	BERT-GPT2	Bert/TransCONV-GPT2
lenses are a kind of object	frog is a kind of object	lenses are a kind of object
the chemical symbol for helium is he	a substance has a physical shape	the chemical symbol for helium is He
a rose is a kind of plant	a window pane is a kind of surface	a rose is a kind of flower
a body of water contains water	a flood has a large amount of rainfall	a body of water contains water
growing is a kind of process	population is a kind of process	growing is a kind of process
air is a kind of gas	farming is a kind of human	air is a kind of gas
action means activity	feed means use	activity means action
soda water is a kind of carbonated beverage	condensing is a kind of change in temperature	soda water is a kind of carbonated beverage
plasma is a kind of state of matter	black probability is a kind of event	plasma is a kind of state of matter
earth is a kind of celestial object	sun is a kind of light	earth is a kind of celestial object
a bee is a kind of living thing	a frog is a kind of amphibian	a bee is a kind of living thing
green is a kind of color	deforestation is a kind of process	green is a kind of color
a wooded area is a kind of forest	a coal mine is a kind of natural resource	a wooded area is a kind of forest

Observation: Comparing vanilla Optimus with Bert-TransCONV(*addition Q)*, the latent space can better encode lexical information.

Hypotheses: language encoder induce information bottleneck (i.e., trade-off between semantics and syntax), dual encoders can alleviate such bottleneck (see our paper for proof).

Empirical analysis

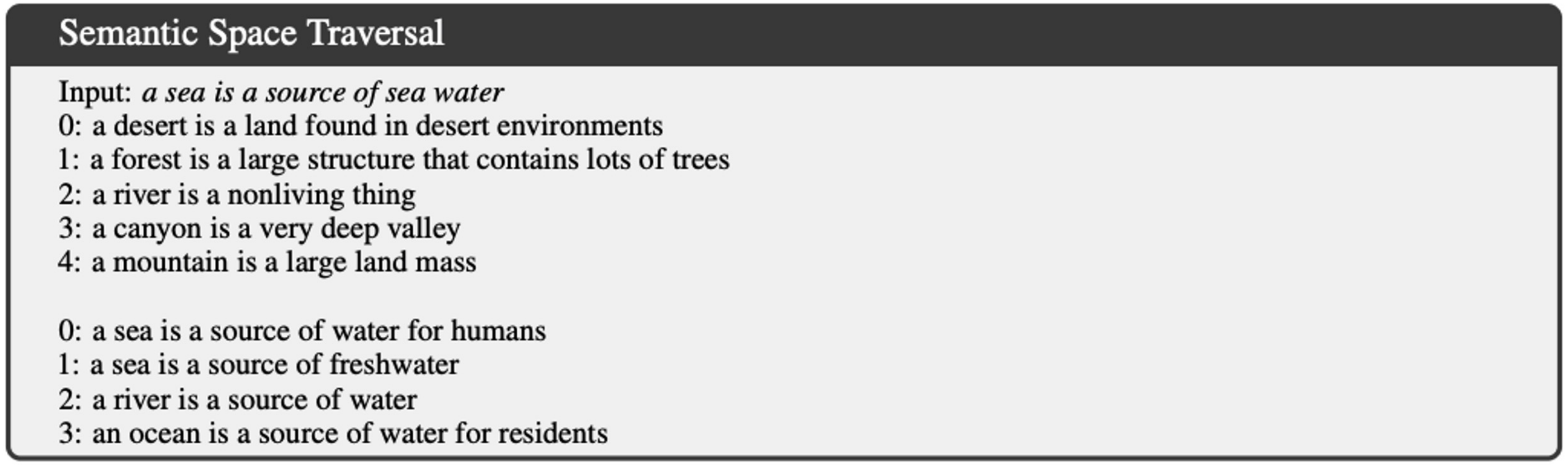
Latent traversal:

Given an input, performing random walk (e.g., *Ornstein- Uhlenbeck*)

Observation:

Graph-language encoders setup leads to better generation control.

Traversing syntax lead to both syntax and semantics changed.

