TRAITEMENT AUTOMATIQUE DES LANGUES AVANCE

Master Informatique

2ème Année – 1er Semestre

Intervenants CM:

Gaël DIAS (gael.dias@unicaen.fr), Marc SPANIOL, Fabrice MAUREL

Intervenants TP:

Navneet AGARWAL, Kirill MILINTSEVICH

GREYC Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN ÉCOLE PUBLIQUE D'INGÉNIEURS CENTRE DE RECHERCHE

cnrs

Plan de l'UE

1. [CM 1] Représentation sémantique de texte [GD]
2. [CM 2] Cohérence textuelle [MS]
3. [CM 3] Modélisation thématique [NA]
4. [CM 4] Résumé de textes et traduction automatique [MS]
5. [CM 5] Génération langagière I [KM]
6. [CM 6] Génération langagière II [KM]
7. [CM 7] TAL multimodal [NA]
8. [CM 8] TAL et web [MS]
9. [CM 9] TAL et handicap visuel [FM]
10. [CM 10] TAL et psychiatrie [GD]

11. [TP 1-5] Génération neuronal de comptes-rendus médicaux [NA - KM]

COURS N°1

Représentation sémantique de texte

GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

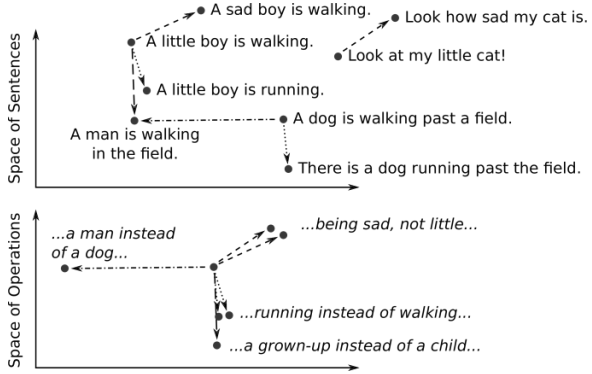
Plan du cours

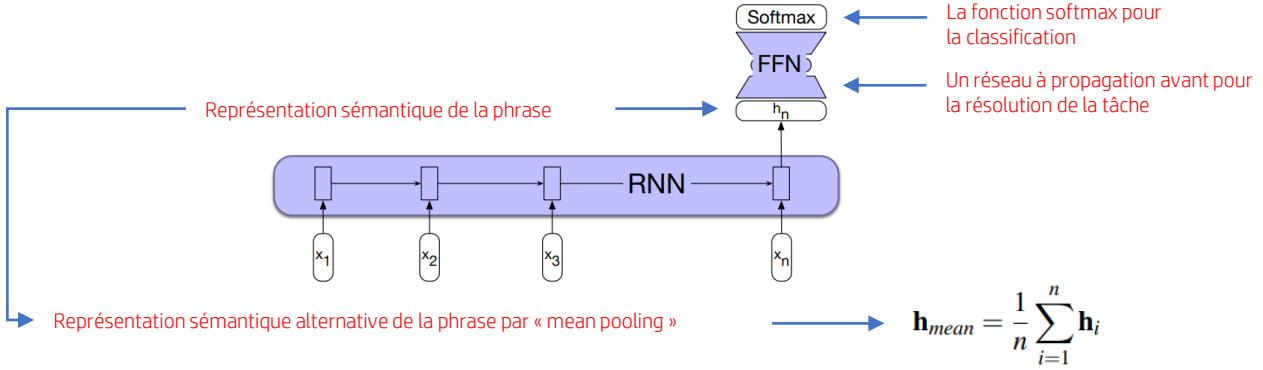
1. Représentations neuronales au niveau de la phrase
2. Les modèles du langage pré-entraînés
3. Amélioration de la représentation phrastique
4. Analyse au niveau du document
5. Les limitations des représentations neuronales
6. Les modèles hiérarchiques
7. Les transformeurs avec récurrence
8. Les transformeurs avec patrons de contenus
9. Les transformeurs avec patrons spécifiques

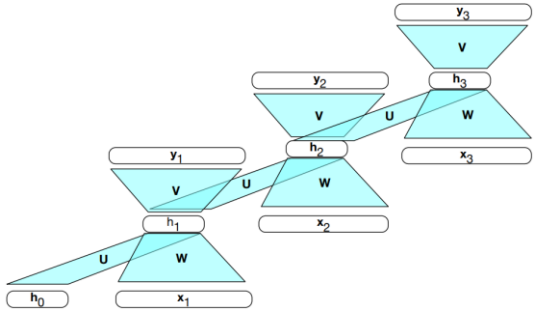
Représentations neuronales au niveau de la phrase

1. Les réseaux récurrents

- Les réseaux de neurones récurrents (RNN) comme les LSTM permettent de traiter des séquences de toute longueur.
- Ils présentent le problème de « vanishing gradient » qui empêche de capturer pleinement la sémantique des phrases.







Représentations neuronales au niveau de la phrase

2. Les transformeurs

- Contrairement aux RNN, les transformeurs ne présentent pas le problème de « vanishing gradient » et sont ainsi capables de représenter la sémantique des phrases plus complètement.
- Par contre, ils ne peuvent pas représenter des phrases de toute longueur puisqu'ils n'encodent pas la récurrence.

Représentation sémantique de la phrase

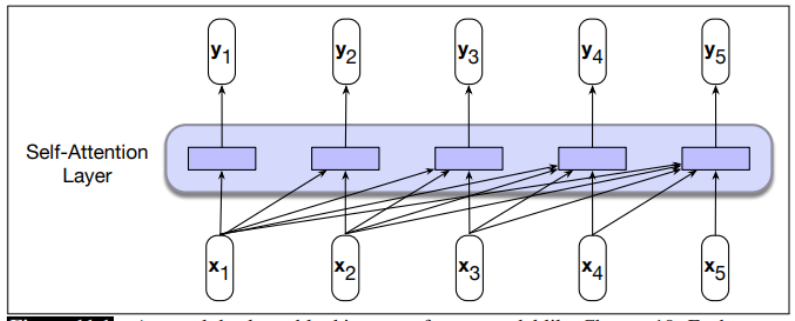
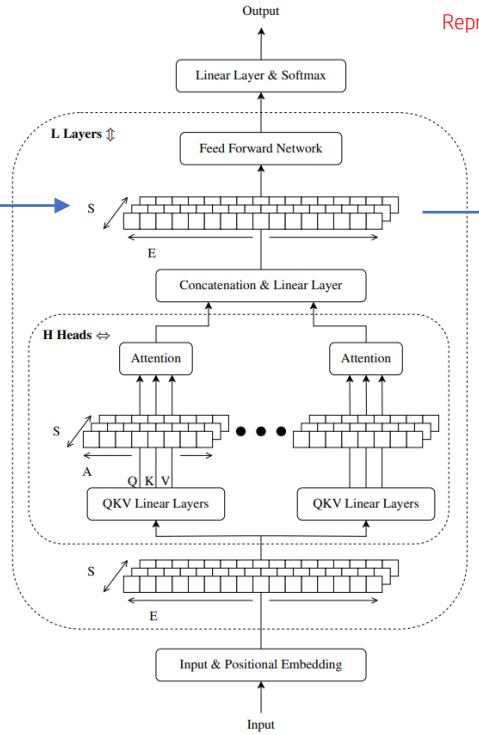


Figure 11.1 A causal, backward looking, transformer model like Chapter 10. Each output is computed independently of the others using only information seen earlier in the context.

Architecture causale : e.g. GPT



Représentation sémantique alternative de la phrase par « mean pooling »

$$\mathbf{h}_{mean} = \frac{1}{n}\sum_{i=1}^{n}\mathbf{h}_i$$

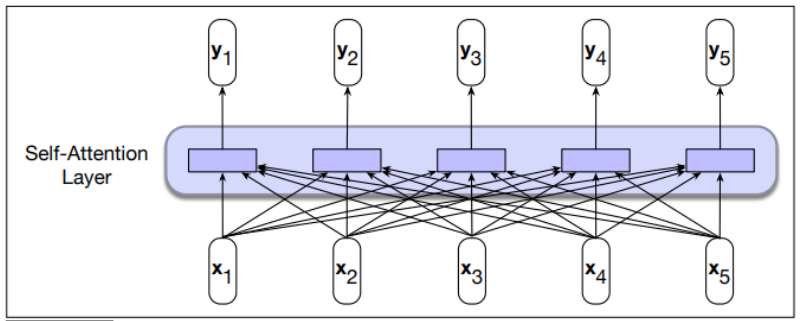


Figure 11.2 Information flow in a bidirectional self-attention model. In processing each element of the sequence, the model attends to all inputs, both before and after the current one.

Architecture bidirectionnelle : e.g. BERT

Les modèles du langage pré-entraînés

1. Idée de base

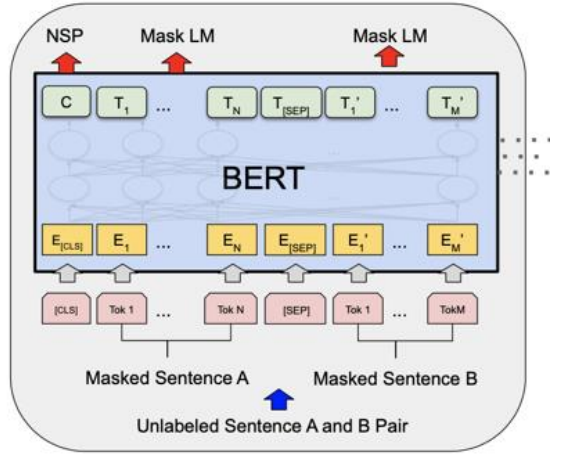
- Plutôt que de travailler avec des modèles dont les poids sont initialisés aléatoirement et de ne se fonder que sur les plongements lexicaux comme base de connaissances, il est souvent préférable de pré-calculer les poids du réseau.
- Ainsi, les modèles pré-entraînés sont des modèles où des connaissances sur la langue ont déjà été apprises de façon non supervisée (autoencodeurs).
- Ces modèles pré-entraînés sont donc des modèles du langage.
- Suivant les modèles pré-entraînés, différentes techniques peuvent être utilisées.

2. Le modèle du langage masqué (« Masked Language Modeling »)

- L'idée est de prédire des tokens d'une phrase qui sont masqués aléatoirement, appelée « cloze task ».

Please turn your homework ____ . Please turn ____ homework in.

- Cette technique est utilisée dans le cadre du modèle BERT (Delvin et al., 2019) basé sur des transformeurs bidirectionnels. En particulier, BERT est composé de 12 blocs chacun avec 12 têtes d'attention et la taille des couches cachées est de 768. Les phrases sont décomposées en tokens qui représentent des sous-mots (30000 tokens pour l'anglais).

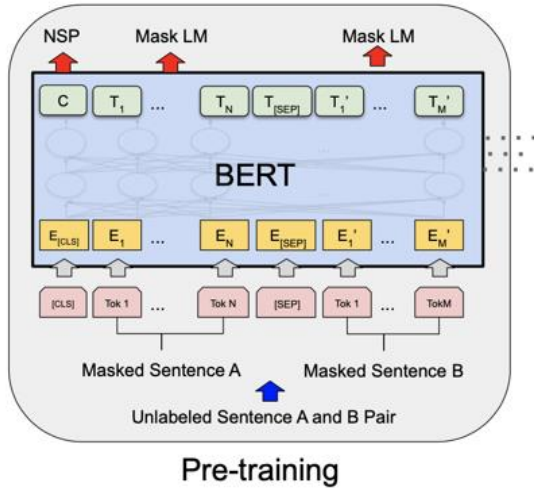


Pre-training

Les modèles du langage pré-entraînés

2. Le modèle du langage masqué (« Masked Language Modeling ») (suite)

- En plus des tokens masqués, certains mots sont remplacés par des mots non pertinents de façon aléatoire. Cela a pour objectif d'améliorer le regroupement sémantique des tokens (idée proche du « contrastive learning »).



Pre-training

3. Interprétation du modèle du langage masqué

- Etant donnée une séquence de tokens, les vecteurs de sortie de chaque token correspondent à des plongements contextualisés c'est-à-dire prenant en compte le contexte de la phrase.
- Ceci est particulièrement important pour les tokens polysémiques (e.g. jaguar) dont les représentations seront différentes selon le contexte de la phrase dans lesquels ils sont inclus.
- Ainsi, le vecteur yi du dernier bloc est le plongement du token xi.
- Il est possible de faire la moyenne des vecteurs yi des derniers blocs du transformeur pour atteindre une meilleure représentation.

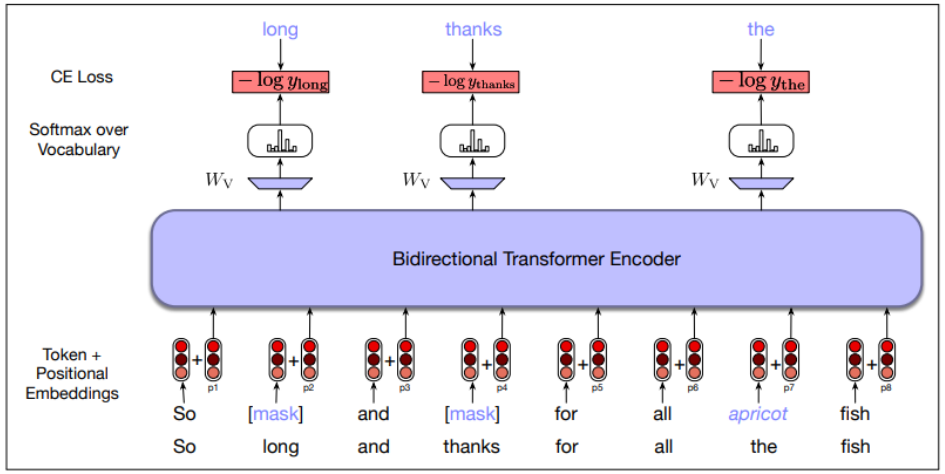


Figure 11.5 Masked language model training. In this example, three of the input tokens are selected, two of which are masked and the third is replaced with an unrelated word. The probabilities assigned by the model to these three items are used as the training loss. (In this and subsequent figures we display the input as words rather than subword tokens; the reader should keep in mind that BERT and similar models actually use subword tokens instead.)

Les modèles du langage pré-entraînés

4. La prédiction de la phrase suivante

- Dans le cadre de BERT, une autre tâche est apprise qui consiste à prédire si une phrase en précède une autre ou non.
- En effet, ceci peut aider à résoudre des tâches comme la détection de paraphrases, l'implication de phrases (« textual entailment ») ou la cohérence textuelle.
- Pour se faire, un token de segment est créé qui facilite la différenciation des deux phrases: [SEP].
- Notez que la prédiction n'est faite que sur le vecteur du token [CLS] qui correspond à la représentation de la phrase dans sa globalité.
- Chaque token est représenté par un plongement lexical, un plongement positionnel et un plongement de segment.

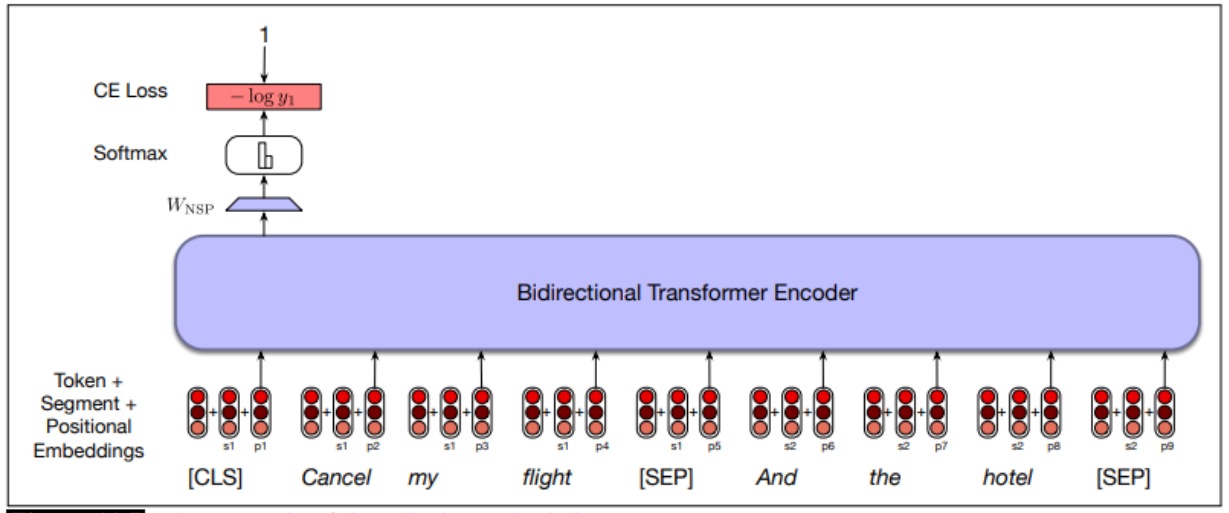
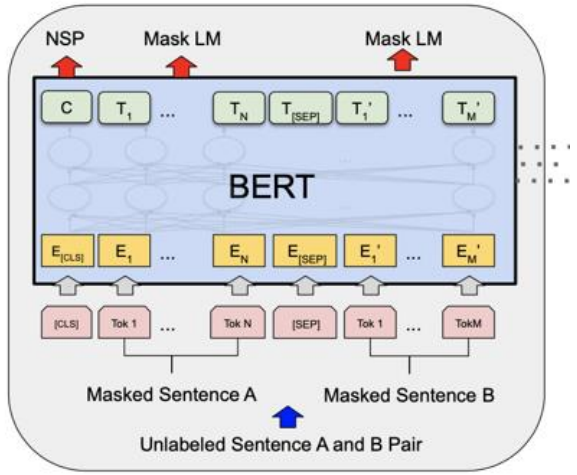


Figure 11.7 An example of the NSP loss calculation.



Pre-training

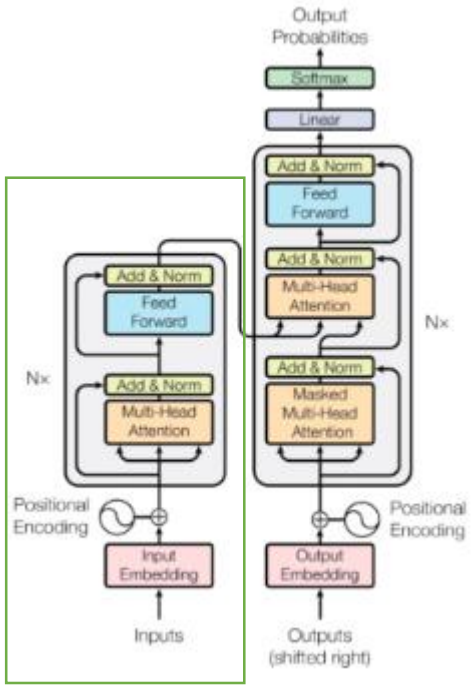
Les modèles du langage pré-entraînés

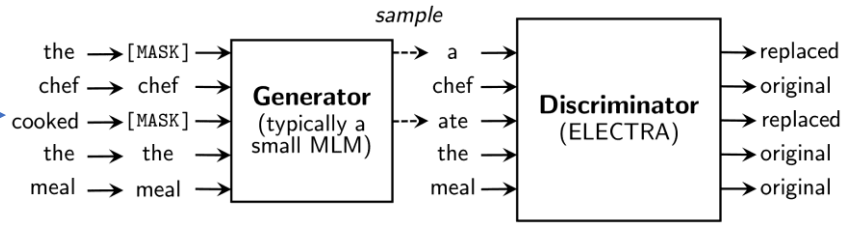
5. Types de modèles pré-entraînés

- Il existe différents types de modèles du langage pré-entraînés, selon qu'ils ont été entraînés sur des tâches de compréhension de la langue (« natural language understanding ») ou des tâches de génération (« natural language generation ») ou les deux à la fois.

6. Modèles de type Encodeur (« natural language understanding »)

- BERT (Delvin et al., 2019)
- DistillBERT (Sanh et al., 2020) – un modèle compact de BERT par distillation
- RoBERTa (Liu et al., 2019) – plus de données, pas de NSP et des phrases longues
- XLM (Lample et Conneau, 2019) – version multilingue avec une nouvelle fonction d'erreur
- ALBERT (Lan et al., 2019) – prédiction de l'ordre des phrases
- ELECTRA (Clark et al., 2020) – architecture de type *Generative Adversarial Network* (GAN)
- DeBERTa (He et al., 2021) – deux vecteurs de représentation des tokens

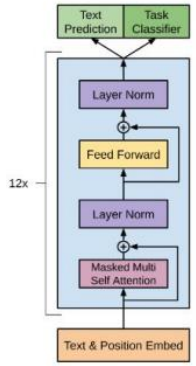


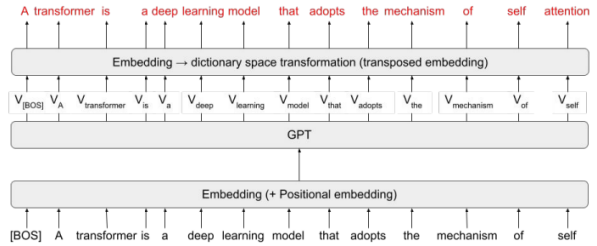


Les modèles du langage pré-entraînés

7. Modèles de type Décodeur (« natural language generation »)

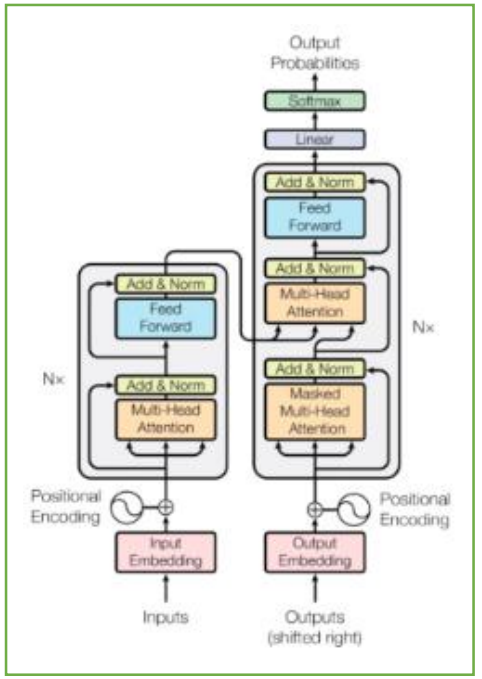
- GPT (Radford et al., 2018) – modèle causal
- CTRL (Keshar et al., 2019) – contrôle du style de génération
- GPT-2 (Solaiman et al., 2019) – peut produire des textes cohérents
- GPT-3 (Brown et al, 2020) – version avec beaucoup plus de paramètres





8. Modèles de type Encodeur - Décodeur (NLU + NLG)

- T5 (Raffel et al., 2020)
- BART (Lewis et al., 2019) – BERT + GPT
- M2M-100 (Fan et al., 2020) – modèle multilingue
- BigBIRD (Zaheer et al, 2020) – optimisé en terme de mémoire utilisée



Le « fine tuning » ou ajustement

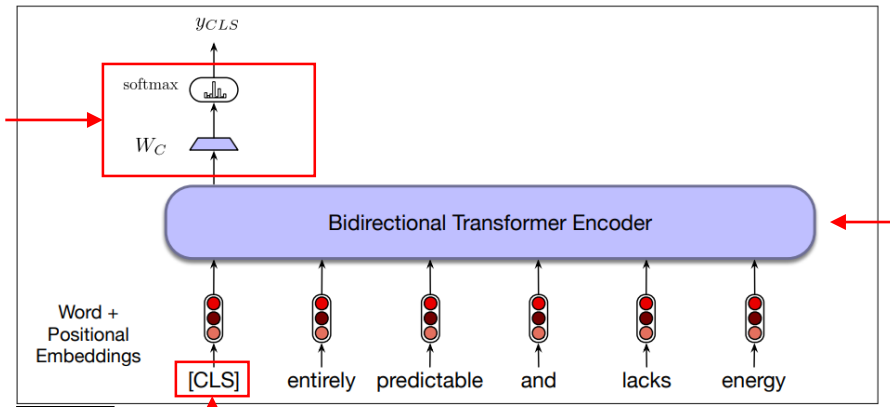
1. Idée générale

- Le « fine tuning » consiste à utiliser les modèles du langage pré-entraînés comme base de connaissance initiale pour finalement ajuster les poids du réseau pour une application donnée, e.g. analyse de sentiments, l'annotation des entités nommées etc.
- Intellectuellement, ceci revient à concevoir un réseau de neurones comme un cerveau qui contient déjà de l'information à sa naissance. Il hérite donc d'une connaissance du langage acquise au long des temps, c'est-à-dire à partir d'une quantité astronomique des textes. Ceci correspond à l'idée initiale de Chomsky mais dans un cadre statistique/probabiliste plutôt que logique.

2. Classification par « fine tuning »

- A l'entraînement, les poids du modèle pré-entraînés sont ajustés pour une petite quantité de données spécifiques à la tâche.

Couche de classification

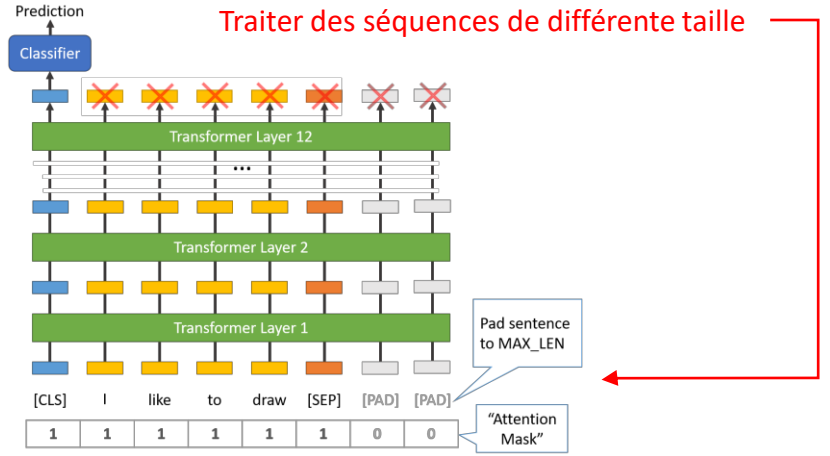


BERT

Token [CLS] sensé représenter sémantiquement la phrase

Figure 11.8 Sequence classification with a bidirectional transformer encoder. The output vector for the [CLS] token serves as input to a simple classifier.

Traiter des séquences de différente taille



Le « fine tuning » ou ajustement

3. Annotation par « fine tuning »

- Dans le cadre d'une annotation au niveau des mots ou des séquences selon le modèle BIO, le « fine tuning » est facile à modéliser avec une particularité en ce qui concerne la tokenization.

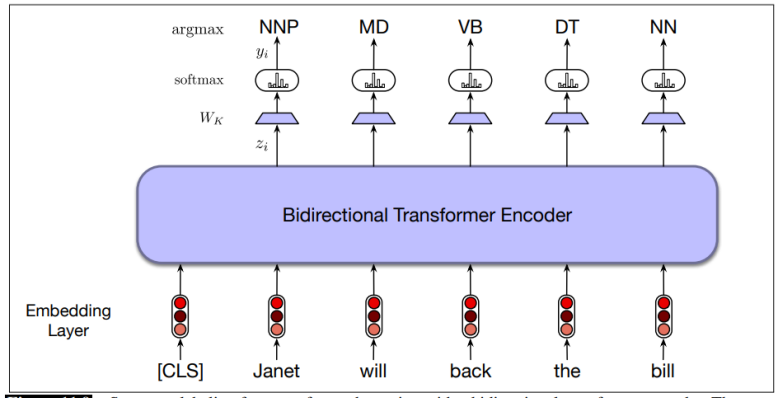


Figure 11.9 Sequence labeling for part-of-speech tagging with a bidirectional transformer encoder. The output vector for each input token is passed to a simple k-way classifier.

[LOC **Mt. Sanitas**] is in [LOC **Sunshine Canyon**]

Mt. Sanitas is in Sunshine Canyon .
B-LOC I-LOC O O B-LOC I-LOC O

BERT se repose sur une tokenization au niveau des sous-mots, donc la tâche d'annotation n'est pas directe.

'Mt', '.', 'San', '##itas', 'is', 'in', 'Sunshine', 'Canyon' '.'

A l'entraînement, chaque sous-mot reçoit l'étiquette du mot complet.

A l'inférence, le mot reçoit le label le plus significatif de son premier sous-mot. Il existe des techniques plus sophistiquées dont le but est de regarder la distribution des étiquettes de tous les sous-mots pour en déduire l'étiquette du mot.

Le « fine tuning » ou ajustement

4. Classification de séquences par « fine tuning »

- Cette tâche consiste à classer des séquences de tokens d'intérêt, comme par exemple les unités polylexicales, des unités étiquetées par des experts (e.g. mémoire épisodique, annotations de psychiatres, etc.). Cette tâche se rapproche des méthodes d'annotation BIO.

Une séquence est représentée par la concaténation de son plongement initial, de son plongement final et d'un plongement calculé par un mécanisme d'auto-attention sur l'ensemble de la séquence.

$$\text{spanRep}_{ij} = [h_i; h_j; g_{i,j}]$$

$$\mathbf{g}_{ij} = \text{SelfAttention}(\mathbf{h}_{i:j})$$

Ainsi, le plongement intermédiaire entre le plongement initial et le plongement final est une moyenne (« average pooling ») de tous les plongements de la séquence pondérés par le mécanisme d'auto-attention.

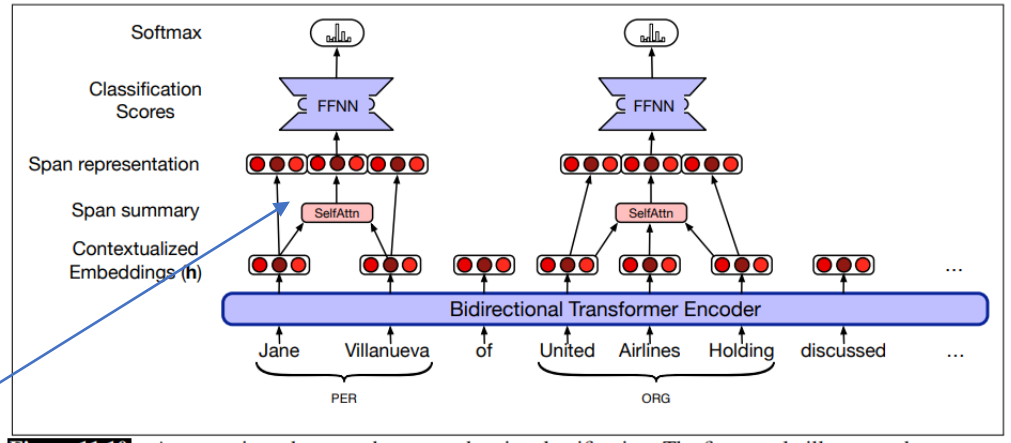
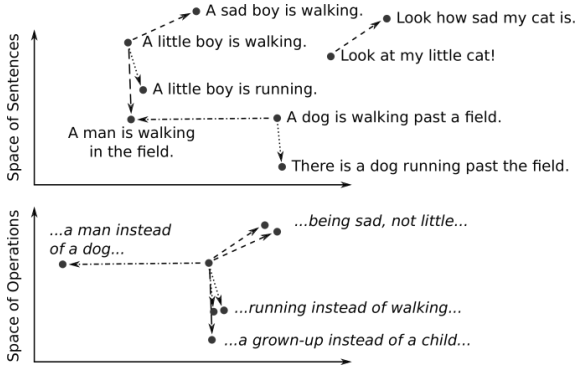


Figure 11.10 A span-oriented approach to named entity classification. The figure only illustrates the computation for 2 spans corresponding to ground truth named entities. In reality, the network scores all of the $\frac{T(T-1)}{2}$ spans in the text. That is, all the unigrams, bigrams, trigrams, etc. up to the length limit.

Toutes les sous-séquences jusqu'à une taille maximale sont classifiées

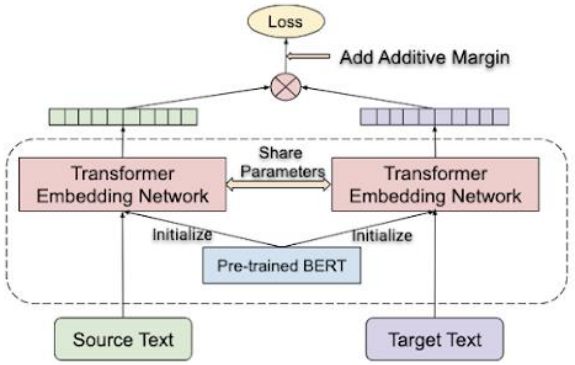
Amélioration de la représentation phrastique

1. Importance de la représentation sémantique
   - Les modèles du langage ont comme principale tâche de correctement représenter les phrases dans un espace sémantique cohérent. Plus cet espace sera juste et plus les tâches pourront être apprises simplement et plus les performances seront élevées.
2. Limites des modèles du langage
   - Les modèles du langage ont été appris sur la base de tâches bien définies. Pour BERT, le « masked language modelling » et pour GPT le « next word prediction ».
   - Or ces tâches ne peuvent garantir la construction d'un espace sémantique cohérent même si elles s'en rapprochent.
   - Des travaux s'attachent à construire des espaces de représentation sémantique, notamment basés sur l'idée de « contrastive learning » (Reimers and Gurevych 2019, Chen et al. 2020)



Please turn ____ homework in.

Please turn your homework ____ .



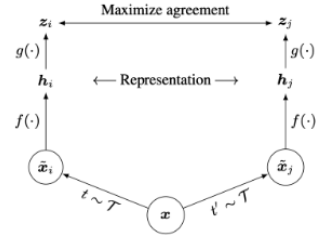


Figure 2. A simple framework for contrastive learning of visual representations. Two separate data augmentation operators are sampled from the same family of augmentations ($t \sim \mathcal{T}$ and $t' \sim \mathcal{T}$) and applied to each data example to obtain two correlated views. A base encoder network $f(\cdot)$ and a projection head $g(\cdot)$ are trained to maximize agreement using a contrastive loss. After training is completed, we throw away the projection head $g(\cdot)$ and use encoder $f(\cdot)$ and representation $\boldsymbol{h}$ for downstream tasks.

Amélioration de la représentation phrastique

1. SentenceBERT (Reimers & Gurevych, 2019)

- SentenceBERT propose d'ajuster un modèle du langage de type BERT en se basant sur une architecture siamoise.
- Pour se faire, l'architecture est entraînée sur les jeux de données SNLI (https://nlp.stanford.edu/projects/snli/) et MultiNLI (https://cims.nyu.edu/~sbowman/multinli/).
- Cette architecture est ensuite testée sans nouvel apprentissage (i.e. « linear probing ») sur la série de jeux de données STS qui évaluent les similarités entre phrases .

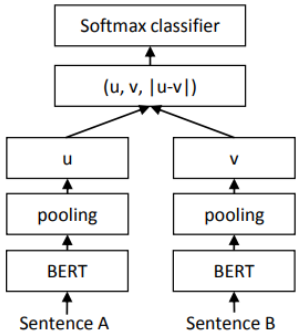


Figure 1: SBERT architecture with classification objective function, e.g., for fine-tuning on SNLI dataset. The two BERT networks have tied weights (siamese network structure).

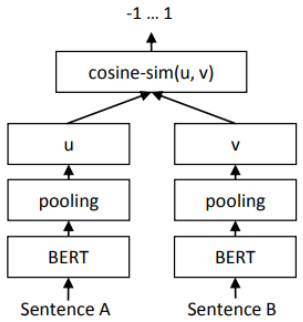


Figure 2: SBERT architecture at inference, for example, to compute similarity scores. This architecture is also used with the regression objective function.

Model	STS12	STS13	STS14	STS15	STS16	STSb	SICK-R	Avg.
Avg. GloVe embeddings	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
Avg. BERT embeddings	38.78	57.98	57.98	63.15	61.06	46.35	58.40	54.81
BERT CLS-vector	20.16	30.01	20.09	36.88	38.08	16.50	42.63	29.19
InferSent - Glove	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder	64.49	67.80	64.61	76.83	73.18	74.92	**76.69**	71.22
SBERT-NLI-base	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
SBERT-NLI-large	72.27	**78.46**	**74.90**	80.99	76.25	**79.23**	73.75	76.55
SRoBERTa-NLI-base	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
SRoBERTa-NLI-large	**74.53**	77.00	73.18	**81.85**	**76.82**	79.10	74.29	**76.68**

Table 1: Spearman rank correlation ρ between the cosine similarity of sentence representations and the gold labels for various Textual Similarity (STS) tasks. Performance is reported by convention as $\rho \times 100$. STS12-STS16: SemEval 2012-2016, STSb: STSbenchmark, SICK-R: SICK relatedness dataset.

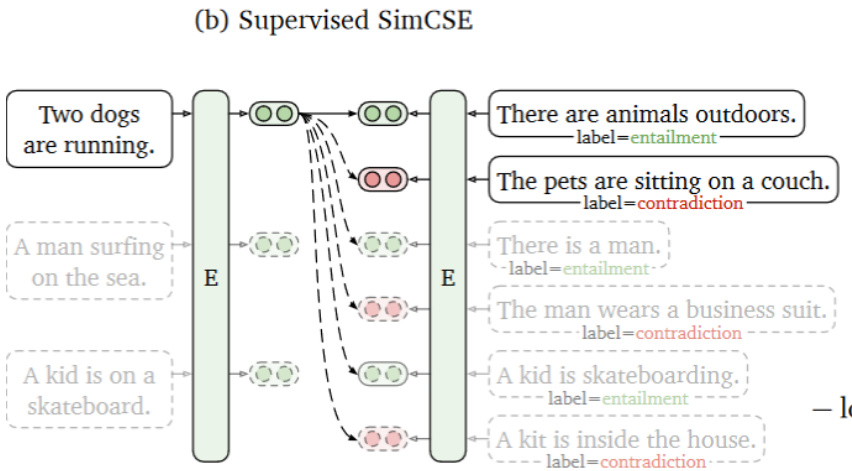
	NLI	STSb
Pooling Strategy		
MEAN	**80.78**	**87.44**
MAX	79.07	69.92
CLS	79.80	86.62
Concatenation		
(u, v)	66.04	-
$(\|u - v\|)$	69.78	-
$(u * v)$	70.54	-
$(\|u - v\|, u * v)$	78.37	-
$(u, v, u * v)$	77.44	-
$(u, v, \|u - v\|)$	**80.78**	-
$(u, v, \|u - v\|, u * v)$	80.44	-

Table 6: SBERT trained on NLI data with the classification objective function, on the STS benchmark (STSb) with the regression objective function. Configurations are evaluated on the development set of the STSb using cosine-similarity and Spearman's rank correlation. For the concatenation methods, we only report scores with MEAN pooling strategy.

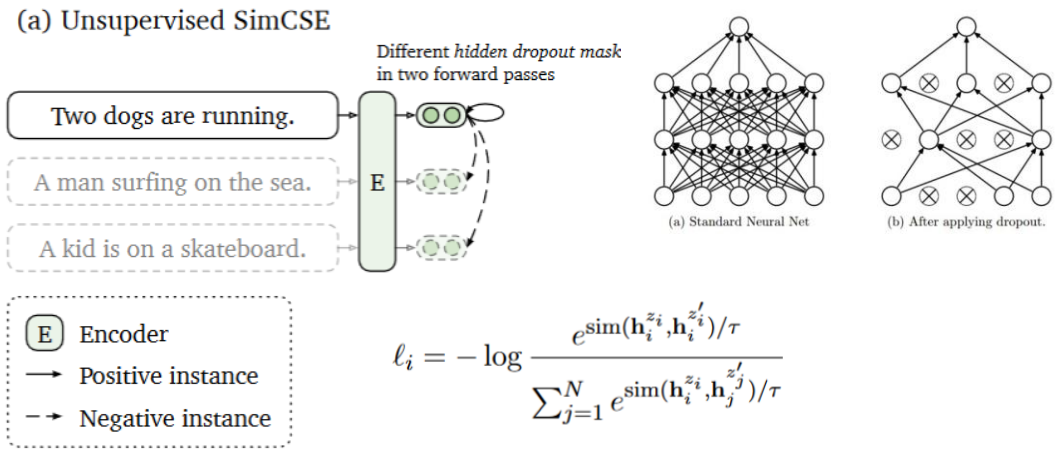
Amélioration de la représentation phrastique

1. SimSCE (Gao et al., 2022)

- SimSCE propose d'ajuster un modèle du langage de type BERT ou RoBERTa en se basant sur deux idées différentes: une architecture non supervisée et une architecture supervisée dans le cadre d'un apprentissage contrastif.
- L'architecture non supervisée reçoit une même phrase deux fois en entrée dans le même batch mais avec deux masques de « droupout » différents. L'objectif d'apprentissage est donc de rapprocher ses deux représentations.
- L'architecture supervisée est entraînée sur les jeux de données SNLI (https://nlp.stanford.edu/projects/snli/) et MultiNLI (https://cims.nyu.edu/~sbowman/multinli/) pour donner les meilleurs résultats.
- Dans l'architecture supervisée, des exemples positifs (« entailment ») et des exemples négatifs (« contradiction ») partagent le même « batch » et les exemples positifs doivent être rapprochés alors que les négatifs éloignés.



$$-\log \frac{e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_i^+)/\tau}}{\sum_{j=1}^{N}\left(e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^+)/\tau}+e^{\mathrm{sim}(\mathbf{h}_i,\mathbf{h}_j^-)/\tau}\right)}$$



$$\ell_i = -\log \frac{e^{\mathrm{sim}(\mathbf{h}_i^{z_i},\mathbf{h}_i^{z_i'})/\tau}}{\sum_{j=1}^{N} e^{\mathrm{sim}(\mathbf{h}_i^{z_i},\mathbf{h}_j^{z_j'})/\tau}}$$

Amélioration de la représentation phrastique

1. SimSCE (Gao et al., 20220) (suite)

Dataset	sample	full
Unsup. SimCSE (1m)	-	82.5
QQP (134k)	81.8	81.8
Flickr30k (318k)	81.5	81.4
ParaNMT (5m)	79.7	78.7
SNLI+MNLI		
entailment (314k)	**84.1**	**84.9**
neutral (314k)[8]	82.6	82.9
contradiction (314k)	77.5	77.6
all (942k)	81.7	81.9
SNLI+MNLI		
entailment + hard neg.	-	**86.2**
+ ANLI (52k)	-	85.0

Table 4: Comparisons of different supervised datasets as positive pairs. Results are Spearman's correlations on the STS-B development set using $\text{BERT}_{\text{base}}$ (we use the same hyperparameters as the final SimCSE model). Numbers in brackets denote the # of pairs. *Sample*: subsampling 134k positive pairs for a fair comparison among datasets; *full*: using the full dataset. In the last block, we use entailment pairs as positives and contradiction pairs as hard negatives (our final model).

