Continuous Representations For Efficient Language Models

Vlad Niculae
Language Technology Lab
Informatics Institute, U. of Amsterdam

BamNLP, 2026

Structured representations and optimization in NLP

or, StroopNLP

discrete structure
(Niculae et al., 2018)
(Niculae et al., 2020)
(Correia et al., 2020)
(Niculae et al., 2025)

continuous structure
(Troshin et al., 2023)
(Tokarchuk et al., 2025a)
(Tokarchuk et al., 2026)

contributes to

controllability
(Troshin et al., 2025a)
(Troshin et al., 2025b)

using long contexts
(Mohammed et al., 2024)
(Mohammed et al., 2025)
(Mohammed et al., 2026)

retrieving similar contexts
(Nachesa et al., 2025)
(Tokarchuk et al., 2025b)

Structured representations and optimization in NLP

or, StroopNLP

discrete structure
(Niculae et al., 2018)
(Niculae et al., 2020)
(Correia et al., 2020)
(Niculae et al., 2025)

continuous structure
(Troshin et al., 2023)
(Tokarchuk et al., 2025a)
(Tokarchuk et al., 2026)

contributes to

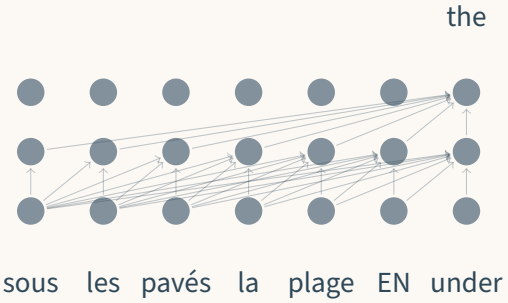
controllability
(Troshin et al., 2025a)
(Troshin et al., 2025b)

using long contexts
(Mohammed et al., 2024)
(Mohammed et al., 2025)
(Mohammed et al., 2026)

retrieving similar contexts
(Nachesa et al., 2025)
(Tokarchuk et al., 2025b)

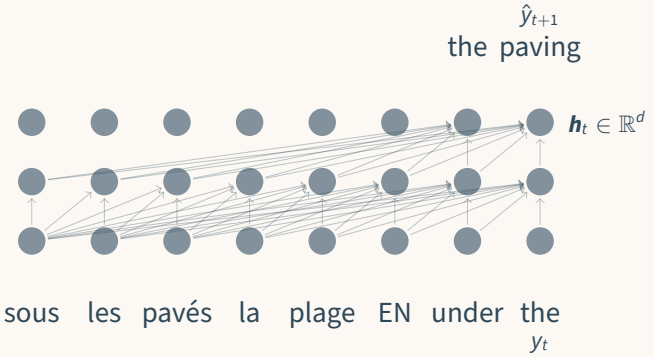
Transformer LM: Next-word prediction

(Vaswani et al., 2017)



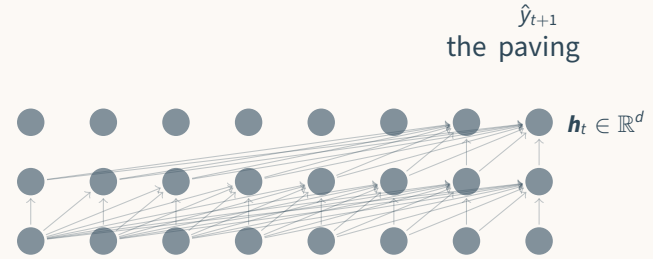
Transformer LM: Next-word prediction

(Vaswani et al., 2017)



Transformer LM: Next-word prediction

(Vaswani et al., 2017)

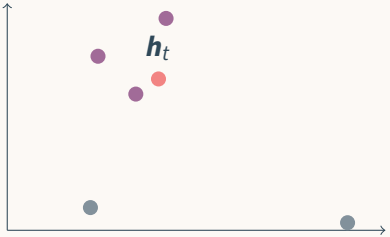


$$\boldsymbol{h}_t = f_\theta(y_1, \ldots, y_t)$$

$\boldsymbol{h}_t$ is a good representation of the entire context $y_{1,\ldots,t}$

$$P(y_{t+1} = \text{paving}) = \frac{\exp\langle \boldsymbol{h}_t, \boldsymbol{w}_{\text{paving}} \rangle}{\sum_{v \in \mathcal{V}} \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_v \rangle}$$

Retrieving contexts by hidden state



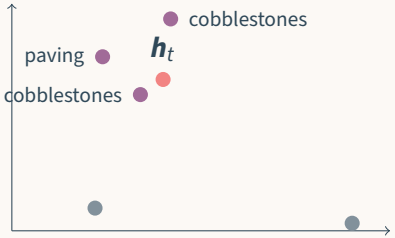
k-nearest neighbors by $d(\boldsymbol{h}_t, \boldsymbol{h})$

Helps understand decisions and mistakes

d	$y_{1,\ldots,t}$	y_{t+1}
.1	Le sable stabilise les pavés autobloquants EN The sand stabilizes the interlocking	**paving**
.8	Ils se promenent sur les pavés pour nostalgie. EN They take a walk on the	**cobblestones**
.9	Ces pavés ont l'air d'avoir été nettoyés EN These	**cobblestones**

k-nearest neighbors language models

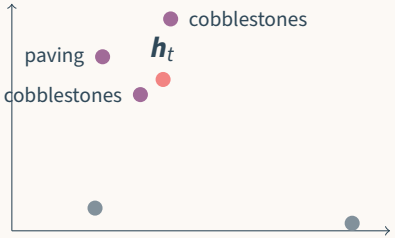
(Khandelwal et al., 2020; Khandelwal et al., 2021)



*k*NN classifier: predict most voted word from similar contexts.

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



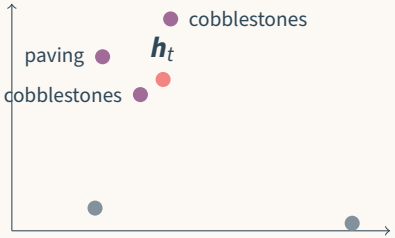
*k*NN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with *k*NN

- better accuracy
- domain adaptation
- but, high compute cost

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



*k*NN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with *k*NN

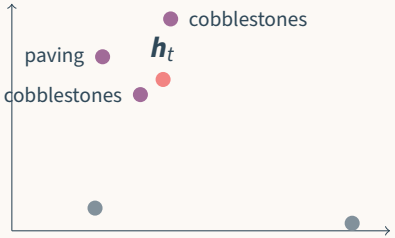
- better accuracy
- domain adaptation
- but, high compute cost

$$p_{\text{LM}}(y_{t+1} = y \mid y_{1:t}) \propto \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle$$

(linear in $\boldsymbol{h}_t$)

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



*k*NN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with *k*NN

- better accuracy
- domain adaptation
- but, high compute cost

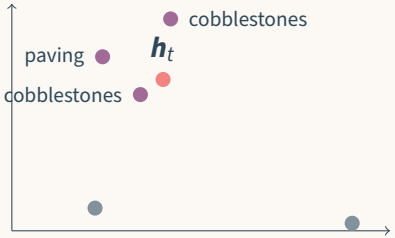
$$p_{\text{LM}}(y_{t+1} = y \mid y_{1:t}) \propto \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle \quad \text{(linear in } \boldsymbol{h}_t\text{)}$$

$$p_{\text{kNN}}(y_{t+1} = y \mid y_{1:t}) \propto \sum_{\boldsymbol{h} \in \text{neighbors of } \boldsymbol{h}_t \text{ with label } y} \text{vote}(\boldsymbol{h}, \boldsymbol{h}_t) \quad \text{(non-linear)}$$

vote can be $\{0, 1\}$ or distance-based; papers use latter

k-nearest neighbors language models

(Khandelwal et al., 2020; Khandelwal et al., 2021)



*k*NN classifier: predict most voted word from similar contexts.