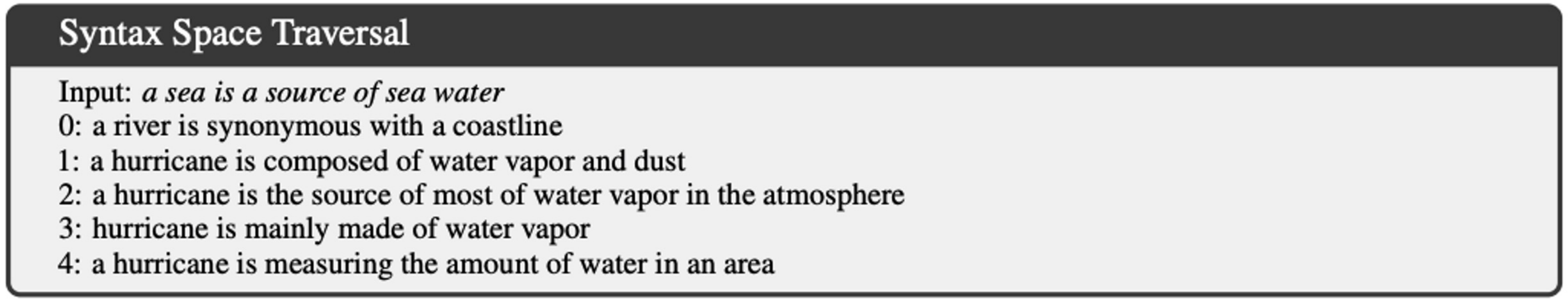


Semantic Space Traversal

Input: *a sea is a source of sea water*
0: a desert is a land found in desert environments
1: a forest is a large structure that contains lots of trees
2: a river is a nonliving thing
3: a canyon is a very deep valley
4: a mountain is a large land mass

0: a sea is a source of water for humans
1: a sea is a source of freshwater
2: a river is a source of water
3: an ocean is a source of water for residents

Table 9: Qualitative evaluation of traversed examples of Optimus (top) and Bert-TransCONV (addition QKV) (bottom).



Syntax Space Traversal

Input: *a sea is a source of sea water*
0: a river is synonymous with a coastline
1: a hurricane is composed of water vapor and dust
2: a hurricane is the source of most of water vapor in the atmosphere
3: hurricane is mainly made of water vapor
4: a hurricane is measuring the amount of water in an area

Table 10: Qualitative evaluation of traversed examples of Bert-TransCONV (addition QKV).

Additional References

Language VAE: literature review

Prior	Latent Space	Model Name	Encoder-Decoder
Fixed	Gaussian sentence	DG-VAE [10]	LSTM
		AdaVAE [9]	GPT2-GPT2
		Optimus [5]	Bert-GPT2
		LLaMaVAE [4]	sentenceT5-LlaMA
		(Bowman et al., 2015) [16]	LSTM
		DELLA [1]	GPT2-GPT2 or Transformer
	semantic-syntax	(Zhang et al., 2024) [3]	Bert-TransCONV-GPT2
		(Bao et al., 2019) [2]	LSTM
		(Chen et al., 2019) [8]	LSTM
		SIVAE [11]	LSTM
	vMF sentence	(Xu and Durrett, 2018) [15]	LSTM

Language VAE: literature review

Prior	Latent Space	Model Name	Encoder-Decoder
Trainable	hierarchical sequence	HRQ-VAE [12]	Transformer
	sequence	T5VQVAE [13]	T5
	single sentence	FlowPrior [14]	LSTM
	single sentence	DPrior [7]	Bert-GPT2
	hyperbolic	APo-VAE [6]	LSTM
	label-content	VAE-DPrior [17]	Bert-GPT2
	CVAE: Gaussian	(Fang et al., 2021)[18]	Transfomer
	CVAE: Gaussian	PPVAE [19]	LSTM
	CVAE: Gaussian	T-CVAE [20]	Transformer