Model	STS12	STS13	STS14	STS15	STS16	STS-B	SICK-R	Avg.
Unsupervised models								
GloVe embeddings (avg.)♣	55.14	70.66	59.73	68.25	63.66	58.02	53.76	61.32
$\text{BERT}_{\text{base}}$ (first-last avg.)	39.70	59.38	49.67	66.03	66.19	53.87	62.06	56.70
$\text{BERT}_{\text{base}}$-flow	58.40	67.10	60.85	75.16	71.22	68.66	64.47	66.55
$\text{BERT}_{\text{base}}$-whitening	57.83	66.90	60.90	75.08	71.31	68.24	63.73	66.28
IS-$\text{BERT}_{\text{base}}$♡	56.77	69.24	61.21	75.23	70.16	69.21	64.25	66.58
CT-$\text{BERT}_{\text{base}}$	61.63	76.80	68.47	77.50	76.48	74.31	69.19	72.05
∗ SimCSE-$\text{BERT}_{\text{base}}$	**68.40**	**82.41**	**74.38**	**80.91**	**78.56**	**76.85**	**72.23**	**76.25**
$\text{RoBERTa}_{\text{base}}$ (first-last avg.)	40.88	58.74	49.07	65.63	61.48	58.55	61.63	56.57
$\text{RoBERTa}_{\text{base}}$-whitening	46.99	63.24	57.23	71.36	68.99	61.36	62.91	61.73
DeCLUTR-$\text{RoBERTa}_{\text{base}}$	52.41	75.19	65.52	77.12	78.63	72.41	**68.62**	69.99
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	**70.16**	**81.77**	**73.24**	**81.36**	**80.65**	**80.22**	68.56	**76.57**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	**72.86**	**83.99**	**75.62**	**84.77**	**81.80**	**81.98**	**71.26**	**78.90**
Supervised models								
InferSent-GloVe♣	52.86	66.75	62.15	72.77	66.87	68.03	65.65	65.01
Universal Sentence Encoder♣	64.49	67.80	64.61	76.83	73.18	74.92	76.69	71.22
$\text{SBERT}_{\text{base}}$♣	70.97	76.53	73.19	79.09	74.30	77.03	72.91	74.89
$\text{SBERT}_{\text{base}}$-flow	69.78	77.27	74.35	82.01	77.46	79.12	76.21	76.60
$\text{SBERT}_{\text{base}}$-whitening	69.65	77.57	74.66	82.27	78.39	79.52	76.91	77.00
CT-$\text{SBERT}_{\text{base}}$	74.84	83.20	78.07	83.84	77.93	81.46	76.42	79.39
∗ SimCSE-$\text{BERT}_{\text{base}}$	**75.30**	**84.67**	**80.19**	**85.40**	**80.82**	**84.25**	**80.39**	**81.57**
$\text{SRoBERTa}_{\text{base}}$♣	71.54	72.49	70.80	78.74	73.69	77.77	74.46	74.21
$\text{SRoBERTa}_{\text{base}}$-whitening	70.46	77.07	74.46	81.64	76.43	79.49	76.65	76.60
∗ SimCSE-$\text{RoBERTa}_{\text{base}}$	**76.53**	**85.21**	**80.95**	**86.03**	**82.57**	**85.83**	**80.50**	**82.52**
∗ SimCSE-$\text{RoBERTa}_{\text{large}}$	**77.46**	**87.27**	**82.36**	**86.66**	**83.93**	**86.70**	**81.95**	**83.76**

Table 5: Sentence embedding performance on STS tasks (Spearman's correlation, "all" setting). We highlight the highest numbers among models with the same pre-trained encoder. ♣: results from Reimers and Gurevych (2019); ♡: results from Zhang et al. (2020); all other results are reproduced or reevaluated by ourselves. For BERT-flow (Li et al., 2020) and whitening (Su et al., 2021), we only report the "NLI" setting (see Table C.1).

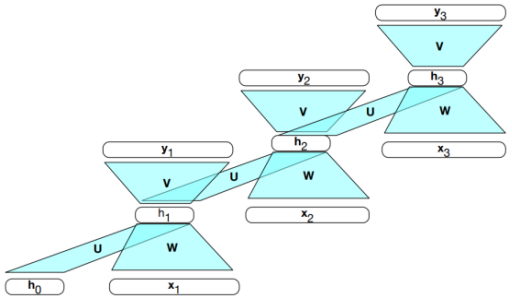
Analyse au niveau du document

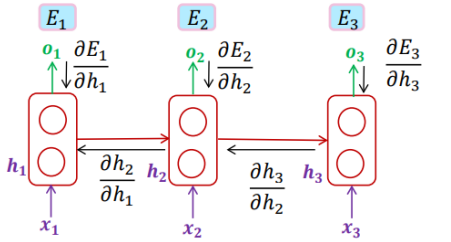
1. Différentes applications
   - Au semestre 1 du Master 1, nous avons étudié le langage au niveau du mot (analyse lexicale).
   - Au semestre 2 du Master 1, nous avons étudié le langage au niveau de la phrase (analyse phrastique)
   - Or, de nombreuses applications du langage naturel traitent des textes qui sont des séquences longues de phrases ou de paragraphes.
2. Applications intra-document
   - Cohérence textuelle : analyse des coréférences
   - Structure du discours : chaînes lexicales, segmentation thématique
   - Structure argumentative : « argument mining »
3. Applications de classification
   - Analyse de sentiments, d'émotions, de fake news, de diagnostic médical, de profilage etc …
4. Applications de génération
   - Traduction automatique
   - Résumé de textes
   - Dialogues

Les limitations des représentations neuronales

1. Représentations des textes par RNN

- Les RNN ont une capacité illimitée de représentation du fait de leur récurrence. Tout texte peut être représenté par un RNN. Par contre, leur capacité à représenter les séquences longues est limitée par la notion de « vanishing gradient ».





Direction of *Backward* pass (via partial derivatives)
--- gradient flow ---

$$\frac{\partial E_3}{\partial W} = \frac{\partial E_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial h_1}\frac{\partial h_1}{\partial W} + \frac{\partial E_3}{\partial h_3}\frac{\partial h_3}{\partial h_2}\frac{\partial h_2}{\partial W} + \frac{\partial E_3}{\partial h_3}\frac{\partial h_3}{\partial h_3}\frac{\partial h_3}{\partial W}$$

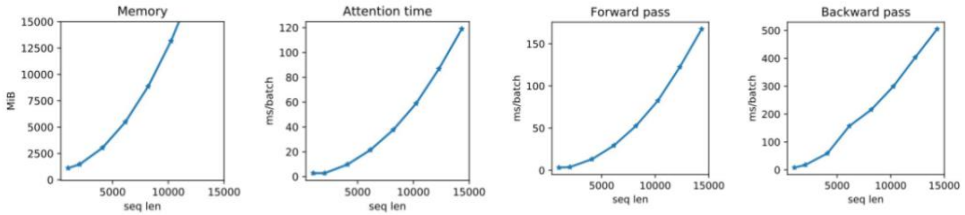
$$= \lll 1 \quad + \quad \ll 1 \quad + \quad < 1$$

Les gradients des dépendances longues sont dominés par les dépendances courtes.

2. Représentation des textes par transformeurs

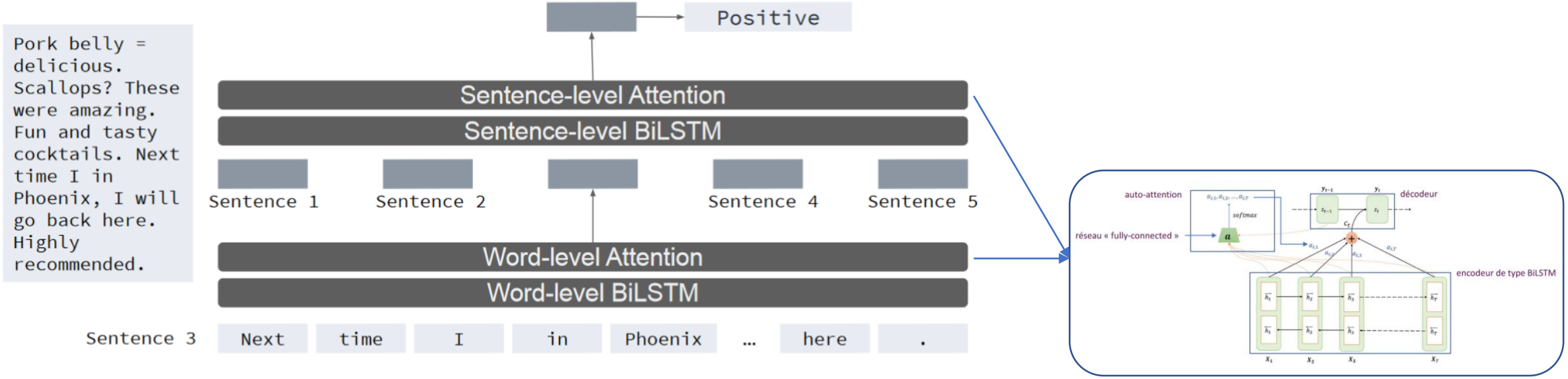
- Les transformeurs ont une capacité limitée de représentation du fait de leur non récurrence. Par exemple, pour BERT le nombre de tokens en entrée est limité à 512.
- Cette limitation est aussi due à la complexité quadratique en coût de calcul: $O(N^2)$ où N est la taille de la séquence.

Pour un transformeur à une tête et un sel bloc sur une carte graphique RTX8000 GPU.



Les modèles hiérarchiques

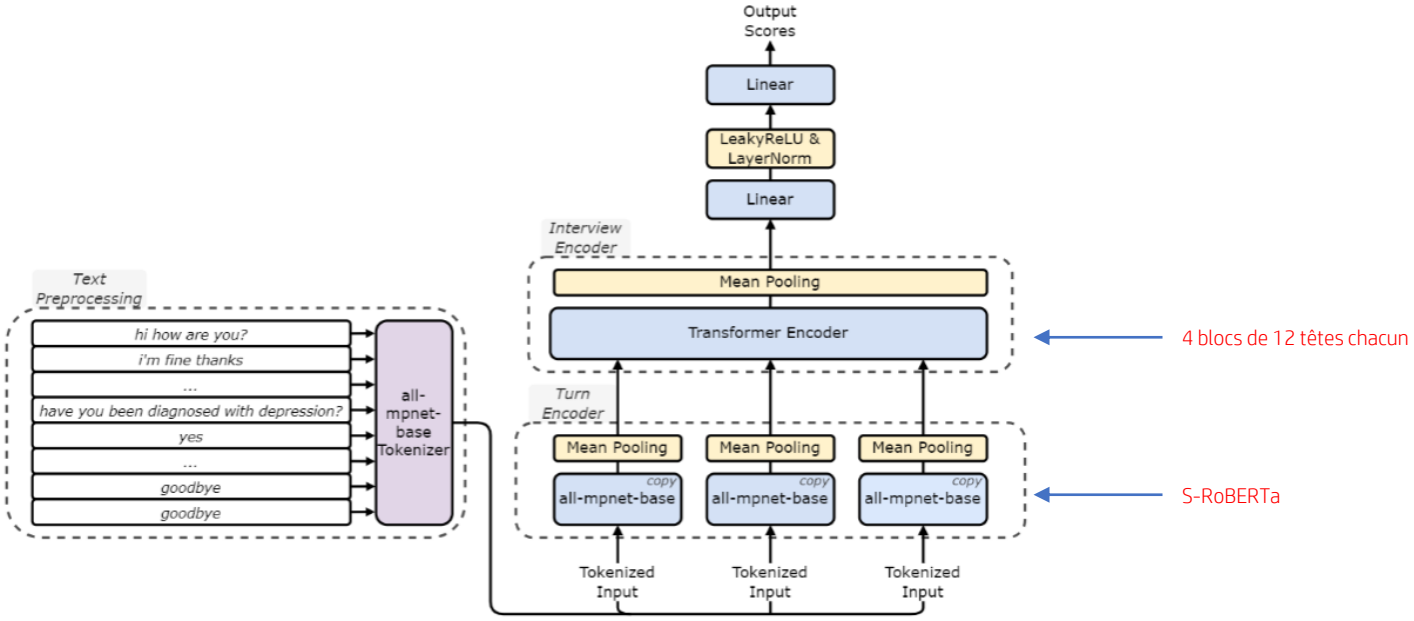
1. Utiliser la nature hiérarchique des textes
   - Les textes sont composés hiérarchiquement de caractères, de mots, de phrases et de paragraphes.
   - Les modèles de représentation prennent en compte cette organisation pour modéliser de longs textes.
2. Une représentation à partir de BiLSTM (Yang et al., 2016)
   - Le texte est d'abord divisé en phrases qui sont elles-mêmes divisées en mots. Chaque phrase est représentée par la dernière couche du BiLSTM initialisé à partir de plongements lexicaux. Toutes les phrases sont ensuite agglomérées dans un BiLSTM dont la dernière couche représente le texte dans son entièreté.



Les modèles hiérarchiques

3. Une représentation à partir de modèles pré-entraînés (Milintsevich et al., 2023)

- Afin de prendre en compte les modèles pré-entraînés de type BERT comme représentation des phrases et les transformeurs comme agrégateurs de sémantique, des architectures hiérarchiques plus performantes peuvent être élaborées.
- Dans ce cas, les phrases sont représentées grâce à des modèles de type SentenceBERT, ici S-RoBERTa qui sont ensuite agglomérées grâce à un transformeur.



Les modèles hiérarchiques

4. Des modèles hiérarchiques pré-entraînés au niveau du document

- (Zhang et al. 2020) proposent de pré-entraîner un modèle hiérarchique à partir de la notion de « masked language modeling » mais au niveau de la phrase.
- L'idée est de construire un modèle capable de représenter sémantiquement un texte de manière auto-supervisée, c'est-à-dire en reconstruisant des phrases masquées et en prédisant la phrase suivante.
- Le modèle HiBERT a été entraîné à partir du corpus GIGA-CM comportant 6.3 millions de documents et 2.8 milliards de tokens.

Pré-entraîné sur GIGA-CM

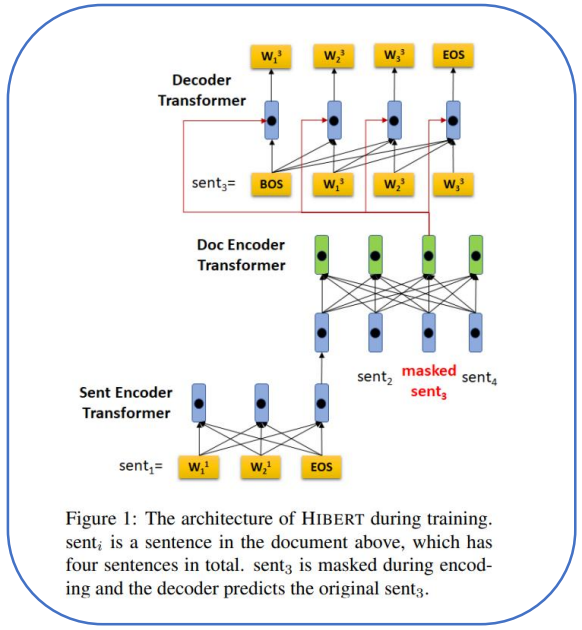


Figure 1: The architecture of HIBERT during training. $sent_i$ is a sentence in the document above, which has four sentences in total. $sent_3$ is masked during encoding and the decoder predicts the original $sent_3$.

Model	R-1	R-2	R-L
Pointer+Coverage	39.53	17.28	36.38
Abstract-ML+RL	39.87	15.82	36.90
DCA	41.69	19.47	37.92
SentRewrite	40.88	17.80	38.54
InconsisLoss	40.68	17.97	37.13
Bottom-Up	41.22	18.68	38.34
Lead3	40.34	17.70	36.57
SummaRuNNer	39.60	16.20	35.30
NeuSum	40.11	17.52	36.39
Refresh	40.00	18.20	36.60
NeuSum-MMR	41.59	19.01	37.98
BanditSum	41.50	18.70	37.60
JECS	41.70	18.50	37.90
LatentSum	41.05	18.77	37.54
HierTransformer	41.11	18.69	37.53
BERT	41.82	19.48	38.30
HIBERT$_S$ (in-domain)	42.10	19.70	38.53
HIBERT$_S$	**42.31**	**19.87**	**38.78**
HIBERT$_M$	**42.37**	**19.95**	**38.83**

Table 1: Results of various models on the CNNDM test set using full-length F1 ROUGE-1 (R-1), ROUGE-2 (R-2), and ROUGE-L (R-L).

Les transformeurs avec récurrence

1. Introduire de la récurrence dans les transformeurs

- (Dai et al. 2019) proposent de sérialiser le traitement des séquences en gardant en mémoire les valeurs d'activation (valeurs d'attention et couches cachées) du segment précédent.
- Ce modèle appelé Transformer-XL introduit aussi la notion de plongement de position relative.

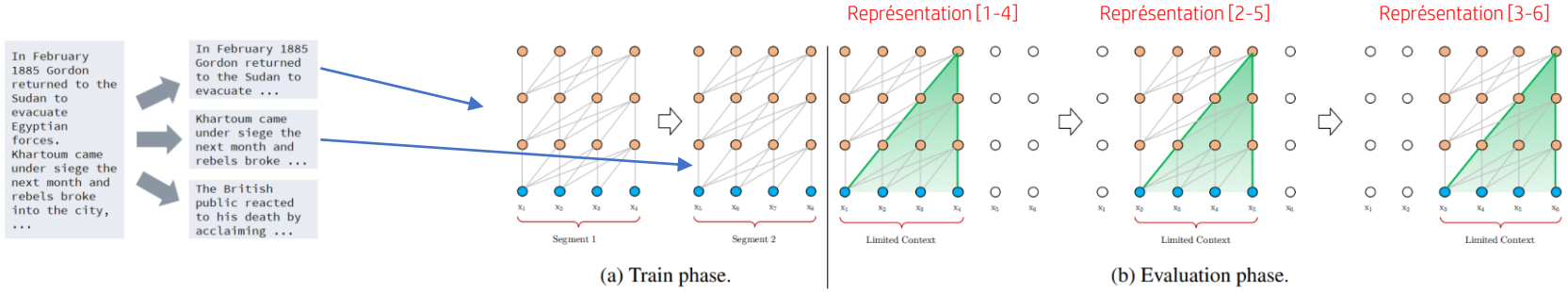


Figure 1: Illustration of the vanilla model with a segment length 4.

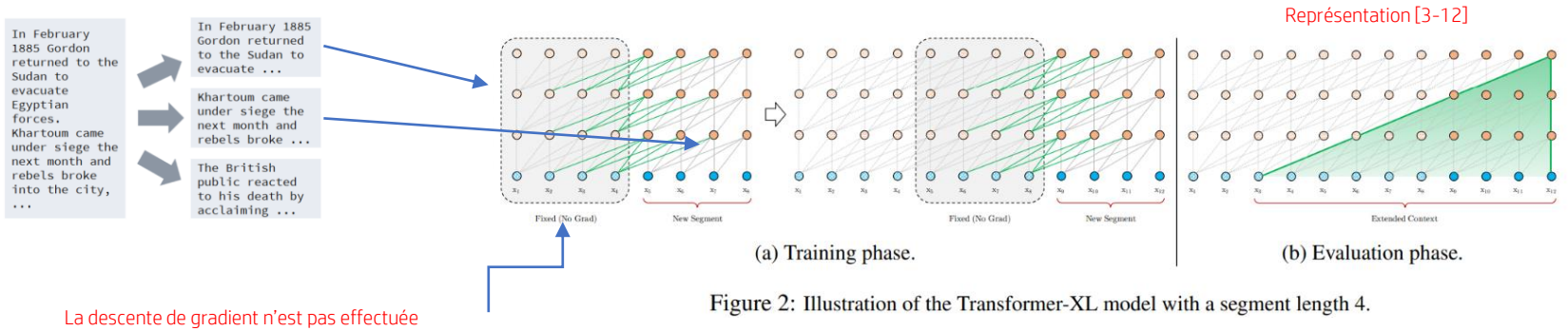


Figure 2: Illustration of the Transformer-XL model with a segment length 4.

La descente de gradient n'est pas effectuée

Model	#Param	PPL
Grave et al. (2016b) - LSTM	-	48.7
Bai et al. (2018) - TCN	-	45.2
Dauphin et al. (2016) - GCNN-8	-	44.9
Grave et al. (2016b) - LSTM + Neural cache	-	40.8
Dauphin et al. (2016) - GCNN-14	-	37.2
Merity et al. (2018) - QRNN	151M	33.0
Rae et al. (2018) - Hebbian + Cache	-	29.9
Ours - Transformer-XL Standard	151M	**24.0**
Baevski and Auli (2018) - Adaptive Input◇	247M	20.5
Ours - Transformer-XL Large	257M	**18.3**

Table 1: Comparison with state-of-the-art results on WikiText-103. ◇ indicates contemporary work.

Performance pour la tâche de « language modeling » (prédire le mot suivant) sur le jeu de données standard WikiText-103.

Les transformeurs avec récurrence

1. Introduire de la récurrence plus lointaine dans les transformeurs

- (Rae et al. 2019) proposent d'étendre la méthodologie de (Dai et al., 2019) afin de prendre en compte un historique plus lointain et ainsi de permettre une récurrence plus longue.
- Pour se faire, deux types de mémoire sont utilisés: une mémoire de court terme et une mémoire compressée.
- Ce modèle appelé « Compressive Transformer » introduit plusieurs fonctions de compression dont l'objectif est de minimiser la différence entre les attentions compressées et les attentions réelles de la mémoire. Cet objectif est optimisé dans un second temps car il est contre-productif avec l'apprentissage du modèle de langage.

Séquence précédente de la précédente comprimée mais accessible pour le calcul de la séquence actuelle

Séquence précédente comme dans (Dai et al., 2019)

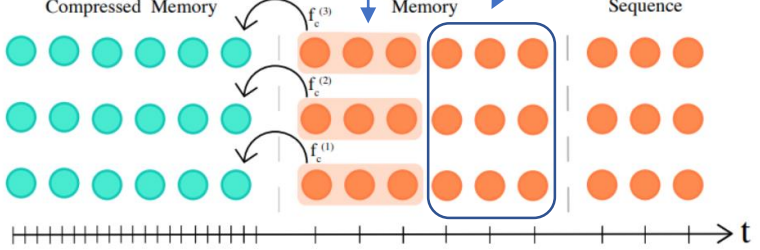


Figure 1: The Compressive Transformer keeps a fine-grained memory of past activations, which are then compressed into coarser *compressed* memories. The above model has three layers, a sequence length $n_s = 3$, memory size $n_m = 6$, compressed memory size $n_{cm} = 6$. The highlighted memories are compacted, with a compression function f_c per layer, to a single compressed memory — instead of being discarded at the next sequence. In this example, the rate of compression $c = 3$.

Table 6: Validation and test perplexities on WikiText-103.

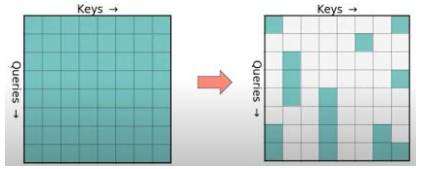
	Valid.	Test
LSTM (Graves et al., 2014)	-	48.7
Temporal CNN (Bai et al., 2018a)	-	45.2
GCNN-14 (Dauphin et al., 2016)	-	37.2
Quasi-RNN Bradbury et al. (2016)	32	33
RMC (Santoro et al., 2018)	30.8	31.9
LSTM+Hebb. (Rae et al., 2018)	29.0	29.2
Transformer (Baevski and Auli, 2019)	-	18.7
18L TransformerXL, M=384 (Dai et al., 2019)	-	18.3
18L TransformerXL, M=1024 (ours)	-	18.1
18L Compressive Transformer, M=1024	**16.0**	**17.1**

Table 7: WikiText-103 test perplexity broken down by word frequency buckets. The most frequent bucket is words which appear in the training set more than $10,000$ times, displayed on the left. For reference, a uniform model would have perplexity $|V| = 2.6e5$ for all frequency buckets. *LSTM comparison from Rae et al. (2018)

	> 10K	1K$-$10K	$100-1$K	< 100	All
LSTM*	12.1	219	1,197	9,725	36.4
TransformerXL (ours)	7.8	61.2	188	1,123	18.1
Compressive Transformer	**7.6**	**55.9**	**158**	**937**	**17.1**
Relative gain over TXL	2.6%	9.5%	21%	19.9%	5.8%

Les transformeurs avec patrons de contenus

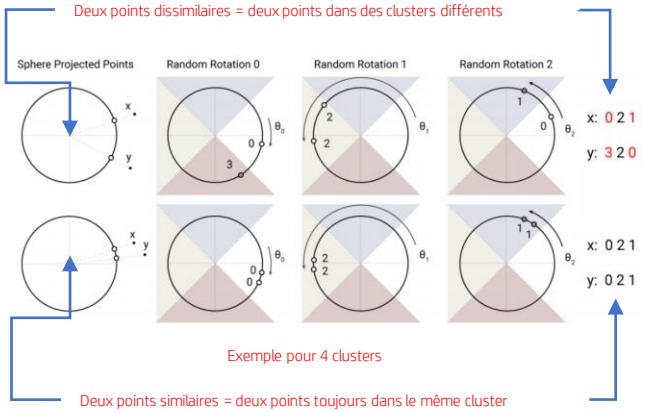
1. Minimiser le spectre d'attention
   - Plutôt que d'introduire de la récurrence, d'autres approches s'appliquent à réduire le spectre des attentions afin de diminuer la complexité de calcul et ainsi permettre d'augmenter la taille des textes en entrée.

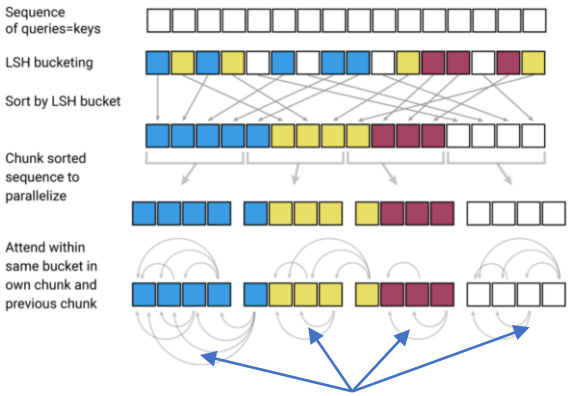


Spectre complet Spectre réduit

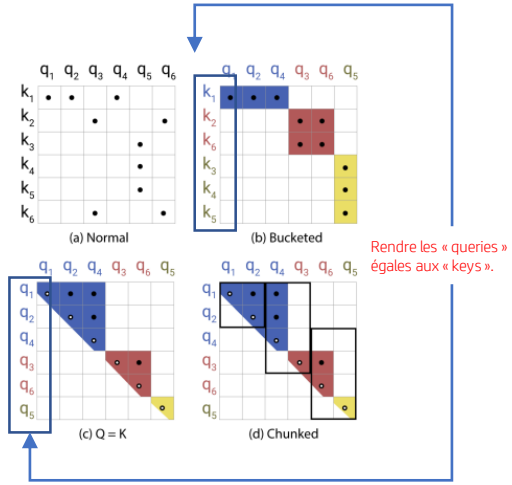
2. Minimiser le spectre d'attention par contenus
   - Les patrons de contenus ont pour objectif de comprendre les régularités dans les données en entrée et de ne focaliser le calcul de l'attention que sur les données en entrée similaires.

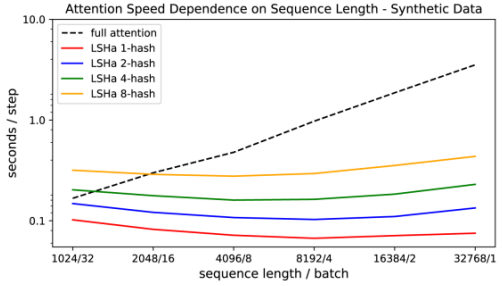
3. Le modèle Reformer
   - (Kitaev et al., 2020) se base sur la méthode de clustering LSH (« Locality Sensitive Hashing ») pour regrouper toutes les requêtes (« queries ») similaires et focaliser l'attention sur ces clusters.





L'attention de ne calcule qu'à l'intérieur d'un même cluster

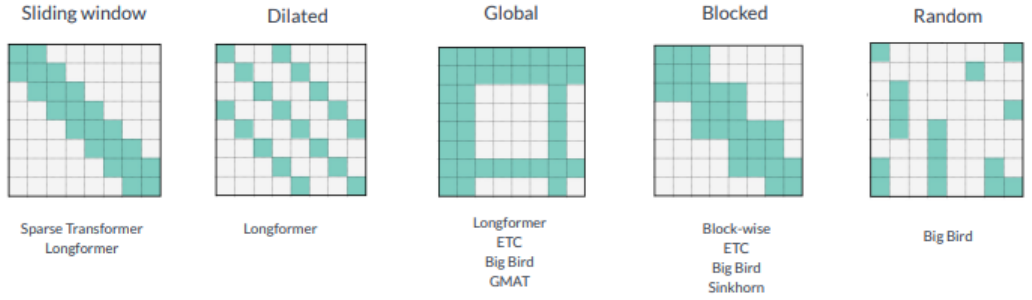




Les transformeurs avec patrons spécifiques

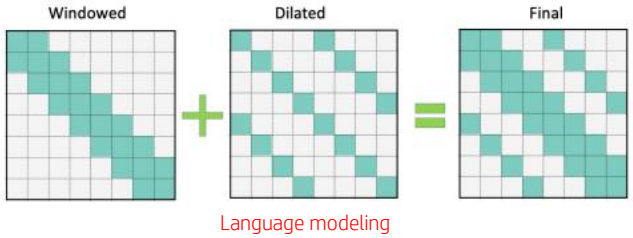
1. Minimiser le spectre d'attention par patrons

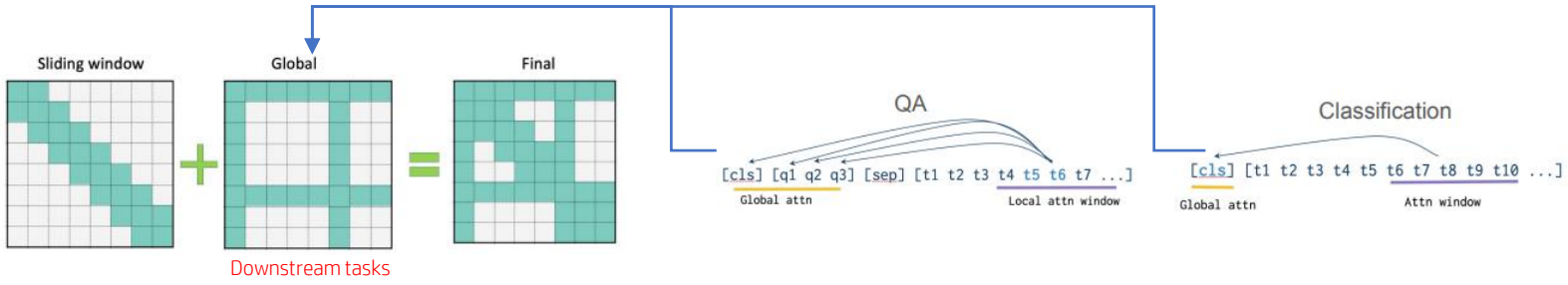
- Plutôt que d'apprendre des régularités sur les données, ce qui rajoute une étape supplémentaire, des patrons d'attention spécifiques peuvent être proposés. Ce sont des architectures basées modèle.



2. Le modèle LongFormer

- (Beltagy et al., 2020) propose plusieurs modèles suivant la tâche à apprendre.
- Aucune information globale n'est utilisée pour la modélisation du langage (« language modeling »).
- L'information globale est utilisée pour l'ajustement à certaines tâches (« downstream tasks »).
- Différents blocs et têtes d'attention peuvent avoir des modèles différents.

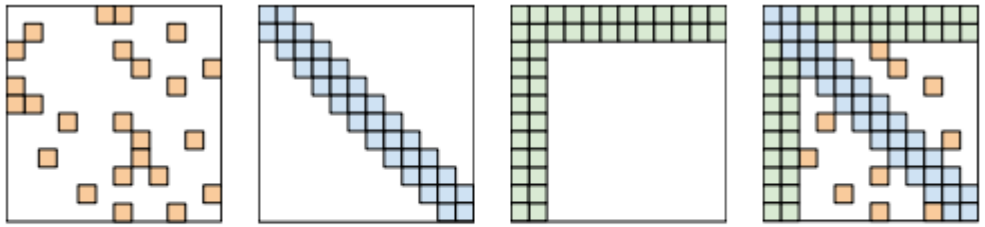




Les transformeurs avec patrons spécifiques

3. Le modèle BigBird-ETC

- (Zaheer et al., 2021) propose une alternative aux LongFormers en introduisant des modèles d'attention aléatoires combinés à des modèles à fenêtres et à attention globale.
- L'attention globale porte son attention de façon symétrique à certaines parties du texte seulement.



(a) Random attention (b) Window attention (c) Global Attention (d) BIGBIRD

Figure 1: Building blocks of the attention mechanism used in BIGBIRD. White color indicates absence of attention. (a) random attention with $r = 2$, (b) sliding window attention with $w = 3$ (c) global attention with $g = 2$. (d) the combined BIGBIRD model.

Model	MLM	SQuAD	MNLI
BERT-base	64.2	88.5	83.4
Random (R)	60.1	83.0	80.2
Window (W)	58.3	76.4	73.1
R + W	62.7	85.1	80.5
Global + R + W	64.4	87.2	82.9

Similaire à BERT pour une même taille d'entrée sans utiliser toutes les attentions.

Model	HotpotQA			NaturalQ		TriviaQA		WikiHop
	Ans	Sup	Joint	LA	SA	Full	Verified	MCQ
HGN [26]	**82.2**	88.5	**74.2**	-	-	-	-	-
GSAN	81.6	88.7	73.9	-	-	-	-	-
ReflectionNet [32]	-	-	-	77.1	**64.1**	-	-	-
RikiNet-v2 [61]	-	-	-	76.1	61.3	-	-	-
Fusion-in-Decoder [39]	-	-	-	-	-	84.4	90.3	-
SpanBERT [42]	-	-	-	-	-	79.1	86.6	-
MRC-GCN [87]	-	-	-	-	-	-	-	78.3
MultiHop [14]	-	-	-	-	-	-	-	76.5
Longformer [8]	81.2	88.3	73.2	-	-	77.3	85.3	81.9
BIGBIRD-ETC	81.2	**89.1**	73.6	**77.8**	57.9	**84.5**	**92.4**	**82.3**

Table 3: Fine-tuning results on **Test** set for QA tasks. The Test results (F1 for HotpotQA, Natural Questions, TriviaQA, and Accuracy for WikiHop) have been picked from their respective leaderboard. For each task the top-3 leaders were picked not including BIGBIRD-etc. **For Natural Questions Long Answer (LA), TriviaQA, and WikiHop, BIGBIRD-ETC is the new state-of-the-art**. On HotpotQA we are third in the leaderboard by F1 and second by Exact Match (EM).

COURS N°1

Représentation sémantique de texte

Questions supplémentaires?

GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

Plan de l'UE

1. **[CM 1]** Représentation sémantique de texte [GD]
2. **[CM 2]** Cohérence textuelle [MS]
3. **[CM 3]** Modélisation thématique [NA]
4. **[CM 4]** Résumé de textes et traduction automatique [MS]
5. **[CM 5]** Génération langagière I [KM]
6. **[CM 6]** Génération langagière II [KM]
7. **[CM 7]** TAL multimodal [NA]
8. **[CM 8]** TAL et web [MS]
9. **[CM 9]** TAL et handicap visuel [FM]
10. **[CM 10]** TAL et psychiatrie [GD]

11. **[TP 1-5]** Génération neuronal de comptes-rendus médicaux [NA - KM]

TRAITEMENT AUTOMATIQUE DES LANGUES AVANCE

Master Informatique

2ème Année - 1er Semestre

Intervenants CM:

Gaël DIAS (gael.dias@unicaen.fr), Marc SPANIOL, Fabrice MAUREL,

Navneet AGARWAL, Kirill MILINTSEVICH

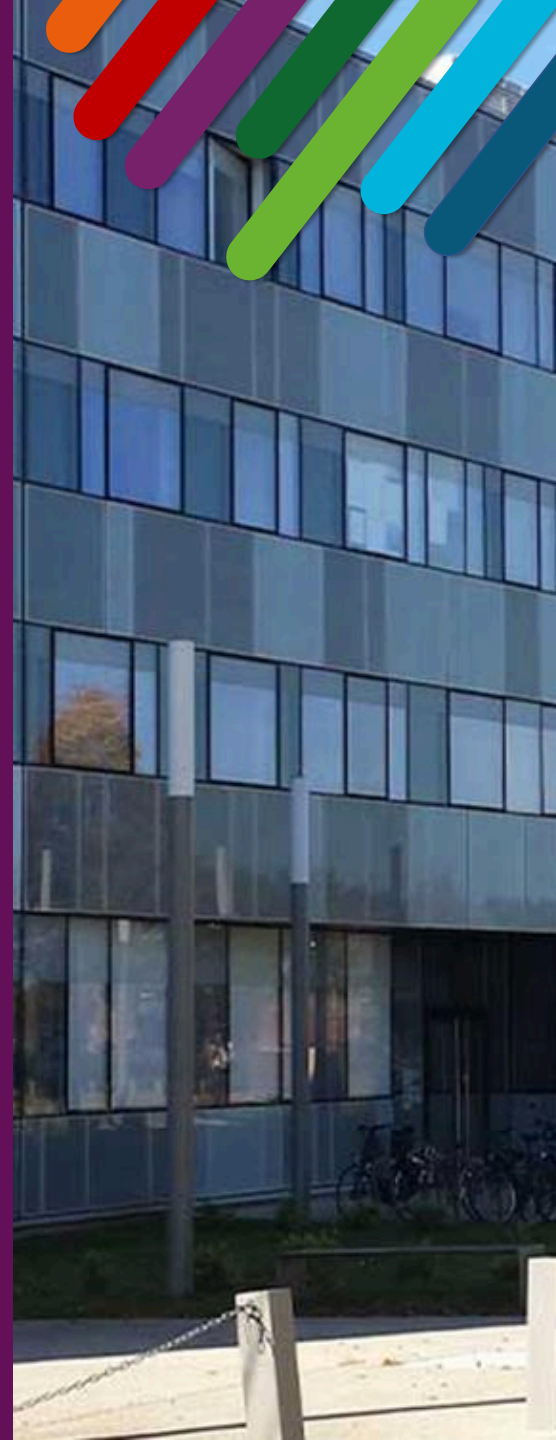
GREYC Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN ÉCOLE PUBLIQUE D'INGÉNIEURS CENTRE DE RECHERCHE

cnrs



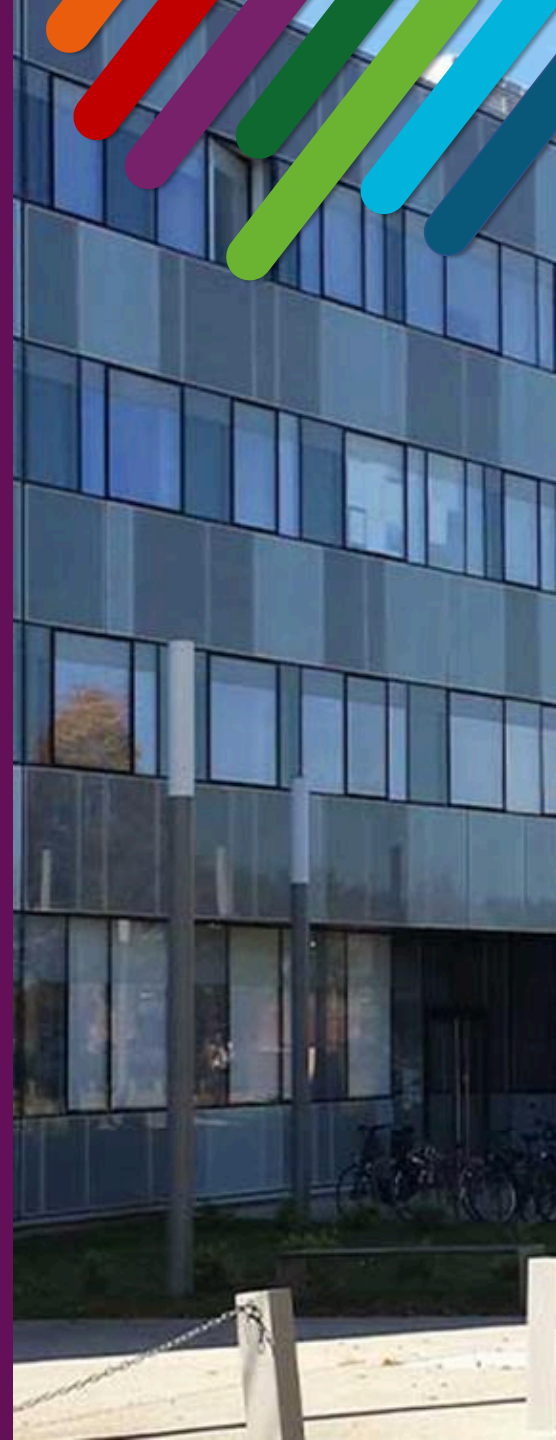
Plan de l'UE

1. **[CM 1]** Représentation sémantique de texte [GD]
2. **[CM 2]** Cohérence textuelle [MS]
3. **[CM 3]** Modélisation thématique [NA]
4. **[CM 4]** Résumé de textes et traduction automatique [MS]
5. **[CM 5]** Génération langagière I [KM]
6. **[CM 6]** Génération langagière II [KM]
7. **[CM 7]** TAL multimodal [NA]
8. **[CM 8]** TAL et web [MS]
9. **[CM 9]** TAL et handicap visuel [FM]
10. **[CM 10]** TAL et psychiatrie [GD]

11. **[TP 1-5]** Génération neuronal de comptes-rendus médicaux [NA - KM]

COURS N°1

Modélisation thématique



GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

Plan du cours

- Motivation
- Topic modeling
- Latent Dirichlet Allocation
- Gibbs Sampling
- Nonnegative Matrix Factorization
- Dynamic topic models
- Correlated topic models
- Structured topic models

Motivation

Suppose you are given a massive corpora and asked to carry out following tasks

- Carry out the initial exploratory analysis of the data
- Organise the documents into thematic categories
- Study how these topics evolved over time
- Find relationships between these categories

Topic Models

Topic models are statistical methods that analyze the words within original texts to discover the themes that run through them, study interactions between these themes and also how they evolve over time.