kNN-LM, kNN-MT: augmenting transformers with *k*NN

- better accuracy
- domain adaptation
- but, high compute cost

$$p_{\text{LM}}(y_{t+1} = y \mid y_{1:t}) \propto \exp\langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle \quad \text{(linear in } \boldsymbol{h}_t\text{)}$$

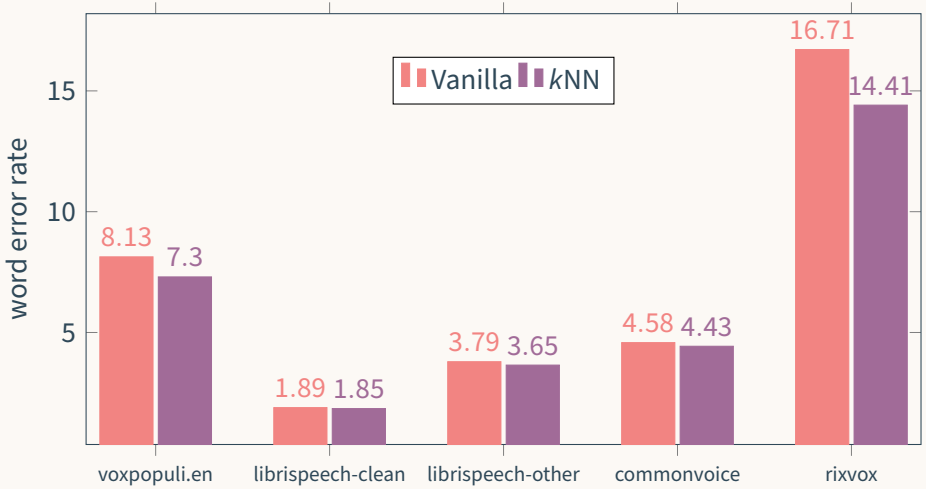
$$p_{\text{kNN}}(y_{t+1} = y \mid y_{1:t}) \propto \sum_{\boldsymbol{h} \in \text{neighbors of } \boldsymbol{h}_t \text{ with label } y} \text{vote}(\boldsymbol{h}, \boldsymbol{h}_t) \quad \text{(non-linear)}$$

$$p = (1 - \lambda) p_{\text{LM}} + \lambda p_{\text{kNN}}$$

vote can be $\{0, 1\}$ or distance-based; papers use latter

Aside/Bonus: kNN-Whisper for ASR

The Whisper model for ASR is also a conditional LM.
Augmenting it with *k*NN helps! (Nachesa et al., 2025)

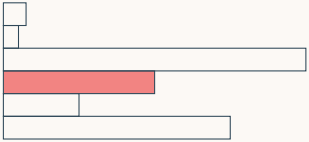


Continuous-output LM

(Kumar et al., 2019)

Standard LM objective: *classification*

$$\log P(y_{t+1} = y \mid y_{1:t}) = \langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle - \log \sum_{y'} \exp \langle \boldsymbol{h}_t, \boldsymbol{w}_{y'} \rangle$$

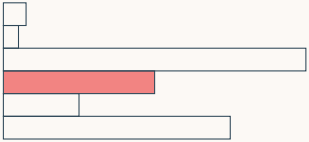


Continuous-output LM

(Kumar et al., 2019)

Standard LM objective: *classification*

$$\log P(y_{t+1} = y \mid y_{1:t}) = \langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle - \log \sum_{y'} \exp \langle \boldsymbol{h}_t, \boldsymbol{w}_{y'} \rangle$$



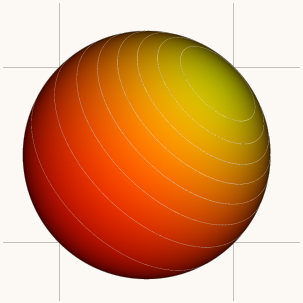
Continuous LM objective: *regression*

$$\log P(y_{t+1} = \boldsymbol{w}_y \mid y_{1:t}) = \langle \boldsymbol{h}_t, \boldsymbol{w}_y \rangle - \log C$$

Langevin probabilistic model on $\mathbb{S}_d$

train: encourage $\boldsymbol{h}_t$ close to $\boldsymbol{w}_y$

test: retrieve nearest-neighbor embeddings



Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

model and $\boldsymbol{w}$	ro-en BLEU$_{\uparrow}$	de-en BLEU$_{\uparrow}$
discrete	31.7	39.3
continuous, pretrained $\boldsymbol{w}$	29.0	32.9

(Romanian-English WMT16, transformer-base, embedding size 128.)

(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

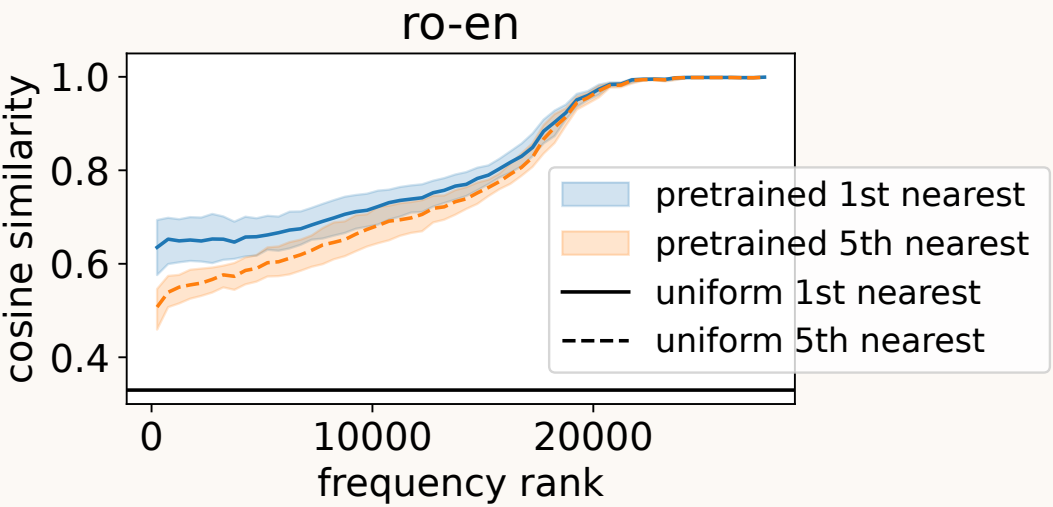
model and $\boldsymbol{w}$	ro-en BLEU$_{\uparrow}$	de-en BLEU$_{\uparrow}$
discrete	31.7	39.3
continuous, pretrained $\boldsymbol{w}$	29.0	32.9
continuous, random unif $\boldsymbol{w}$	28.8	33.9

(Romanian-English WMT16, transformer-base, embedding size 128.)

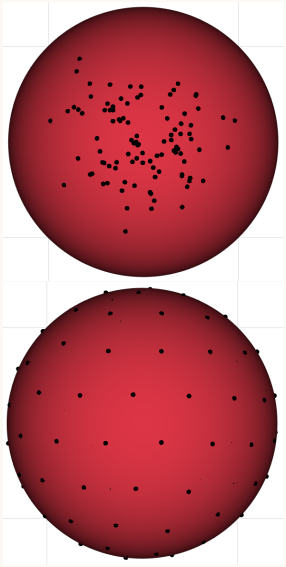
(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

Analysis: geometry, distribution by frequency



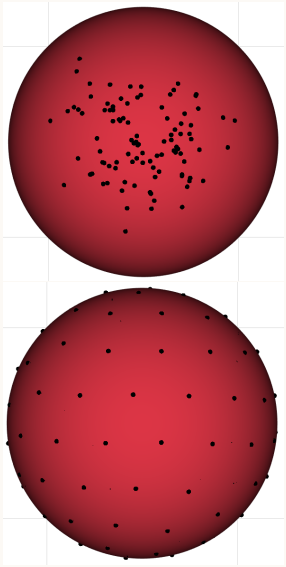
Dispersion on the sphere



Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{W}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Dispersion on the sphere