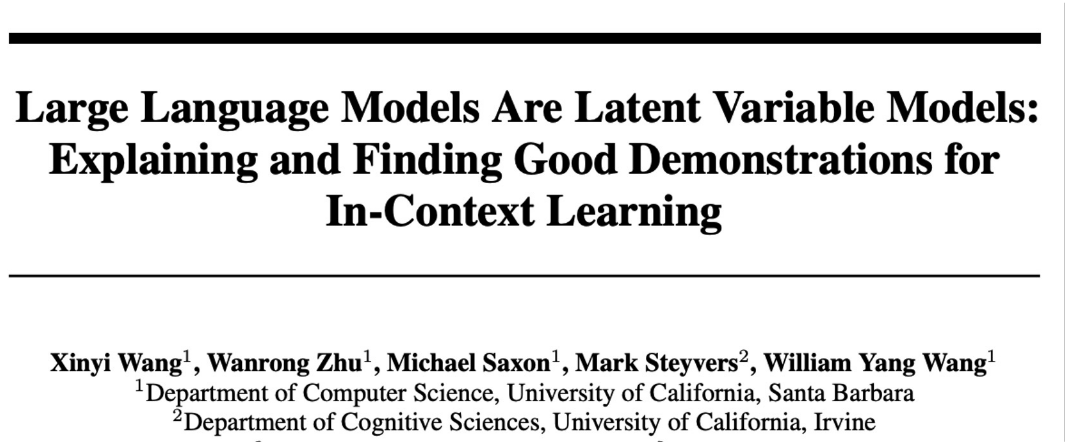
Language VAE: literature review

[1] Jinyi Hu, Xiaoyuan Yi, Wenhao Li, Maosong Sun, and Xing Xie. 2022. Fuse It More Deeply! A Variational Transformer with Layer-Wise Latent Variable Inference for Text Generation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 697–716, Seattle, United States. Association for Computational Linguistics.
[2] Yu Bao, Hao Zhou, Shujian Huang, Lei Li, Lili Mou, Olga Vechtomova, Xin-yu Dai, and Jiajun Chen. 2019. Generating Sentences from Disentangled Syntactic and Semantic Spaces. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6008–6019, Florence, Italy. Association for Computational Linguistics.
[3] Zhang, Y., Valentino, M., Carvalho, D. S., Pratt-Hartmann, I., & Freitas, A. (2023). Graph-Induced Syntactic-Semantic Spaces in Transformer-Based Variational AutoEncoders. *arXiv preprint arXiv:2311.08579*.
[4] Zhang, Y., Carvalho, D. S., Pratt-Hartmann, I., & Freitas, A. (2023). LlaMaVAE: Guiding Large Language Model Generation via Continuous Latent Sentence Spaces. *arXiv preprint arXiv:2312.13208*.
[5] Chunyuan Li, Xiang Gao, Yuan Li, Baolin Peng, Xiujun Li, Yizhe Zhang, and Jianfeng Gao. 2020. Optimus: Organizing Sentences via Pre-trained Modeling of a Latent Space. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4678–4699, Online. Association for Computational Linguistics.
[6] Shuyang Dai, Zhe Gan, Yu Cheng, Chenyang Tao, Lawrence Carin, and Jingjing Liu. 2021. APo-VAE: Text Generation in Hyperbolic Space. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 416–431, Online. Association for Computational Linguistics.
[7] Xianghong Fang, Jian Li, Lifeng Shang, Xin Jiang, Qun Liu, and Dit-Yan Yeung. 2022. Controlled Text Generation Using Dictionary Prior in Variational Autoencoders. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 97–111, Dublin, Ireland. Association for Computational Linguistics.
[8] Mingda Chen, Qingming Tang, Sam Wiseman, and Kevin Gimpel. 2019. A Multi-Task Approach for Disentangling Syntax and Semantics in Sentence Representations. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2453–2464, Minneapolis, Minnesota. Association for Computational Linguistics.
[9] Tu, H., Yang, Z., Yang, J., & Huang, Y. (2022). Adavae: Exploring adaptive gpt-2s in variational auto-encoders for language modeling. *arXiv preprint arXiv:2205.05862*.
[10] Zhang, J., Bai, J., Lin, C., Wang, Y., & Rong, W. (2022). Improving variational autoencoders with density gap-based regularization. *Advances in Neural Information Processing Systems*, *35*, 19470-19483.
[11] Xinyuan Zhang, Yi Yang, Siyang Yuan, Dinghan Shen, and Lawrence Carin. 2019. Syntax-Infused Variational Autoencoder for Text Generation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2069–2078, Florence, Italy. Association for Computational Linguistics.
[12] Tom Hosking, Hao Tang, and Mirella Lapata. 2022. Hierarchical Sketch Induction for Paraphrase Generation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2489–2501, Dublin, Ireland. Association for Computational Linguistics.
[13] Yingji Zhang, Danilo Carvalho, Marco Valentino, Ian Pratt-Hartmann, and Andre Freitas. 2024. Improving Semantic Control in Discrete Latent Spaces with Transformer Quantized Variational Autoencoders. In *Findings of the Association for Computational Linguistics: EACL 2024*, pages 1434–1450, St. Julian’s, Malta. Association for Computational Linguistics.
[14] Xiaoan Ding and Kevin Gimpel. 2021. FlowPrior: Learning Expressive Priors for Latent Variable Sentence Models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 3242–3258, Online. Association for Computational Linguistics.
[15] Xu, J., & Durrett, G. (2018). Spherical latent spaces for stable variational autoencoders. *arXiv preprint arXiv:1808.10805*.
[16] Bowman, S. R., Vilnis, L., Vinyals, O., Dai, A. M., Jozefowicz, R., & Bengio, S. (2015). Generating sentences from a continuous space. *arXiv preprint arXiv:1511.06349*.
[17] Zhuang Li, Lizhen Qu, Qiongkai Xu, Tongtong Wu, Tianyang Zhan, and Gholamreza Haffari. 2022. Variational Autoencoder with Disentanglement Priors for Low-Resource Task-Specific Natural Language Generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 10335–10356, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.
[18] Fang, L., Zeng, T., Liu, C., Bo, L., Dong, W., & Chen, C. (2021). Transformer-based conditional variational autoencoder for controllable story generation. *arXiv preprint arXiv:2101.00828*.
[19] Duan, Y., Xu, C., Pei, J., Han, J., & Li, C. (2019). Pre-train and plug-in: Flexible conditional text generation with variational auto-encoders. *arXiv preprint arXiv:1911.03882*.
[20] Wang, T., & Wan, X. (2019, August). T-CVAE: Transformer-based conditioned variational autoencoder for story completion. In *IJCAI* (pp. 5233-5239).