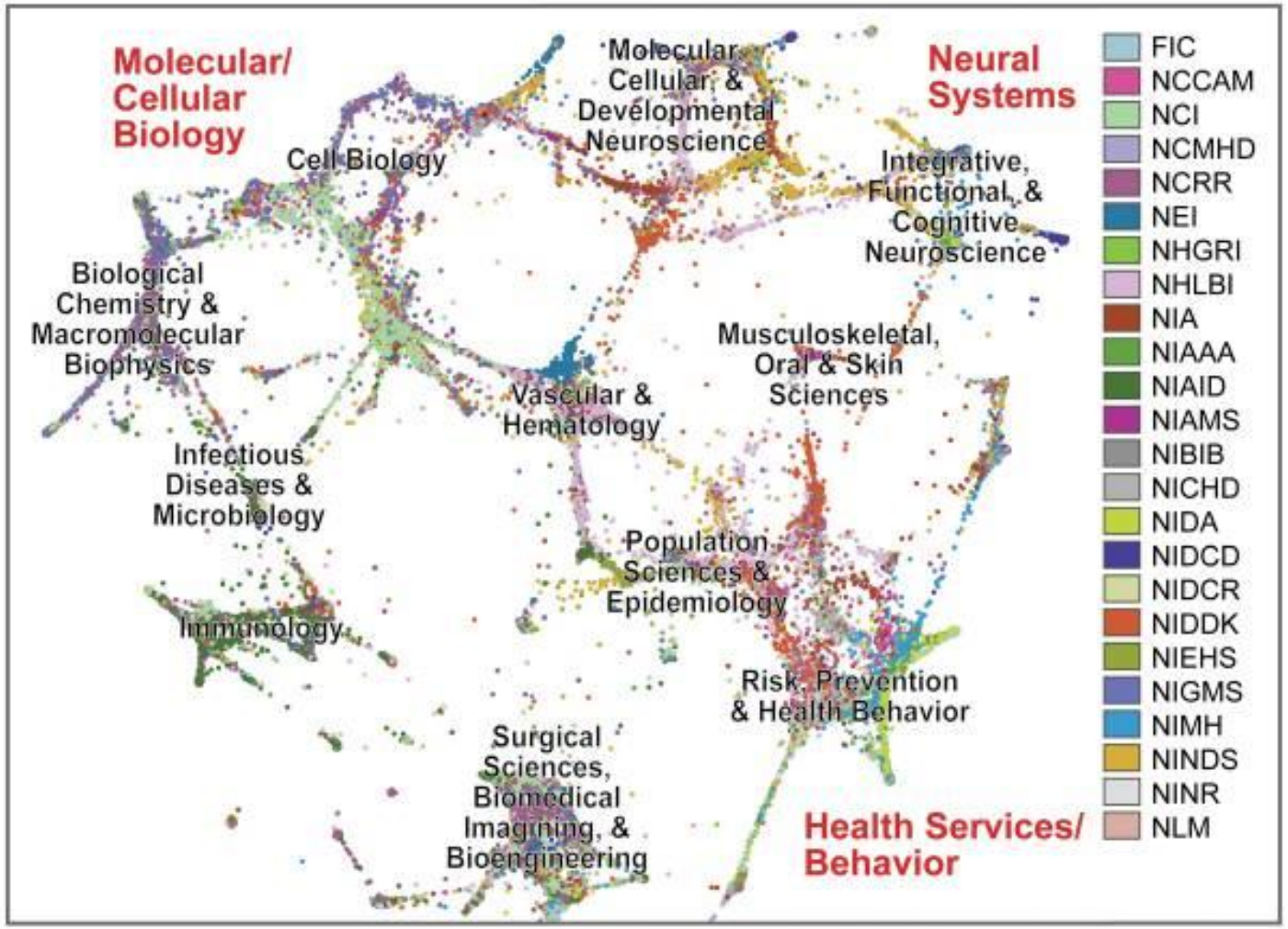
- Unsupervised methods that do not require prior annotations or labeling of documents
- Can be applied to massive collections of documents.
- Applied primarily to text corpora, but concepts are more general
- The topics emerge from the analysis of the original text without the need for human intervention in the learning process.

Topic Models

Map of National Institute of Health grants

year: 2010

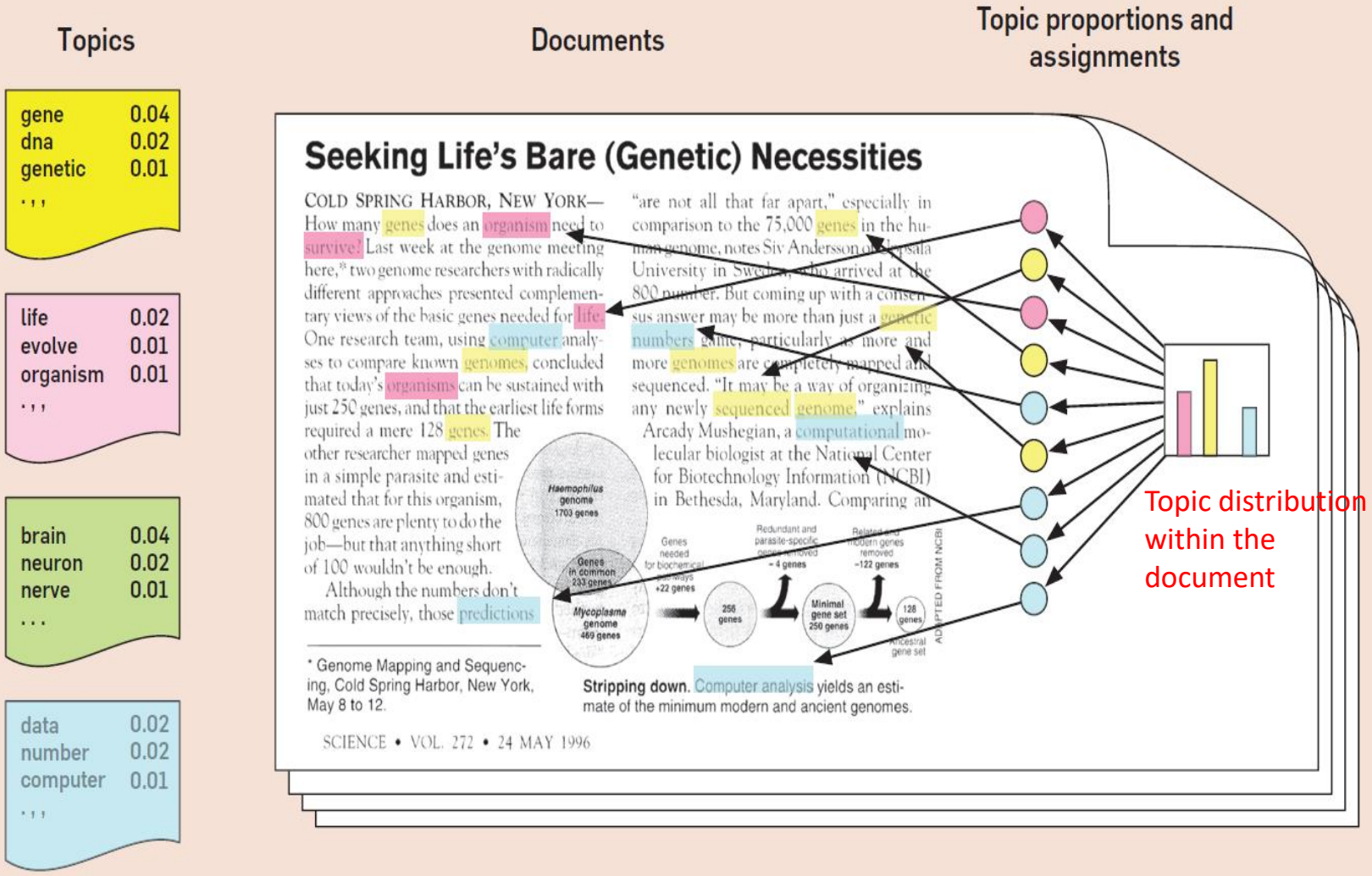
documents: 80,000



https://www.ncbi.nlm.nih.gov/pmc/articles/PMC5361216/

Latent Dirichlet Allocation

- Topics are defined to be a distribution over a fixed vocabulary (vocabulary of entire dataset).
- Each document is defined using distributon over topics and each topic is in-turn a distribution over words in the vocabulary
- Topic distribution defines the contribution of each topic towards the document



Blei et al. 2012

Latent Dirichlet Allocation

- Dimensionality reduction:
  - # documents: N
  - # words in vocabulary: V

documents can be represented using document-term matrix $\mathbb{R}^{N \times V}$

	W_1	W_2			W_V
D_1					
D_2					
D_N					

Assume T topics are learned from this data.

Documents are represented as distribution over topics $\mathbb{R}^{N \times T}$

	t_1	t_2			t_T
D_1					
D_2					
D_N					

- Unsupervised learning: can be compared to clustering.

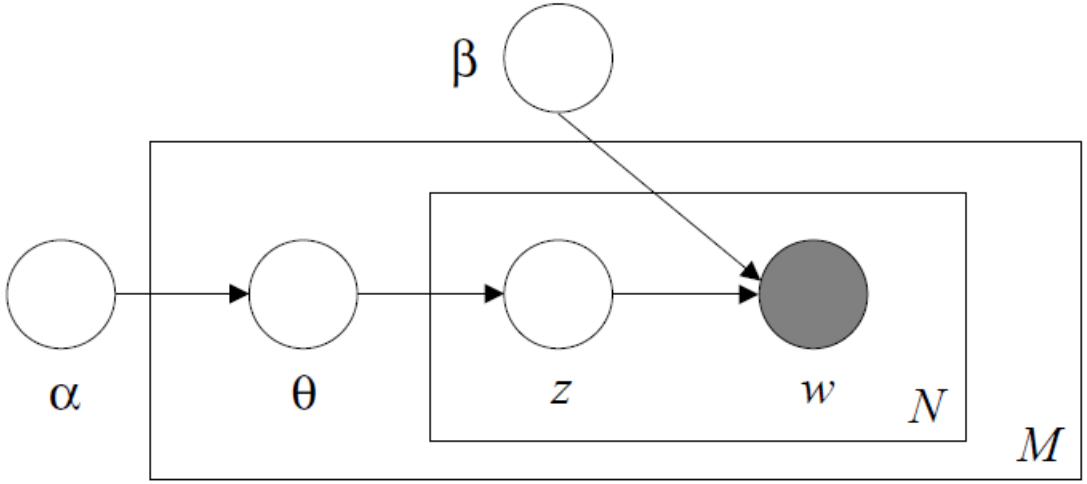
Words are clustered together to form topics based on their co-occurance patterns

Documents are clustered based on their topic distributions.

Blei et al., JMLR 2003

Latent Dirichlet Allocation

Latent Dirichlet Allocation (LDA) is a generative probablistic model of a corpus. Within LDA, documents are represented as random mixtures over latent topics, where each topic is characterized by a distribution over words.



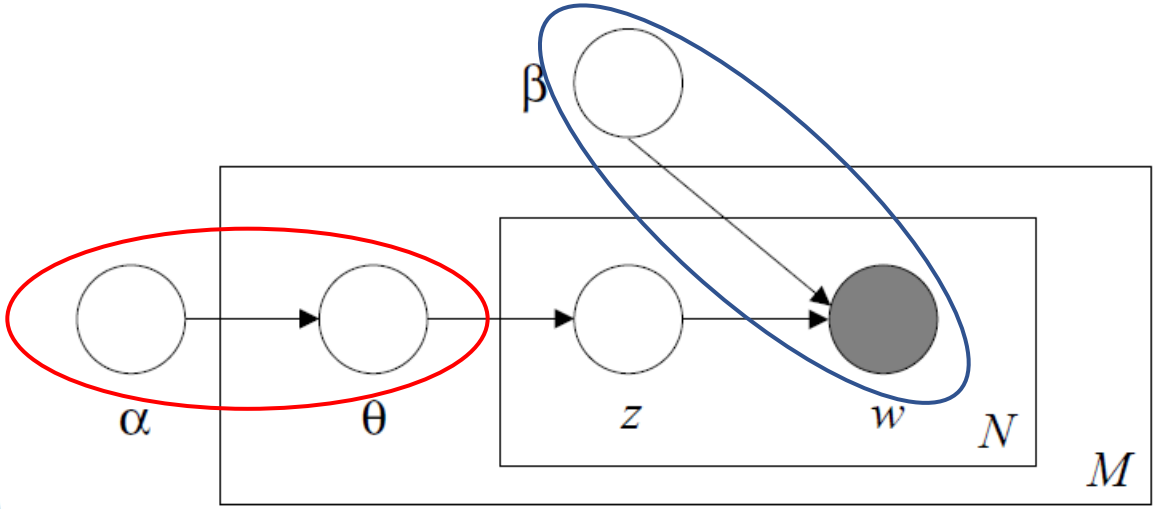
Lets assume we want to generate M documents:

1. Choose N: number of words in the document
2. Choose θ ~Dir(α): topic proportions for document w
3. For each of the N words:
   a) Choose a topic z_n ~ Multinomial(θ): topic assignment for document w
   b) Choose a word w_n from $p(w|z_n, \beta)$

Latent Dirichlet Allocation

Given α and β, the joint distribution of a topic mixture θ, set of *N* topics *z*, and *N* words *w is given by:*

$$p(\theta, \mathbf{z}, \mathbf{w} \mid \alpha, \beta) = p(\theta \mid \alpha) \prod_{n=1}^{N} p(z_n \mid \theta) p(w_n \mid z_n, \beta),$$

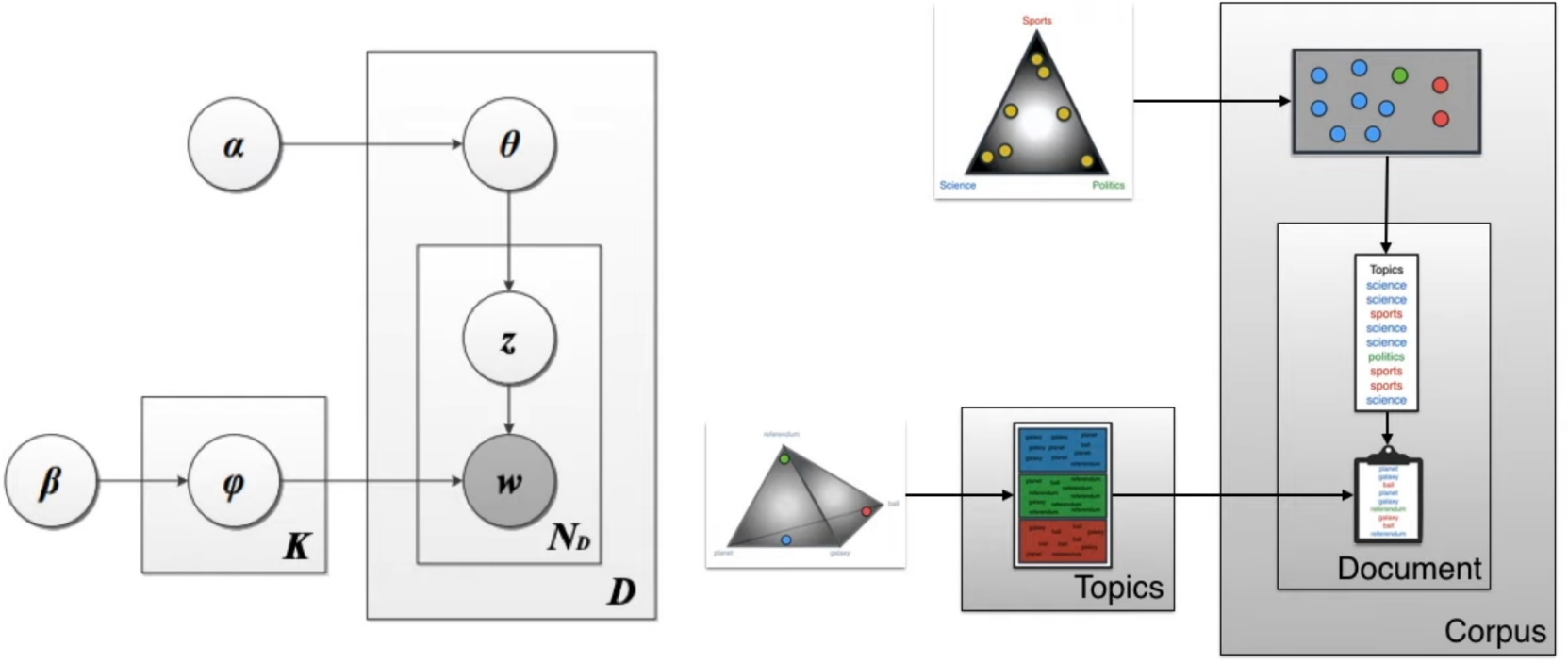


Integrating over θ and summing over *z*, we obtain the marginal distribution of a document:

$$p(\mathbf{w} \mid \alpha, \beta) = \int p(\theta \mid \alpha) \left(\prod_{n=1}^{N} \sum_{z_n} p(z_n \mid \theta) p(w_n \mid z_n, \beta) \right) d\theta.$$

Finally, taking product of marginal probabilities of single documents, we obtain the probability of a corpus:

$$p(D \mid \alpha, \beta) = \prod_{d=1}^{M} \int p(\theta_d \mid \alpha) \left(\prod_{n=1}^{N_d} \sum_{z_{dn}} p(z_{dn} \mid \theta_d) p(w_{dn} \mid z_{dn}, \beta) \right) d\theta_d.$$


α
θ
z
β
φ
w
K
N_D
D
Sports
Science
Politics
Topics
science
science
sports
science
science
politics
sports
sports
science
Topics
Document
Corpus

Latent Dirichlet Allocation

LDA is part of a larger field of *probabilistic modeling*.

We treat the data as arising from a generative process that includes *hidden variables*.

This generative process defines a joint probability distribution over both the observed and hidden variables

We perform data analysis by using joint distribution to compute conditional distribution of the hidden variables given the observed variables.

This conditional distribution is called *posterior distribution*.

observed variables: w

hidden variables: θ, z, β

$$p(\theta, z, \beta | w) = \frac{p(\theta, z, \beta, w)}{p(w)}$$

Latent Dirichlet Allocation

- The posterior cannot be computed because the denominator is intractable.
- Topic modeling algorithms form an approximation of the equation by adapting an alternative distribution over the latent topic structures to be close to the true posterior.
- Topic modeling algorithms generally fall into two categories:
  - Sampling based algorithms
  - Variational algorithms

- The most commonly used sampling algorithm for topic modeling is *Gibbs Sampling*

Gibbs Sampling

- Gibbs sampling procedure considers each word token in the text collection in turn.
- Estimate the probability of assigning the current word token to each topic, conditioned on the topic assignments to all the other word tokens.

$$P(z_i = j \mid \mathbf{z}_{-i}, w_i, d_i, \cdot) \propto \frac{C^{WT}_{w_i j} + \beta}{\sum_{w=1}^{W} C^{WT}_{wj} + W\beta} \frac{C^{DT}_{d_i j} + \alpha}{\sum_{t=1}^{T} C^{DT}_{d_i t} + T\alpha}$$

- From this conditional distribution, a topic is sampled and stored as the new assignment for this word token.

Griffiths and Steyvers (2004)

Gibbs Sampling

- n_{dk}: number of words assigned to topic k in document d
- n_{kw}: number of times word w is assigned to topic k
- n_k: number of times any word is assigned to topic k

Input: words $\mathbf{w} \in$ documents $\mathbf{d}$
Output: topic assignments $\mathbf{z}$ and counts $n_{d,k}, n_{k,w}$, and n_k
begin
  randomly initialize $\mathbf{z}$ and increment counters
  foreach *iteration* **do**
    for $i = 0 \rightarrow N - 1$ **do**
      $word \leftarrow w[i]$
      $topic \leftarrow z[i]$
      $n_{d,topic}$-=1; $n_{word,topic}$-=1; n_{topic}-=1
      for $k = 0 \rightarrow K - 1$ **do**
        $p(z = k|\cdot) = (n_{d,k} + \alpha_k)\frac{n_{k,w}+\beta_w}{n_k+\beta \times W}$
      end
      $topic \leftarrow$ sample from $p(z|\cdot)$
      $z[i] \leftarrow topic$
      $n_{d,topic}$+=1; $n_{word,topic}$+=1; n_{topic}+=1
    end
  end
  return $\mathbf{z}$, $n_{d,k}, n_{k,w}, n_k$
end

Gibbs Sampling

Example

Assume we have some document with random word-topic assignment

India	enters	world	cup	final
1	3	1	2	4

We have count matrix C^{WT}

	1	2	3	4
India	70	5	0	8
enters	2	3	15	6
world	28	4	12	1
cup	6	43	6	0
final	7	0	9	31

Gibbs Sampling

Example

Assume we have some document with random word-topic assignment

India	enters	world	cup	final
1	3	1	2	4

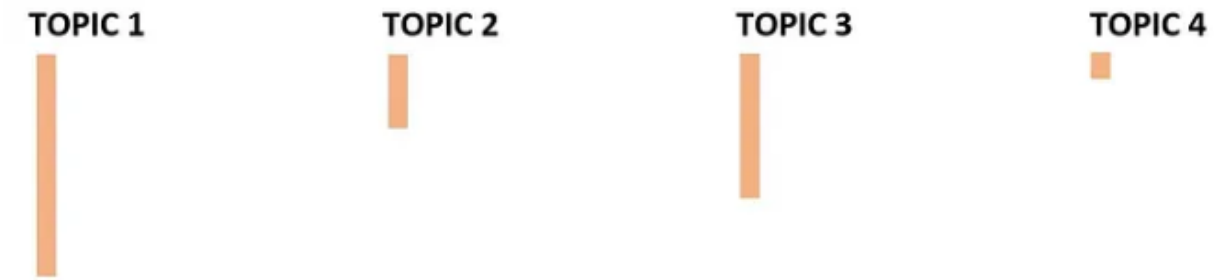
We have count matrix C^{WT}

	1	2	3	4
India	70	5	0	8
enters	2	3	15	6
world	28	4	12	1
cup	6	43	6	0
final	7	0	9	31

Gibbs Sampling

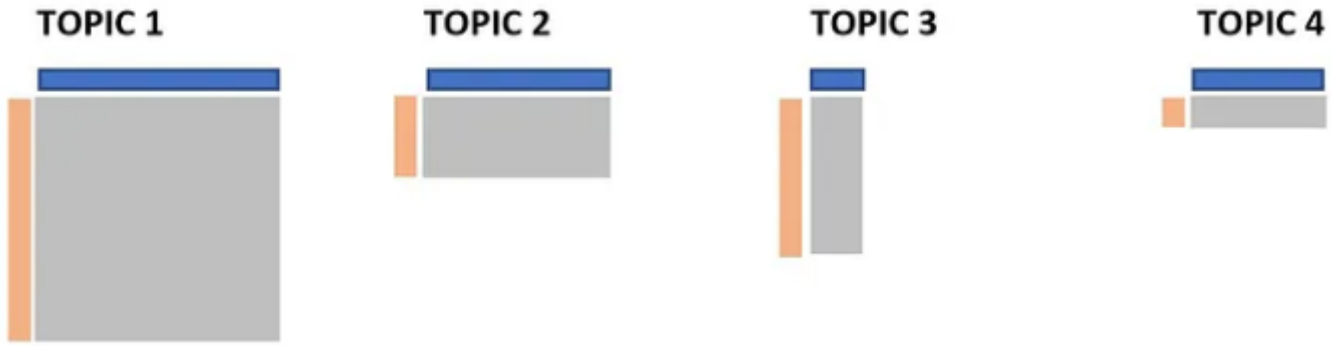
- *Consider the conribution of each topic towards this document*

- *Next, we take how many times each topic is assigned to this word*

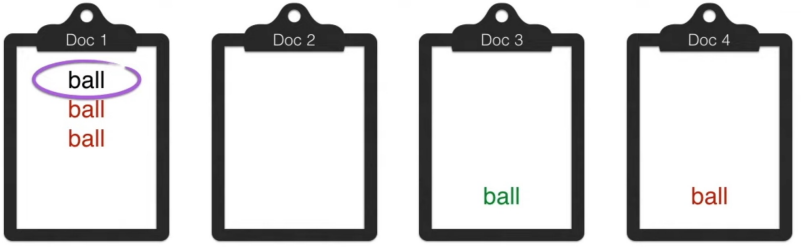


Gibbs Sampling

- *Multiply these values to get conditional probabilities*



- *Finally, pick one of the topics from this distribution and update the variables accordingly.*
- *Repeat this for every word.*



Topic 1

How much is Topic 1 in Doc 1?

2

How much is ‘ball’ in Topic 1?

0

Topic 2

How much is Topic 2 in Doc 1?

0

How much is ‘ball’ in Topic 2?

1

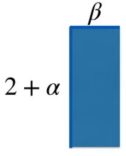
Topic 3

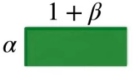
How much is Topic 3 in Doc 1?

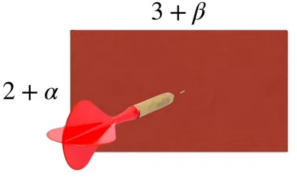
2

How much is ‘ball’ in Topic 3?

3







Topic 1

How much is Topic 1 in Doc 1?

$2+\alpha$

How much is ‘ball’ in Topic 1?

$0+\beta$

Topic 2

How much is Topic 2 in Doc 1?

$0+\alpha$

How much is ‘ball’ in Topic 2?

$1+\beta$

ball

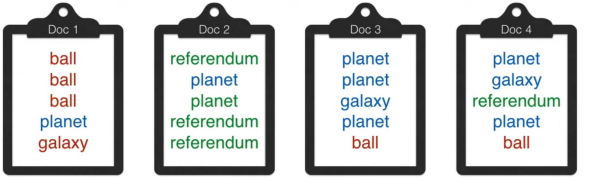
Topic 3

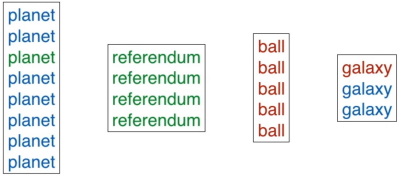
How much is Topic 3 in Doc 1?

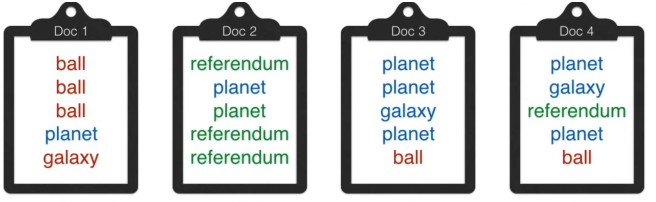
$2+\alpha$

How much is ‘ball’ in Topic 3?

$3+\beta$





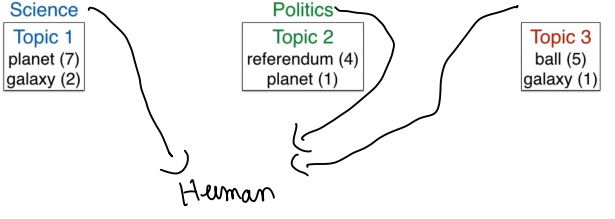


80% Topic 3
20% Topic 1

80% Topic 2
20% Topic 1

80% Topic 1
20% Topic 3

60% Topic 1
20% Topic 2
20% Topic 3



Nonnegative Matrix Factorization (NMF)

1. Nonnegative Matrix Factorization (NMF) factors input nonnegative matrix V into two nonnegative matrices W and H.

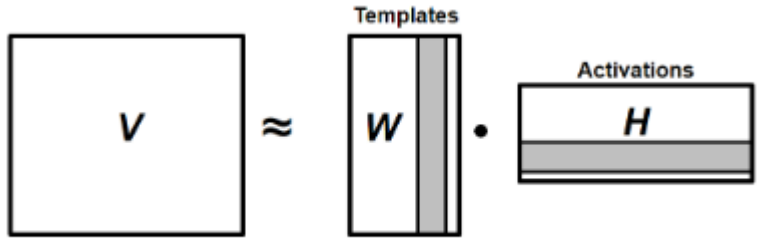


Figure 8.20b from [Müller, FMP, Springer 2015]

$$V \approx W \cdot H.$$

V×D V×T T×D

- W and H are required to have much lower rank that the original matrix V.
- Columns of V contains V-dimensional data vectors
- Columns of W are the word distribution for each topic.
- Rows of H are the activation of given topic across all documents.
- In most cases the factorization does not have an exact solution and requires optimization procedures to find numerical approximations.

$$||V - WH||^2$$

LDA vs NMF

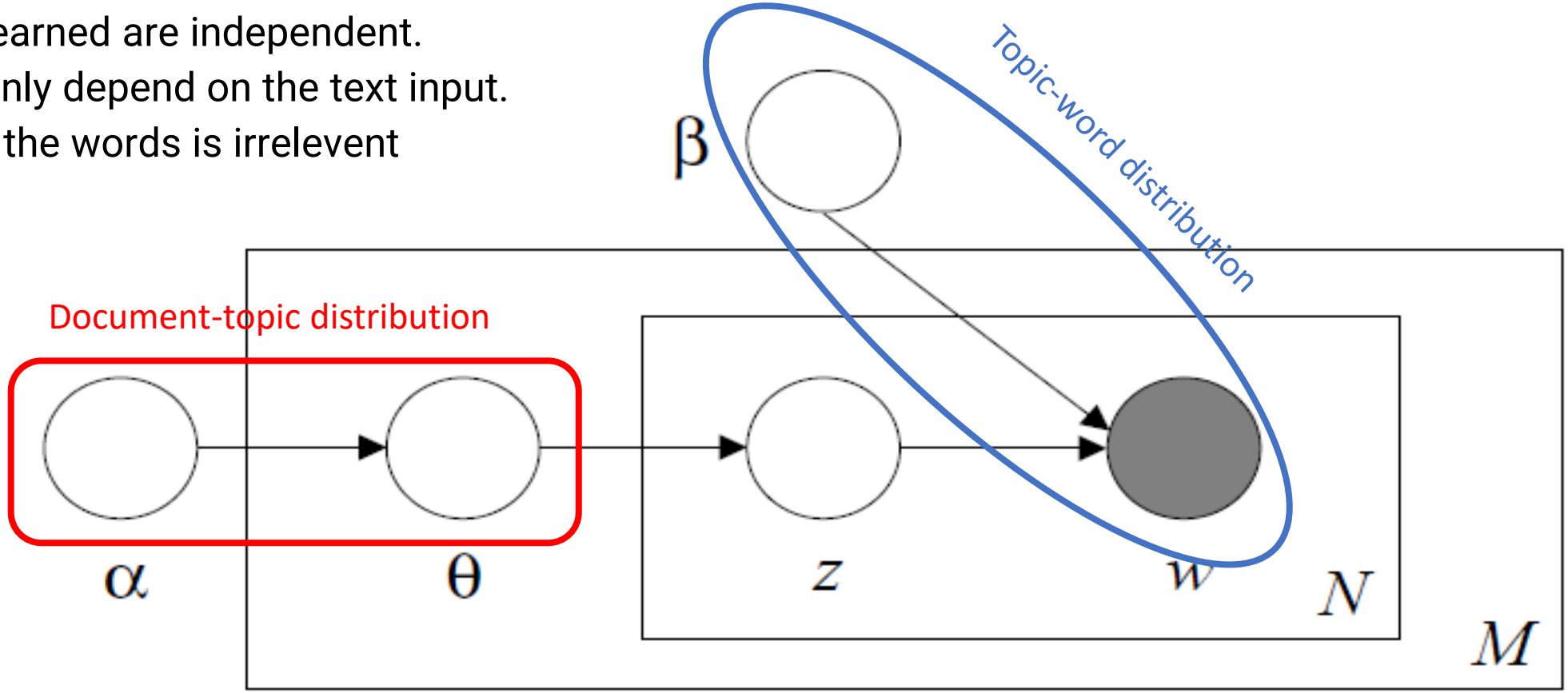
- LDA is probabilistic while NMF uses matrix factorization.
- LDA extracts independent topics from word distributions. Therefore, topics that are dissimilar in the document may not be identified seperately.
- NMF learns dissimilar topics, but can cause difficulties in interpreting findings.
- NMF usually performs better with short texts, like social media data.
- For both LDA and NMF, the results are highly dependent on hyperparameter tuning.

LDA vs NMF

	LDA		NMF	
No.	Topic/content	Keywords	Topic/content	Keywords
1	Government response	ban, travelgov, potus, dv2021, loveisnottourism, whcovidresponse, end, visa, please, vp	Government response	whcovidresponse, potus, loveisnottourism, cdcdirector, presssec, vp, cdctravel, cdcgov, liftthetravelban, cdctravel cdcdirector
2	Association for Molecular Pathology (AMP) / mask and virus	amp, travel, come, spread, mask, place, follow, stay, keep, virus	Association for Molecular Pathology (AMP) / desire to travel	covid, travel, people, amp, want, covid travel, time, travel covid, like, year
3	R_t value / India, UK, Europe	rt, travel, country, India, uk, covid, government, list, eu, news	R_t value	rt, covid, travel, https, covid19, traveler, rt ollysmithtravel, traveler, httpstco, ollysmithtravel
4	Travel restriction / England and Scotland	travel, covid, restriction, city, team, England, despite, event, expect, Scotland	Travel restriction	restriction, travel restriction, covid travel, covid19 travel, ease, covid restriction, travel, lift, covid19 restriction, restriction lift
5	Vaccination / border between Canada and the USA	vaccinate, covid19, international, traveler, travel, vaccination, Canada, border, US, fully	Travel ban / India and UK	ban, India, travel ban, travel India, uk, list, country, ban travel, red, variant
6	Quarantine and lockdown / Australia	traveler, day, quarantine, variant, allow, return, lockdown, Australia, break, two	General about travel / Canada	covid19, travel, covid19 travel, international, travel covid19, country, pandemic, international travel, vaccination, Canada
7	COVID-19 cases / USA	case, new, travel, health, state, tourism, public, number, close, include	Vaccination and quarantine	vaccinate, fully, fully vaccinate, vaccinate covid19, traveler, vaccinate traveler, traveler, quarantine, cdc, require
8	Flight / COVID-19 test	test, travel, need, positive, covid, flight, negative, air, take, airport	COVID-19 cases / New Zealand	case, new, covid case, covid19 case, new case, rise, Zealand, New Zealand, report, case covid19
9	Death / Florida	covid, die, death, cause, florida, child, spike, shoot, traveler002, flu	COVID-19 test	test, covid test, negative, positive, test travel, test positive, PCR, covid19 test, day, result
10	China and USA	travel, covid, call, china, business, 2020, trump, usa, dr	Vaccination pass	vaccine, covid19 vaccine, covid vaccine, passport, vaccine passport, require, vaccine travel, dose, mandate, vaccination
11	Unspecific I	not, covid, vaccine, people, do, travel, get, make, still, would		
12	Unspecific II	travel, may, covid, 2, please, 1, help, show, 3, pass		
13	Unspecific III	covid19, travel, due, pandemic, world, today, first, update, coronavirus, safe		
14	Unspecific IV	covid, be, go, travel, time, get, want, one, year, see		

Topic model variations

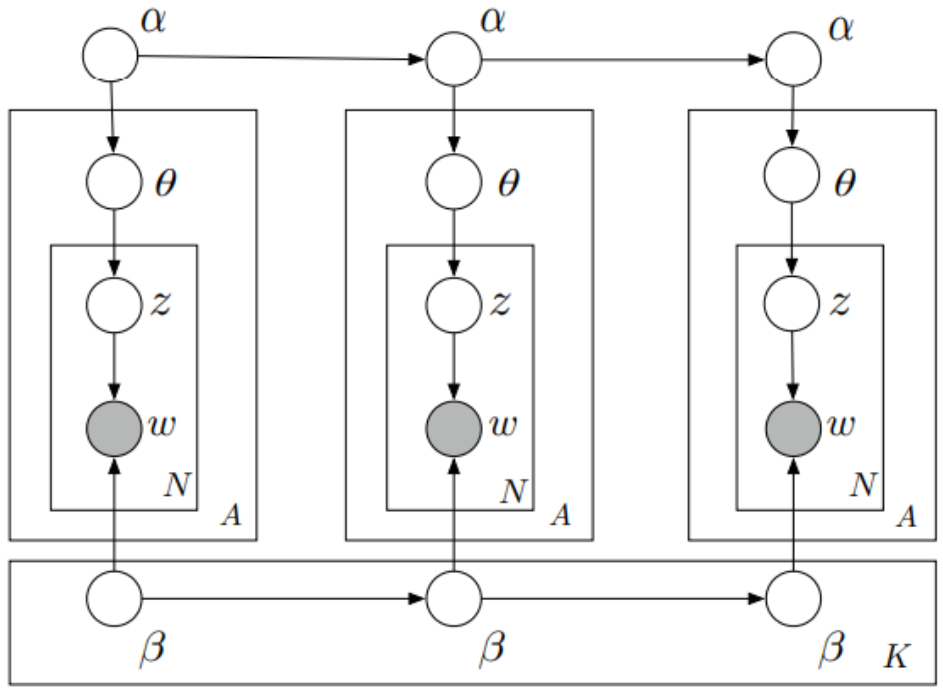
1. Order of documents does not matter.
2. Topics learned are independent.
3. Topics only depend on the text input.
4. Order of the words is irrelevent



Blei, David M., and John D. Lafferty. "Dynamic topic models." *Proceedings of the 23rd international conference on Machine learning*. 2006.

Topic model variations

1. Dynamic topic model
   - LDA assumes that order of documents does not matter.
   - This assumption may be unrealistic when considering long running collections that span years or centuries.
   - Dynamic topic model solves this problem by dividing documents based on time slots.
   - Topic models learned for each time slot are dependent on respective topic from previous time slot.

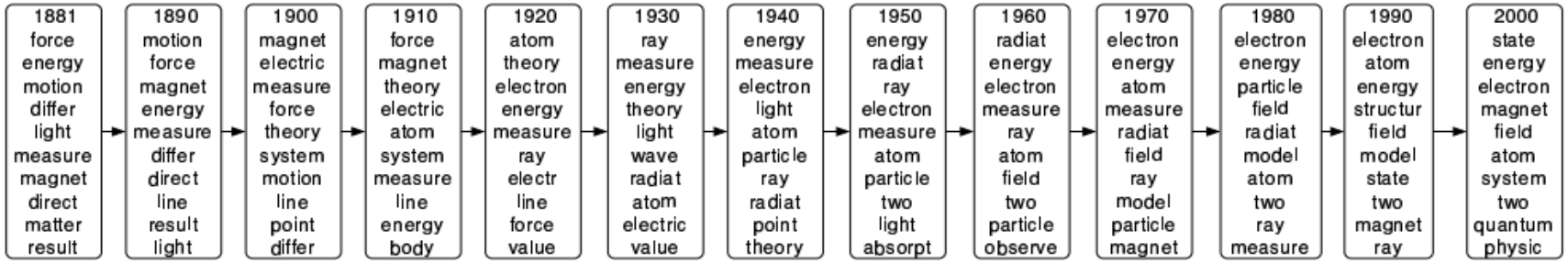


1. Draw topics $\beta_t \mid \beta_{t-1} \sim \mathcal{N}(\beta_{t-1}, \sigma^2 I)$.
2. Draw $\alpha_t \mid \alpha_{t-1} \sim \mathcal{N}(\alpha_{t-1}, \delta^2 I)$.
3. For each document:
   (a) Draw $\eta \sim \mathcal{N}(\alpha_t, a^2 I)$
   (b) For each word:
      i. Draw $Z \sim Mult(\pi(\eta))$.
      ii. Draw $W_{t,d,n} \sim Mult(\pi(\beta_{t,z}))$.

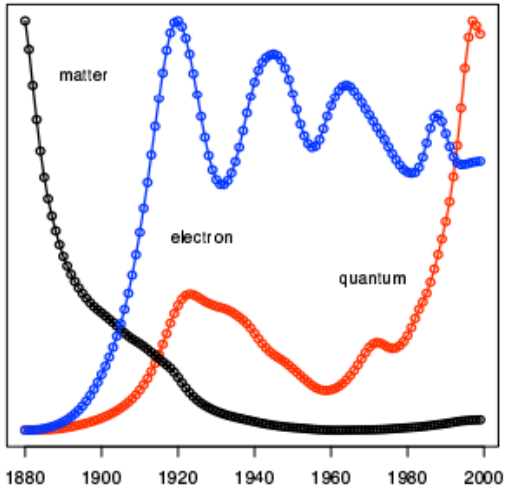
Blei, David M., and John D. Lafferty. "Dynamic topic models." *Proceedings of the 23rd international conference on Machine learning*. 2006.