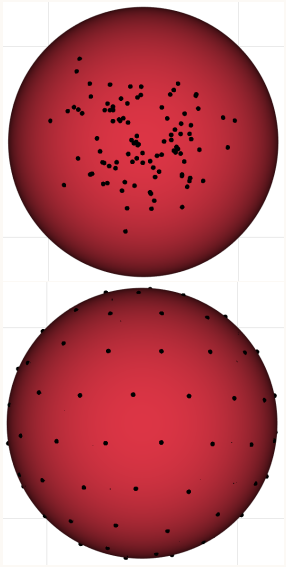
Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{W}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Measures:

- Minimum distance: $\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$
- Spherical variance: $1 - \|\boldsymbol{\mu}\|$ where $\boldsymbol{\mu} = \sum_i \boldsymbol{w}_i / n$

Dispersion on the sphere



Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{W}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

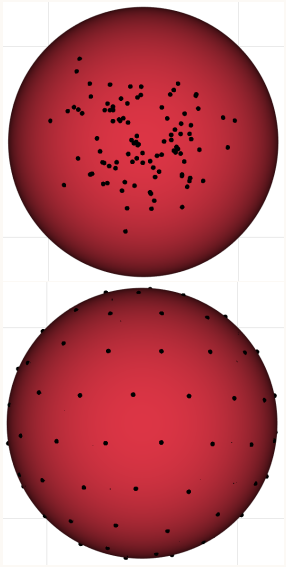
Measures:

- Minimum distance: $\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$
- Spherical variance: $1 - \|\boldsymbol{\mu}\|$ where $\boldsymbol{\mu} = \sum_i \boldsymbol{w}_i / n$

Related problems:

- Thomson dispersion: electrostatic charges
- Spherical codes (quantizing the sphere)
- Sphere packing / the kissing problem

Dispersion on the sphere



Optimal dispersion according to Tammes (1930),

$$\max_{\boldsymbol{W}} \min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Measures:

- Minimum distance: $\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$
- Spherical variance: $1 - \|\boldsymbol{\mu}\|$ where $\boldsymbol{\mu} = \sum_i \boldsymbol{w}_i / n$

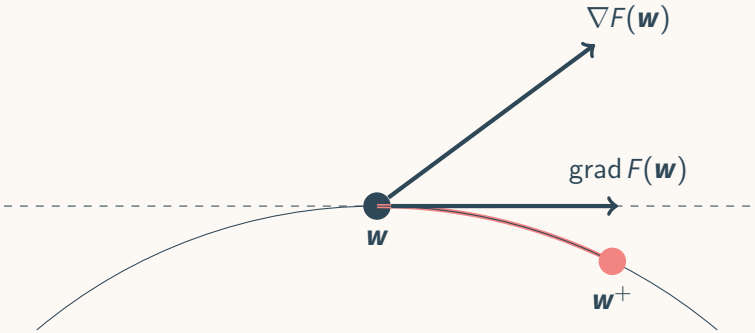
Related problems:

- Thomson dispersion: electrostatic charges
- Spherical codes (quantizing the sphere)
- Sphere packing / the kissing problem

Despite symmetry, exact solution generally unknown.
We want to trade off dispersion with a *task loss*:

$$L(\boldsymbol{W}) + \alpha R(\boldsymbol{W})$$

Riemannian optimization on $\mathbb{S}_m$



Scary math and notation but simple intuition:
walk along the surface in the direction of the gradient.

$$\boldsymbol{w}^+ = \operatorname{Exp}_{\boldsymbol{w}}(\operatorname{grad} F(\boldsymbol{w}))$$

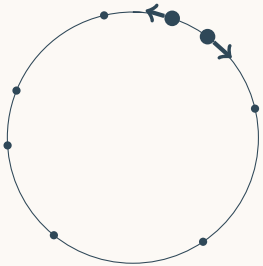
Useful measures are not necessarily optimization-friendly

min. dist. (Tammes objective)

$$R_{\text{Tammes}}(\boldsymbol{W}) = -\min_{i \neq j} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

$$\operatorname{grad}_{\boldsymbol{w}_k} R_{\text{Tammes}}(\boldsymbol{W}) = 0$$

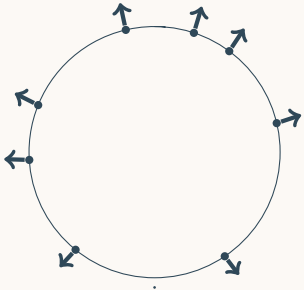
for almost all k except the two closest ones.



spherical variance

$$R_{\text{var}}(\boldsymbol{W}) = 1 - \left\| \sum_k \boldsymbol{w}_k / n \right\|$$

Eucl. gradients are normal;
so all Riemannian gradients are 0.



Common approaches in literature

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\text{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\text{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\text{MHE},k} = \frac{1}{n(n-1)}\sum_{i\neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\text{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\text{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\text{MHE},k} = \frac{1}{n(n-1)}\sum_{i\neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

Quadratic complexity $O(mn^2)$

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\text{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\text{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\text{MHE},k} = \frac{1}{n(n-1)}\sum_{i\neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

Quadratic complexity $O(mn^2)$

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\text{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\text{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

We show: $R_{\text{Tammes}} \leq R_{\text{MM}} \leq R_{\text{KoLeo}} - 1$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

$$R_{\text{MHE},k} = \frac{1}{n(n-1)}\sum_{i\neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

Common approaches in literature

Quadratic complexity $O(mn^2)$

per-point min distance:

(Mettes et al., 2019; Z. Wang et al., 2021)

$$R_{\text{MM}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

(Sablayrolles et al., 2019; Leonenko, 1987)

$$R_{\text{KoLeo}}(\boldsymbol{W}) = -\frac{1}{n}\sum_i \log \min_{j\neq i} d(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

We show: $R_{\text{Tammes}} \leq R_{\text{MM}} \leq R_{\text{KoLeo}} - 1$

kernel style:

minimum hyperspherical energy: (Liu et al., 2018; Gautam et al., 2013; Liu et al., 2021; T. Wang et al., 2020; Thomson, 1904)

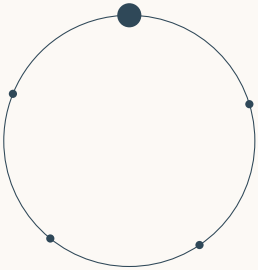
$$R_{\text{MHE},k} = \frac{1}{n(n-1)}\sum_{i\neq j} k(\boldsymbol{w}_i, \boldsymbol{w}_j)$$

We show: $R_{\text{MHE},k}$ is the maximum mean discrepancy between the empirical measure $\boldsymbol{W}$ and the uniform measure on the sphere.

Sliced dispersion

(Bonet et al., 2023; Tokarchuk et al., 2025a)

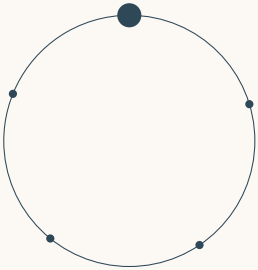
Observation: on $\mathbb{S}_1$, optimally dispersed configurations are some rotation of:



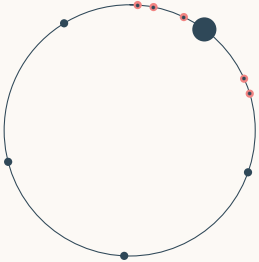
Sliced dispersion

(Bonet et al., 2023; Tokarchuk et al., 2025a)

Observation: on $\mathbb{S}_1$, optimally dispersed configurations are some rotation of:



Given some suboptimal configuration, we can define its distance to the closest dispersed one:



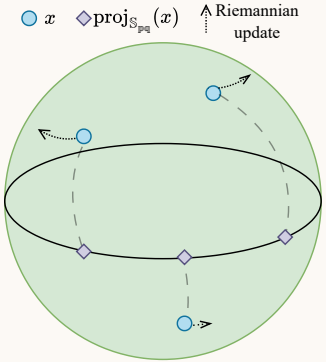
Intuition: sort angles;
map 1-to-1 around mean.
Computation: $O(n + \text{sort}(n))$.

Sliced dispersion on $\mathbb{S}_m$

Slicing along any great circle should preserve dispersion on average.