Trends

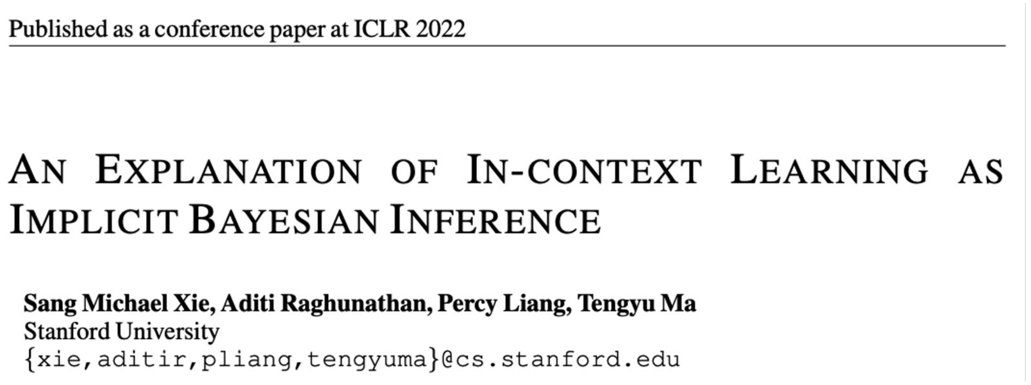
Closer integration with transformer interpretability

Mechanistic interpretability, disentanglement and transformer theory.