Topic model variations

1. Dynamic topic model
   - Subset of 30,000 articles from *Science*, 250 from each of the 120 years between 1881 and 1999



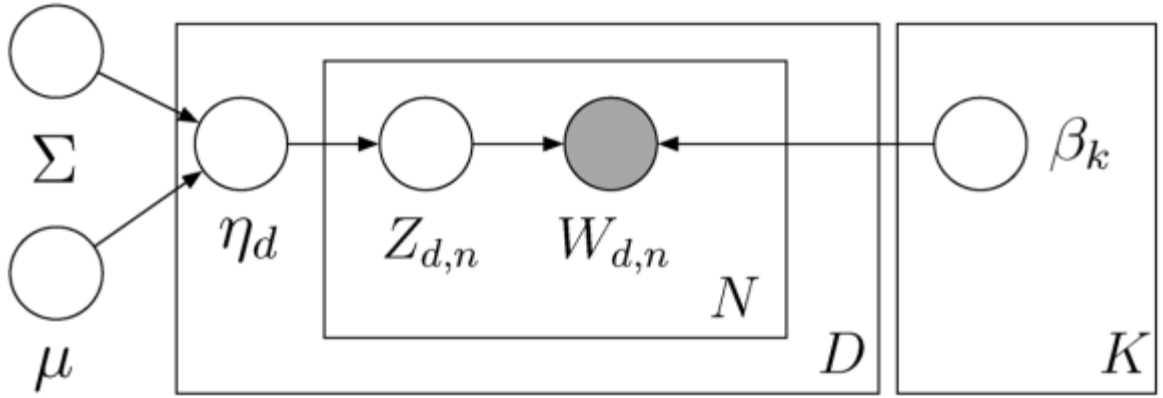
"Atomic Physics"



1881 On Matter as a form of Energy
1892 Non-Euclidean Geometry
1900 On Kathode Rays and Some Related Phenomena
1917 ``Keep Your Eye on the Ball''
1920 The Arrangement of Atoms in Some Common Metals
1933 Studies in Nuclear Physics
1943 Aristotle, Newton, Einstein. II
1950 Instrumentation for Radioactivity
1965 Lasers
1975 Particle Physics: Evidence for Magnetic Monopole Obtained
1985 Fermilab Tests its Antiproton Factory
1999 Quantum Computing with Electrons Floating on Liquid Helium

Topic model variations

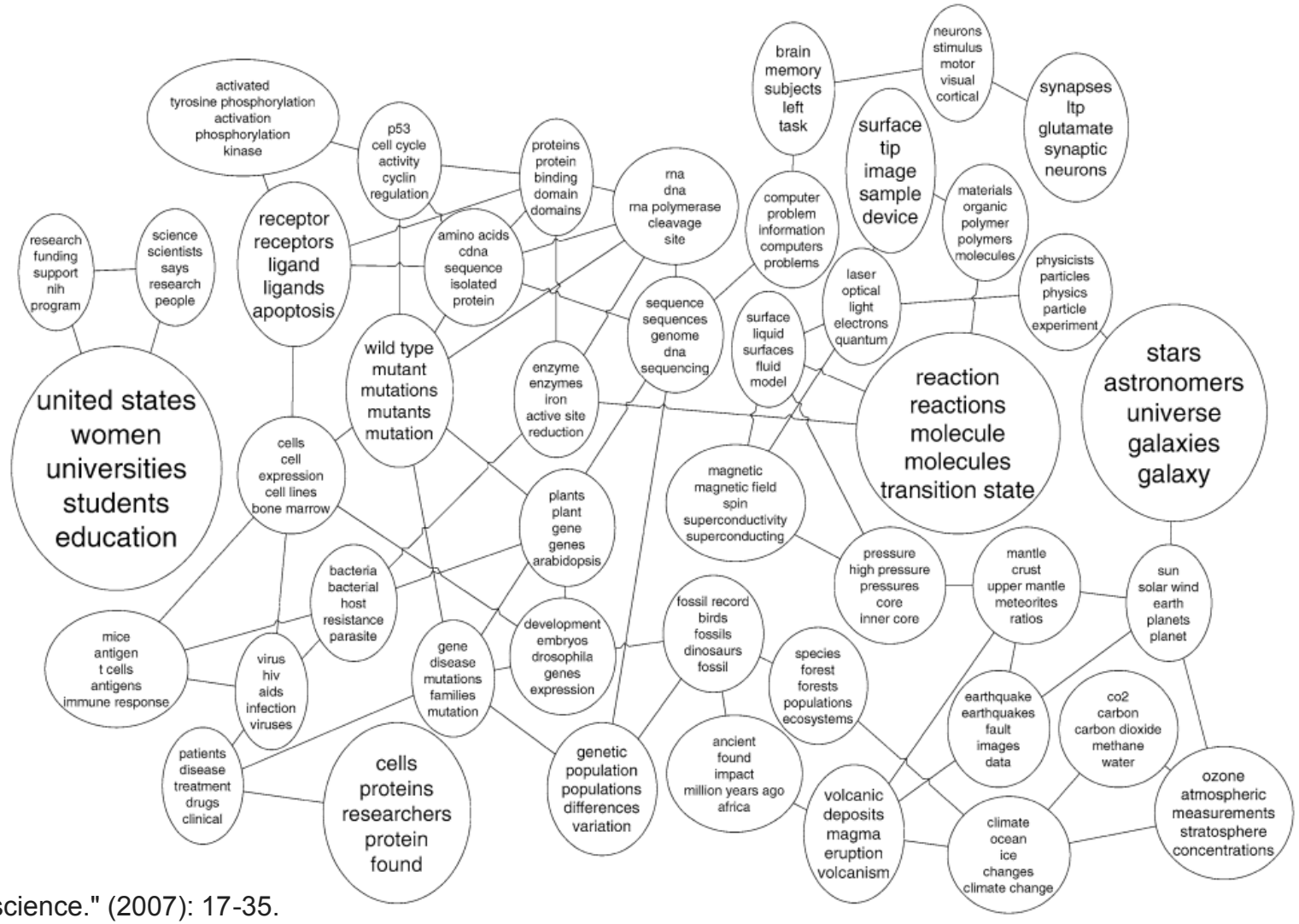
1. Correlated topic model
   - Within LDA, topics are sampled from a Dirichlet distribution are independent which is not realistic for real document collections.
   - CTM draws real values random vectors from multivariate Gaussian distribution inducing dependencies between the components.



Blei, David M., and John D. Lafferty. "A correlated topic model of science." (2007): 17-35.

Topic model variations

1. Correlated topic model
   - Topic graph learned from 16,351 OCR articles from science (1990-1999)



Blei, David M., and John D. Lafferty. "A correlated topic model of science." (2007): 17-35.

Topic model variations

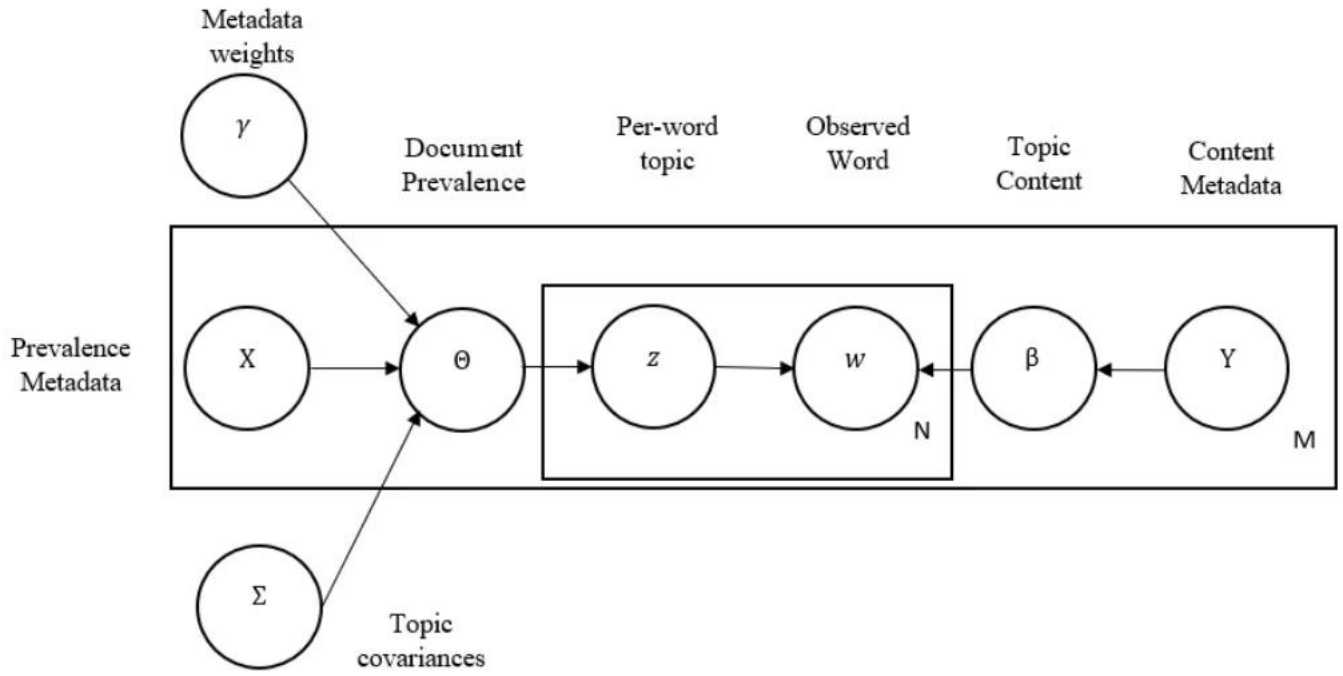
1. Structured Topic Model (STM)

- LDA learns only based on input text.
- Certain sources may be more likely to write about politics.
- Metadata can include date published, author, publication, likes on social media, etc.
- Within LDA our topic distribution comes from Dirichlet distrubution.
- STM defines topic distributions based on document metadata
- We need to go from X_i, *1xp* metadata vector to *1xk* vector of topic distribution.
- We multiply X with pxk weight matrix t.

Les transformeurs avec récurrence

1. Introduire de la récurrence dans les transformeurs
   - (Dai et al. 2019) proposent de sérialiser le traitement des séquences en gardant en mémoire les valeurs d'activation (valeurs d'attention et couches cachées) du segment précédent.
   - Ce modèle appelé Transformer-XL introduit aussi la notion de plongement de position relative.

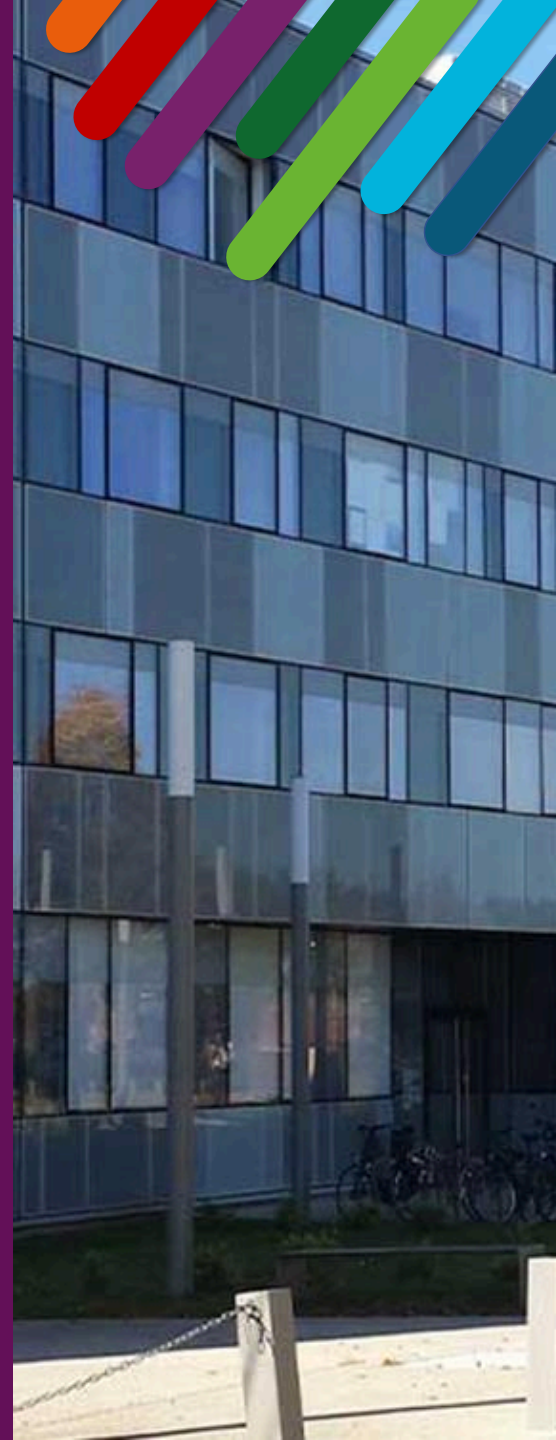
$$\theta_i \sim \text{LogisticNormal}(\tau X_i, \Sigma)$$



COURS N°1

Modélisation thématique

Questions supplémentaires?



GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

Plan de l'UE

1. **[CM 1]** Représentation sémantique de texte [GD]
2. **[CM 2]** Cohérence textuelle [MS]
3. **[CM 3]** Modélisation thématique [NA]
4. **[CM 4]** Résumé de textes et traduction automatique [MS]
5. **[CM 5]** Génération langagière I [KM]
6. **[CM 6]** Génération langagière II [KM]
7. **[CM 7]** TAL multimodal [NA]
8. **[CM 8]** TAL et web [MS]
9. **[CM 9]** TAL et handicap visuel [FM]
10. **[CM 10]** TAL et psychiatrie [GD]

11. **[TP 1-5]** Génération neuronal de comptes-rendus médicaux [NA - KM]

TRAITEMENT AUTOMATIQUE DES LANGUES AVANCE

Master Informatique

2ème Année - 1er Semestre

Intervenants CM:

Gaël DIAS (gael.dias@unicaen.fr), Marc SPANIOL, Fabrice MAUREL

Navneet AGARWAL, Kirill MILINTSEVICH

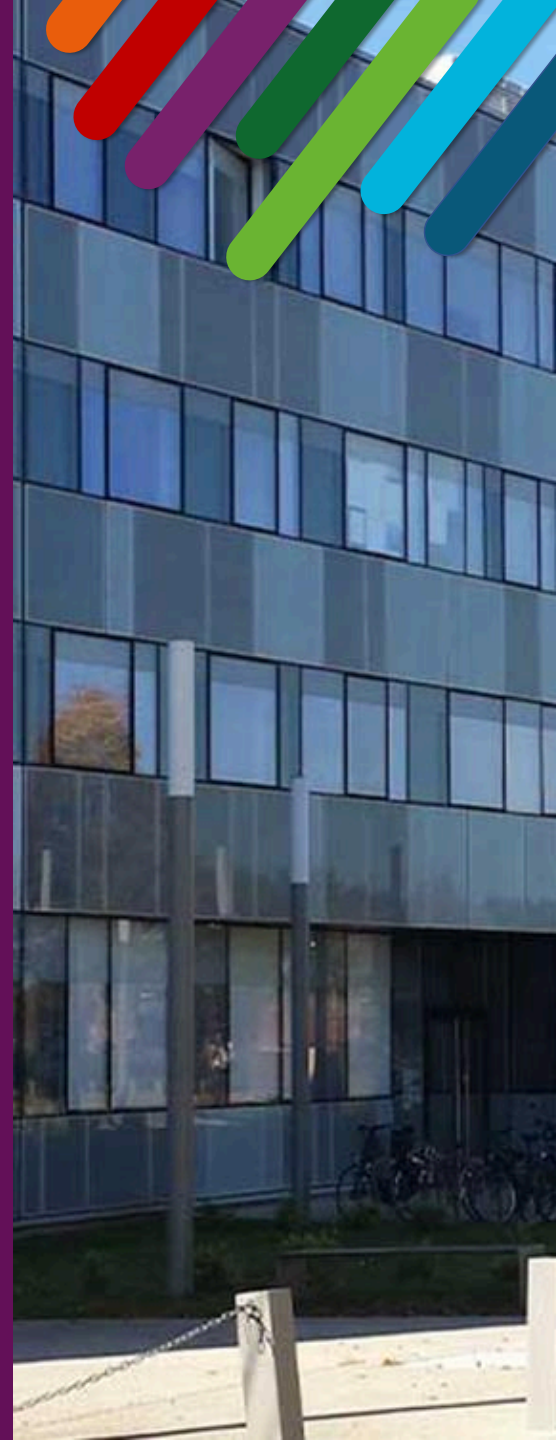
GREYC Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN ÉCOLE PUBLIQUE D'INGÉNIEURS CENTRE DE RECHERCHE

cnrs



Natural Language Generation I

Kirill Milintsevich | 11.01.2024

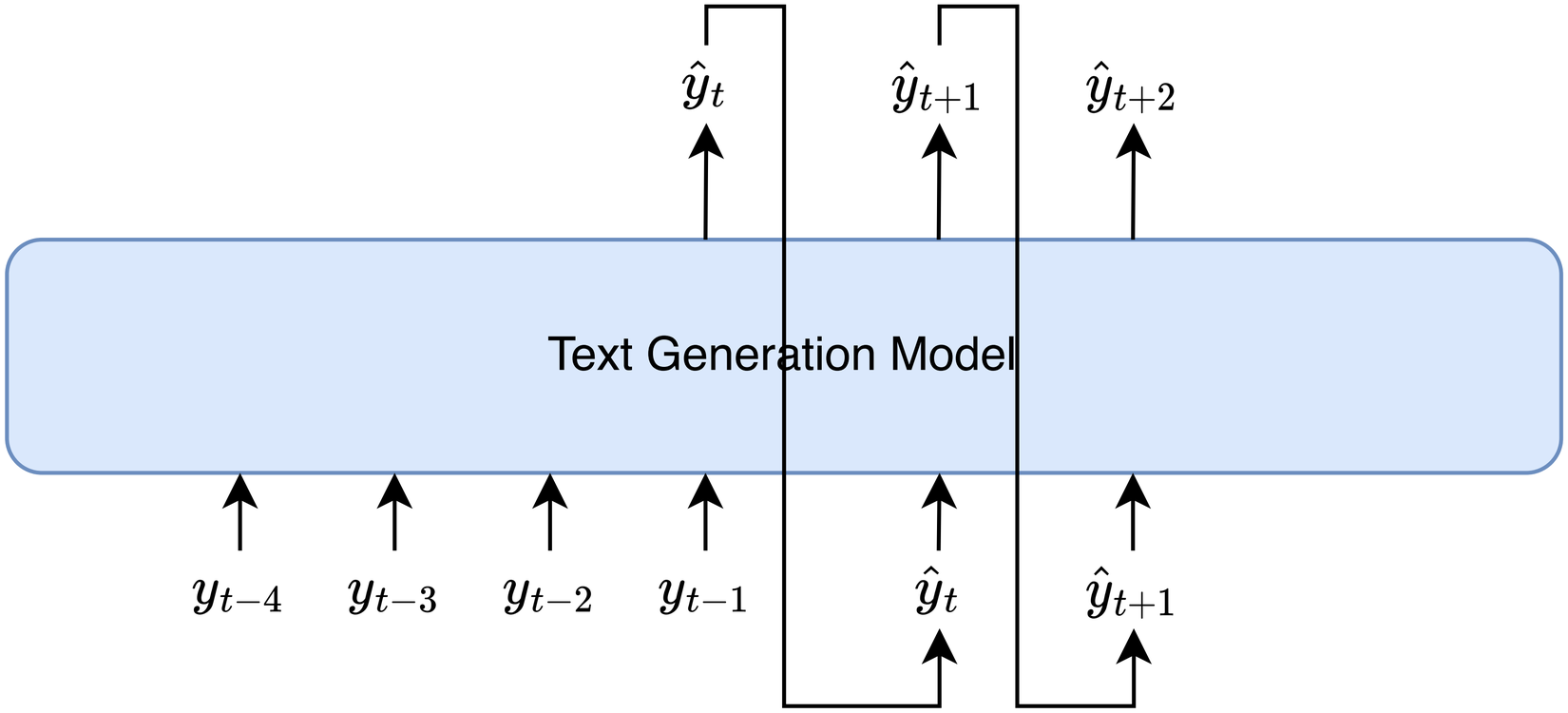
NLG Tasks

- Machine Translation
- Dialogue Systems
- Summarisation
- Data-to-text Generation
- Visual Description

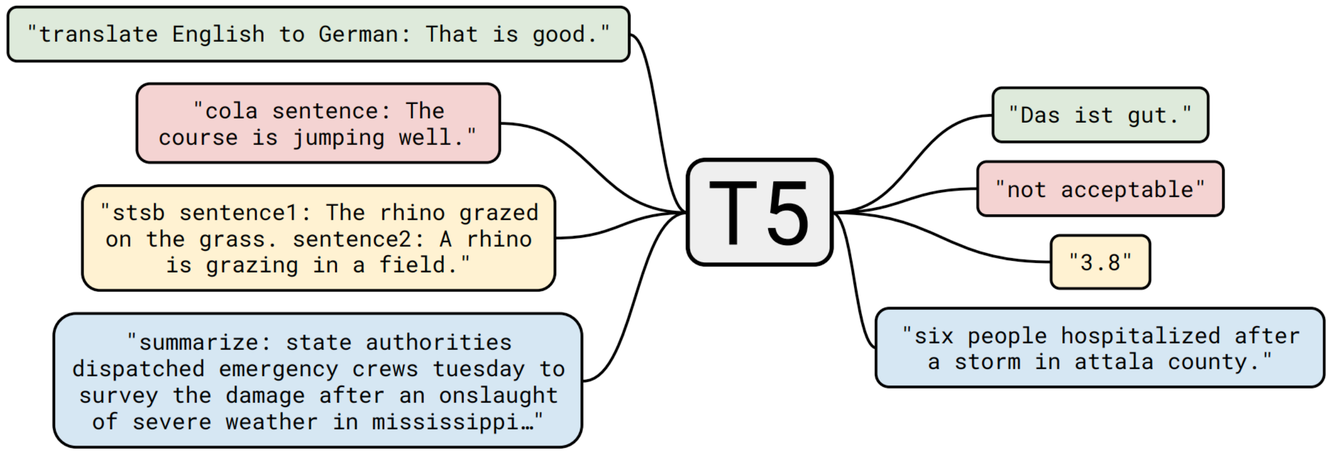
Autoregressive Models

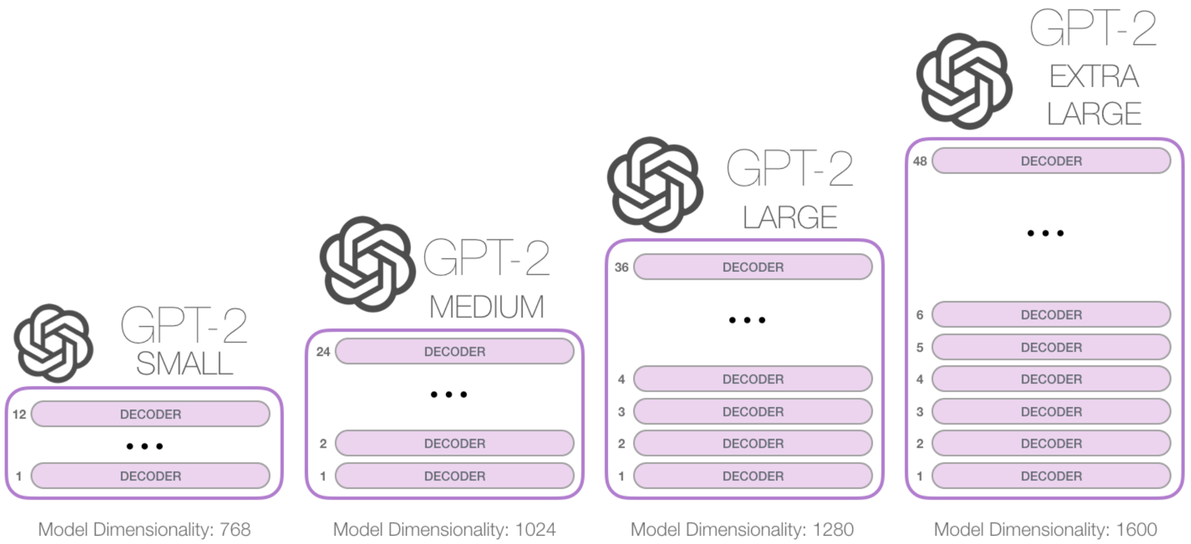
Autoregressive Models

- In autoregressive text generation models, at each time step t, our model takes in a sequence of tokens of text as input $y_{<t}$ and outputs a new token, $\hat{y}_t$

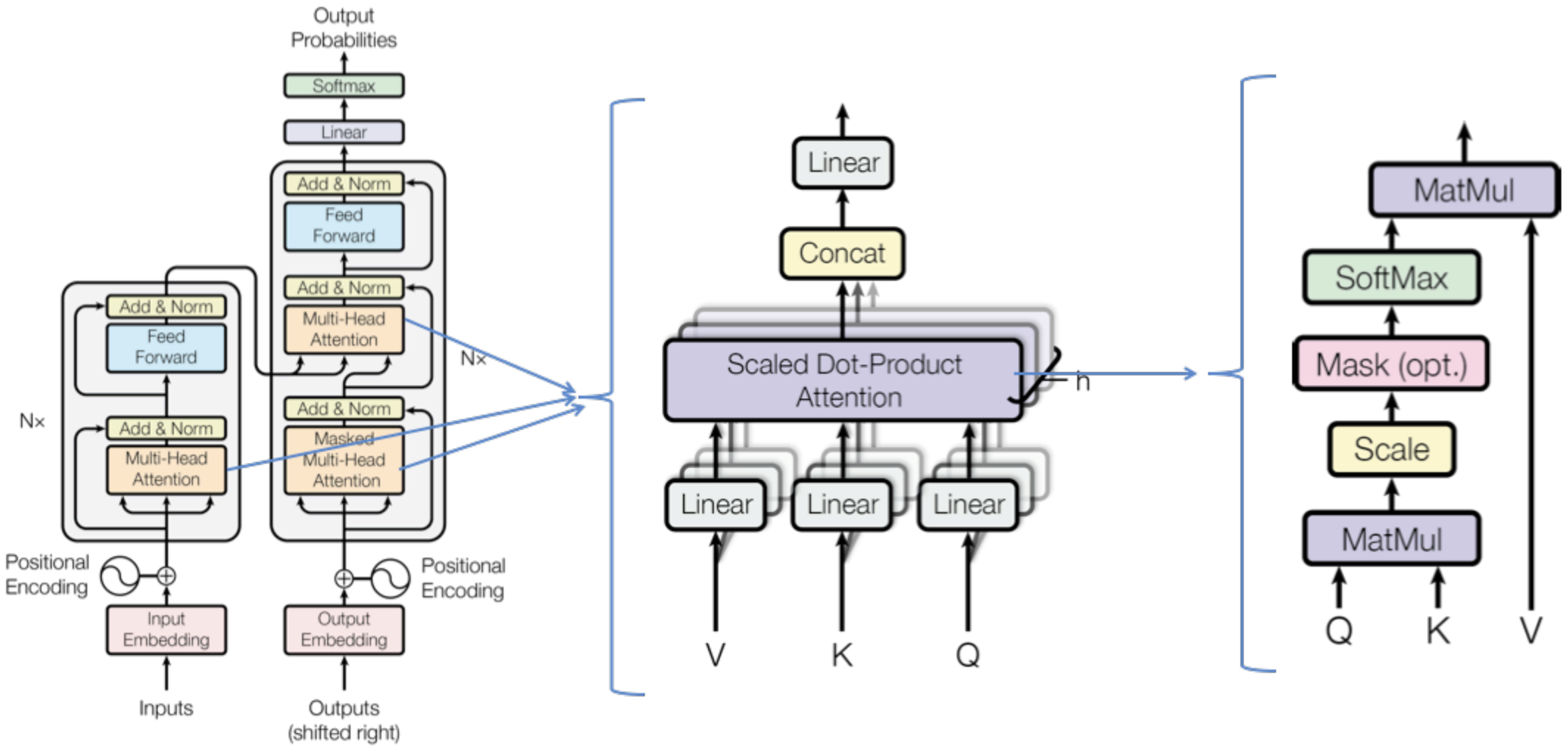


SOTA Autoregressive Models





Inside These Models



During a Single Step

- At each time step t, our model computes a vector of scores for each token in our vocabulary, $S \in \mathbb{R}^V$:

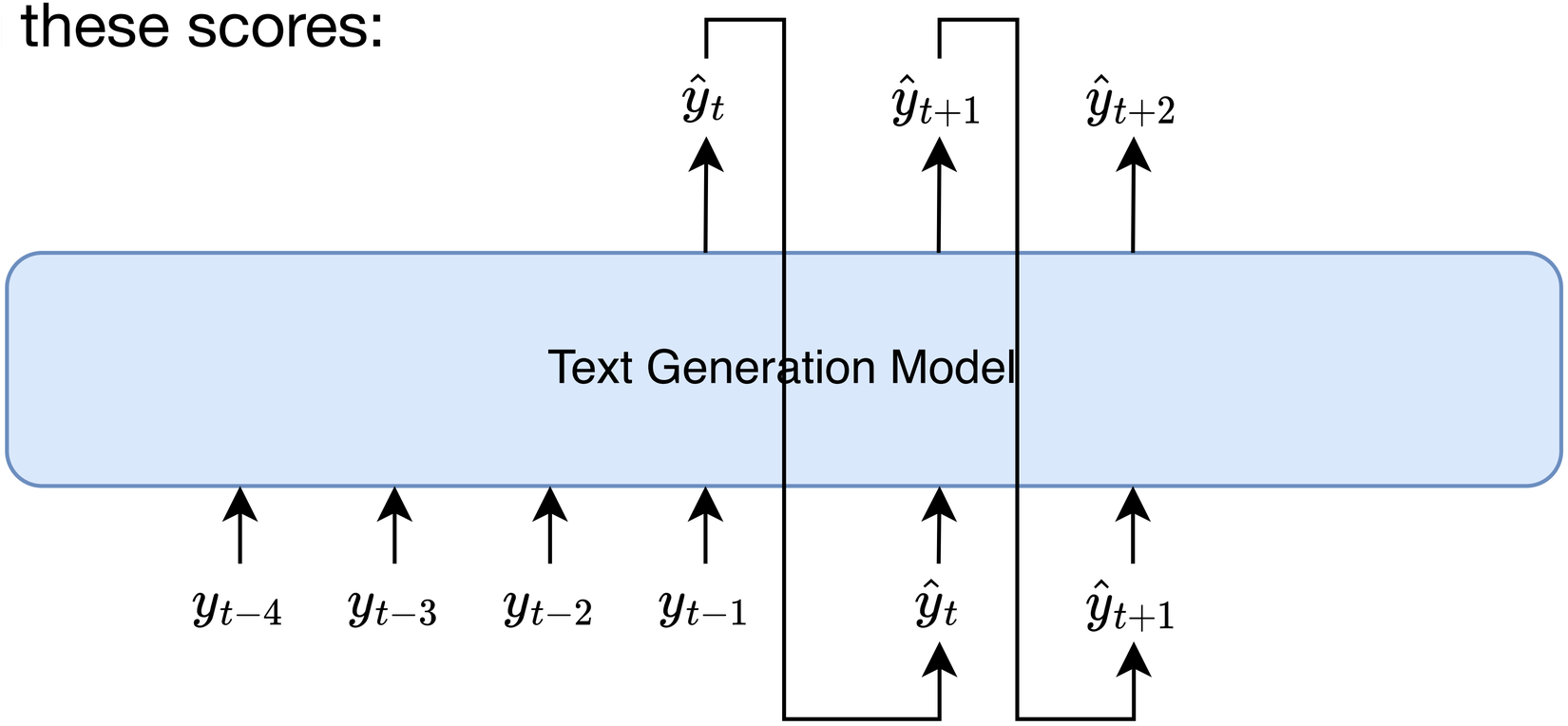
$$S = f(\{y_{<t}\}, \theta)$$

- Then, we compute a probability distribution P over $w \in V$ using these scores:

$$P(y_t \mid \{y_{<t}\}) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

Autoregressive Models

- At each time step t, our model computes a vector of scores for each token in our vocabulary, $S \in \mathbb{R}^V$. Then, we compute a probability distribution P over $w \in V$ using these scores:



Inference and Training

- At inference time, our decoding algorithm defines a function to select a token from this distribution:

$$\hat{y}_t = g(P(y_t \mid \{y_{<t}\}))$$

- We train the model to minimize the negative loglikelihood of predicting the next token in the sequence:

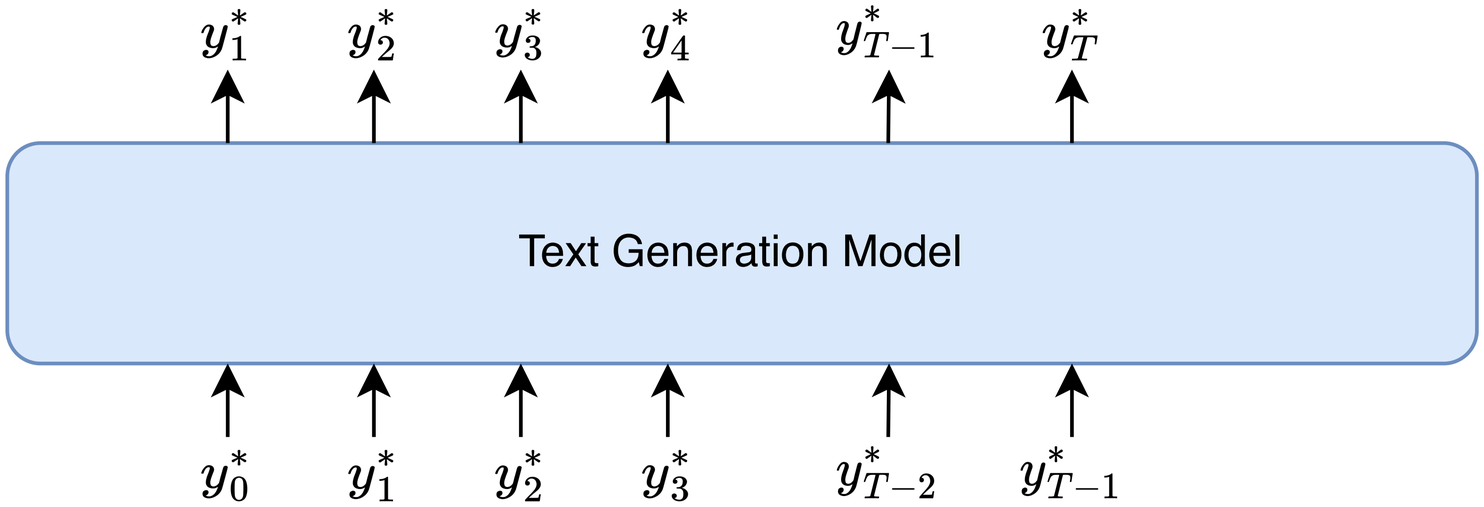
$$L_t = -\log P(y_t^* \mid \{y_{<t}^*\})$$

Maximum Likelihood Training

Teacher Forcing

- Trained to generate the next word y_t^* given a set of preceding words $\{y^*\}_{<t}$

$$L = -\sum_{t=1}^{T^t} \log P(y_t^* \quad \{y_{<t}^*\})$$



Decoding

- At each time step t, our model computes a vector of scores for each token in our vocabulary, $S \in \mathbb{R}^V$:

$$S = f(\{y_{<t}\}, \theta)$$

- Then, we compute a probability distribution P over $w \in V$ using these scores:

$$P(y_t \mid \{y_{<t}\}) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

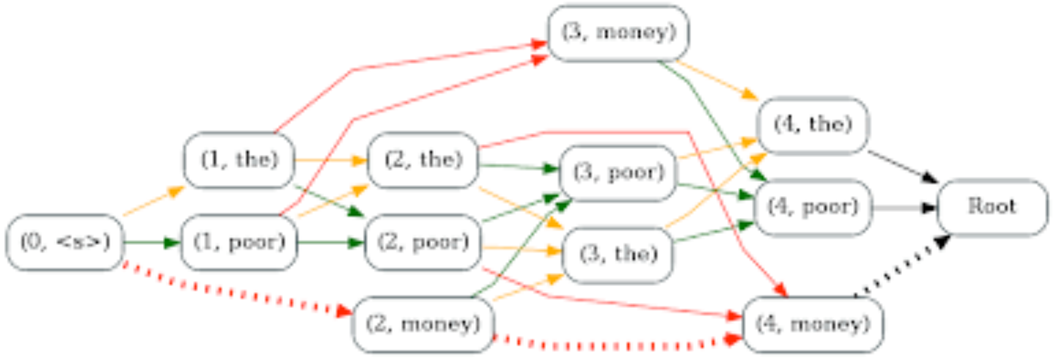
- Our decoding algorithm defines a function to select a token from this distribution:

$$\hat{y}_t = g(P(y_t \mid \{y_{<t}\}))$$

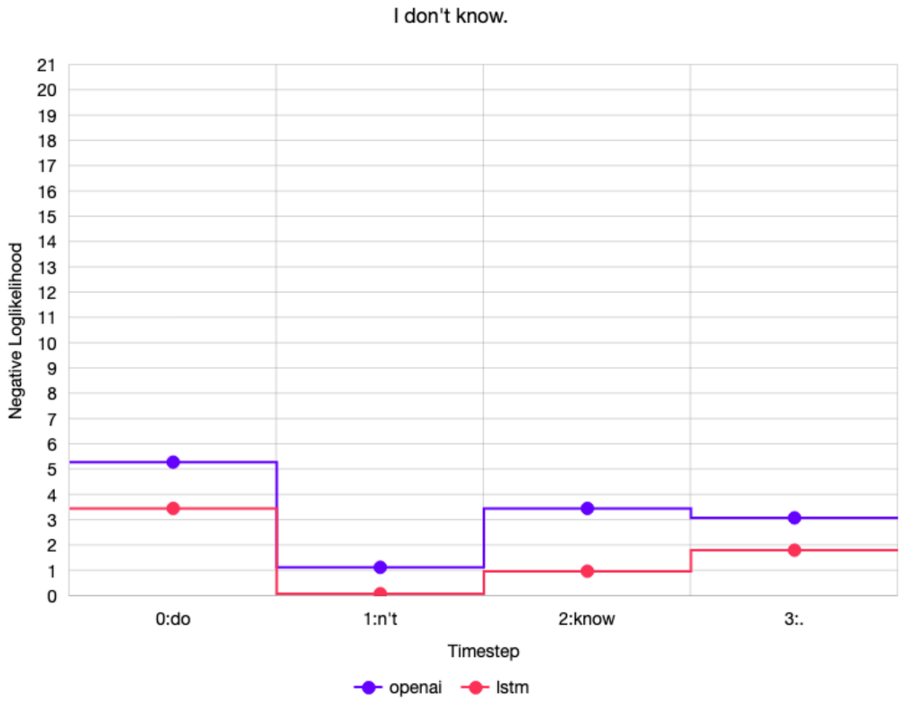
Greedy Methods

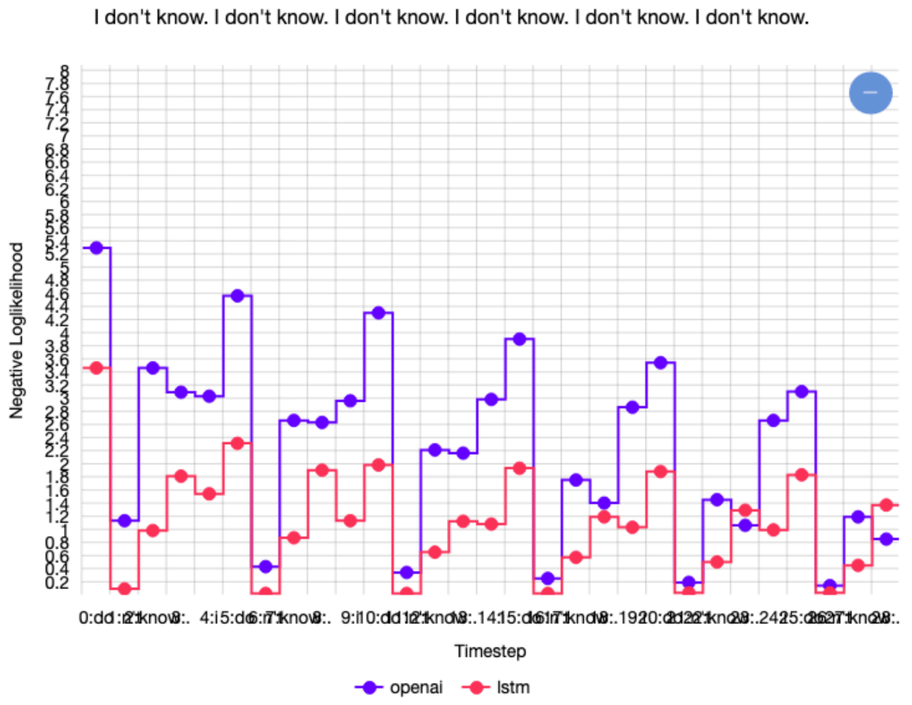
- Argmax Decoding
  - Selects the highest probability token in $P(y_t \mid y_{<t})$

- Beam Search



Repetition in Greedy Methods





Sampling

- Sample a token from the distribution of tokens

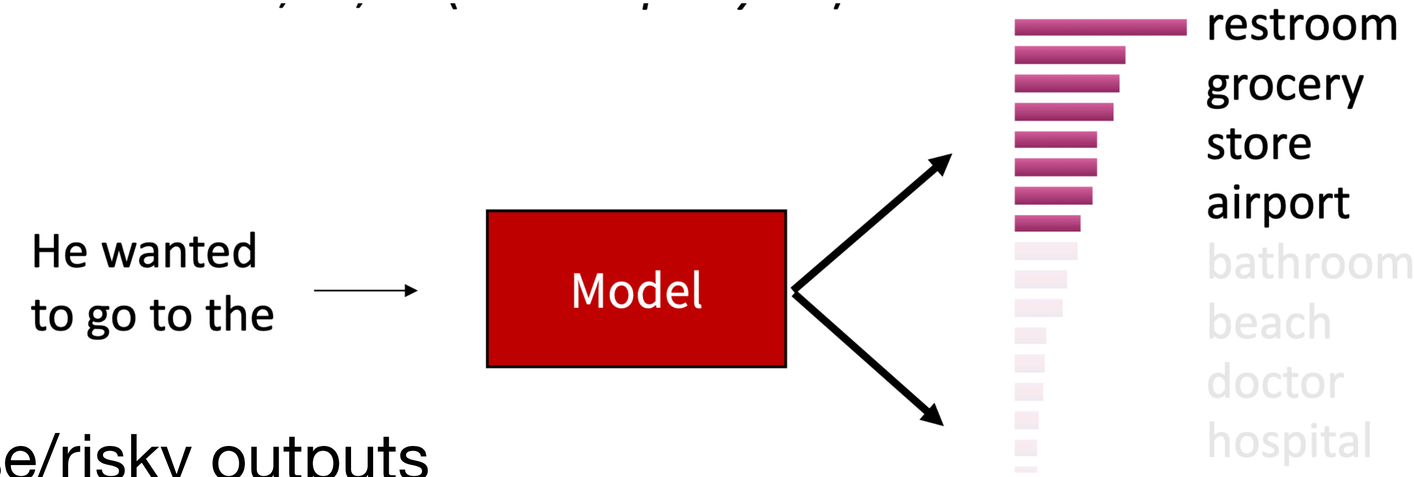
$$\hat{y}_t \sim P(y_t = w \mid \{y\}_{<t})$$

Top-k sampling

- Problem: Vanilla sampling makes every token in the vocabulary an option
  - Even if most of the probability mass in the distribution is over a limited set of options, the tail of the distribution could be very long
  - Many tokens are probably irrelevant in the current context
  - Why are we giving them individually a tiny chance to be selected?
  - Why are we giving them as a group a high chance to be selected?
- Solution: Top-k sampling
  - Only sample from the top k tokens in the probability distribution

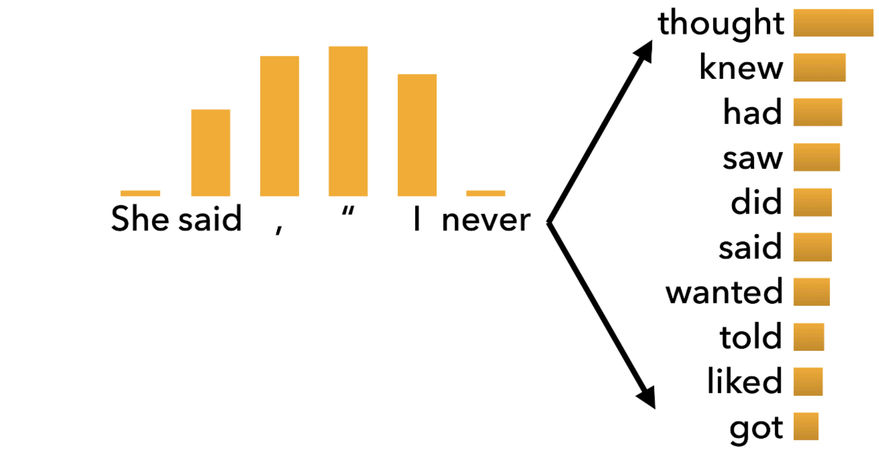
Top-k sampling

- Only sample from the top k tokens in the probability distribution
- Common values are k = 5, 10, 20 (but it's up to you!)