R_{sliced}:
expectation over great circles (p, q)
of distance to optimal 1d configuration

Complexity:
$O(mn)$ to project to great circle
($O(n)$ if axis-aligned)
$+O(\text{sort}(n))$ to disperse.



Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

model and $\boldsymbol{w}$	ro-en BLEU$_{\uparrow}$	de-en BLEU$_{\uparrow}$
discrete	31.7	39.3
continuous, pretrained $\boldsymbol{w}$	29.0	32.9
continuous, random unif $\boldsymbol{w}$	28.8	33.9
continuous, dispersed	**30.1**	**36.6**

(Romanian-English WMT16, transformer-base, embedding size 128.)

(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

Continuous-output machine translation: embedding choice

(Tokarchuk et al., 2024; Tokarchuk et al., 2026)

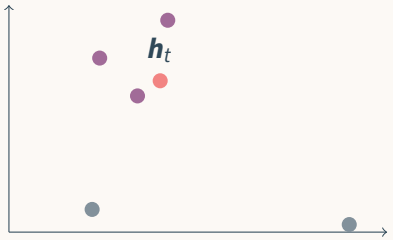
model and $\boldsymbol{w}$	ro-en BLEU$_{\uparrow}$	de-en BLEU$_{\uparrow}$
discrete	31.7	39.3
discrete dispersed	32.4	39.2
continuous, pretrained $\boldsymbol{w}$	29.0	32.9
continuous, random unif $\boldsymbol{w}$	28.8	33.9
continuous, dispersed	**30.1**	**36.6**

(Romanian-English WMT16, transformer-base, embedding size 128.)

(English-German WMT19, transformer-big, embedding size 1024.)

(bold best continuous model)

From CoNMT to kNN-MT



	CoNMT	kNN-MT
	Both: retrieve next word through lookup of nearest key vector	
keys	word embeddings on $\mathbb{S}_m$	context representations on $\mathbb{R}^m$
$n \approx$	10^4	10^5–10^7

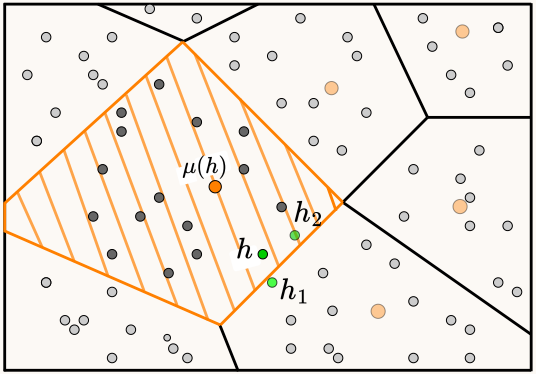
Due to higher n, we must use efficient approximate nearest neighbors methods.

Approximate nearest neighbor retrieval

Inverse vector file + product quantization (IVFPQ, Johnson et al., 2019): a SOTA method

IVF: cluster the keys using *k*-means; treat each Voronoi cell as a separate smaller (centered) data store.

PQ: inside each cell, split the *m* dimensions into subspaces and quantize them to 8 bit per key.



Approximate nearest neighbor retrieval

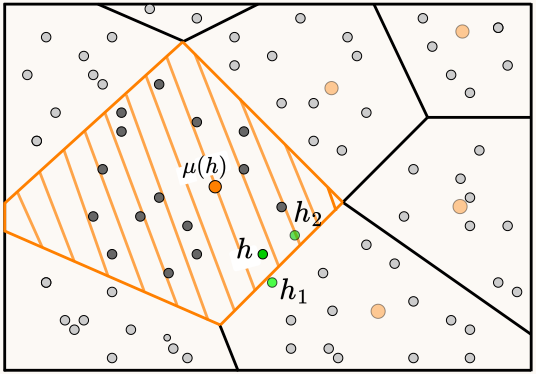
Inverse vector file + product quantization (IVFPQ, Johnson et al., 2019): a SOTA method

IVF: cluster the keys using k-means; treat each Voronoi cell as a separate smaller (centered) data store.

PQ: inside each cell, split the m dimensions into subspaces and quantize them to 8 bit per key.

At lookup time:

- find the n_{probes} closest centroids
- search exhaustively within their Voronoi cells.



Approximate nearest neighbor retrieval

Inverse vector file + product quantization (IVFPQ, Johnson et al., 2019): a SOTA method

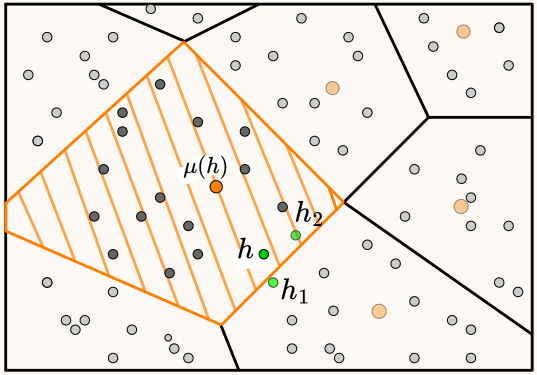
IVF: cluster the keys using k-means; treat each Voronoi cell as a separate smaller (centered) data store.

PQ: inside each cell, split the m dimensions into subspaces and quantize them to 8 bit per key.

At lookup time:

- find the n_{probes} closest centroids
- search exhaustively within their Voronoi cells.

Hypothesis: IVFPQ performs better with dispersed keys.



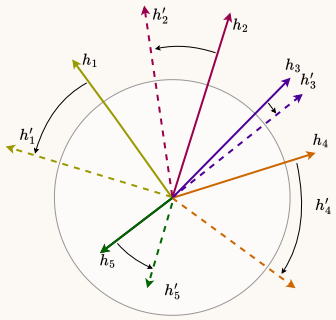
Angular dispersion and isotropy

(Tokarchuk et al., 2025b)

Given a set of hidden states $\boldsymbol{h} \in \mathbb{R}^m$, consider the dispersion of their *directions*

$$\boldsymbol{h}/\|\boldsymbol{h}\| \in \mathbb{S}_m.$$

Intuition: Distribution of $\boldsymbol{h}$ spherically symmetric around origin implies angular dispersion.



Synthetic validation

Generative process.

Draw $n = 10M$ directions from mixture of 5 Power Sphericals on $\mathbb{S}_{128}$, with varying concentration.

Assign to each direction a uniform length on $[1, 100]$

Synthetic validation

Generative process.

Draw $n = 10M$ directions from mixture of 5 Power Sphericals on $\mathbb{S}_{128}$, with varying concentration.

Assign to each direction a uniform length on $[1, 100]$

Evaluation.

Fit IVFPQ datastore with 2048 cells, 8 neighbors, batch size 10, $n_{\text{probes}} = 32$, and measure over $10K$ random queries:

- imbalance factor $IF = K \sum_{i=1}^{K} \left(\frac{n_i}{n}\right)^2$
- throughput (requests per second)

Synthetic validation

Generative process.

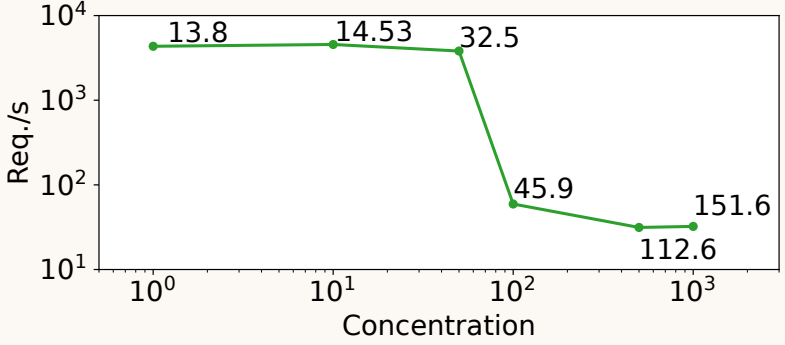
Draw $n = 10M$ directions from mixture of 5 Power Sphericals on $\mathbb{S}_{128}$, with varying concentration.