Large Language Models Are Latent Variable Models: Explaining and Finding Good Demonstrations for In-Context Learning

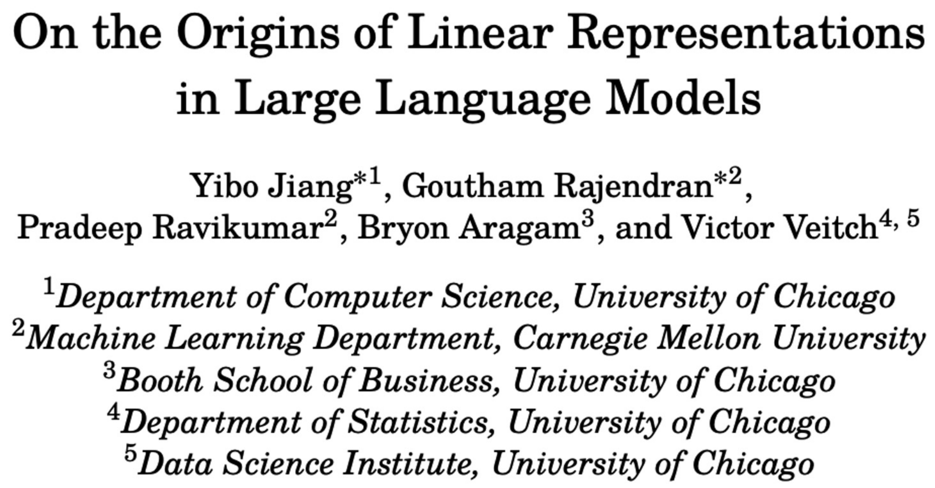
Xinyi Wang[1], **Wanrong Zhu**[1], **Michael Saxon**[1], **Mark Steyvers**[2], **William Yang Wang**[1]
[1]Department of Computer Science, University of California, Santa Barbara
[2]Department of Cognitive Sciences, University of California, Irvine



Published as a conference paper at ICLR 2022

AN EXPLANATION OF IN-CONTEXT LEARNING AS IMPLICIT BAYESIAN INFERENCE

Sang Michael Xie, Aditi Raghunathan, Percy Liang, Tengyu Ma
Stanford University
{xie,aditir,pliang,tengyuma}@cs.stanford.edu

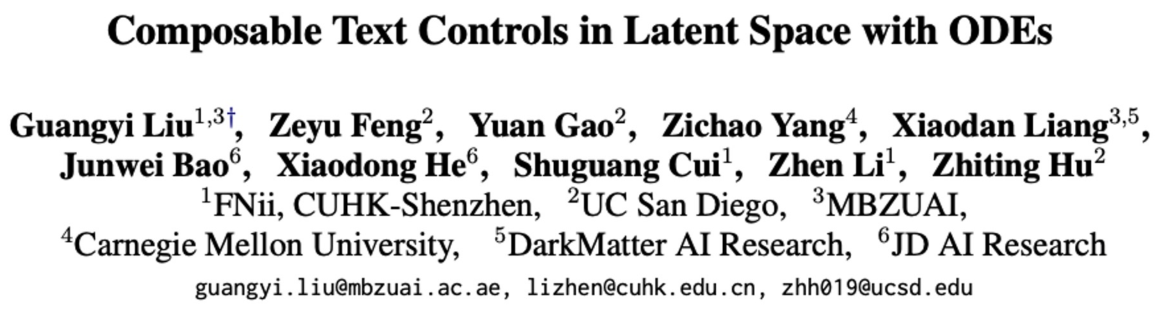


On the Origins of Linear Representations in Large Language Models

Yibo Jiang*[1], Goutham Rajendran*[2], Pradeep Ravikumar[2], Bryon Aragam[3], and Victor Veitch[4, 5]

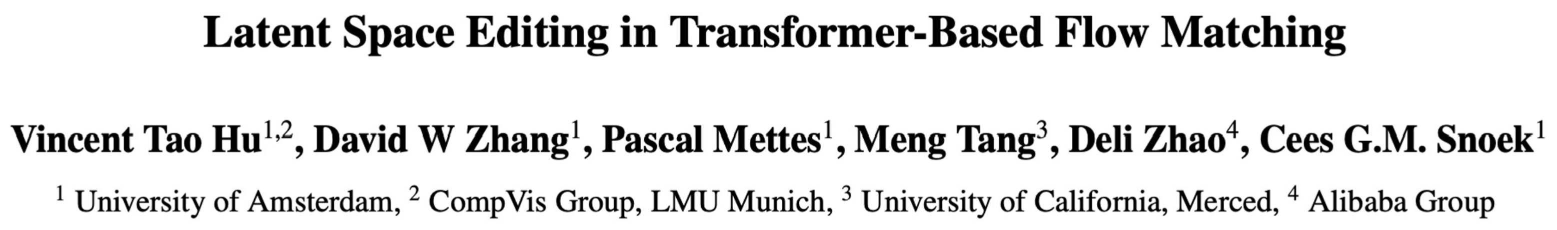
[1]*Department of Computer Science, University of Chicago*
[2]*Machine Learning Department, Carnegie Mellon University*
[3]*Booth School of Business, University of Chicago*
[4]*Department of Statistics, University of Chicago*
[5]*Data Science Institute, University of Chicago*