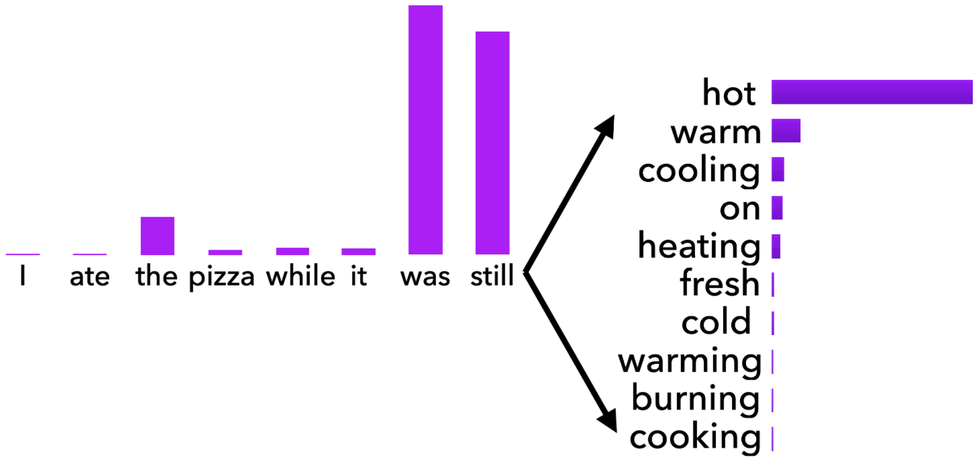


- Increase k for more diverse/risky outputs
- Decrease k for more generic/safe outputs

Top-k sampling



Top-*k* sampling can cut off too ***quickly***!



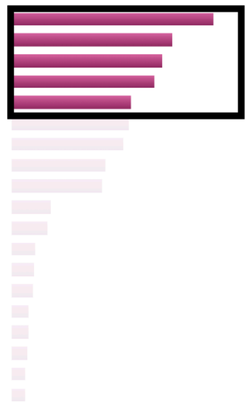
Top-*k* sampling can also cut off too ***slowly***!

Top-p (nucleus) sampling

- Problem: The probability distributions we sample from are dynamic
  - When the distribution P_t is flatter, a limited k removes many viable options
  - When the distribution P_t is peakier, a high k allows for too many options to have a chance of being selected
- Solution: Top-p sampling
  - Sample from all tokens in the top p cumulative probability mass (i.e., where mass is concentrated)
  - Varies k depending on the uniformity of P_t

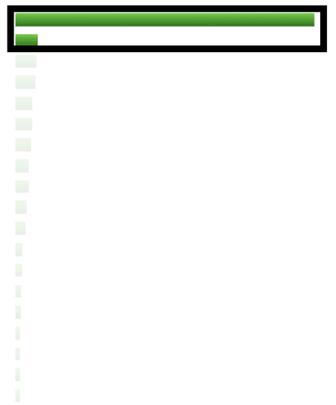
Top-p (nucleus) sampling

$P_t^1(y_t = w \mid \{y\}_{<t})$



$P_t^2(y_t = w \mid \{y\}_{<t})$

$P_t^3(y_t = w \mid \{y\}_{<t})$



Softmax Temperature

- On timestep t, the model computes a prob distribution P_t by applying the softmax function to a vector of scores $s \in \mathbb{R}^V$

$$P(y_t \mid \{y_{<t}\}) = \frac{\exp(S_w)}{\sum_{w' \in V} \exp(S_{w'})}$$

- You can apply a temperature hyperparameter τ to the softmax to rebalance P_t:

$$P(y_t \mid \{y_{<t}\}) = \frac{\exp(S_w/\tau)}{\sum_{w' \in V} \exp(S_{w'}/\tau)}$$

Softmax Temperature

- Raise the temperature $\tau > 1$: P_t becomes more uniform
  - More diverse output (probability is spread around vocab)
- Lower the temperature $\tau < 1$: P_t becomes more spiky
  - Less diverse output (probability is concentrated on top words)

Re-ranking

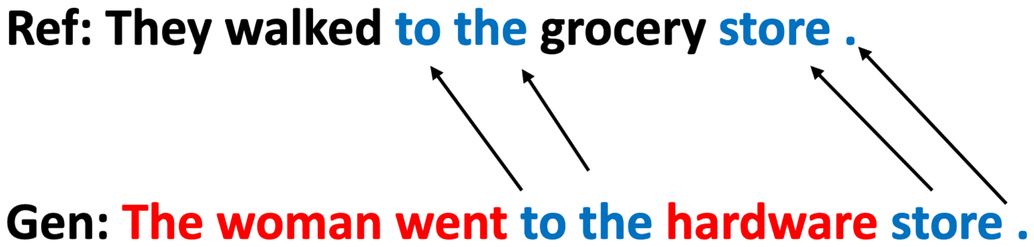
- Decode a bunch of sequences
  - 10 candidates is a common number
- Define a score to approximate quality of sequences and re-rank by this score
  - Simplest is to use perplexity
    - Careful! Remember that repetitive methods can generally get high perplexity.

Decoding

- Decoding is still a challenging problem in natural language generation
- Human language distribution is noisy and doesn't reflect simple properties (i.e., probability maximization)
- Different decoding algorithms can allow us to inject biases that encourage different properties of coherent natural language generation
- Some of the most impactful advances in NLG of the last few years have come from simple, but effective, modifications to decoding algorithms

Evaluation

Content Overlap Metrics



- Compute a score that indicates the similarity between generated and gold-standard (human-written) text
- Fast and efficient and widely used
- Two broad categories:
  - N-gram overlap metrics (e.g., BLEU, ROUGE, METEOR, CIDEr, etc.)
  - Semantic overlap metrics (e.g., PYRAMID, SPICE, SPIDEr, etc.)

Content Overlap Metrics

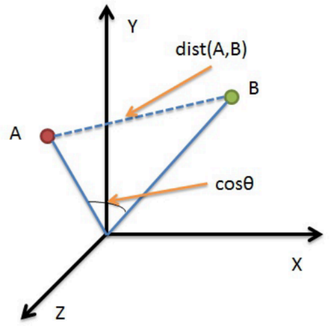
- They’re not ideal for machine translation
- They get progressively much worse for tasks that are more open-ended than machine translation
  - Worse for summarization, where extractive methods that copy from documents are preferred
  - Much worse for dialogue, which is more open-ended that summarization
  - Much, much worse story generation, which is also open-ended, but whose sequence length can make it seem you’re getting decent scores!

Dans le cas de la summarization, où des méthodes extractives (copier des parties du texte source) sont souvent préférées, les métriques de recouvrement de contenu peuvent ne pas bien refléter la qualité du résumé, car la redondance peut être considérée comme une bonne pratique.

Les métriques de recouvrement de contenu peuvent être encore moins adaptées pour évaluer la qualité des dialogues. Les conversations sont souvent évaluées en fonction de la cohérence, de la pertinence contextuelle, et de la manière dont elles maintiennent une interaction significative, des aspects qui ne sont pas capturés efficacement par la simple comparaison du contenu.

La génération d'histoires est une tâche très ouverte où la créativité et la cohérence narrative sont cruciales. Les métriques de recouvrement de contenu peuvent ne pas saisir ces aspects, en particulier lorsque les histoires sont longues et complexes.

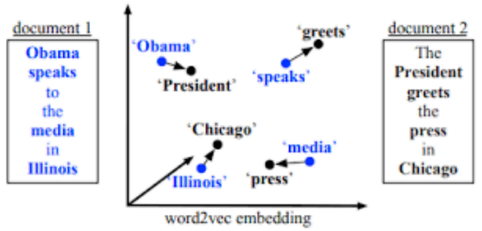
Model-Based Metrics



Vector Similarity:

Embedding based similarity for semantic distance between text.

- **Embedding Average** (Liu et al., 2016)
- **Vector Extrema** (Liu et al., 2016)
- **MEANT** (Lo, 2017)
- **YISI** (Lo, 2019)

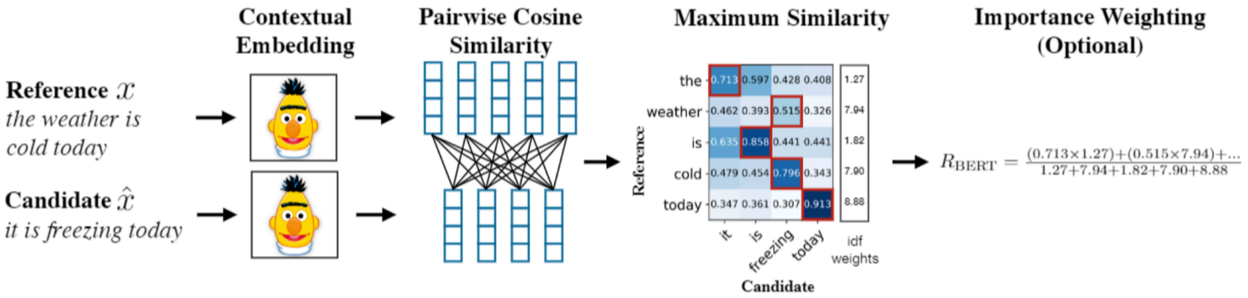


Word Mover's Distance:

Measures the distance between two sequences (e.g., sentences, paragraphs, etc.), using word embedding similarity matching.
(Kusner et.al., 2015; Zhao et al., 2019)

BERTSCORE:

Uses pre-trained contextual embeddings from BERT and matches words in candidate and reference sentences by cosine similarity.
(Zhang et.al. 2020)



Human Evaluation

- Ask humans to evaluate the quality of generated text
- Human judgments are regarded as the gold standard
- Humans are inconsistent

Evaluation

- Content overlap metrics provide a good starting point for evaluating the quality of generated text, but they’re not good enough on their own.
- Model-based metrics are can be more correlated with human judgment, but behavior is not interpretable
- Human judgments are critical.
  - Only ones that can directly evaluate factuality
  - But humans are inconsistent!

Non-autoregressive models

Non-autoregressive Models

Application	Example Source (S) and target (T) text	Use seq2seq
Machine translation	S: Turing studied at King's College, where he was awarded first-class honours in mathematics. T: Turing studierte am King's College, wo er erstklassige Auszeichnungen in Mathematik erhielt.	
Summarization	S: Court members Deborah Poritz and Peter Verniero did not participate in the Nelson case. T: Court members didn't participate in the case.	
Sentence fusion	S: Turing was born in 1912. Turing died in 1954. T: Turing was born in 1912 and he died in 1954.	
Grammar correction	S: New Zealand have a cool weather. T: New Zealand has cool weather.	

Non-autoregressive Models

- Most NLP tasks apart from Machine Translation are monolingual
- Sources and targets often overlap
  - Generating the target from scratch is wasteful
  - Can reconstruct most of the target from the source via basic operations like KEEP, DELETE, INSERT

Turing	was	born	in	1912	.	Turing	died	in	1954	.
KEEP	KEEP	KEEP	KEEP	KEEP	DEL INS	PRON	KEEP	KEEP	KEEP	KEEP
Turing	was	born	in	1912	and	he	died	in	1954	.

Applications

- Grammatical Error Correction
- Text Simplification
- Sentence fusion
- Style transfer
- Text normalisation
- Text summarisation
- Automatic post-editing for machine translation

Advantages

- Text editing models need less training data
- They are faster at the inference
- They are more faithful
- We can control what model adds or removes
- We can incorporate external knowledge

Conclusion

Conclusion

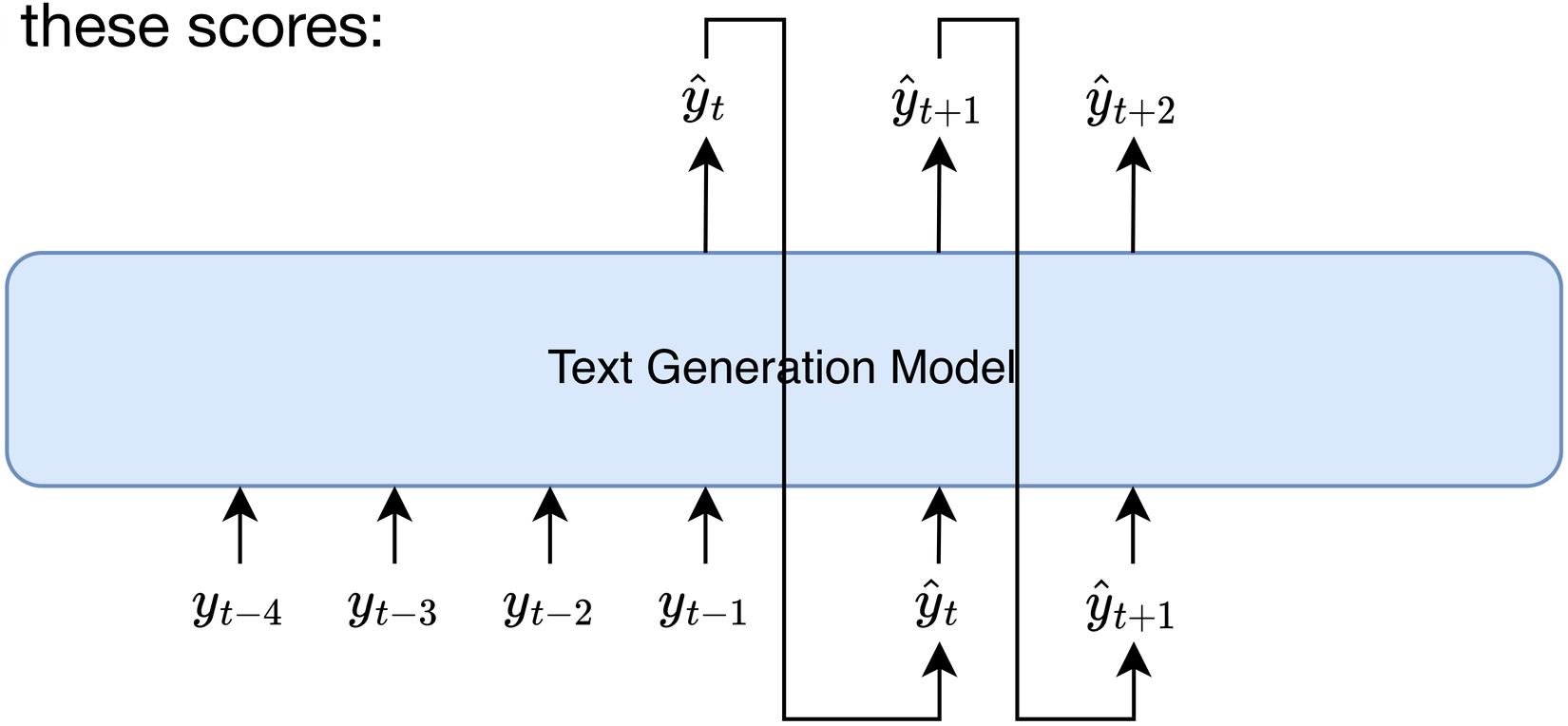
- Natural Language Generation made a huge progress in the recent years
- NLG tasks cover a vast field of NLP problems (technically, any NLP task can be converted into a text generation!)
- Evaluating NLG models is challenging
- Training a performant NLG model requires a lot of data

Natural Language Generation II

Kirill Milintsevich | 12.01.2024

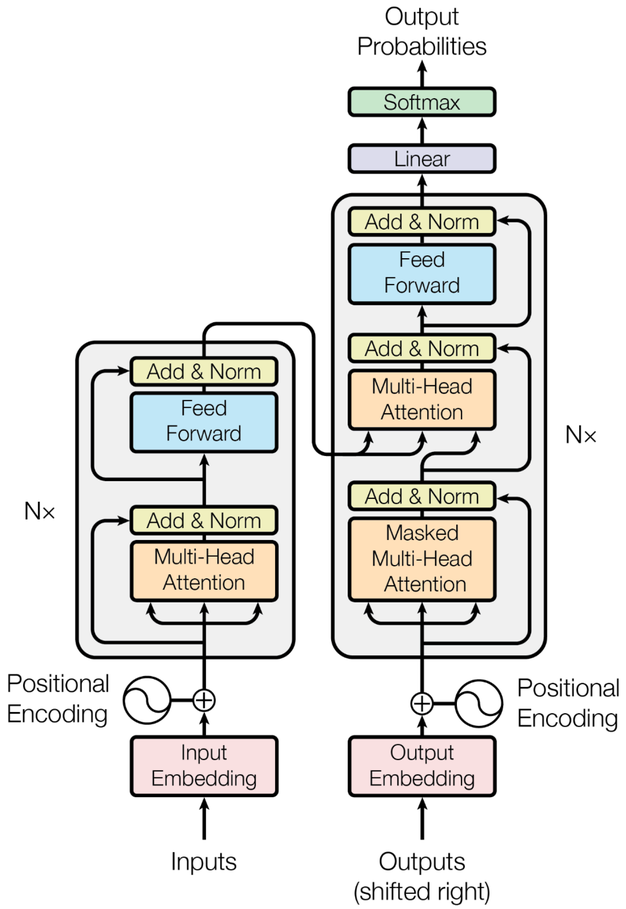
Autoregressive Models (Reminder)

- At each time step t, our model computes a vector of scores for each token in our vocabulary, $S \in \mathbb{R}^V$. Then, we compute a probability distribution P over $w \in V$ using these scores:



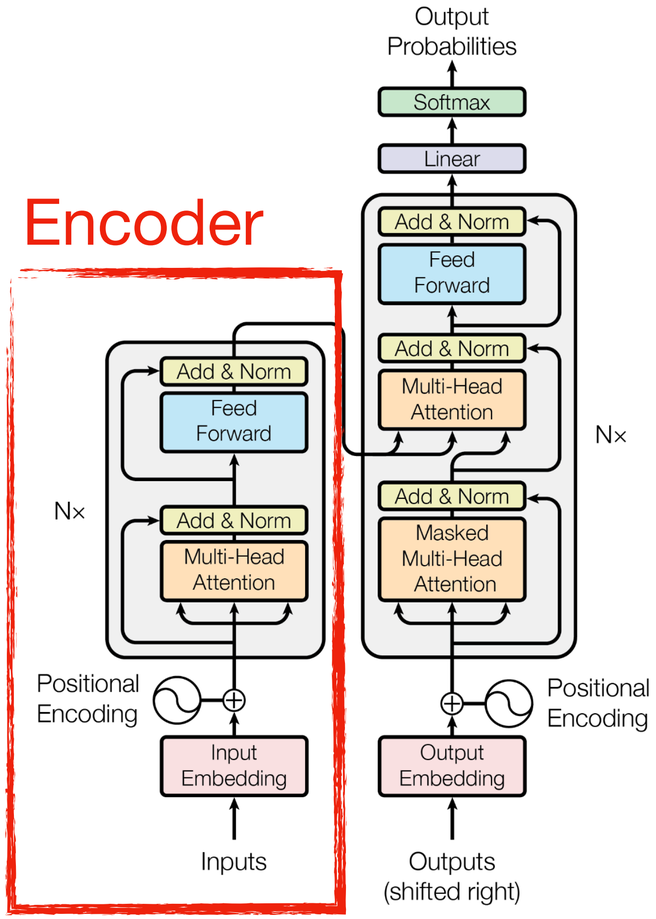
Transformer Architecture

Vaswani, Ashish, et al. "Attention is all you need." (2017)



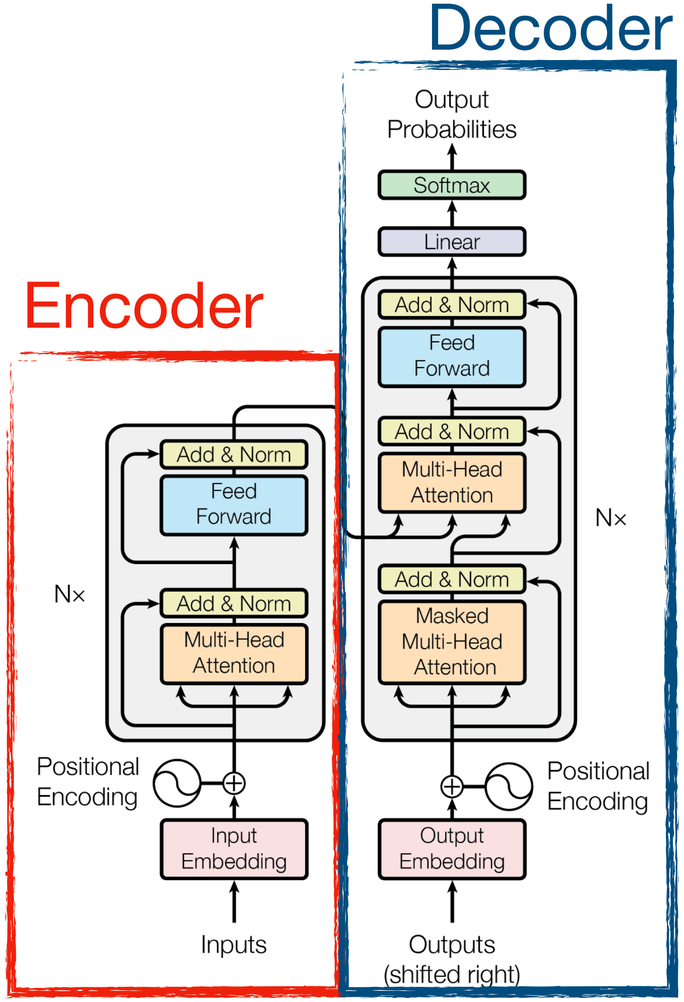
Transformer Architecture

Vaswani, Ashish, et al. "Attention is all you need." (2017)

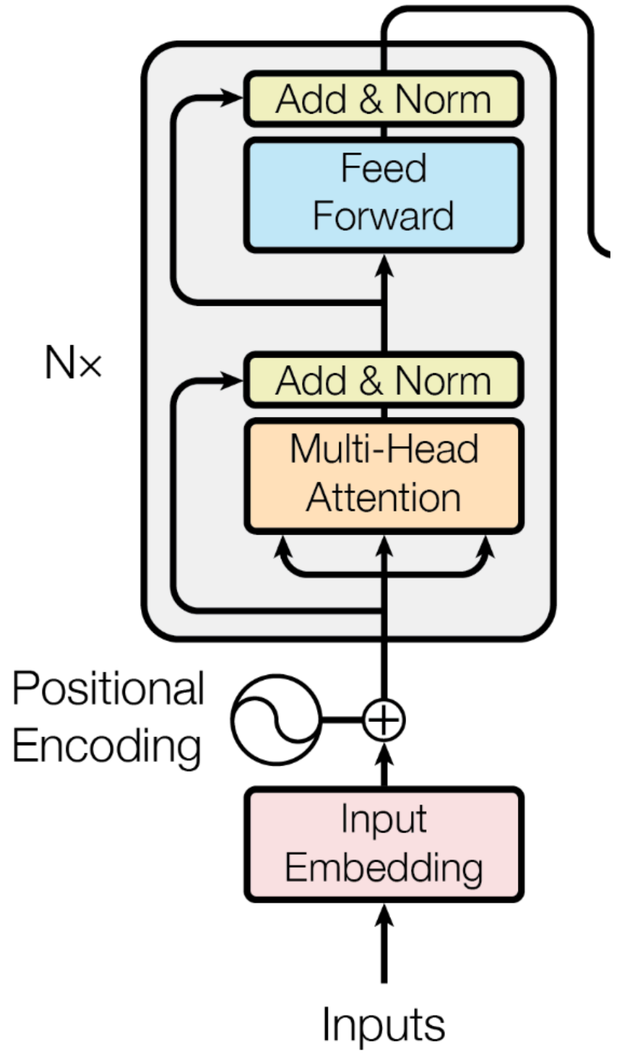


Transformer Architecture

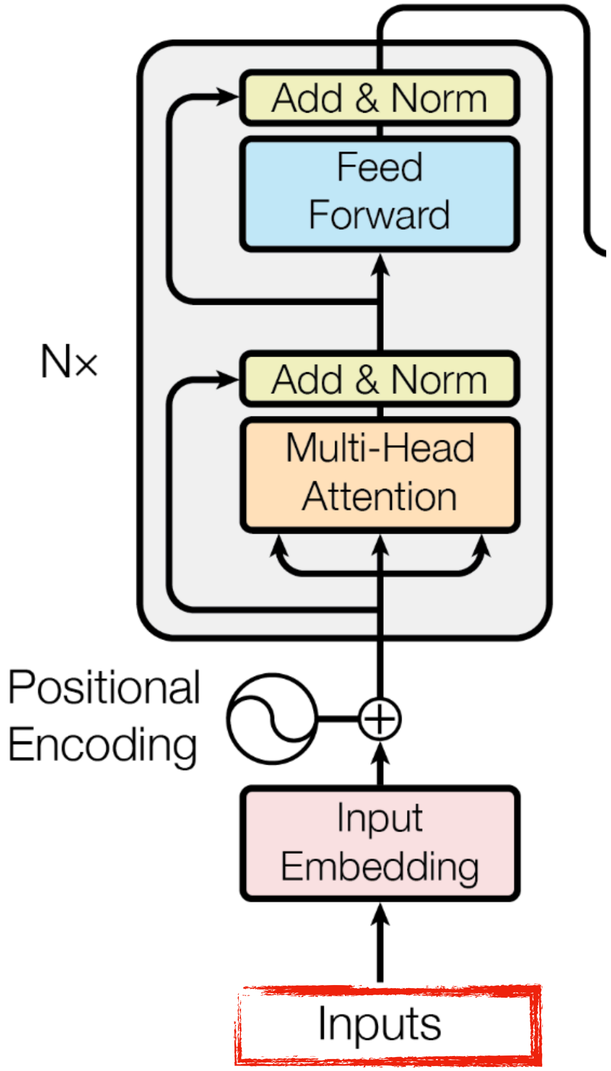
Vaswani, Ashish, et al. "Attention is all you need." (2017)



Transformer Encoder



Transformer Encoder



- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization

Byte-Pair Encoding

Training

- First, we compute the unique set of words in the corpus
- Split each word into characters
- Start merging most frequent pairs

- ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)
- ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

Byte-Pair Encoding

Training

- First, we compute the unique set of words in the corpus
- Split each word into characters
- Start merging most frequent pairs
- The most frequent pair is `(“u”, “g”)`

- ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)
- ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

Vocabulary: `["b", "g", "h", "n", "p", "s", "u", "ug"]`

Corpus: `("h" "ug", 10), ("p" "ug", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "ug" "s", 5)`

Byte-Pair Encoding

Training

- First, we compute the unique set of words in the corpus
- Split each word into characters
- Start merging most frequent pairs
- The most frequent pair is `("u", "n")`

- ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)
- ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

Vocabulary: `["b", "g", "h", "n", "p", "s", "u", "ug", "un"]`

Corpus: `("h" "ug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("h" "ug" "s", 5)`

Byte-Pair Encoding

Training

- First, we compute the unique set of words in the corpus
- Split each word into characters
- Start merging most frequent pairs
- The most frequent pair is `("h", "ug")`

- ("hug", 10), ("pug", 5), ("pun", 12), ("bun", 4), ("hugs", 5)
- ("h" "u" "g", 10), ("p" "u" "g", 5), ("p" "u" "n", 12), ("b" "u" "n", 4), ("h" "u" "g" "s", 5)

Vocabulary: `["b", "g", "h", "n", "p", "s", "u", "ug", "un", "hug"]`

Corpus: `("hug", 10), ("p" "ug", 5), ("p" "un", 12), ("b" "un", 4), ("hug" "s", 5)`

Byte-Pair Encoding

Tokenization

1. Normalization
2. Pre-tokenization
3. Splitting the words into individual characters
4. Applying the merge rules learned in order on those splits

Byte-Pair Encoding

Tokenization

1. Normalization
2. Pre-tokenization
3. Splitting the words into individual characters
4. Applying the merge rules learned in order on those splits

Learned rules:

```
("u", "g") -> "ug"

("u", "n") -> "un"

("h", "ug") -> "hug"
```

Byte-Pair Encoding

Tokenization

1. Normalization
2. Pre-tokenization
3. Splitting the words into individual characters
4. Applying the merge rules learned in order on those splits

Learned rules:

```
("u", "g") -> "ug"

("u", "n") -> "un"

("h", "ug") -> "hug"
```

“bug” -> “b”, “ug” | “mug” -> “[UNK]”, “ug”

WordPiece Tokenization

Training

- Same as BPE but different merging rule

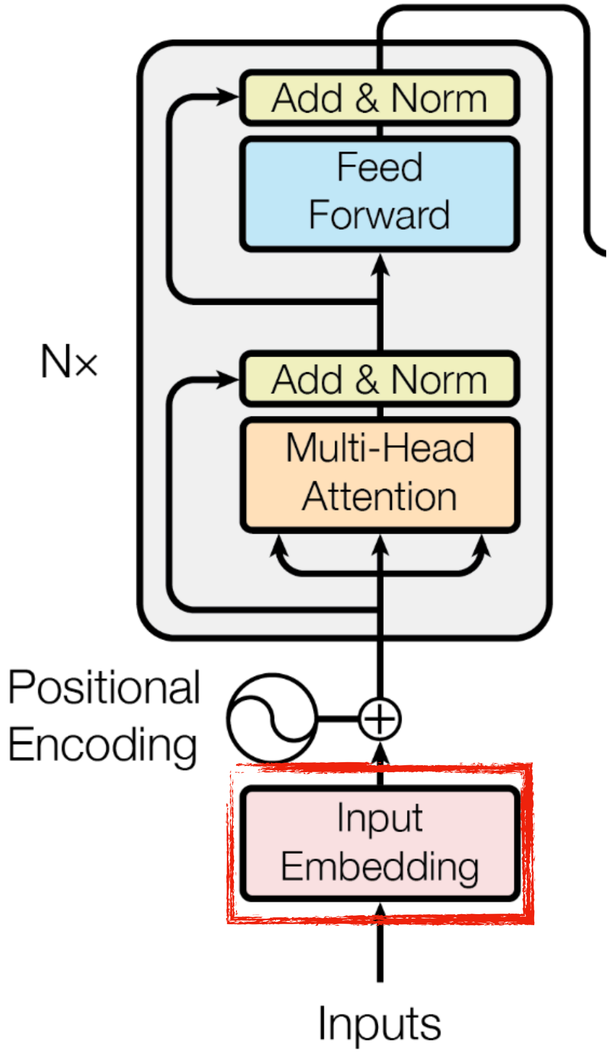
$$\text{score} = \frac{\text{freq\_of\_pair}}{\text{freq\_of\_first\_element} \times \text{freq\_of\_second\_element}}$$

WordPiece Tokenization

Tokenization

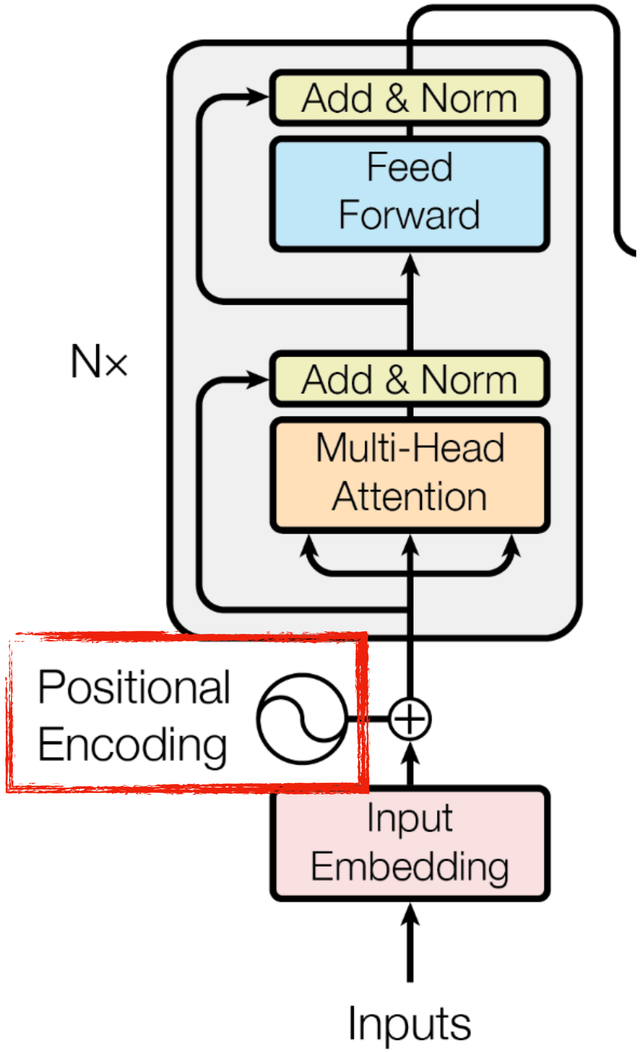
- Same as BPE but the WordPiece starts by searching for the longest sub-token in the vocabulary

Transformer Encoder



- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer

Transformer Encoder



- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer
- Positional encoding to add information about words' position in the sequence

Positional Encoding

- Since Transformer doesn't have recursion it lacks information about words' positions

Let t be the desired position in an input sentence, $\overrightarrow{p_t} \in \mathbb{R}^d$ be its corresponding encoding, and d be the encoding dimension (where $d \equiv_2 0$) Then $f : \mathbb{N} \to \mathbb{R}^d$ will be the function that produces the output vector $\overrightarrow{p_t}$ and it is defined as follows:

$$\overrightarrow{p_t}^{(i)} = f(t)^{(i)} := \begin{cases} \sin(\omega_k . t), & \text{if } i = 2k \\ \cos(\omega_k . t), & \text{if } i = 2k+1 \end{cases}$$

where

$$\omega_k = \frac{1}{10000^{2k/d}}$$

https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

Positional Encoding

Intuition

0 :	0	0	0	0
1 :	0	0	0	1
2 :	0	0	1	0
3 :	0	0	1	1
4 :	0	1	0	0
5 :	0	1	0	1
6 :	0	1	1	0
7 :	0	1	1	1

8 :	1	0	0	0
9 :	1	0	0	1
10 :	1	0	1	0
11 :	1	0	1	1
12 :	1	1	0	0
13 :	1	1	0	1
14 :	1	1	1	0
15 :	1	1	1	1

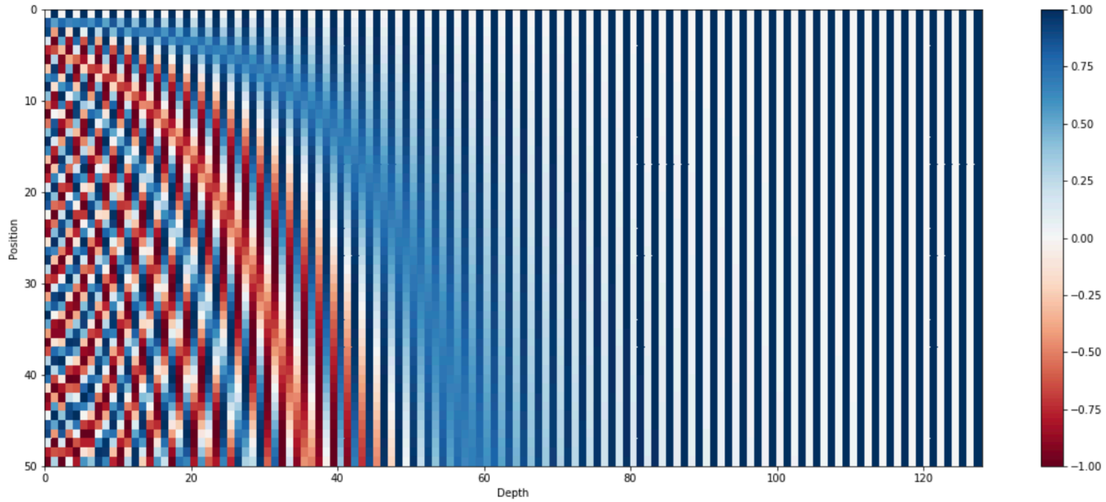
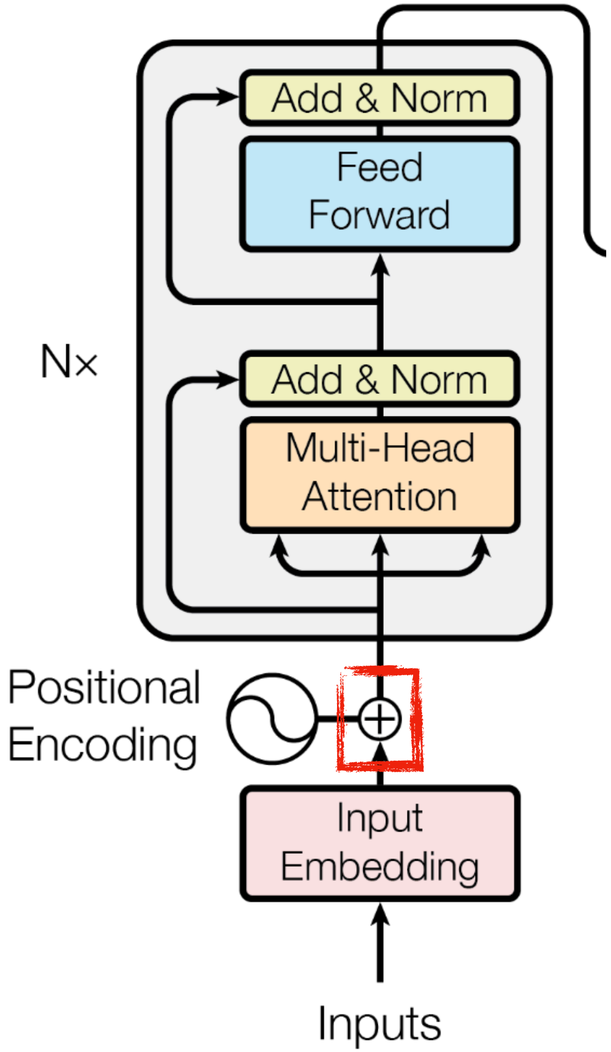


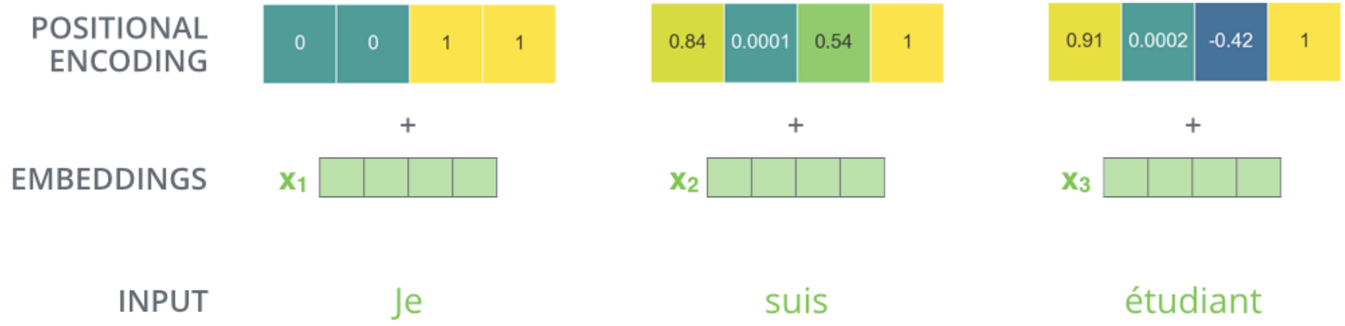
Figure 2 - The 128-dimensional positonal encoding for a sentence with the maximum lenght of 50. Each row represents the embedding vector $\overrightarrow{p_t}$

https://kazemnejad.com/blog/transformer_architecture_positional_encoding/

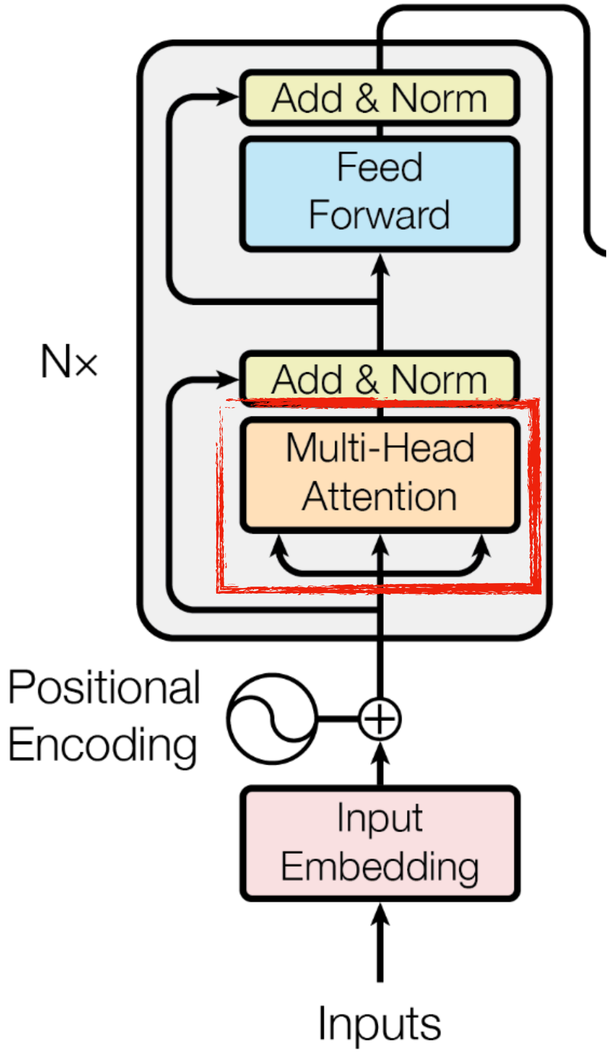
Transformer Encoder



- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer
- Positional encoding to add information about words' position in the sequence
- Positional encoding is added to the embedding



Transformer Encoder



- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer
- Positional encoding to add information about words' position in the sequence
- Positional encoding is added to the embedding
- MHA used to redistribute the information among the inputs (contextualization)

Multi-Head Self-Attention

Vaswani, Ashish, et al. "Attention is all you need." (2017)