Assign to each direction a uniform length on $[1, 100]$

Evaluation.

Fit IVFPQ datastore with 2048 cells, 8 neighbors, batch size 10, $n_{\text{probes}} = 32$, and measure over $10K$ random queries:

- imbalance factor $IF = K \sum_{i=1}^{K} \left(\frac{n_i}{n}\right)^2$
- throughput (requests per second)



Can we make hidden states dispersed?

Not parameters, but network outputs:
$\boldsymbol{h}_t = f_\theta(y_1, \ldots, y_t)$

Transformers are trained on minibatches,
we don't see all contexts at once.

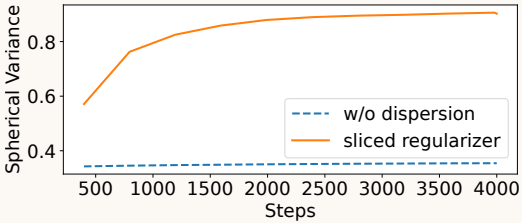
Doubly-stochastic sliced dispersion:
Each training update disperses a subset of
the collection along a random great circle.

Can we make hidden states dispersed?

Not parameters, but network outputs:
$\boldsymbol{h}_t = f_\theta(y_1, \ldots, y_t)$

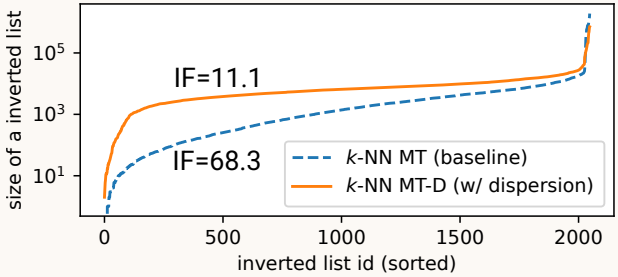
Transformers are trained on minibatches, we don't see all contexts at once.

Doubly-stochastic sliced dispersion:
Each training update disperses a subset of the collection along a random great circle.



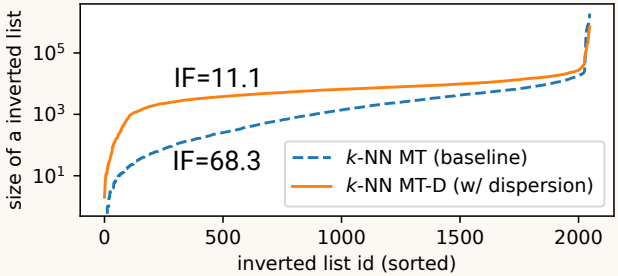
Dispersed kNN-MT

Romanian-English WMT16, transformer-base, $m = 128$.



Dispersed kNN-MT

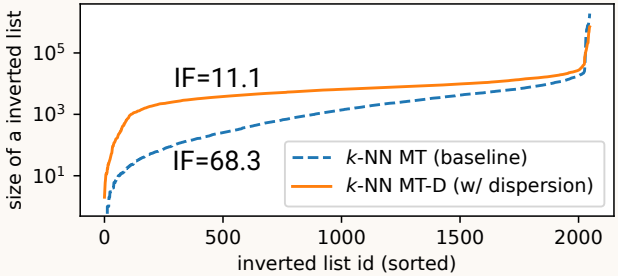
Romanian-English WMT16, transformer-base, $m = 128$.



model	**#probes**	**BLEU**$_{(\uparrow)}$	**COMET**$_{(\uparrow)}$	**tok/s**$_{(\uparrow)}$
baseline	-	31.5	78.95	75
*k*NN MT	32	32.4	79.89	12
*k*NN MT-D	32	32.6	79.91	53

Dispersed kNN-MT

Romanian-English WMT16, transformer-base, $m = 128$.

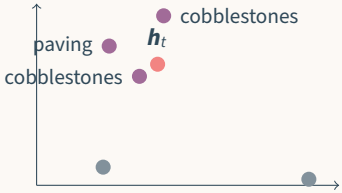


model	#probes	BLEU(↑)	COMET(↑)	tok/s(↑)
baseline	-	31.5	78.95	75
*k*NN MT	32	32.4	79.89	12
*k*NN MT	8	32.2	79.69	28
*k*NN MT-D	32	32.6	79.91	53
*k*NN MT-D	8	**32.6**	**79.93**	**63**

Continuous representations for efficient language models

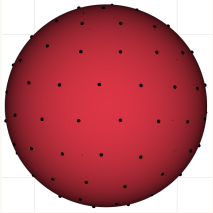
***k*NN language models:**

a powerful model, more adaptable, more interpretable ($\rightarrow$ also for speech recognition).



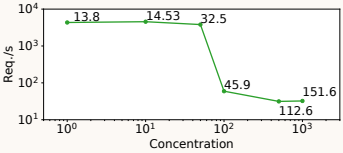
***dispersion*:**

a centuries-old problem still relevant in modern ML.



***substantial improvements*:**

speedups and accuracy boost thanks to going back to basics.



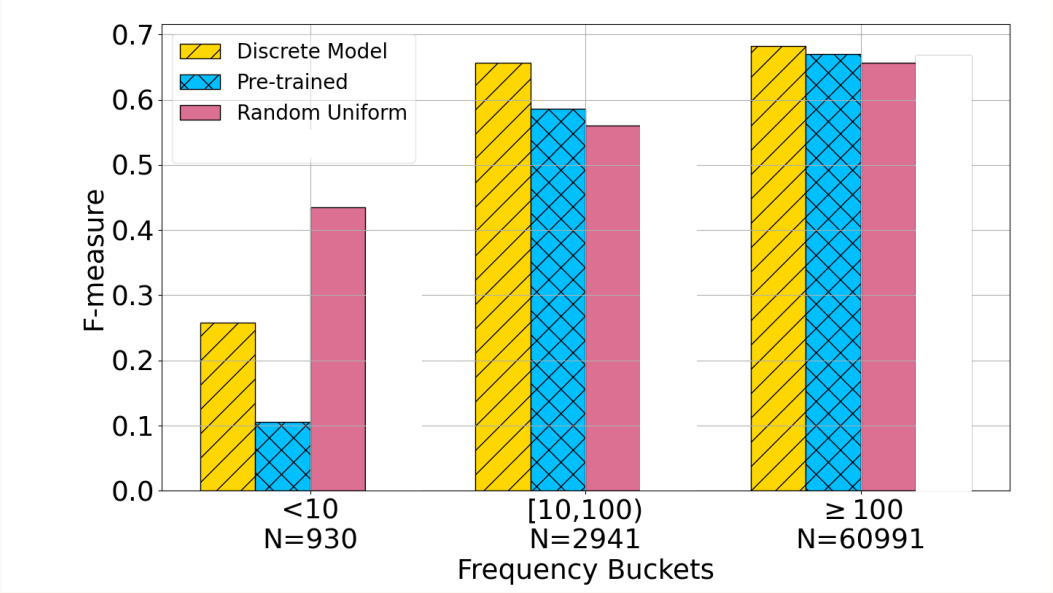
Immediate perspectives:

- Seek alternatives to IVFPQ built directly for dispersion.
- Explore the spectrum between CoNMT and *k*NN-MT.

Long-term perspectives: Geometry of representations from single tokens to phrases; “reasoning” over such structured / searchable spaces.