Multi-step semantic control as a dynamical systems model



Composable Text Controls in Latent Space with ODEs

Guangyi Liu[1,3†], Zeyu Feng[2], Yuan Gao[2], Zichao Yang[4], Xiaodan Liang[3,5], Junwei Bao[6], Xiaodong He[6], Shuguang Cui[1], Zhen Li[1], Zhiting Hu[2]
[1]FNii, CUHK-Shenzhen, [2]UC San Diego, [3]MBZUAI, [4]Carnegie Mellon University, [5]DarkMatter AI Research, [6]JD AI Research
guangyi.liu@mbzuai.ac.ae, lizhen@cuhk.edu.cn, zhh019@ucsd.edu



Latent Space Editing in Transformer-Based Flow Matching

Vincent Tao Hu[1,2], David W Zhang[1], Pascal Mettes[1], Meng Tang[3], Deli Zhao[4], Cees G.M. Snoek[1]
[1] University of Amsterdam, [2] CompVis Group, LMU Munich, [3] University of California, Merced, [4] Alibaba Group

continuous normalizing flow:

https://veryunknown.com/post/continuous-normalizing-flows/

https://jmtomczak.github.io/blog/18/18_fm.html

Conclusions

- Today we focused at the interface between formal semantic models and neural models.
- Emphasizing two dimensions: **interpretability** and **control**.
- We focused on mechanisms that allows for close semantic integration, with an emphasis on VAEs as an architecture.
- This allows for a complementary perspective to the current empirical norm: less task-oriented and more representation centered (fundamental linguistic and inference properties).

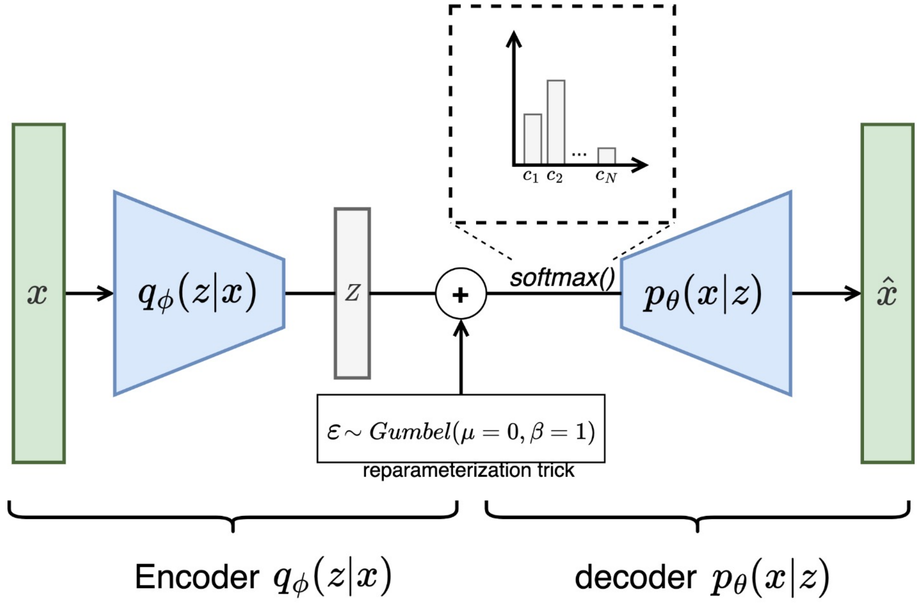
Appendix

Discretisation: 3. Gumbel Softmax trick

Recap: In VAE, stochastic sampling from a distribution will stop the deterministic backward propagation.