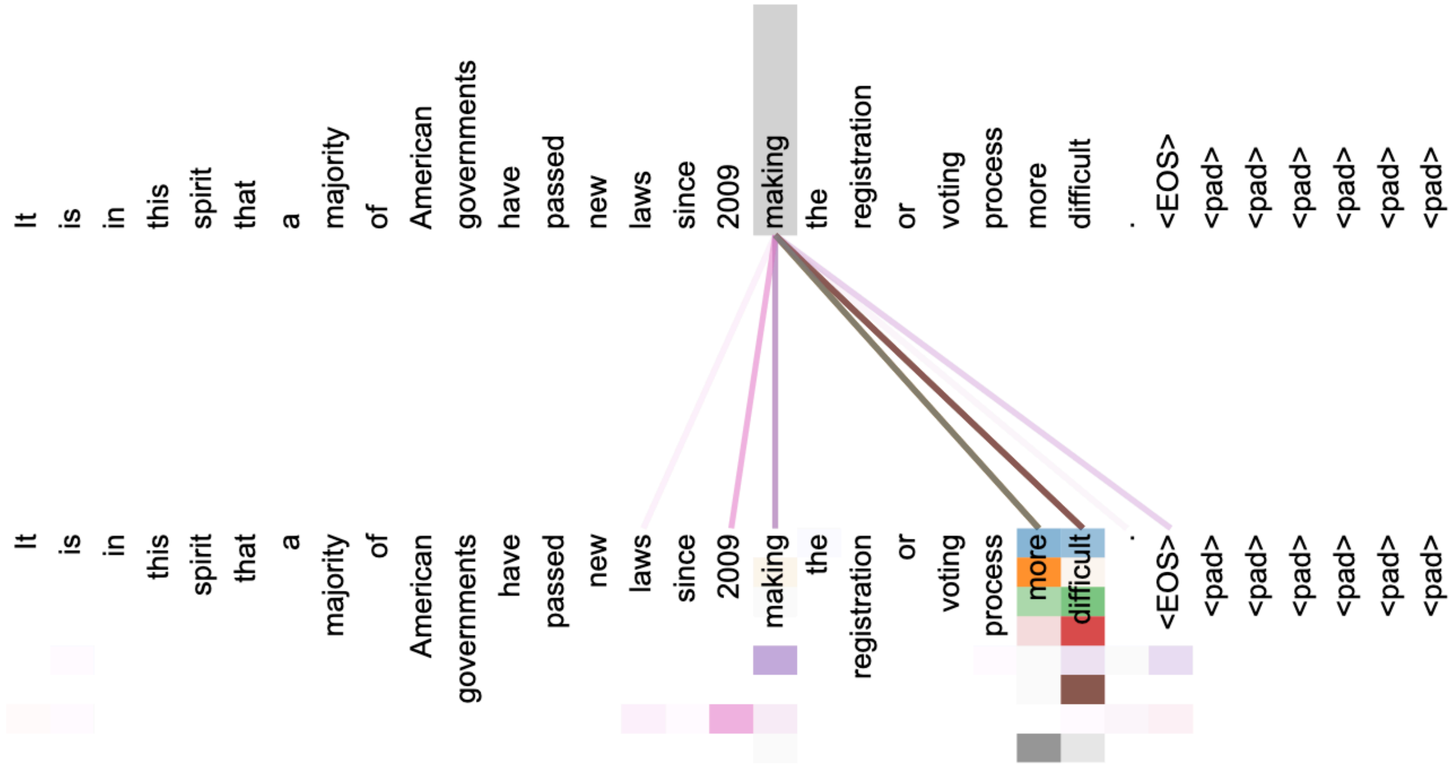
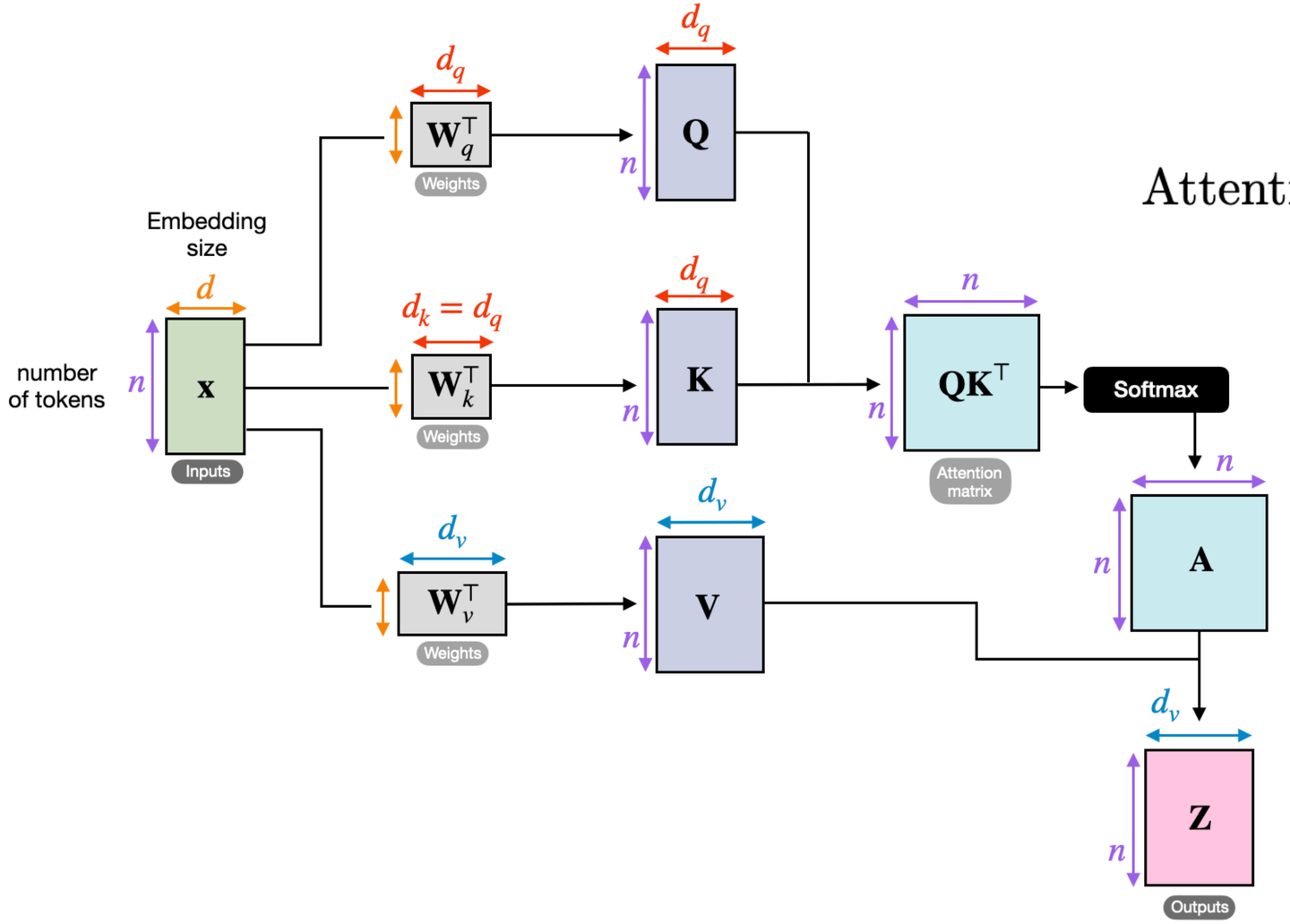


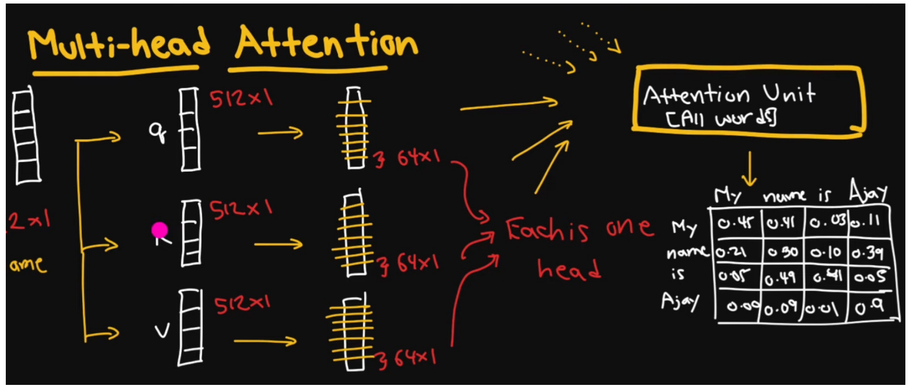
Figure 3: An example of the attention mechanism following long-distance dependencies in the encoder self-attention in layer 5 of 6. Many of the attention heads attend to a distant dependency of the verb 'making', completing the phrase 'making...more difficult'. Attentions here shown only for the word 'making'. Different colors represent different heads. Best viewed in color.

Multi-Head Self-Attention

https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html

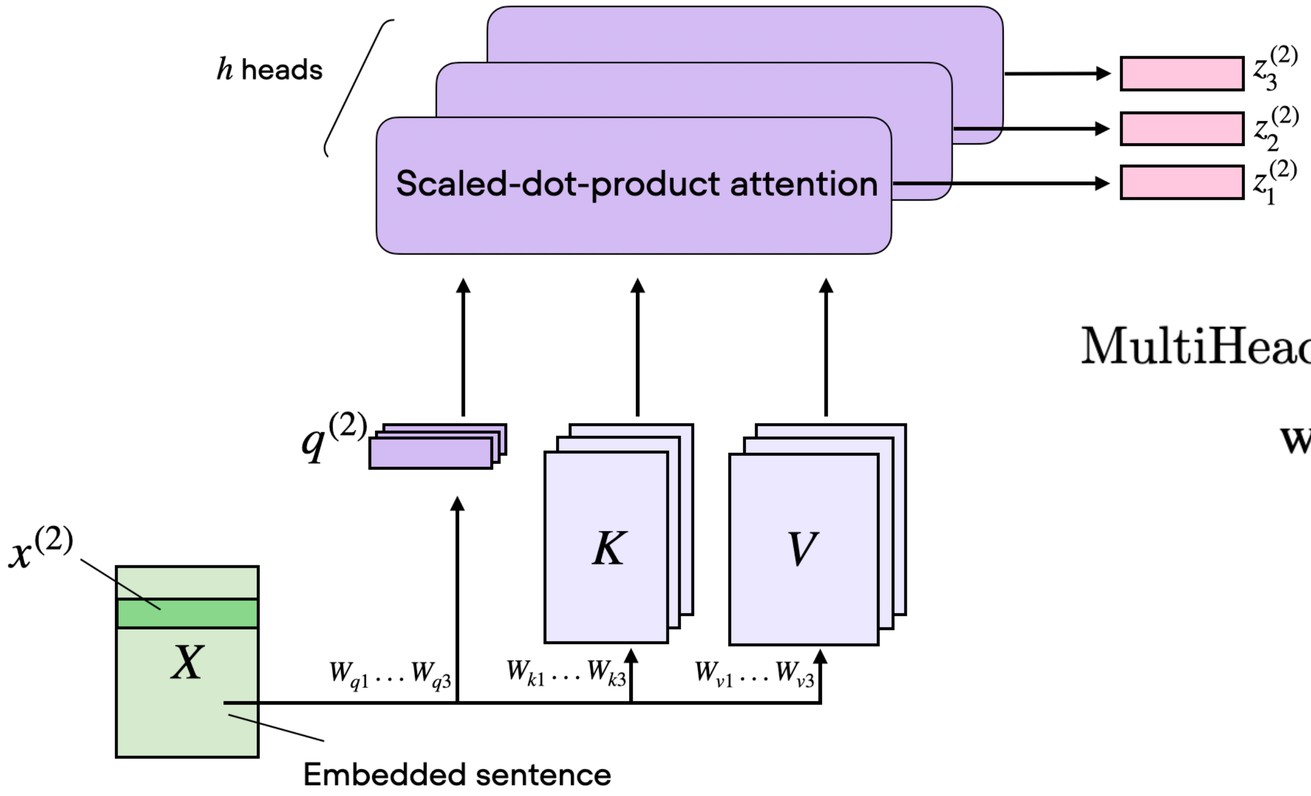


$$\text{Attention}(Q, K, V) = \text{softmax}(\frac{QK^T}{\sqrt{d_k}})V$$



Multi-Head Self-Attention

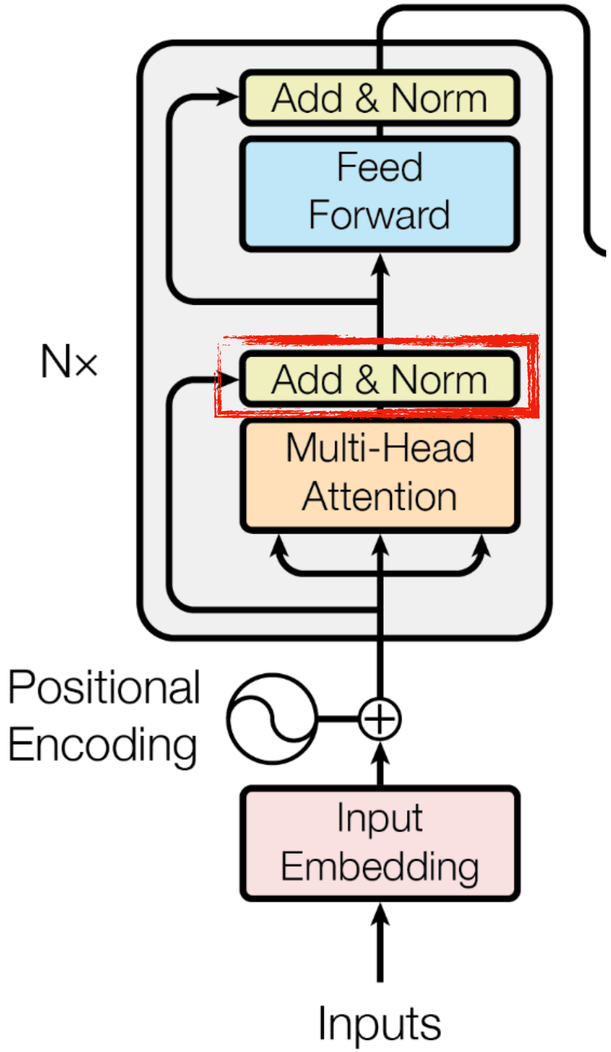
https://sebastianraschka.com/blog/2023/self-attention-from-scratch.html



$$\text{MultiHead}(Q, K, V) = \text{Concat}(\text{head}_1, ..., \text{head}_\text{h})W^O$$

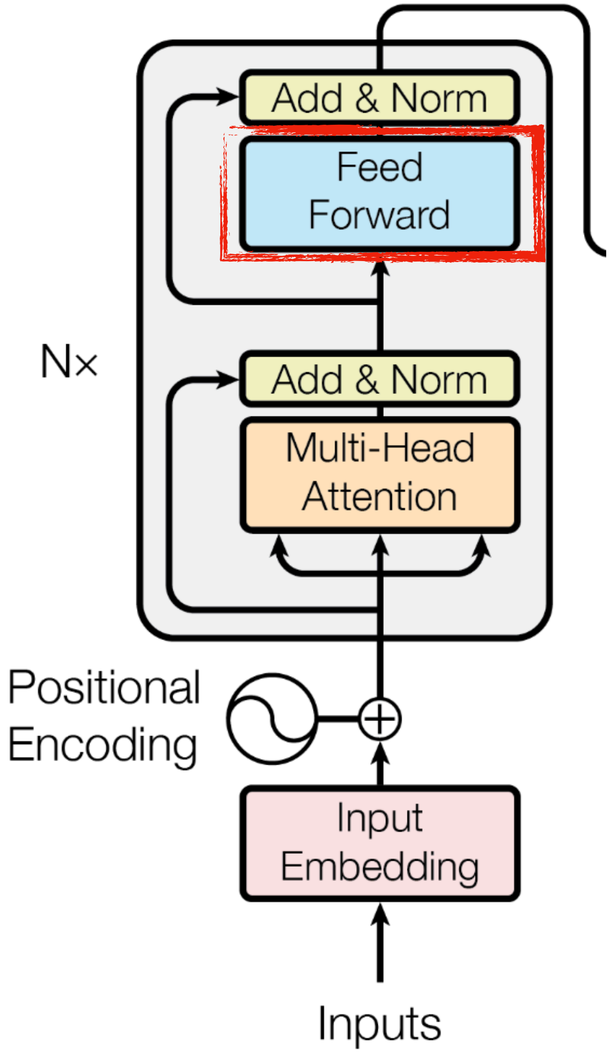
$$\text{where head}_\text{i} = \text{Attention}(QW_i^Q, KW_i^K, VW_i^V)$$

Transformer Encoder



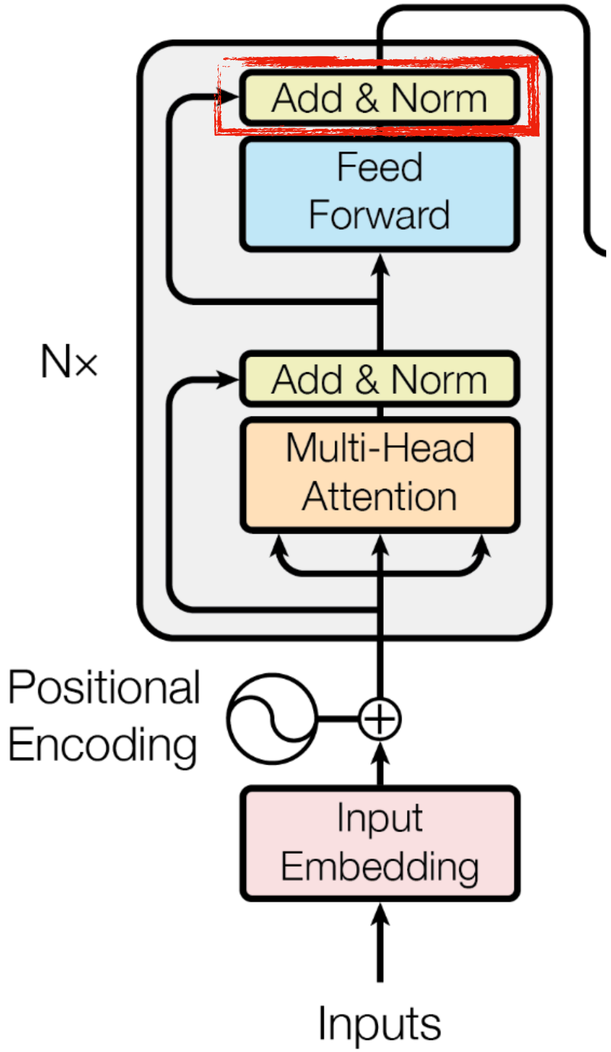
- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer
- Positional encoding to add information about words' position in the sequence
- Positional encoding is added to the embedding
- MHA used to redistribute the information among the inputs (contextualization)
- Add the input information to the result of MHA and normalize

Transformer Encoder



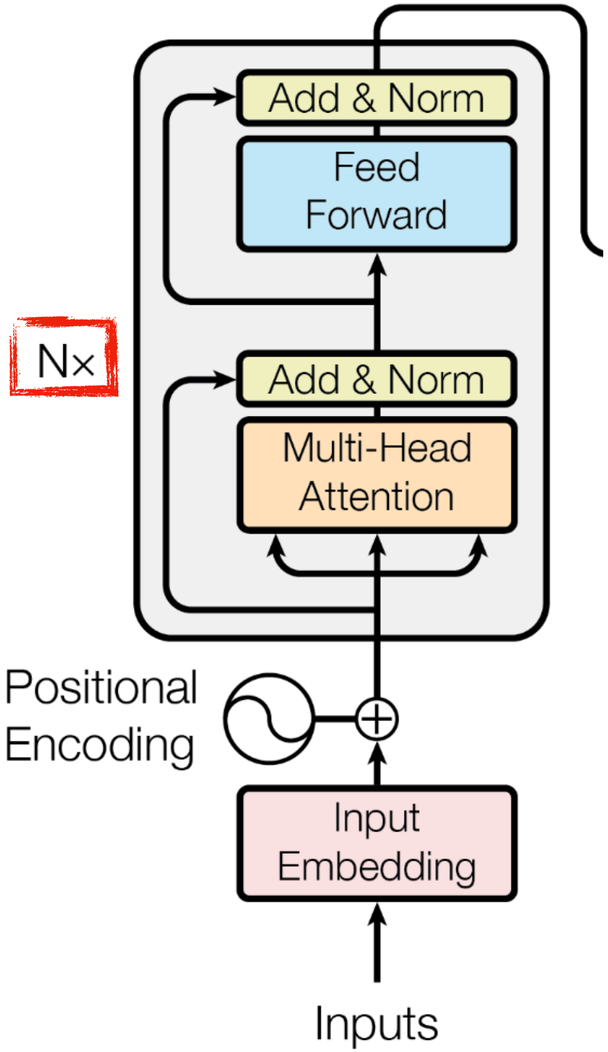
- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer
- Positional encoding to add information about words' position in the sequence
- Positional encoding is added to the embedding
- MHA used to redistribute the information among the inputs (contextualization)
- Add the input information to the result of MHA and normalize
- Just a regular feed-forward network

Transformer Encoder



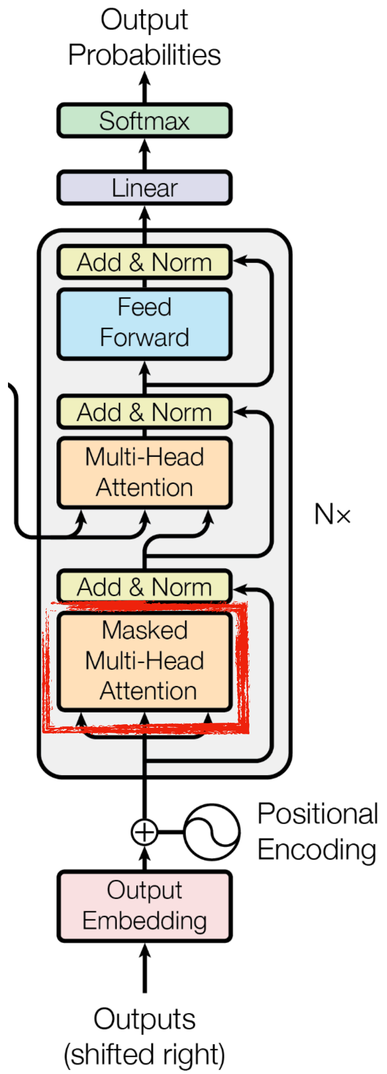
- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer
- Positional encoding to add information about words' position in the sequence
- Positional encoding is added to the embedding
- MHA used to redistribute the information among the inputs (contextualization)
- Add the input information to the result of MHA and normalize
- Just a regular feed-forward network
- Another residual connection and layer normalization

Transformer Encoder

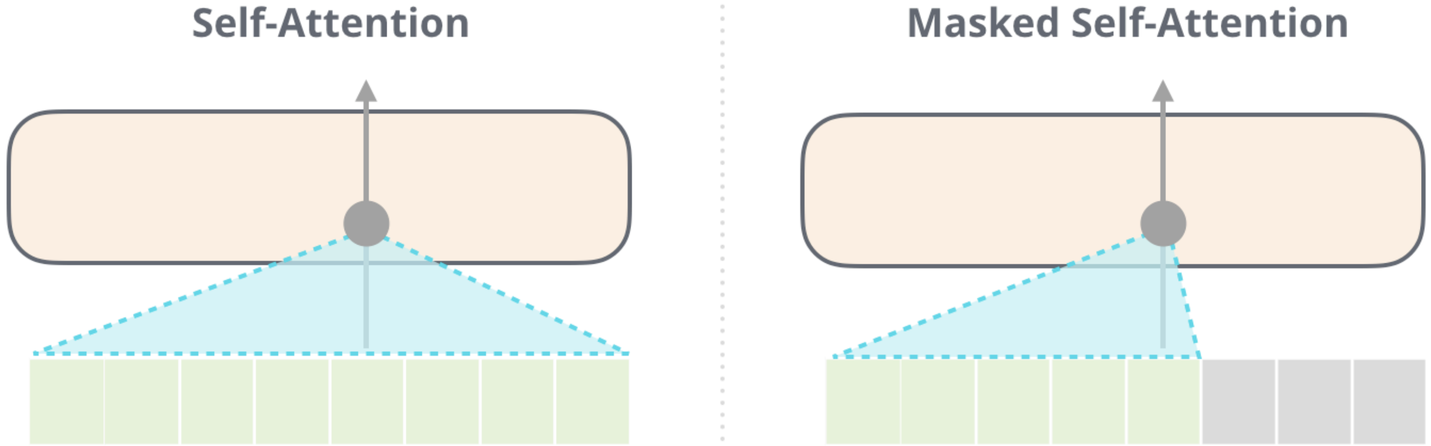


- Byte-pair Encoding (GPT, GPT-2, RoBERTa, BART, and DeBERTa) or WordPiece (BERT) tokenization
- Randomly initialized learnable embedding layer
- Positional encoding to add information about words' position in the sequence
- Positional encoding is added to the embedding
- MHA used to redistribute the information among the inputs (contextualization)
- Add the input information to the result of MHA and normalize
- Just a regular feed-forward network
- Another residual connection and layer normalization
- Repeat N times

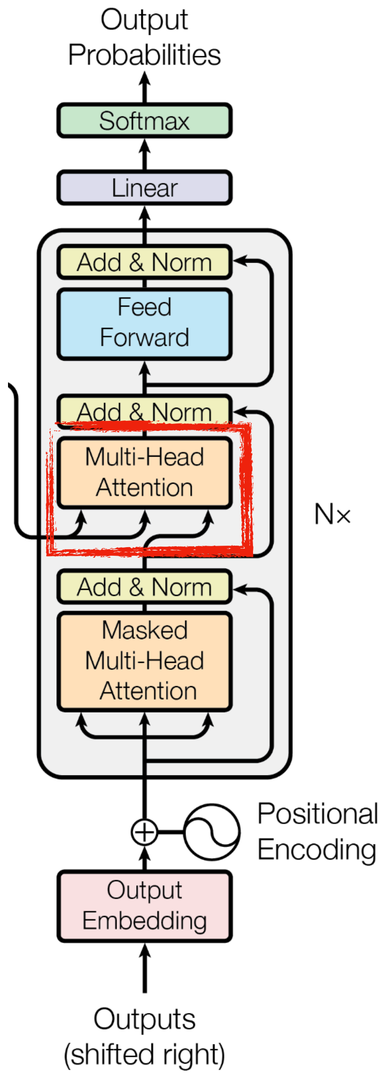
Transformer Decoder



- Has the same elements as encoder
- Masked MHA is used so that the model cannot “look into the future”

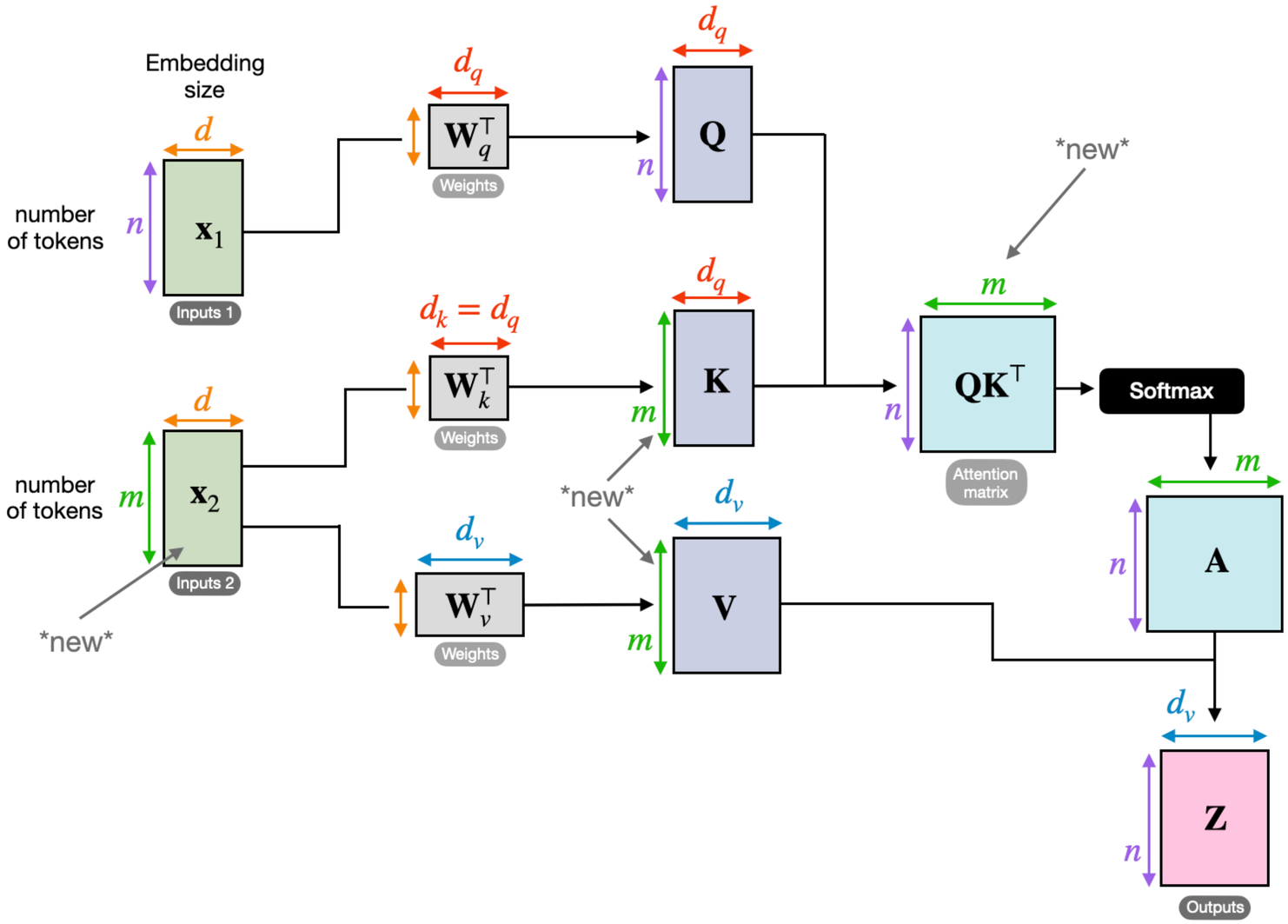


Transformer Decoder

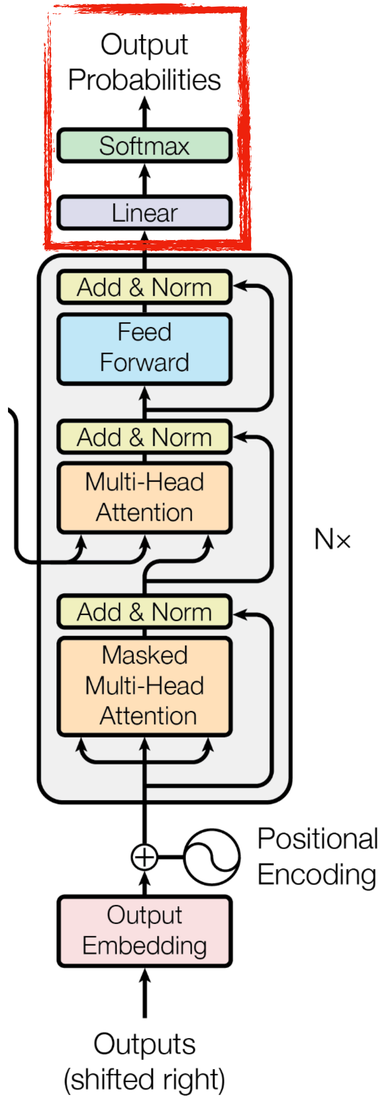


- Has the same elements as encoder
- Masked MHA is used so that the model cannot “look into the future”
- Cross-attention with encoder outputs

Cross-Attention



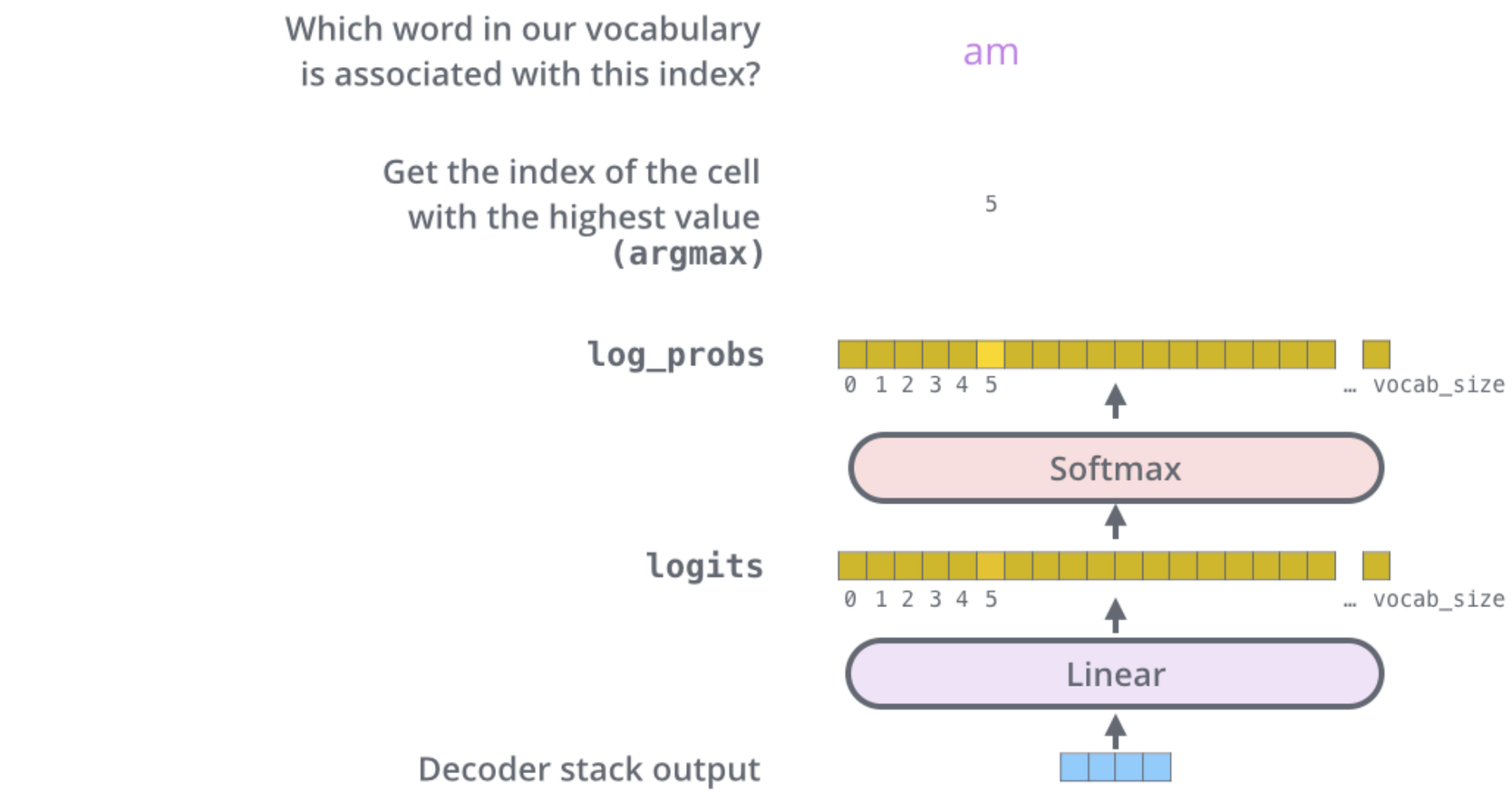
Transformer Decoder



- Has the same elements as encoder
- Masked MHA is used so that the model cannot “look into the future”
- Cross-attention with encoder outputs
- Final layer to output the next token probabilities

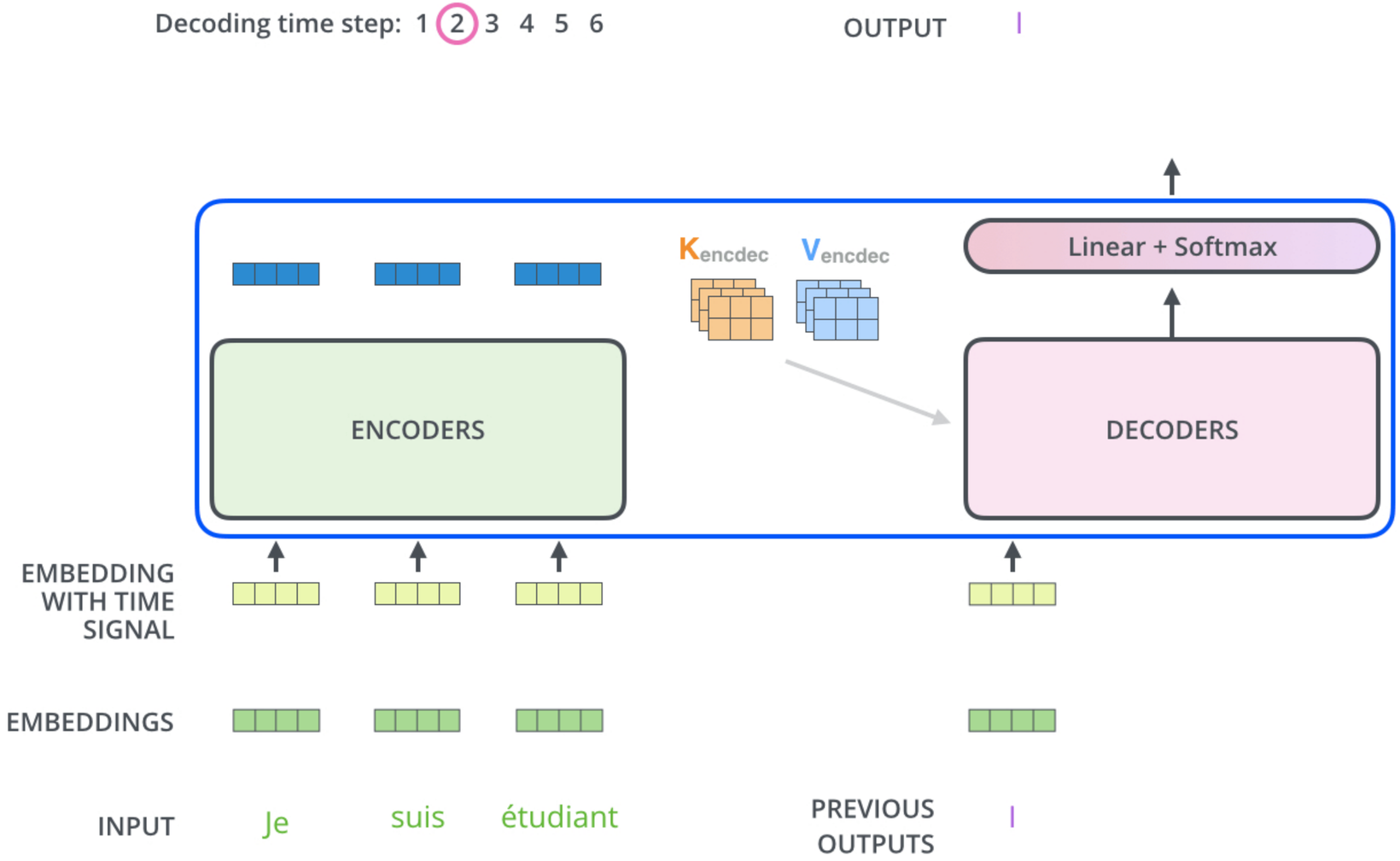
Transformer Final Layer

https://jalammar.github.io/illustrated-transformer/



Encoder-Decoder in Practice

https://jalammar.github.io/illustrated-transformer/

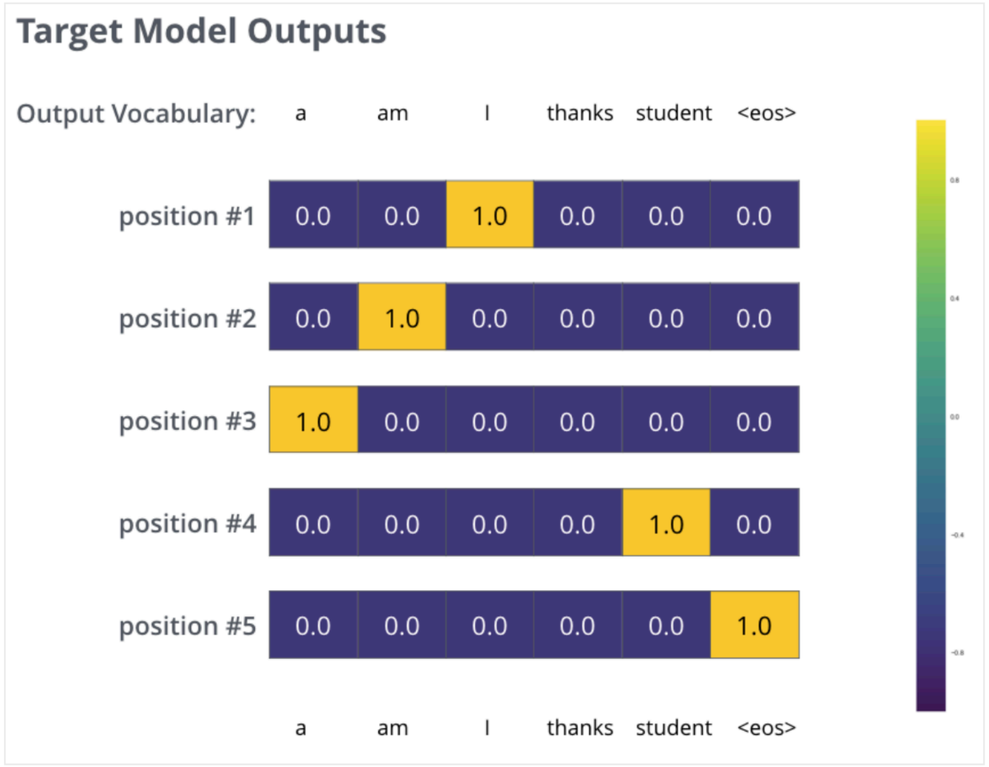


Training

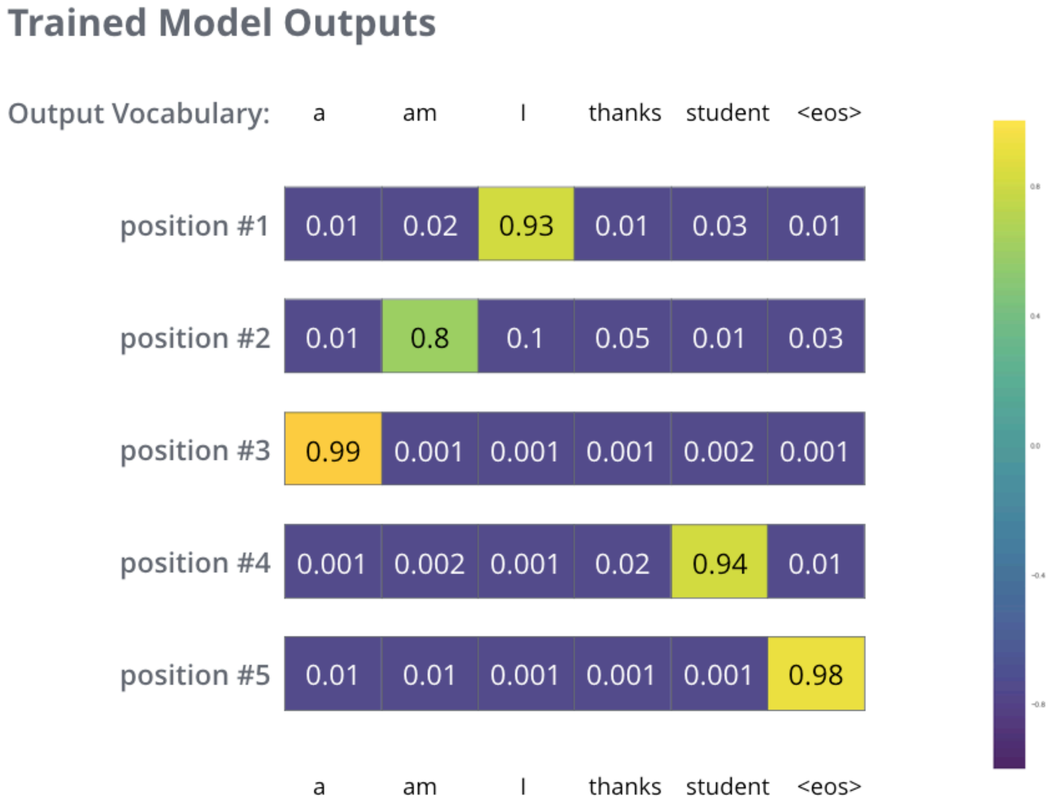
Training

Cross-Entropy Loss

$$H(P^* \mid P) = -\sum_i \underbrace{P^*(i)}_{\text{TRUE CLASS DISTIRBUTION}} \log \underbrace{P(i)}_{\text{PREDICTED CLASS DISTIRBUTION}}$$



The targeted probability distributions we'll train our model against in the training example for one sample sentence.