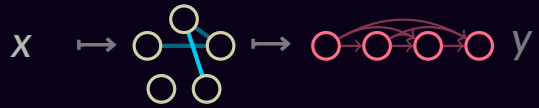
extra

Analysis: errors by frequency



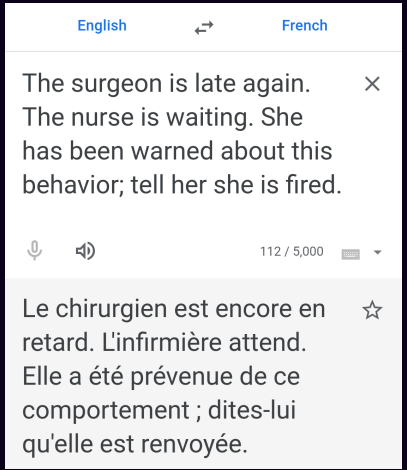
ConStructPlan

controllability through structured plans for language generation



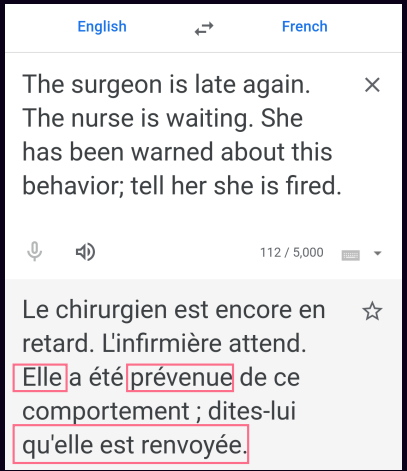
Vlad Niculae
assistant professor, University of Amsterdam

Natural language generation



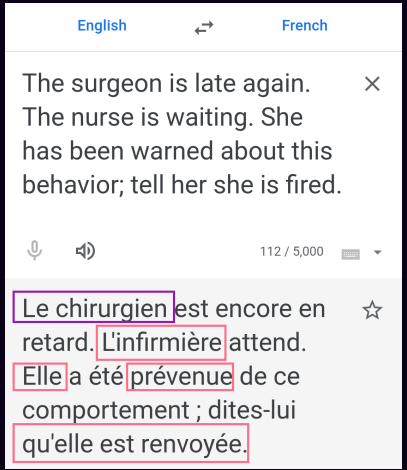
Large language models: increasingly powerful.

Natural language generation



Large language models: increasingly powerful.

Natural language generation

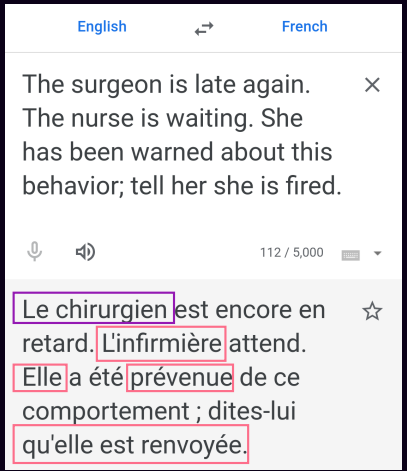


Large language models: increasingly powerful.

Yet terribly self-inconsistent.

Can't control global attributes. Fundamental issue, hampers trust.

Natural language generation



Large language models: increasingly powerful.

Yet terribly self-inconsistent.

Can't control global attributes. Fundamental issue, hampers trust.

Not an isolated example but a pattern (Mohammed and Niculae, 2025).

Natural language generation

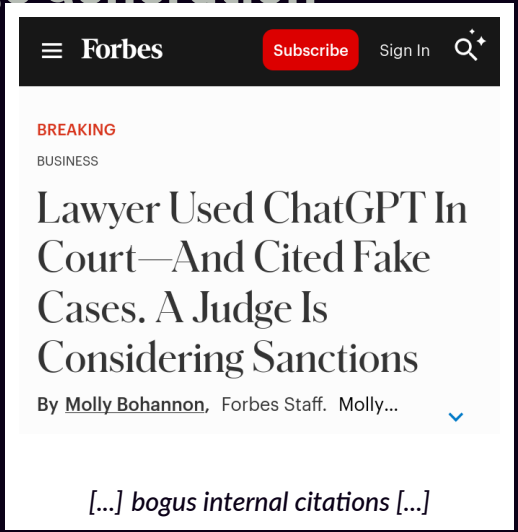

PBS NEWS WEEKEND

How language translation technology is jeopardizing Afghan asylum-seekers

By — Ali Rogin
By — Andrew Corkery

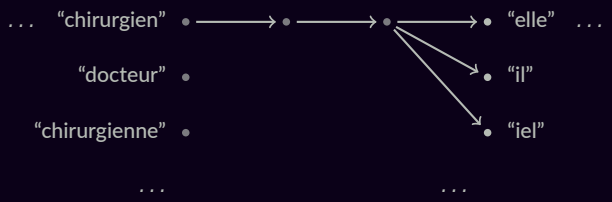
[... a] translation done by a machine mistranslated, actually, "I" as "we". [...] enough for the judge to reject the case. [...]

Natural language generation



current language models:

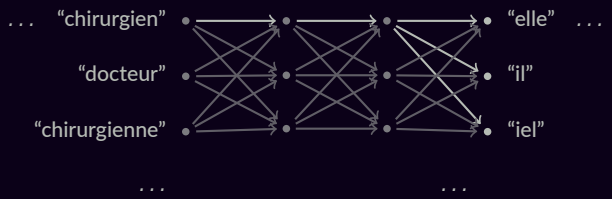
predict next word
left-to-right, locally.



current language models:

predict next word
left-to-right, locally.

can't "backtrack" to satisfy constraints:
exponentially many possible paths.



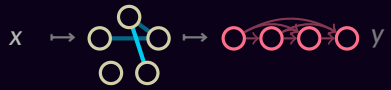
current language models:

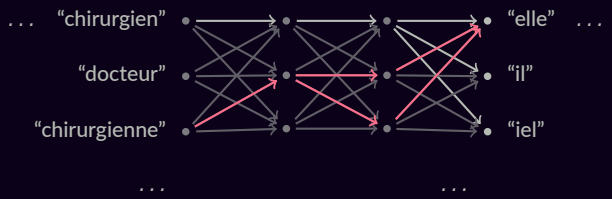
predict next word
left-to-right, locally.

can't "backtrack" to satisfy constraints:
exponentially many possible paths.

ConStructPlan

resolve global matters
in a planning stage.





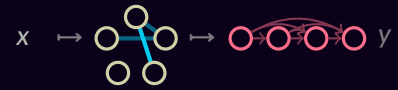
current language models:

predict next word
left-to-right, locally.

can't "backtrack" to satisfy constraints:
exponentially many possible paths.

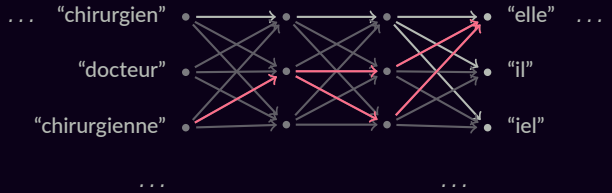
ConStructPlan

resolve global matters
in a planning stage.

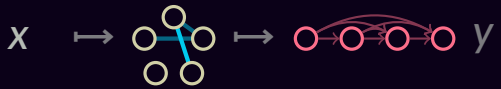


planning restricts the output space;
allows L2R architectures to work.

control and intervene at plan level.

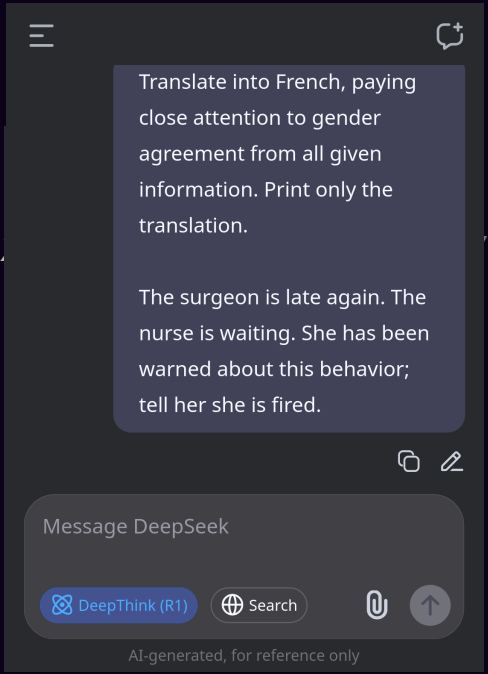


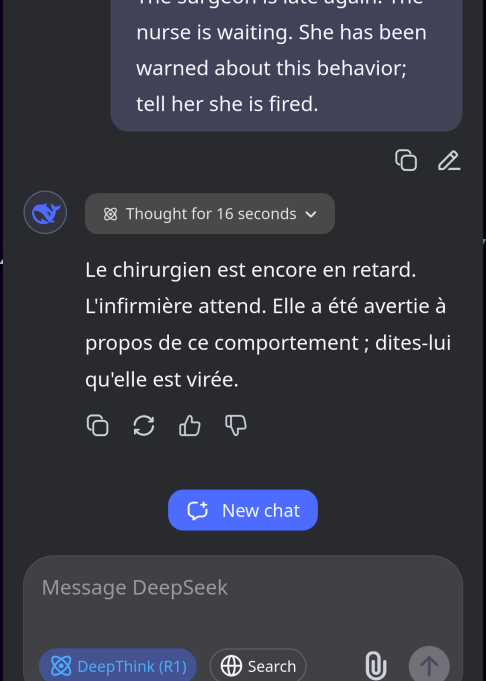
ConStructPlan



What should plans look like?