Therefore, we use reparameterization trick (i.e., sampling a noise following a standard Gaussian distribution).

Gumbel softmax trick: Now, we want to sample from a categorical distribution. We can also sample a noise from Gumbel distribution.



Discretisation: 3. Gumbel Softmax trick

Proof: *why adding a Gumbel noise is the same as sampling from a categorical distribution?*

The output of the encoder is $[x_1, \ldots, x_k, \ldots, x_N]$ where each element represents a category and has its corresponding probability $[p_1, \ldots, p_k, \ldots, p_N]$. Gumbel softmax trick adds a noise G_k to the output to get a new output $[z_1, \ldots, z_k, \ldots, z_N]$ where $z_k = x_k + G_k$ and $G_k \sim \mathit{Gumbel}(\mu = 0, \beta = 1)$ and choose the category with the biggest z_k. Therefore, we only need to prove: $p(z_k \geq z_i) = p_k$ where $i \neq k$.

$$
\begin{aligned}
& p(z_k \geq z_i) \\
& = p(z_1 \geq z_k) \times p(z_2 \geq z_k) \times \cdots \times p(z_N \geq z_k) \\
& = \prod_{i \neq k} e^{-e^{-(z_k - x_i)}}, \text{CDF:} e^{-e^{-\frac{x-\mu}{\beta}}} \\
& = \underbrace{\prod_{i \neq k} e^{-e^{-(z_k - x_i)}}}_{\text{constant to} z_k} \int e^{-(z_k - x_k) - e^{-(z_k - x_k)}} dz_k \\
& = \int \left(\prod_{i \neq k} e^{-e^{-(z_k - x_i)}} \right) e^{-(z_k - x_i) - e^{-(z_k - x_k)}} dz_k \\
& = \int \left(e^{-\sum_{i \neq k} e^{-(z_k - x_i)}} \right) \times \left(e^{-(z_k - x_k) - e^{-(z_k - x_k)}} \right) dz_k
\end{aligned}
$$

$$
\begin{aligned}
& = \int e^{\left(-\sum_{i \neq k} e^{-(z_k - x_i)}\right) - (z_k - x_k) - e^{-(z_k - x_k)}} dz_k \\
& = \int e^{\left[\left(-\sum_{i \neq k} e^{-(z_k - x_i)}\right) - e^{-(z_k - x_k)}\right] - (z_k - x_k)} dz_k \\
& = \int e^{\left(-\sum_i e^{-(z_k - x_i)}\right) - (z_k - x_k)} dz_k \\
& = \int e^{\left(-\sum_i e^{x_i} \times e^{-z_k}\right) - (z_k - x_k)} dz_k \\
& = \int e^{\left(-\sum_i e^{x_i}\right) \times e^{-z_k} - z_k + x_k} dz_k \\
& = \int e^{\left(-e^{-z_k + In\left(\sum_i e^{x_i}\right)}\right) - z_k + x_k} dz_k
\end{aligned}
$$

$$
\begin{aligned}
& = \int e^{\left[\left(-e^{-(z_k - In(\sum_i e^{x_i}))}\right) - (z_k - In\left(\sum_i e^{x_i}\right)) - In\left(\sum_i e^{x_i}\right) + x_k\right]} dz_k \\
& = e^{\left[-In\left(\sum_i e^{x_i}\right) + x_k\right]} \int e^{\left[\left(-e^{-(z_k - In(\sum_i e^{x_i}))}\right) - (z_k - In\left(\sum_i e^{x_i}\right))\right]} dz_k \\
& = e^{-In\left(\sum_i e^{x_i}\right)} \times e^{x_k} \times \int e^{\left[\left(-e^{-(z_k - In(\sum_i e^{x_i}))}\right) - (z_k - In\left(\sum_i e^{x_i}\right))\right]} dz_k \\
& = \frac{e^{x_k}}{\sum_i e^{x_i}} \times \underbrace{\int e^{\left[\left(-e^{-(z_k - In(\sum_i e^{x_i}))}\right) - (z_k - In\left(\sum_i e^{x_i}\right))\right]} dz_k}_{\text{integration of pdf}} \\
& = \frac{e^{x_k}}{\sum_i e^{x_i}}
\end{aligned}
$$