Training

- Usually involves pre-training and fine-tuning
- Pre-training is done on large amount of general purpose texts
- Fine-tuning is on a smaller but more specific corpus

Evaluation

BLEU Score

Mathematically, the BLEU score is defined as:

$$\text{BLEU} = \underbrace{\min\Big(1, \exp\big(1 - \frac{\text{reference-length}}{\text{output-length}}\big)\Big)}_{\text{brevity penalty}} \underbrace{\Big(\prod_{i=1}^{4} precision_i\Big)^{1/4}}_{\text{n-gram overlap}}$$

with

$$precision_i = \frac{\sum_{\text{snt} \in \text{Cand-Corpus}} \sum_{i \in \text{snt}} \min(m^i_{cand}, m^i_{ref})}{w^i_t = \sum_{\text{snt'} \in \text{Cand-Corpus}} \sum_{i' \in \text{snt'}} m^{i'}_{cand}}$$

where

- m^i_{cand} is the count of i-gram in candidate matching the reference translation
- m^i_{ref} is the count of i-gram in the reference translation
- w^i_t is the total number of i-grams in candidate translation

BLEU Score

BLEU Score	Interpretation
< 10	Almost useless
10 - 19	Hard to get the gist
20 - 29	The gist is clear, but has significant grammatical errors
30 - 40	Understandable to good translations
40 - 50	High quality translations
50 - 60	Very high quality, adequate, and fluent translations
> 60	Quality often better than human

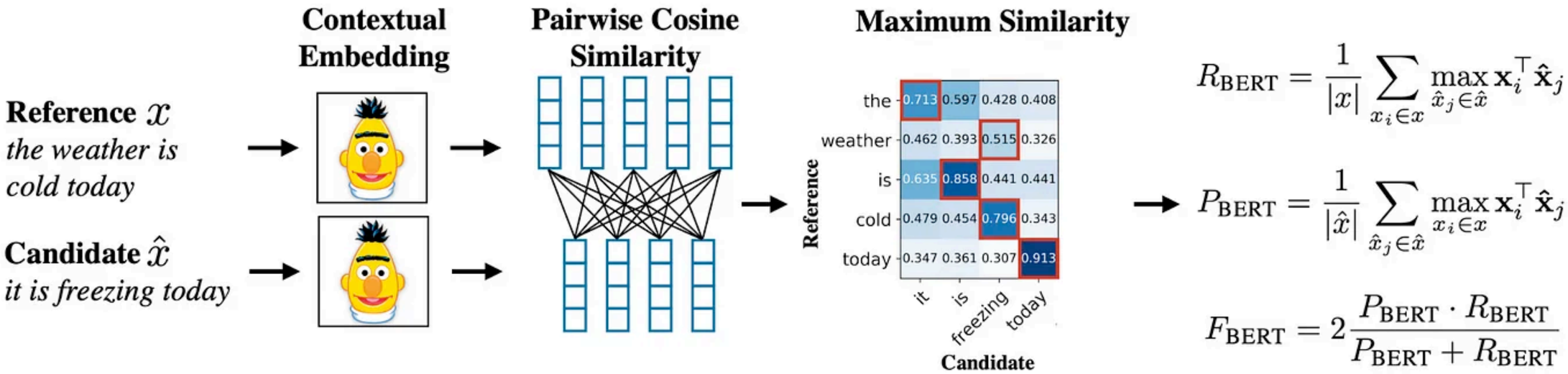
ROUGE Score

$$RECALL = \frac{Overlapping\ number\ of\ n-grams}{Number\ of\ n-grams\ in\ the\ reference}$$

$$PRECISION = \frac{Overlapping\ number\ of\ n-grams}{Number\ of\ n-grams\ in\ the\ candidate}$$

$$F1 = \frac{2 \times Precision \times Recall}{(Precision + Recall)}$$

BERT Score



Source: Bertscore: Evaluating text generation with bert

Examples

Vanilla Transformer

Vaswani, Ashish, et al. "Attention is all you need.” (2017)

- Trained for machine translation
  - WMT 2014 English-German dataset (4.5 million sentence pairs)
  - WMT 2014 English-French dataset (36 million sentence pairs)

Vanilla Transformer

Results

Table 2: The Transformer achieves better BLEU scores than previous state-of-the-art models on the English-to-German and English-to-French newstest2014 tests at a fraction of the training cost.

Model	BLEU		Training Cost (FLOPs)	
	EN-DE	EN-FR	EN-DE	EN-FR
ByteNet [18]	23.75			
Deep-Att + PosUnk [39]		39.2		$1.0 \cdot 10^{20}$
GNMT + RL [38]	24.6	39.92	$2.3 \cdot 10^{19}$	$1.4 \cdot 10^{20}$
ConvS2S [9]	25.16	40.46	$9.6 \cdot 10^{18}$	$1.5 \cdot 10^{20}$
MoE [32]	26.03	40.56	$2.0 \cdot 10^{19}$	$1.2 \cdot 10^{20}$
Deep-Att + PosUnk Ensemble [39]		40.4		$8.0 \cdot 10^{20}$
GNMT + RL Ensemble [38]	26.30	41.16	$1.8 \cdot 10^{20}$	$1.1 \cdot 10^{21}$
ConvS2S Ensemble [9]	26.36	**41.29**	$7.7 \cdot 10^{19}$	$1.2 \cdot 10^{21}$
Transformer (base model)	27.3	38.1	**$3.3 \cdot 10^{18}$**	
Transformer (big)	**28.4**	**41.8**	$2.3 \cdot 10^{19}$	

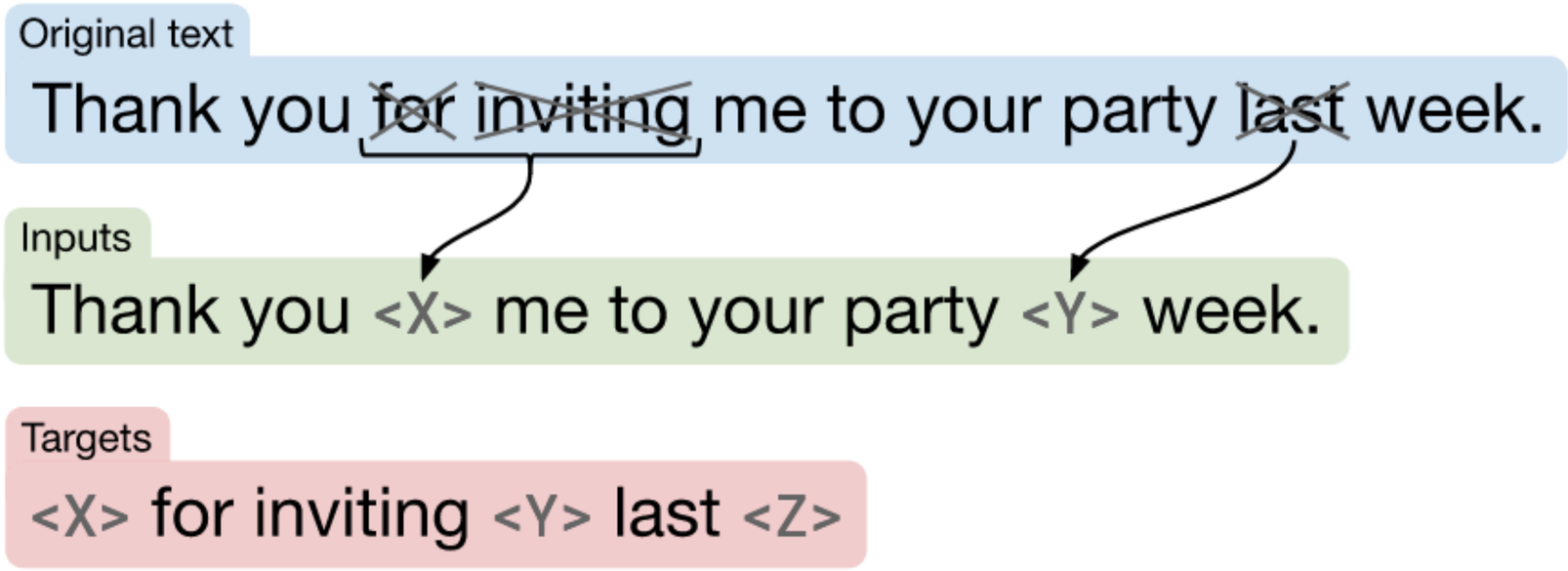
T5

Raffel, Colin, et al. "Exploring the limits of transfer learning with a unified text-to-text transformer.” (2020)

- Encoder-decoder based large language model (LLM)
- Pre-trained on Colossal Clean Crawled Corpus (C4) - about 750 GB of cleaned English text (about 34 billion tokens)
- Use span-corruption objective for pre-training
- Fine-tuning on GLUE and SuperGLUE tasks

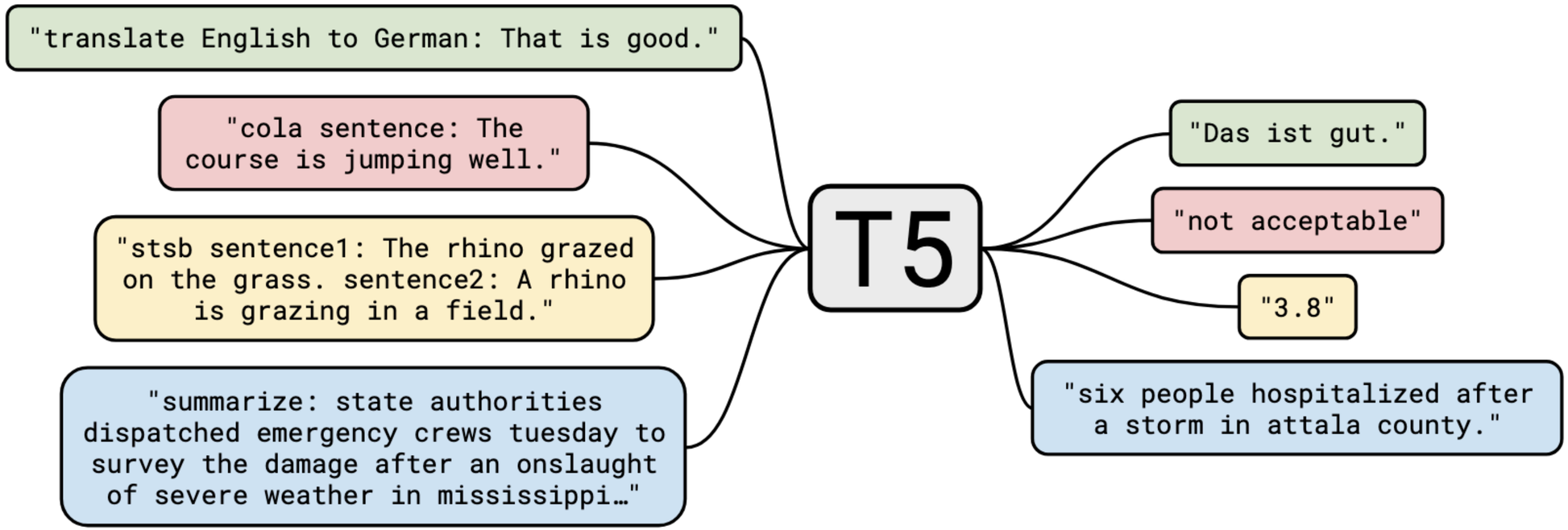
T5

Pre-training



T5

Fine-tuning



T5

Results

Model	GLUE Average	CoLA Matthew's	SST-2 Accuracy	MRPC F1	MRPC Accuracy	STS-B Pearson	STS-B Spearman
Previous best	89.4[a]	69.2[b]	97.1[a]	**93.6**[b]	**91.5**[b]	92.7[b]	92.3[b]
T5-Small	77.4	41.0	91.8	89.7	86.6	85.6	85.0
T5-Base	82.7	51.1	95.2	90.7	87.5	89.4	88.6
T5-Large	86.4	61.2	96.3	92.4	89.9	89.9	89.2
T5-3B	88.5	67.1	97.4	92.5	90.0	90.6	89.8
T5-11B	**90.3**	**71.6**	**97.5**	92.8	90.4	**93.1**	**92.8**

Model	QQP F1	QQP Accuracy	MNLI-m Accuracy	MNLI-mm Accuracy	QNLI Accuracy	RTE Accuracy	WNLI Accuracy
Previous best	74.8[c]	**90.7**[b]	91.3[a]	91.0[a]	**99.2**[a]	89.2[a]	91.8[a]
T5-Small	70.0	88.0	82.4	82.3	90.3	69.9	69.2
T5-Base	72.6	89.4	87.1	86.2	93.7	80.1	78.8
T5-Large	73.9	89.9	89.9	89.6	94.8	87.2	85.6
T5-3B	74.4	89.7	91.4	91.2	96.3	91.1	89.7
T5-11B	**75.1**	90.6	**92.2**	**91.9**	96.9	**92.8**	**94.5**

Model	SQuAD EM	SQuAD F1	SuperGLUE Average	BoolQ Accuracy	CB F1	CB Accuracy	COPA Accuracy
Previous best	90.1[a]	95.5[a]	84.6[d]	87.1[d]	90.5[d]	95.2[d]	90.6[d]
T5-Small	79.10	87.24	63.3	76.4	56.9	81.6	46.0
T5-Base	85.44	92.08	76.2	81.4	86.2	94.0	71.2
T5-Large	86.66	93.79	82.3	85.4	91.6	94.8	83.4
T5-3B	88.53	94.95	86.4	89.9	90.3	94.4	92.0
T5-11B	**91.26**	**96.22**	**88.9**	**91.2**	**93.9**	**96.8**	**94.8**

Model	MultiRC F1a	MultiRC EM	ReCoRD F1	ReCoRD Accuracy	RTE Accuracy	WiC Accuracy	WSC Accuracy
Previous best	84.4[d]	52.5[d]	90.6[d]	90.0[d]	88.2[d]	69.9[d]	89.0[d]
T5-Small	69.3	26.3	56.3	55.4	73.3	66.9	70.5
T5-Base	79.7	43.1	75.0	74.2	81.5	68.3	80.8
T5-Large	83.3	50.7	86.8	85.9	87.8	69.3	86.3
T5-3B	86.8	58.3	91.2	90.4	90.7	72.1	90.4
T5-11B	**88.1**	**63.3**	**94.1**	**93.4**	**92.5**	**76.9**	**93.8**

Model	WMT EnDe BLEU	WMT EnFr BLEU	WMT EnRo BLEU	CNN/DM ROUGE-1	CNN/DM ROUGE-2	CNN/DM ROUGE-L
Previous best	**33.8**[e]	**43.8**[e]	**38.5**[f]	43.47[g]	20.30[g]	40.63[g]
T5-Small	26.7	36.0	26.8	41.12	19.56	38.35
T5-Base	30.9	41.2	28.0	42.05	20.34	39.40
T5-Large	32.0	41.5	28.1	42.50	20.68	39.75
T5-3B	31.8	42.6	28.2	42.72	21.02	39.94
T5-11B	32.1	43.4	28.1	**43.52**	**21.55**	**40.69**

FlanT5

Chung, Hyung Won, et al. "Scaling instruction-finetuned language models." (2022)

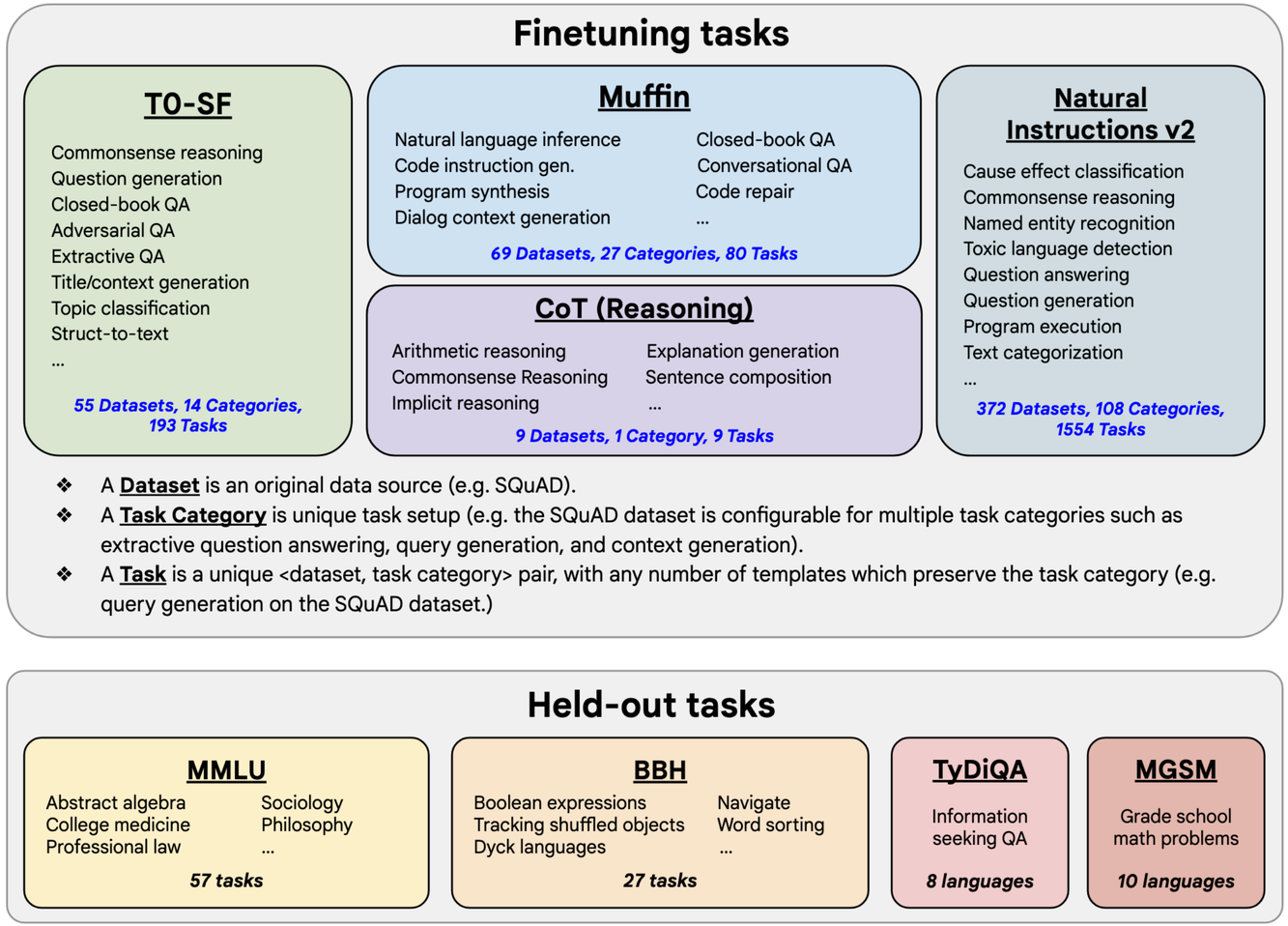


Figure 2: Our finetuning data comprises 473 datasets, 146 task categories, and 1,836 total tasks. Details for the tasks used in this paper is given in Appendix F.

FlanT5

Fine-tuning

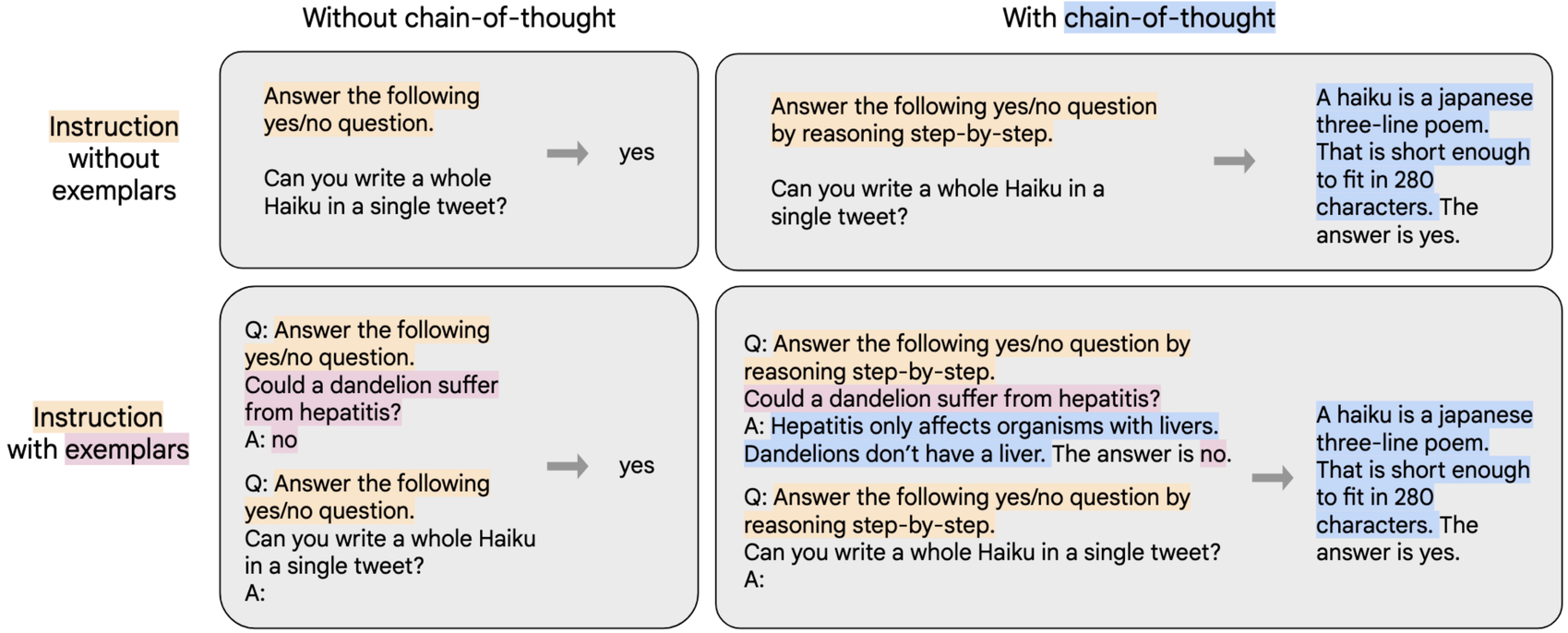
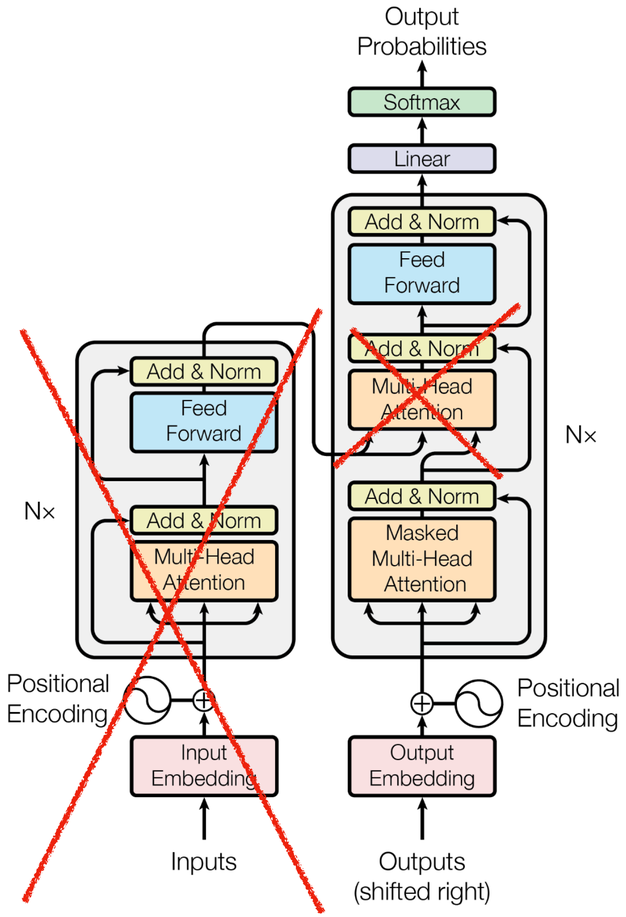


Figure 3: Combinations of finetuning data formats in this work. We finetune with and without exemplars, and also with and without chain-of-thought. In addition, we have some data formats without instructions but with few-shot exemplars only, like in Min et al. (2022) (not shown in the figure). Note that only nine chain-of-thought (CoT) datasets use the CoT formats.

GPT-2

Radford, Alec, et al. "Language models are unsupervised multitask learners.” (2019)



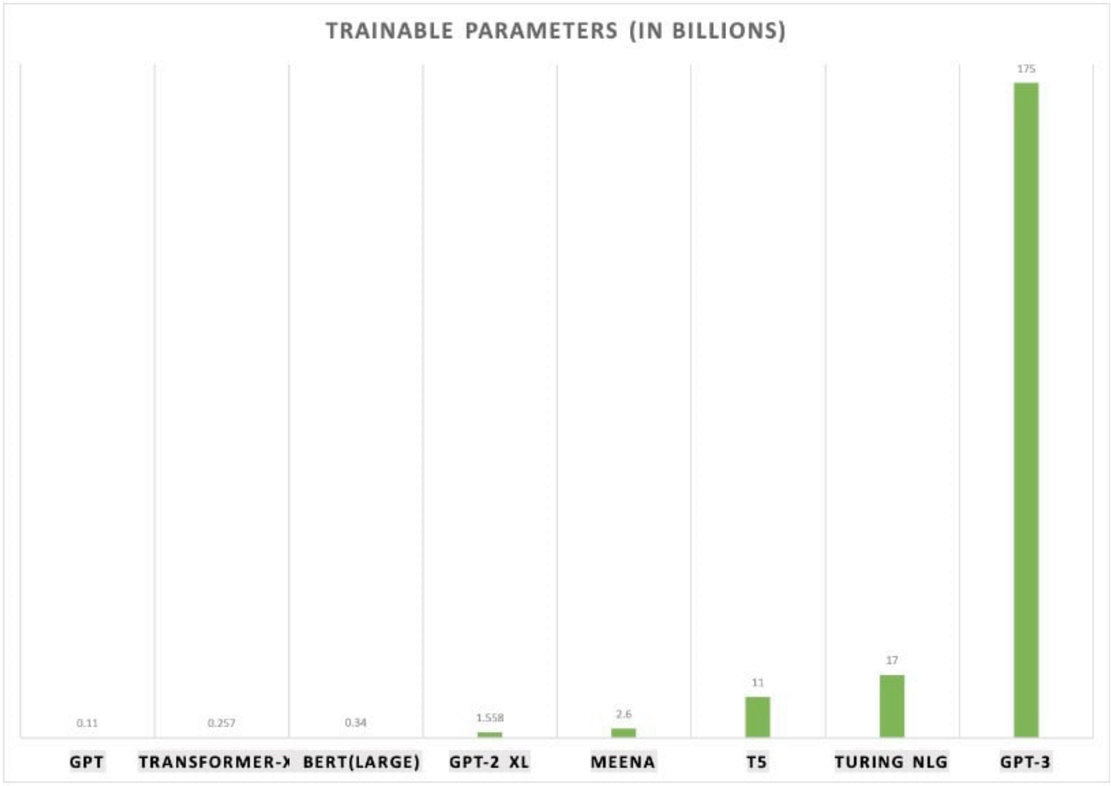
GPT-2

- Decoder-only large language model (LLM)
- Pre-trained on WebText (8 million documents ~ 40 GB of text)
- Some differences with the Vanilla Transformer
  - Layer normalization was moved to the input of each sub-block
  - Additional layer normalization was added after the final self-attention block

GPT-3

Brown, Tom, et al. "Language models are few-shot learners.” (2020)

- Same architecture as GPT-2 but much (much!) bigger
- A huge amount of training data (from 550GB to



Instruct GPT

Ouyang, Long, et al. "Training language models to follow instructions with human feedback." (2022)

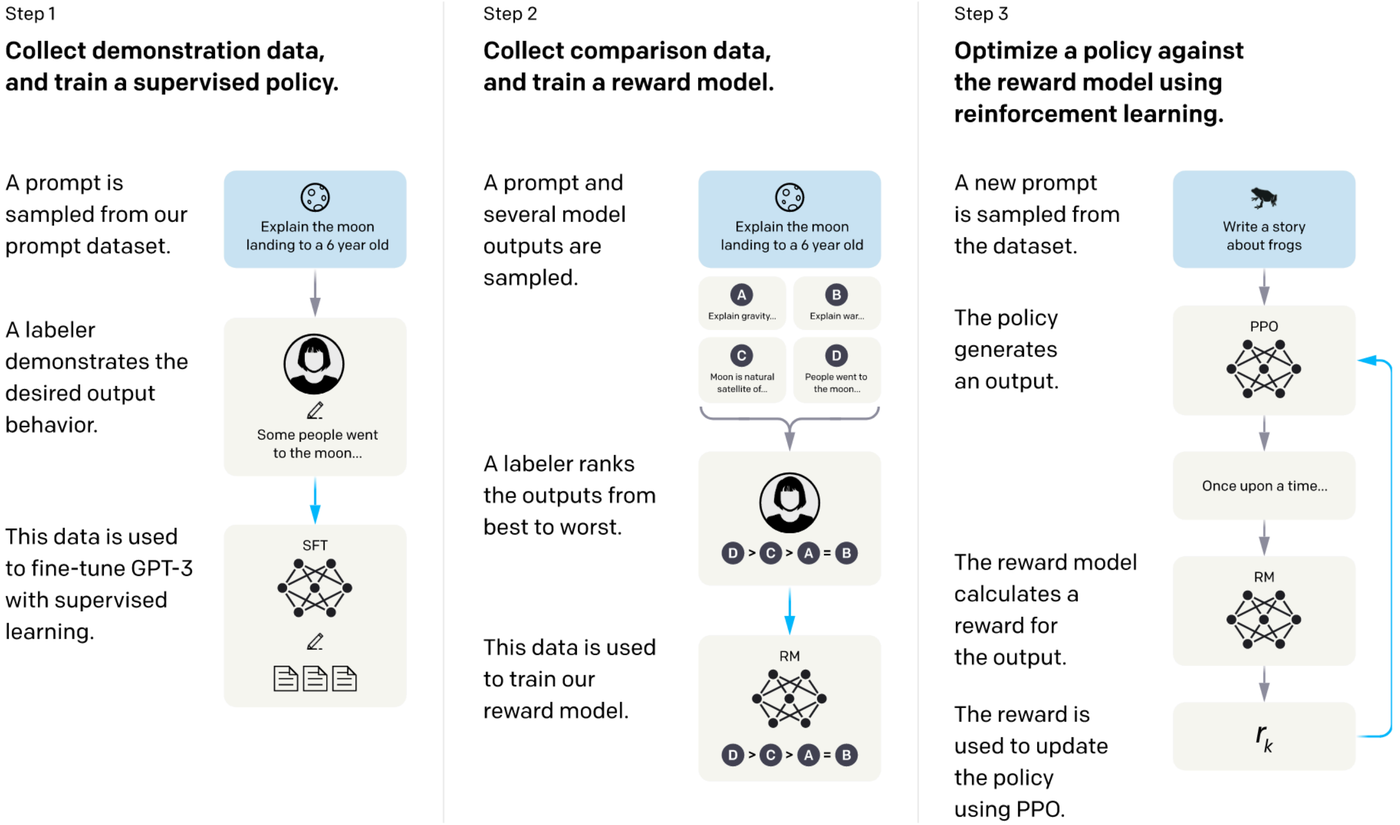


Figure 2: A diagram illustrating the three steps of our method: (1) supervised fine-tuning (SFT), (2) reward model (RM) training, and (3) reinforcement learning via proximal policy optimization (PPO) on this reward model. Blue arrows indicate that this data is used to train one of our models. In Step 2, boxes A-D are samples from our models that get ranked by labelers. See Section 3 for more details on our method.

LLaMA

- Large training dataset (1.4 trillion tokens)
- Differences with Vanilla Transformer:
  - Pre-normalization (like GPT-3)
  - SwiGLU activation function instead of ReLU
  - Rotary embeddings instead of positional embeddings

Mistral

- Dataset is unknown
- Architecture similar to LLaMA with several changes:
  - Sliding window attention that allows input and output sequences up to 131K tokens
  - Various optimization techniques

Conclusion

- NLG is an active research field
- A lot of current NLP tasks can be solved with text generation
- Task-specific models are still better for many tasks but large language models are more versatile and generalizable
- Entry threshold to the LLM is very high (required a lot of recourses to train and use)

TRAITEMENT AUTOMATIQUE DES LANGUES AVANCE

Master Informatique

2ème Année - 1er Semestre

Intervenants CM:

Gaël DIAS (gael.dias@unicaen.fr), Marc SPANIOL, Fabrice MAUREL,

Navneet AGARWAL, Kirill MILINTSEVICH

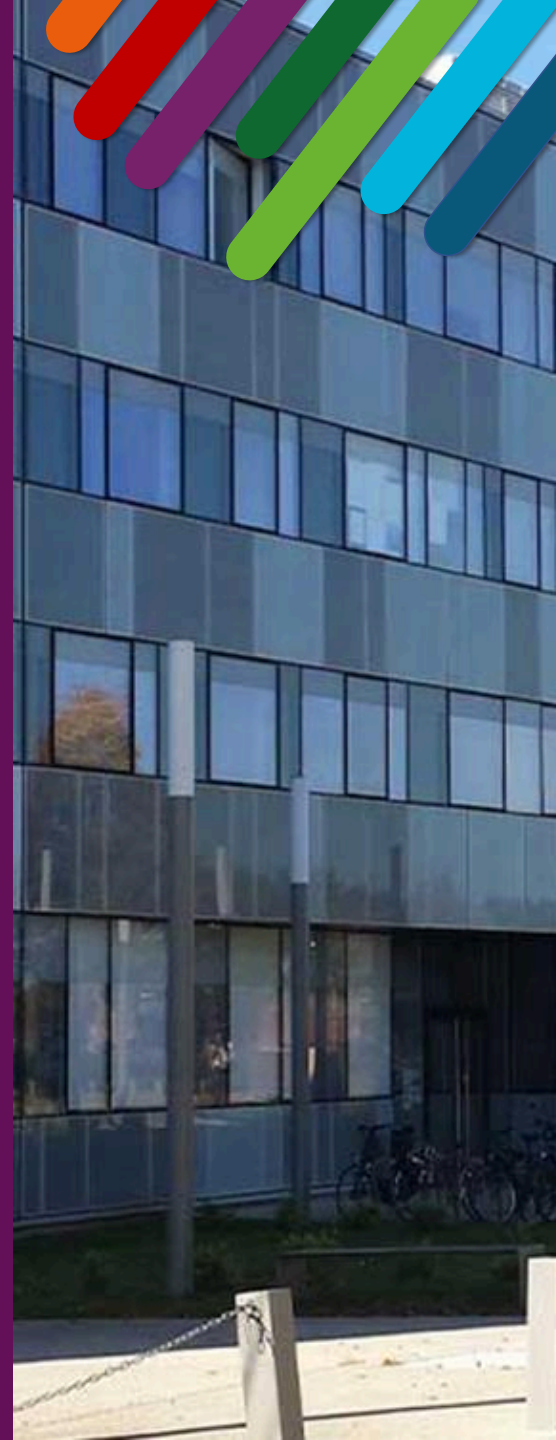
GREYC Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN ÉCOLE PUBLIQUE D'INGÉNIEURS CENTRE DE RECHERCHE

cnrs

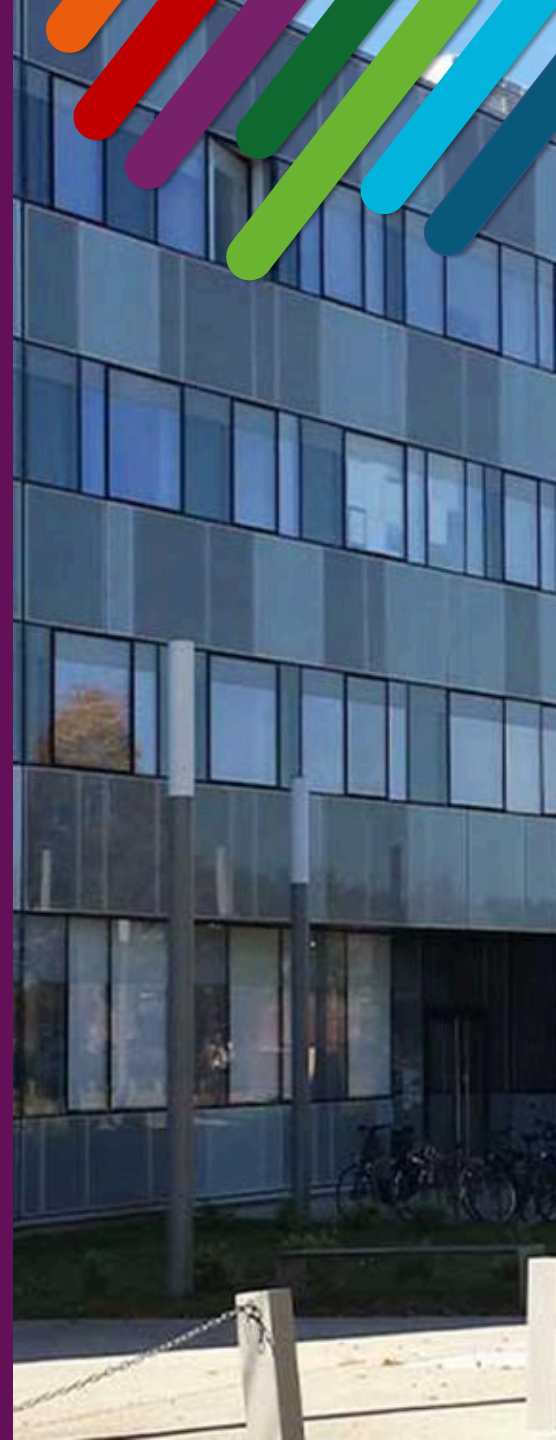


Plan de l'UE

1. **[CM 1]** Représentation sémantique de texte [GD]
2. **[CM 2]** Cohérence textuelle [MS]
3. **[CM 3]** Modélisation thématique [NA]
4. **[CM 4]** Résumé de textes et traduction automatique [MS]
5. **[CM 5]** Génération langagière I [KM]
6. **[CM 6]** Génération langagière II [KM]
7. **[CM 7]** Sentiment Analysis, Image Captioning [NA]
8. **[CM 8]** TAL et web [MS]
9. **[CM 9]** TAL et handicap visuel [FM]
10. **[CM 10]** TAL et psychiatrie [GD]

11. **[TP 1-5]** Génération neuronal de comptes-rendus médicaux [NA - KM]

Sentiment Analysis



GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs

Sentiment Analysis

Sentiment analysis (often refered to as opinion mining) is the process of gathering and analyzing people's opinions, thoughts and impressions regarding various topics, products, subjects and services.

- Rapid growth of internet based applications such as social media and blogs.
- People's opinions can be beneficial for
  - Corporations: product reviews on amazon, sentiment towards a certain feature or service.
  - Governments: public opinion on policies, expected voting tendencies before election.
  - Entertainment industry: movie reviews, game reviews.
  - Travel: reviews of restaurants, hotels, tourist attractions, etc.