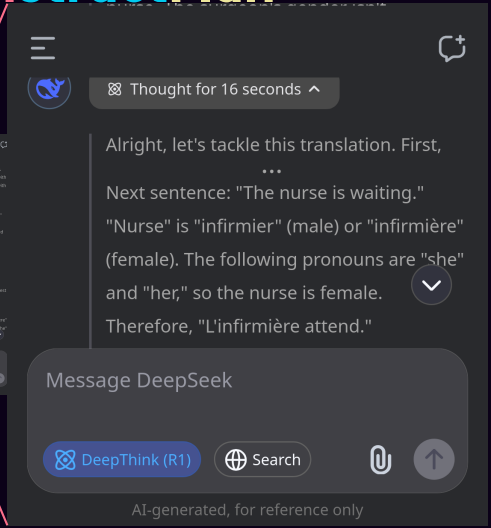
Recent shake-ups in the field tell us
what they should *not* look like.


Translate into French, paying close attention to gender agreement from all given information. Print only the translation.
The surgeon is late again. The nurse is waiting. She has been warned about this behavior; tell her she is fired.
Message DeepSeek
DeepThink (R1)
Search
AI-generated, for reference only


nurse is waiting. She has been warned about this behavior; tell her she is fired.
Thought for 16 seconds
Le chirurgien est encore en retard. L'infirmière attend. Elle a été avertie à propos de ce comportement ; dites-lui qu'elle est virée.
New chat
Message DeepSeek
DeepThink (R1)
Search

ConStructPlan

X



Innovative aspects of ConStructPlan

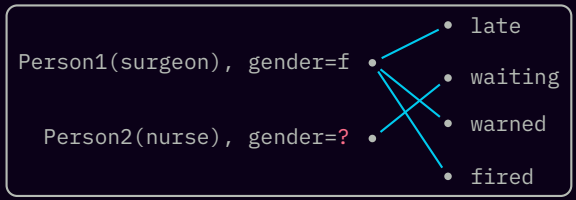
WP1.
latent modeling of
structured plans

Plans should encode exactly what's needed for generating the output correctly.

Innovative aspects of ConStructPlan

WP1.
latent modeling of
structured plans

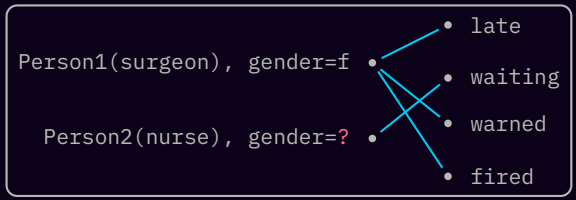
Plans should encode exactly what's needed for generating the output correctly.



Innovative aspects of ConStructPlan

WP1.
latent modeling of
structured plans

Plans should encode exactly what's needed for generating the output correctly.

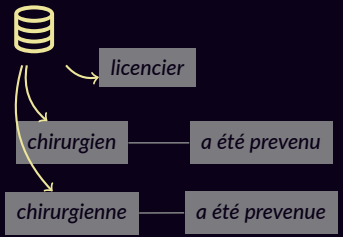


Challenging desiderata

compact, efficient to process (sparsity)
adaptive to the query (optimization)
hierarchical interpretable form (structure)

Innovative aspects of ConStructPlan

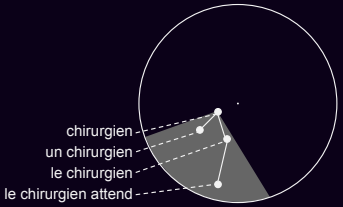
WP2.
planning with
retrieved fragments



Improve generation with retrieval from datasets.
Innovation: modularity;
retrieve globally, at planning stage.

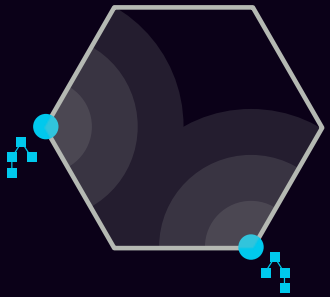
Challenges:

- store & search overlapping, variable-length chunks.
- combine chunks into outputs.



Innovative aspects of ConStructPlan

WP3.
plan arithmetic
for **controllability**

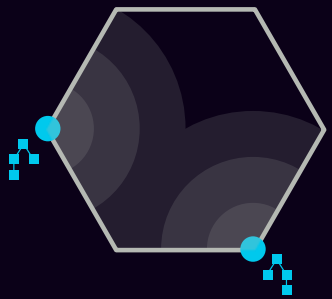


Operations over plans:

- guide a plan towards increasing a reward;
- enforce a hard constraint;
- combine two or more plans.

Innovative aspects of ConStructPlan

WP3.
plan arithmetic
for **controllability**



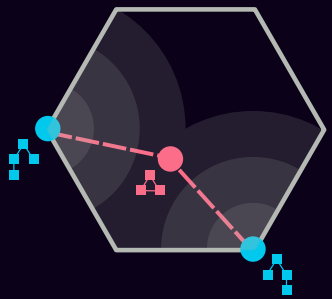
Operations over plans:

- guide a plan towards increasing a reward;
- enforce a hard constraint;
- combine two or more plans. ←

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \tfrac{1}{2}\, d(z, z_1) + \tfrac{1}{2}\, d(z, z_2)$$

Innovative aspects of ConStructPlan

WP3.
plan arithmetic
for **controllability**



Operations over plans:

- guide a plan towards increasing a reward;
- enforce a hard constraint;
- combine two or more plans. ←

$$\underset{z \in \mathcal{Z}}{\text{minimize}} \quad \tfrac{1}{2}\, d(z, z_1) + \tfrac{1}{2}\, d(z, z_2)$$

Discrete/continuous structure (Niculae et al., 2018, 2020)

Challenges:

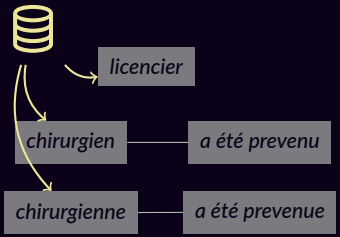
- choice of geometry $d(z, z')$
- efficient large-scale algorithms
  (for local, private use).

Innovative aspects of ConStructPlan

WP1.
latent modeling of
structured plans

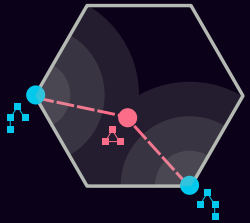
hierarchical, adaptive

WP2.
planning with
retrieved fragments



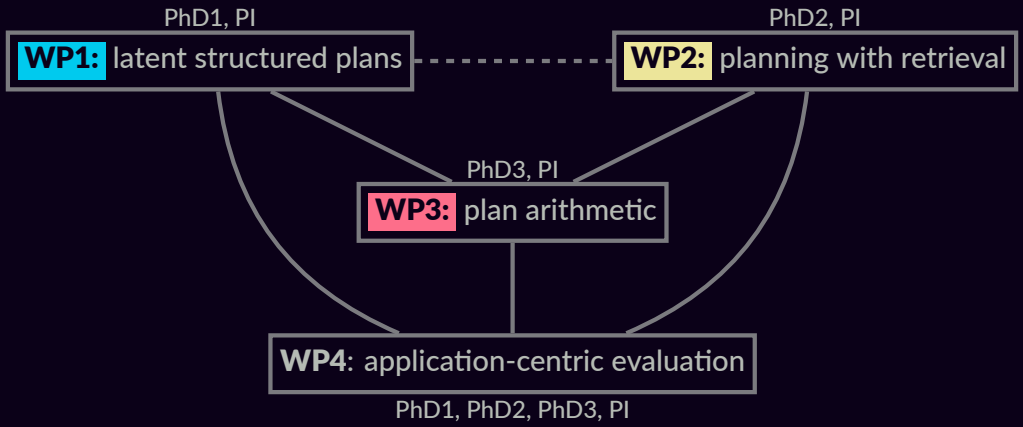
powerful, traceable

WP3.
plan arithmetic for
controllability



efficient operations
in *plan* geometry

Organization & team



focus: (document) translation, (controllable) summarization

My journey to ConStructPlan

My journey to ConStructPlan

open-source
core team

My journey to ConStructPlan

open-source
core team

Cornell University

best CS PhD dissertation

My journey to ConStructPlan

open-source
core team

Cornell University

best CS PhD dissertation

Senior Assistant Professor,
ELLIS Scholar.

Supervising 4 PhDs (co-sup. 9)

- junior researcher talent scheme
- Horizon UTTER coordinator
- WPs in consortia *Hybrid Intelligence* & *ROBUST*.

UvA

My journey to ConStructPlan

open-source
core team

scikit learn

Cornell University

best CS PhD dissertation

Senior Assistant Professor,
ELLIS Scholar.

UvA

Supervising 4 PhDs (co-sup. 9)

junior researcher talent scheme

Horizon UTTER coordinator

WPs in consortia *Hybrid Intelligence* & *ROBUST*.

Publication record: 6542 citations; h-index 21
37 top-venue publications (15 ML, 20 NLP).
Collaborations:

My journey to ConStructPlan

open-source
core team

scikit learn

Cornell University

best CS PhD dissertation

Senior Assistant Professor,
ELLIS Scholar.

UvA

Supervising 4 PhDs (co-sup. 9)

junior researcher talent scheme

Horizon UTTER coordinator

WPs in consortia *Hybrid Intelligence* & *ROBUST*.

Publication record: 6542 citations; h-index 21
37 top-venue publications (15 ML, 20 NLP).
Collaborations:

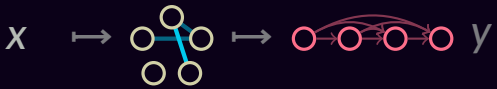
Recognition in structure prediction for NLP
monograph (Niculae et al, F&T Sig. Proc., 2025)
invited tutorial (ACL 2019, RANLP 2020);
summer school lecturer Athens NLP (2024, 2025)

ConStructPlan

a new paradigm

efficient *modular* *controllable*

for language generation
through structured plans.



thank you!

extra slides

WP4 Application-driven evaluation

Specific, practical applications.
Mainly: machine translation and summarization.

conditional generation:

- document translation
  efficient context use
- aspect-based summarization
  plan adaptivity

controlled generation:

- constrained translation:
  lexical constraints,
  subtitle translation
- controllable summarization
  length, toxicity, sentiment
- grammar-constrained generation
  code generation, math problem solving

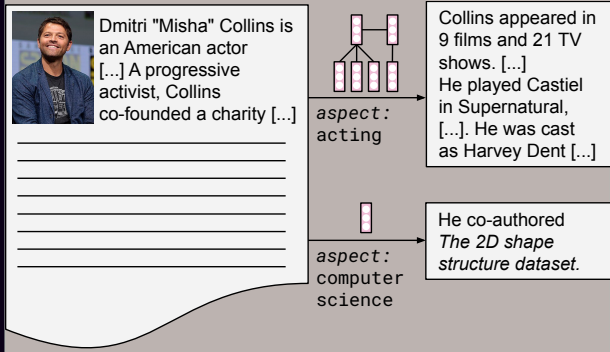
WP1. Latent structured plans

Challenges.

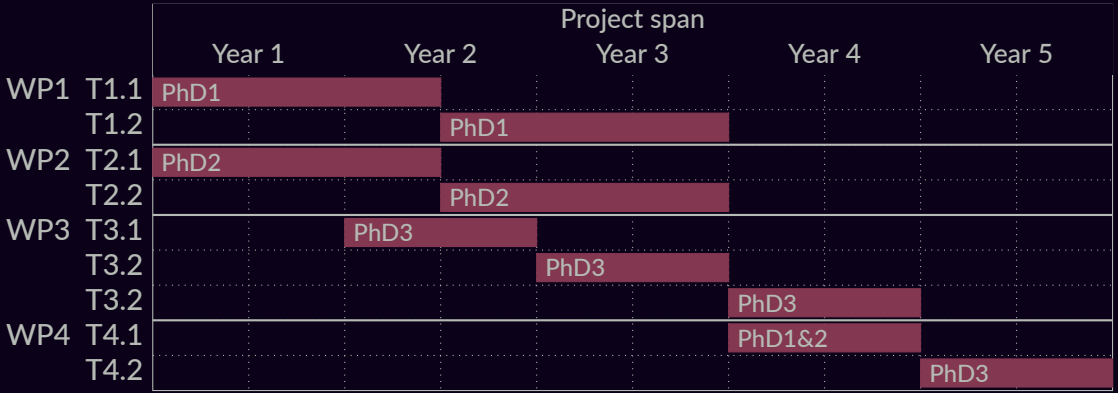
- compact, but interpretable.
- comprehensive, sufficient:
  the plan compresses the output distribution.
- hierarchical and sequential structure
- contextual adaptivity of plan size and complexity

Approach.

- New methods for structured and discrete latent variable learning.
- Adaptivity through sparsity.



Planning


Project span
Year 1
Year 2
Year 3
Year 4
Year 5
WP1
T1.1
PhD1
T1.2
PhD1
WP2
T2.1
PhD2
T2.2
PhD2
WP3
T3.1
PhD3
T3.2
PhD3
T3.2
PhD3
WP4
T4.1
PhD1&2
T4.2
PhD3

Mitigating risks

Expected results and impact

- More reliable and efficient language generation systems.
- New algorithms, probabilistic models, and implementations.
- Lower barrier to LLM research through modularity.
- Research results applicable to other fields, e.g., molecule generation.

Implementation context

Advisory board

Yulia Tsvetkov
Associate professor,
U. of Washington, USA

Mario T. Figueiredo
Professor,
Instituto Superior Tecnico, Portugal

Mirella Lapata
Professor,
U. of Edinburgh, UK

Implementation context

Advisory board

Yulia Tsvetkov
Associate professor,
U. of Washington, USA

Mario T. Figueiredo
Professor,
Instituto Superior Tecnico, Portugal

Mirella Lapata
Professor,
U. of Edinburgh, UK

Collaborations and synergies

UTTER
Horizon Europe, ends 2025

- continue work on context-aware translation,
- build on outputs (TowerLLM, EuroLLM)

TAIM Lab: Trustworthy AI for Media

- collaborate with Maastricht and RTL Netherlands
- exploit applications in subtitling

Key limitations

- current LMs lack controllability.

Key limitations

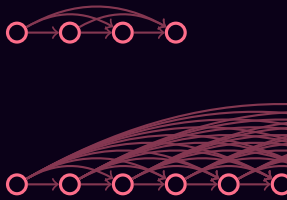
- current LMs lack controllability.
- two-stage generation helps, but current approach inefficient, fragile.

Key limitations

- current LMs lack controllability.
- two-stage generation helps, but current approach inefficient, fragile.
- monolithic: one, huge, server-side model.

Key limitations

- current LMs lack controllability.
- two-stage generation helps, but current approach inefficient, fragile.
- monolithic: one, huge, server-side model.
- optimized for single answer, not distributions.



ConStructPlan in one picture