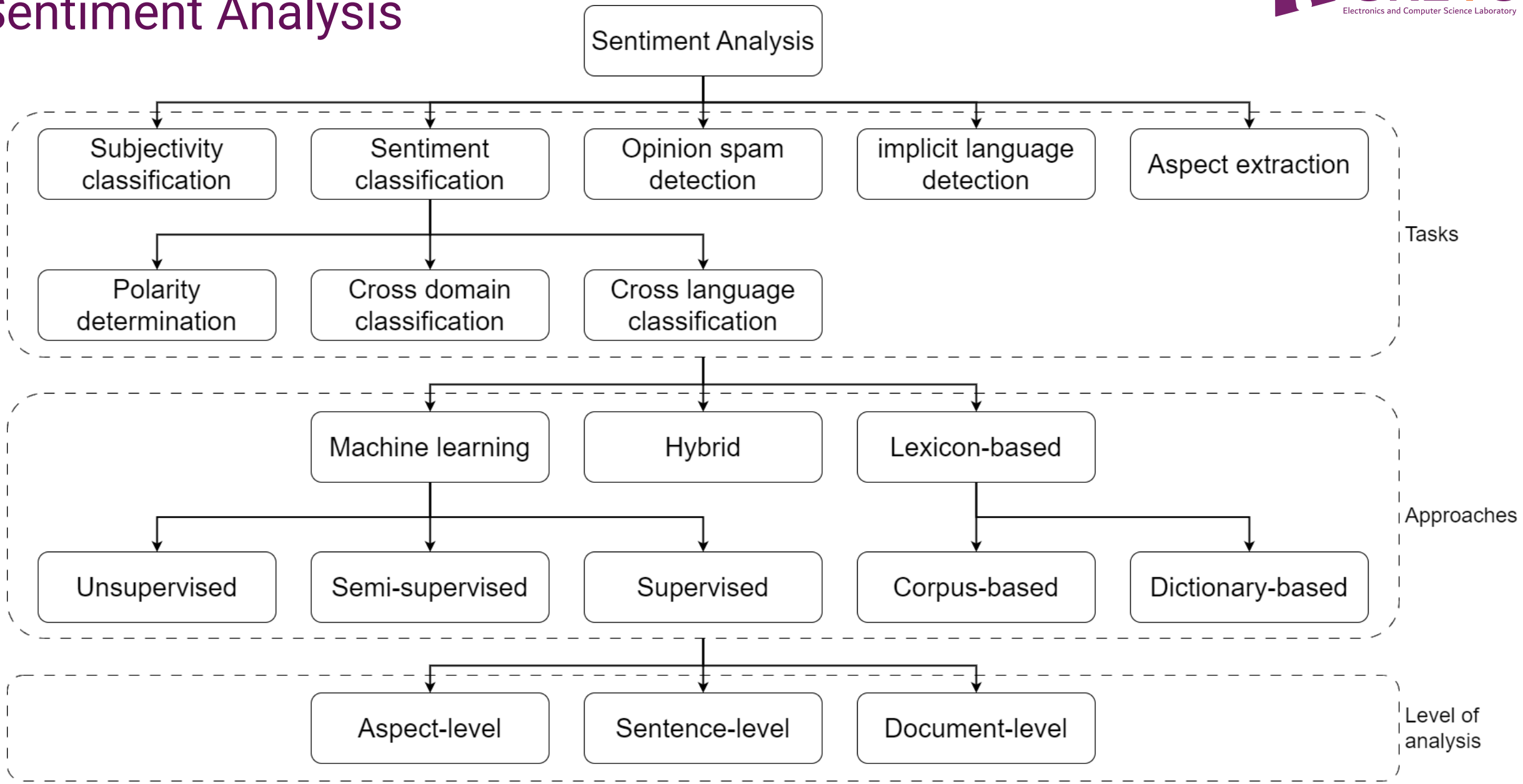
Sentiment Analysis

Sentiment analysis is a broad concept that consists of many different tasks, approaches and types of analysis.

Cambri at al. argue that a holistic approach is required , and only classification or categorization is not sufficient. They present it as a 3 layered problem that includes 15 NLP problems:

- Syntactic layer: Microtext normalization, sentence boundary disambiguation, POS tagging, text chunking and lemmatization.
- Semantics layer: Word sense disambiguation, concept extraction, named entity recognition, anaphora resoluion and subjectivity detection.
- Pragmatics layer: Personality recognition, sarcasm detection, metaphor understanding, aspect extraction and polarity detection.

Sentiment Analysis



Tasks

❑ Sentiment classification

- Most widely known and researched task in sentiment analysis.
- It can be divided into three major sub-tasks: polarity classification, cross-domain classification and cross-language classification.
- Polarity is usually classified as positive or negative with some researchers including a third category neutral.
- Cross-domain classification models transfer knowledge learned from data-rich source domain to a target domain where data and\or labels are limited.
  - Extract domain invarient features whose distribution in the source domain is close to that in target domain
- Cross-language classification fulfills the same function but for languages.
  - An example can be to train the model in source language with abundant data and testing it on target language by translating the input to source language.

Tasks

- Subjectivity classification
  - The goal of subjectivity classifiation is to restrict unwanted objective data objects for further processing.
  - It detects subjective clues, words that carry emotion or subjective notion like ‘expensive’, ‘easy’, ‘better’, etc.
  - These clues are used to classify objects as subjective or objective.

Ce livre coûte 10 euros → Cannot be used for sentiment analysis

Ce livre est cher → Can be used for sentiment analysis

Tasks

❑ Opinion spam detection

- Opinion spams refer to fake or false reviews intelligently written to either promote or discredit a product.
- Three main features are considered within this context:
  - Review content: the actual text of the review
  - Meta-data: information like IP address, geo-location, user-id, etc.
  - Real-life knowledge: this meathod utilizes learned experiences to classify spam. For example, if a aproduct has good reputation and suddenly inferior ratings are given over a period, reviews of that period might be suspected.

Tasks

❑ Implicit language detection

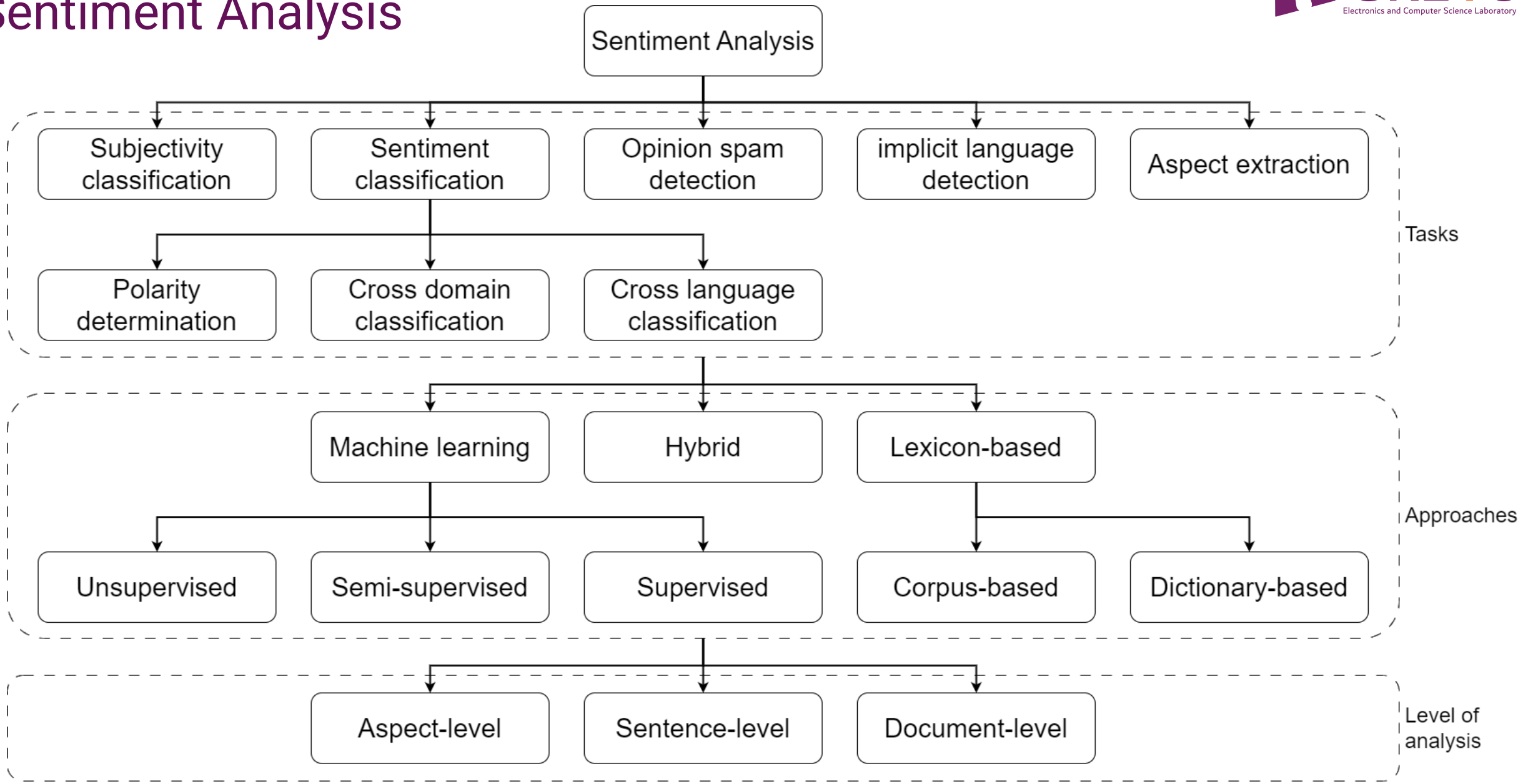
- Implicit language includes humor, sarcasm ans irony.
- There is vagueness and ambiguity in this form of speech, which is sometimes hard to detect.
- An implicit meaning can sometimes completly flip the polarity of a sentence.
- Example, « I love pain », pain is a factual word with negative polarity. The contradiction between pain and love can indicate sarcasm.
- Traditional methods for detection include exploring emoticons, expressions of laughter and heavy punctuation mark usage.

Tasks

❑ Aspect extraction

- Aspect extraction refers to retriving the target entity and the aspects of the target entity in the document. The target entity can be a person, product, event, etc.
- Aspect extraction is particularly important in sentiment analysis of social media and blogs that often do not have predefined topics.
- Most traditional method for this is frequency based where most frequent nouns and compound nouns are considered as candidates for aspects.
  - Not all nouns are aspects
- Syntax-based methods find aspects by means of syntactic relations they are in. For example, identifying aspects that are preceded by a modifying adjective that is a sentiment word.
  - Many relations need to be found for complete coverage

Sentiment Analysis



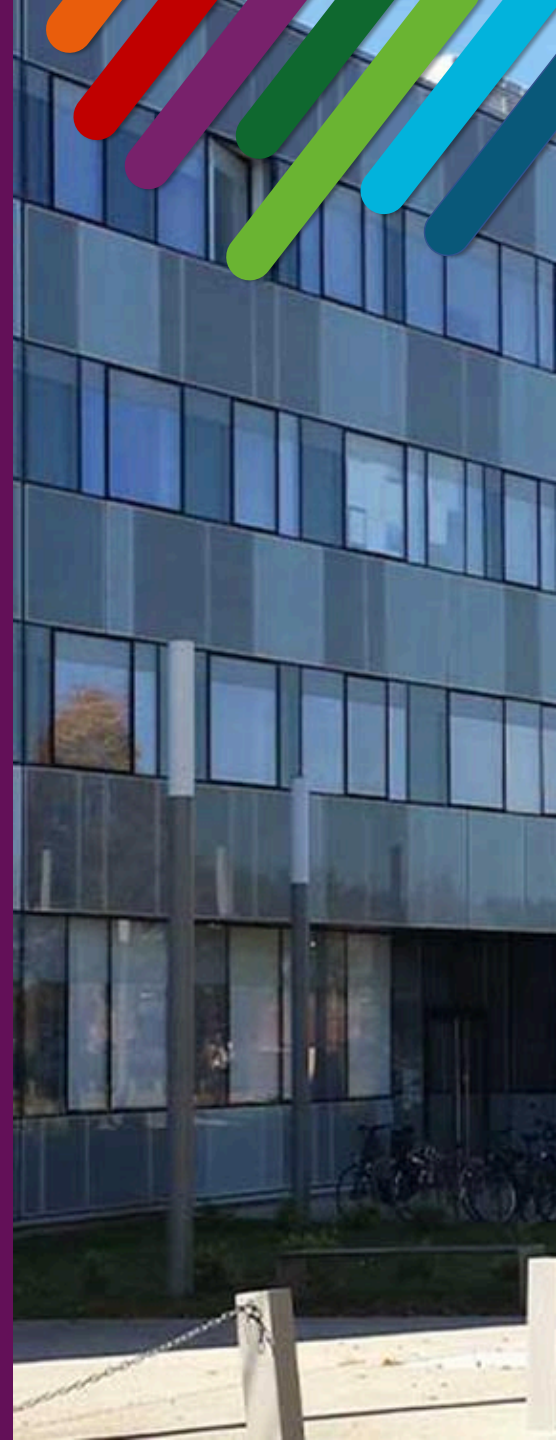
Lexicon based approaches

- Traditional approach for sentiment analysis that scans through the text for words that express positive or negative feelings to humans.
- It shows to be extremly dependent on domain of interest due to differences in language usage between domains.
- There are two main approaches to creating sentiment lexicons: dictionary based and corpus based.
- Dictionary based appraches start with an initial list of terms and iteratively expand the lexicon by adding synonyms and antonyms of the the terms to the list.
- They work best for general purpose use.
- Corpus based approaches starts with general purpose list of words and finds others terms from domain specific copus based on co-occurring word patterns.

Machine learning based approaches

- Machine learning based approaches can be divided into three categories: unsupervised, semi-supervised and supervised.
- Unsupervised approaches group unlabelled data into groups based on similarity to each other.
- Semi-supervised methods use both labelled and unlabelled data in training process.
- In cross-domain or cross-language classification, domain invarient features can be learned with unlabelled data and then labelled data can be used for fine-tuning the model.
- Supervised learning usually provides the best performance but also requires significantly more human effort in order to generate the labels.
- Machine learning approaches are especially popular for aspect extraction task with topic modelling being the most commonly used approach.

Image Captioning



GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

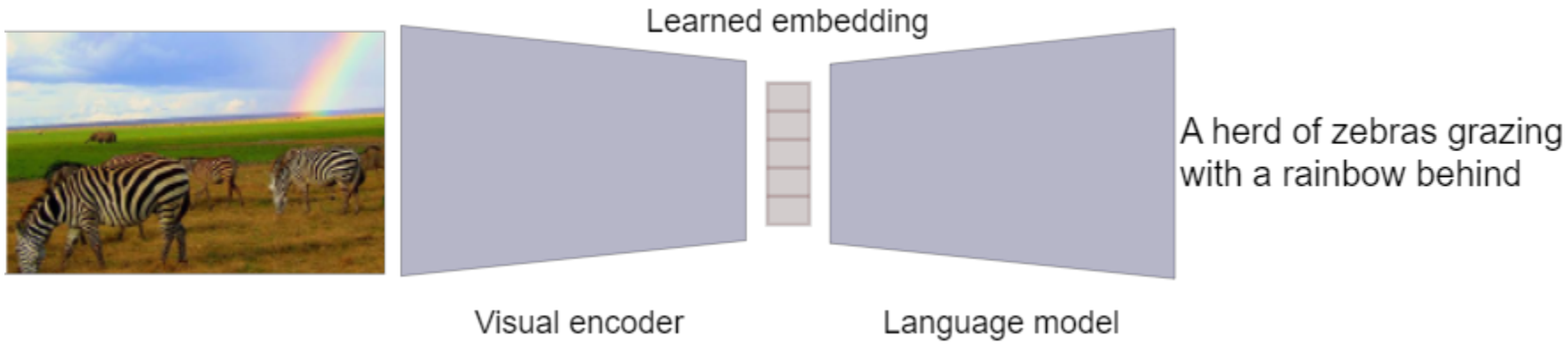
cnrs

Image captioning

Image captioning is the task of describing the visual content of an image in natural language.

- Visual component: model for understanding the visual data.
- Language component: model for generating meaningful and syntactically correct text based on learned image representation.

In its standard configuration the task is a image-to-sequence problem whose inputs are pixels and output is text.



Visual Encoding

Providing an effective encoding of the visual content is the first task within image captioning pipeline.

- Non-attentive methods based on global CNN features
- Additive attention methods
  - Grid based
  - Region based
- Graph-based methods
- Self-attentive methods employing transformer based paradigms.
  - Region based
  - Patch based
  - Image-text early fusion

Global CNN features

With advent of CNNs, all models consuming visual inputs have been improved in terms of performance

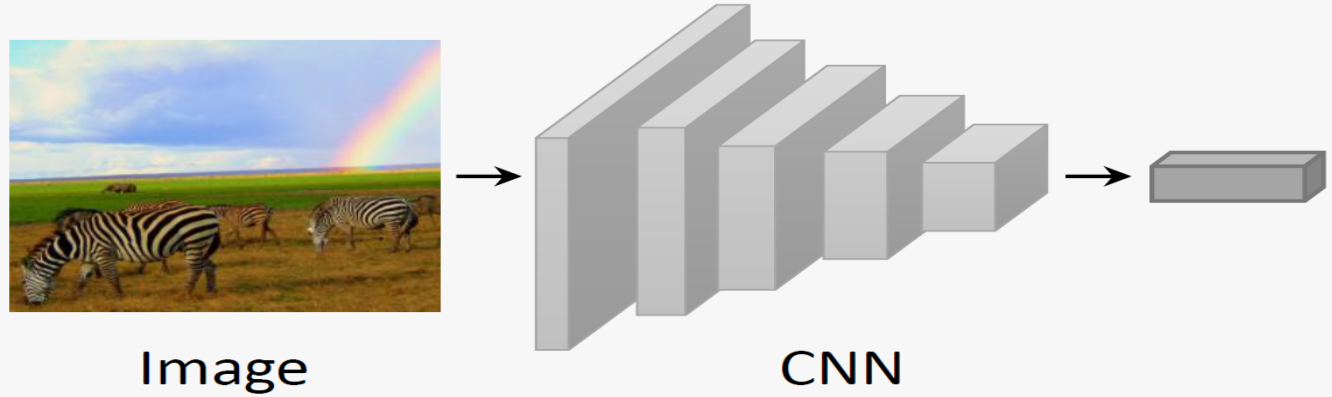
In the most simple recipe, the activation of one of the last layers of a CNN is employed to extract high-level representations, which are then used within language models for generating final text.

Advantages

- Simplicity
- Compactness of representations

Disadvantages

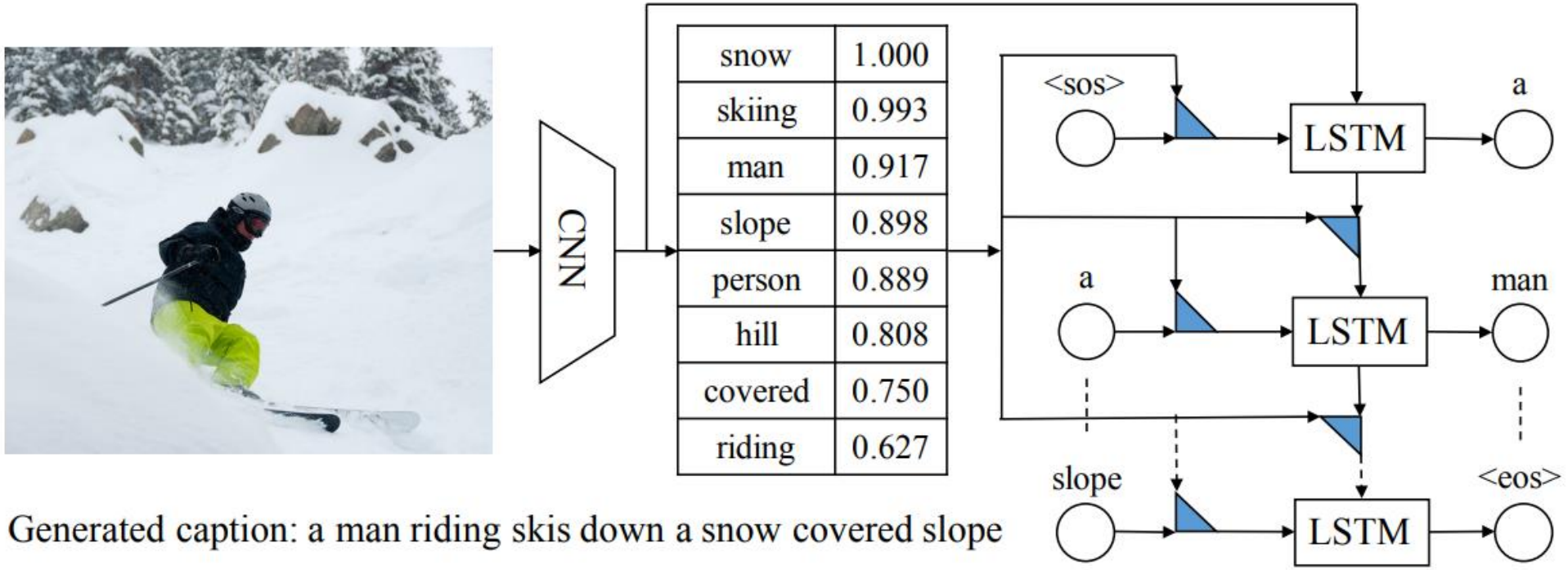
- Excessive compression of information
- Lack of granularity
- Unable to produce specific and fine-grained descriptions



Global CNN features

Gan et al., CVPR 2017

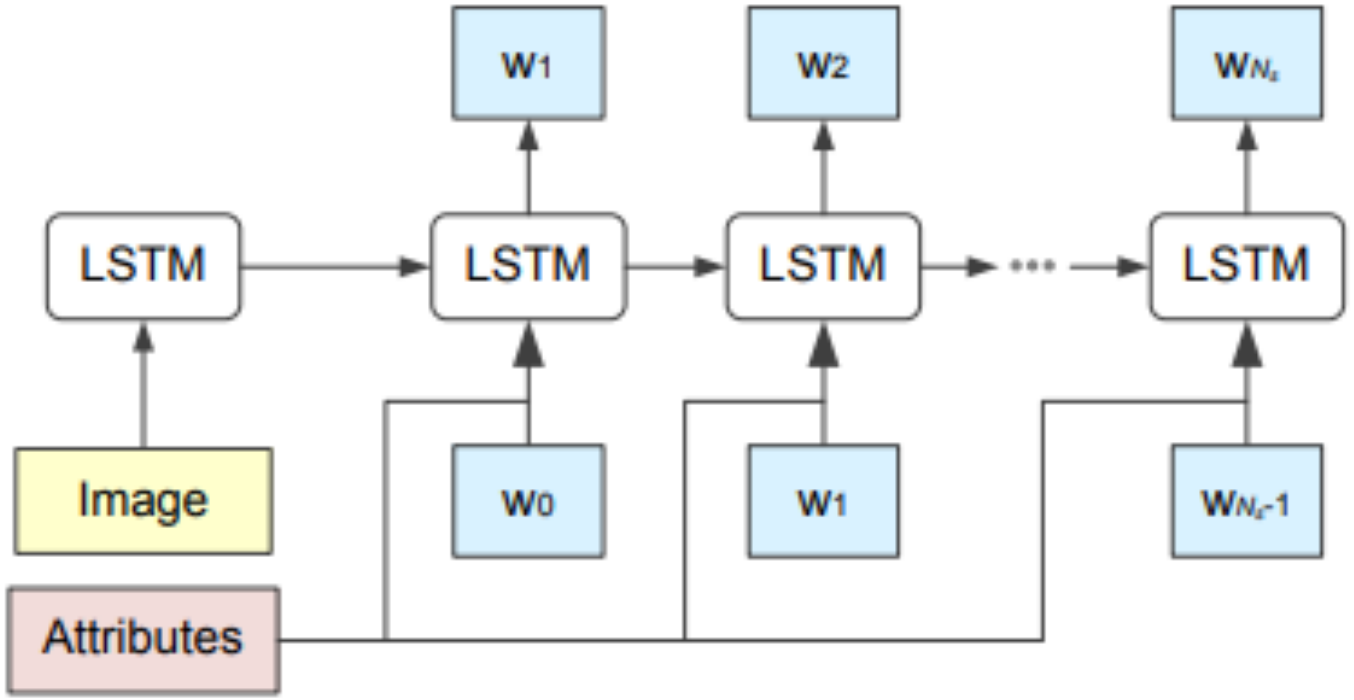
- ResNet-152 pre-trained on imagenet dataset



Global CNN features

Yao et al., Boosting image captioning with attributes, ICCV 2017

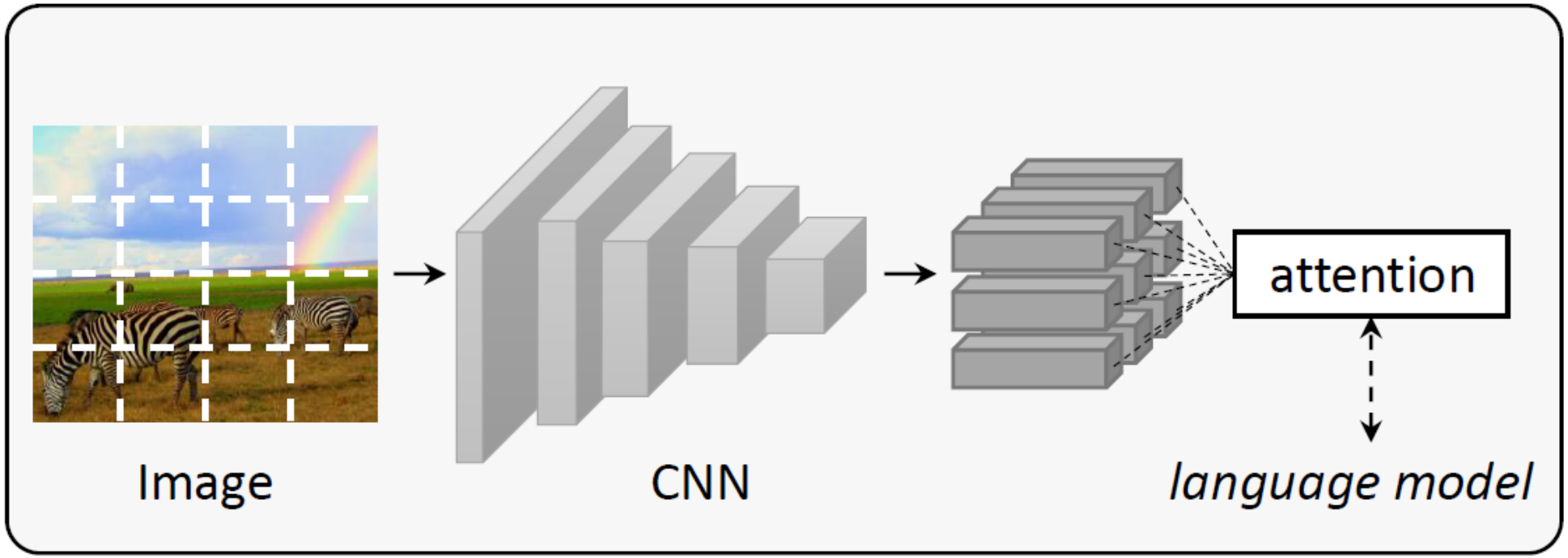
- 1,000 most common words on COCO as the high-level attributes and train the attribute detectors with MIL model (Fang et al., 2015)
- GoogleNet for image encoding



Attention based models

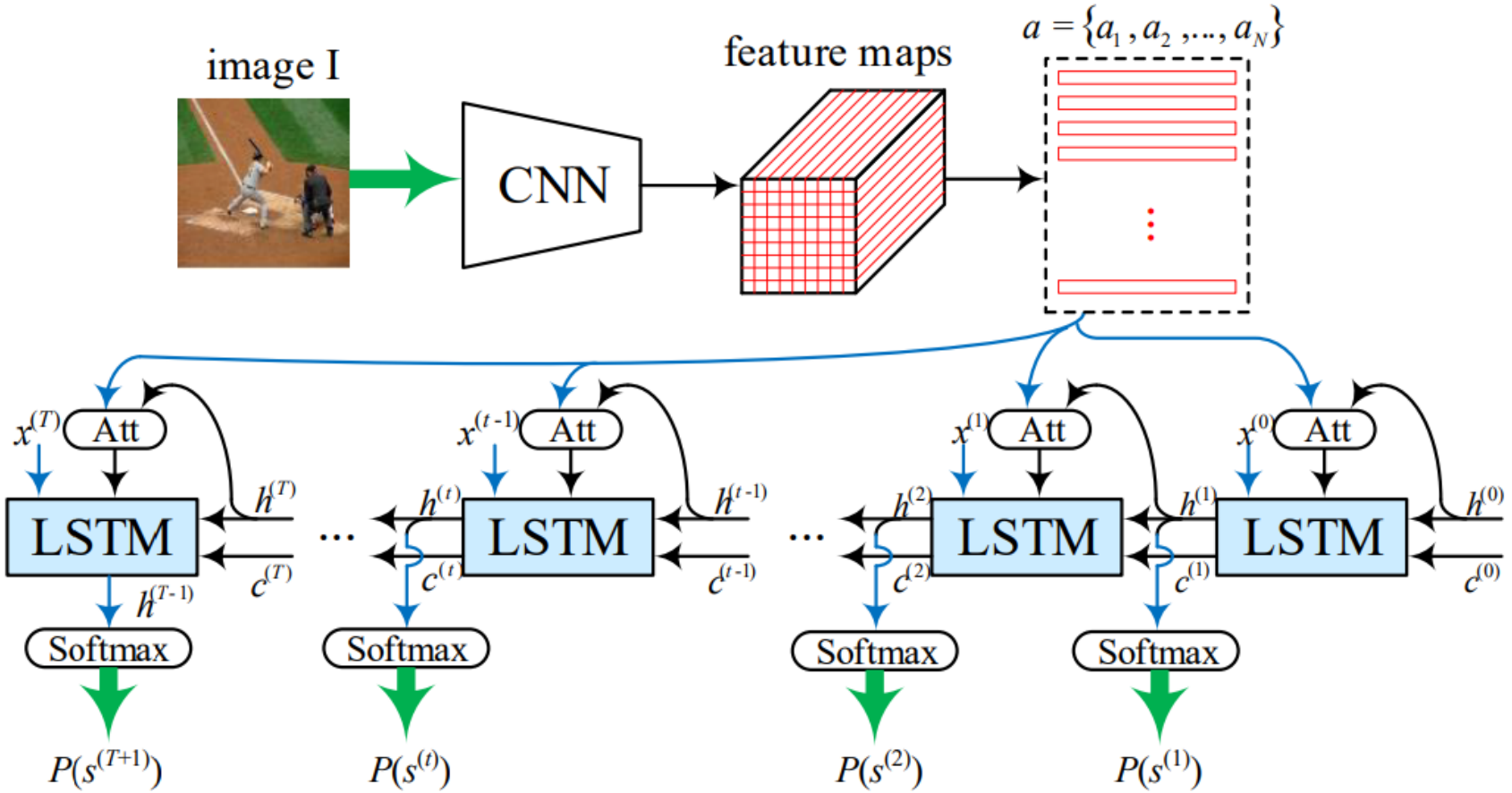
Grid based attention

- Improves upon the drawbacks of global representations and increases the granularity level of visual encoding
- Motivated from use of attention in machine translation.



Attention based models

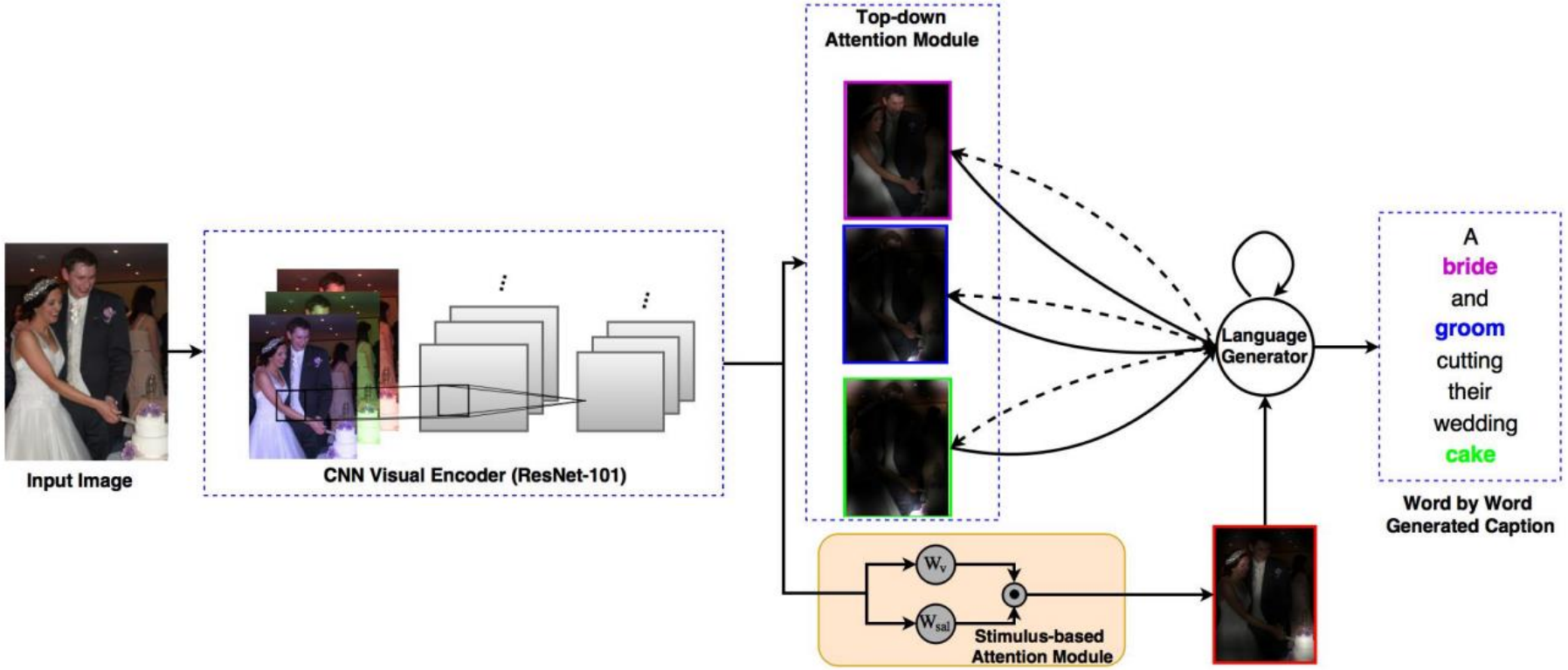
Ge et al., Exploring Overall Contextual Information for Image Captioning in Human-Like Cognitive Style, ICCV 2019



Attention based models

Chen et al., Boosted attention: leveraging human attention for image captioning, ECCV, 2018

- Improve the attention mechanism by incorporating human attention input into the model.



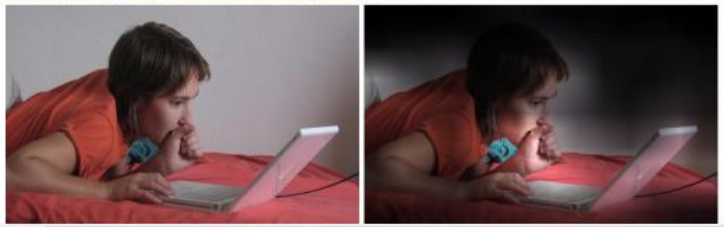
Attention based models

a parking meter on a street with palm trees.
a street sign on the side of the road.
1. A close up of a crosswalk sign in the middle of the road.
2. A tatter street sign sits in the crosswalk.
3. The yield to pedestrians sign is all scratched up.

a green and yellow bus parked next to a street sign.
a sign is on the side of a road
1. A green sign says Thruway one fourth mile.
2. A road sign stands next to the road.
3. a street sign below a bunch of power lines

a plate of food that is sitting on a table.
a bird sitting on top of a plate of food.
1. A plate topped with bread, greens and pasta and a bird.
2. there are two birds standing on the plate of food.
3. A bird attempting to bite a piece of sandwich bread.

a man is standing next to a motorcycle.
a man riding a motorcycle with a mountain in the background.
1. A man in a red shirt and a red hat is on a motorcycle on a hill side.
2. A man riding on the back of a motorcycle.
3. Man riding a motor bike on a dirt road on the countryside.

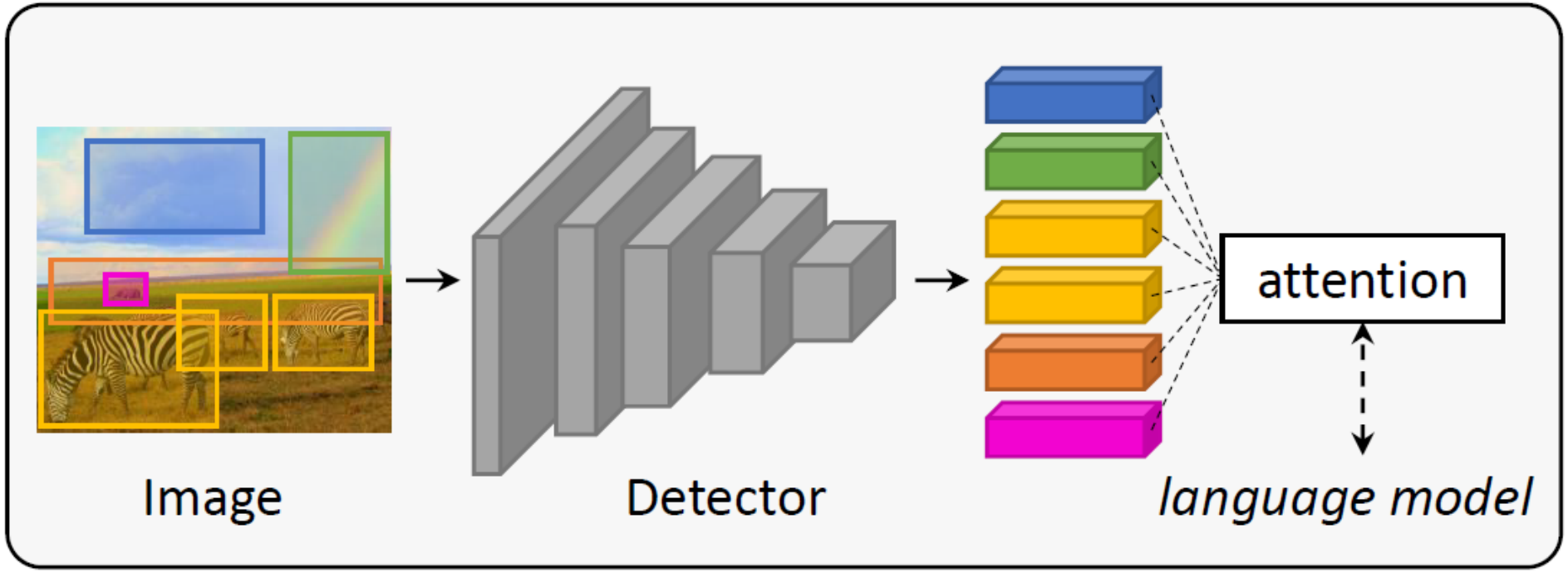


a woman sitting on a bed with a red shirt.
a woman sitting on a bed with a laptop.
1. there is a woman laying in a bed using a lap top.
2. A girl on a bed studying something on her laptop.
3. a woman using a white laptop on the bed.

Attention based models

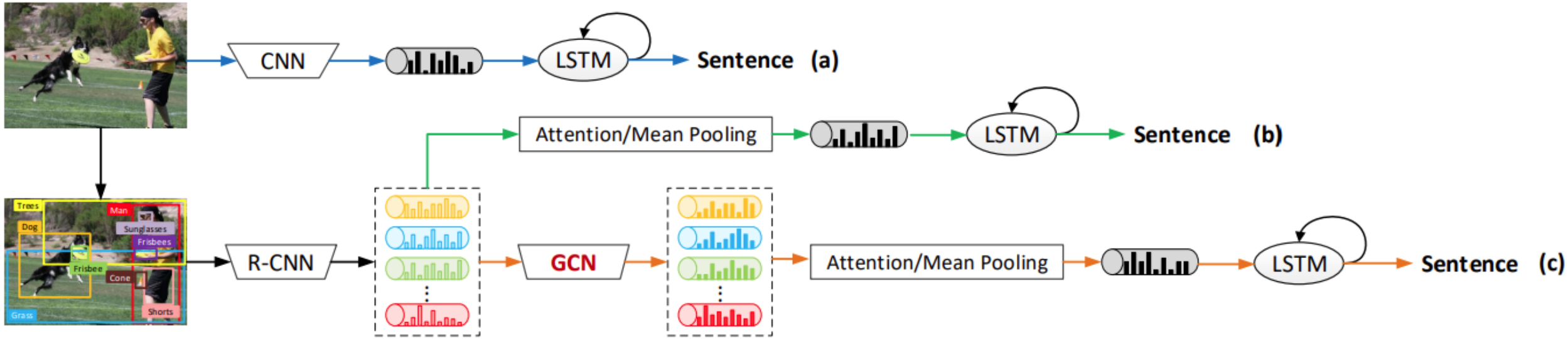
Region based attention

- A natural division of image is not in terms of grid but rather into regions.
- This is evident from saliency maps obtained for human vision.
- Region based attention models divide the image into logical regions rather than grids and calculate attention over these regions.



Graph based models

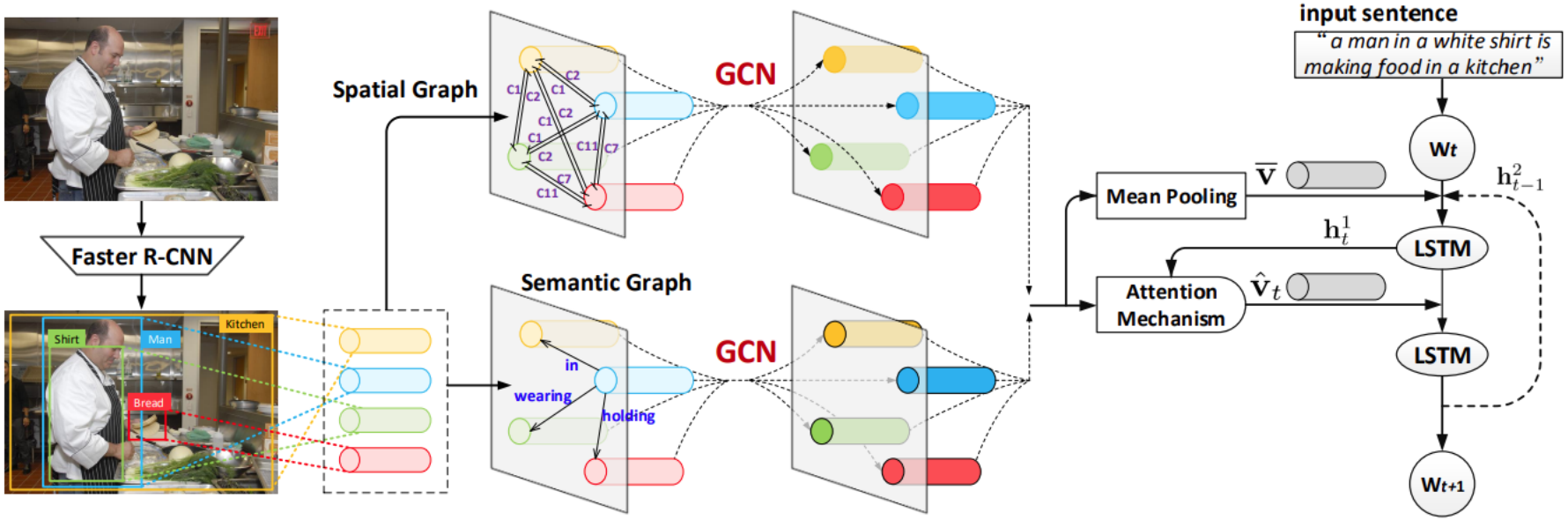
- Region based attention treats all regions equally without taking into account the interactions between them.
- Some studies consider using graphs based models for improved encoding of image regions by incorporating relations between different regions.



Graph based models

Spatial and semantic graphs

- A semantic relation classifier is trained on a large corpus and directly used for semantic graph generation.
- Spatial graphs are built and assigned depending on their Intersection over Union (IoU), relative distance and angle.



Yao T, Pan Y, Li Y, Mei T. Exploring visual relationship for image captioning. In Proceedings of the European conference on computer vision (ECCV) 2018 (pp. 684-699).

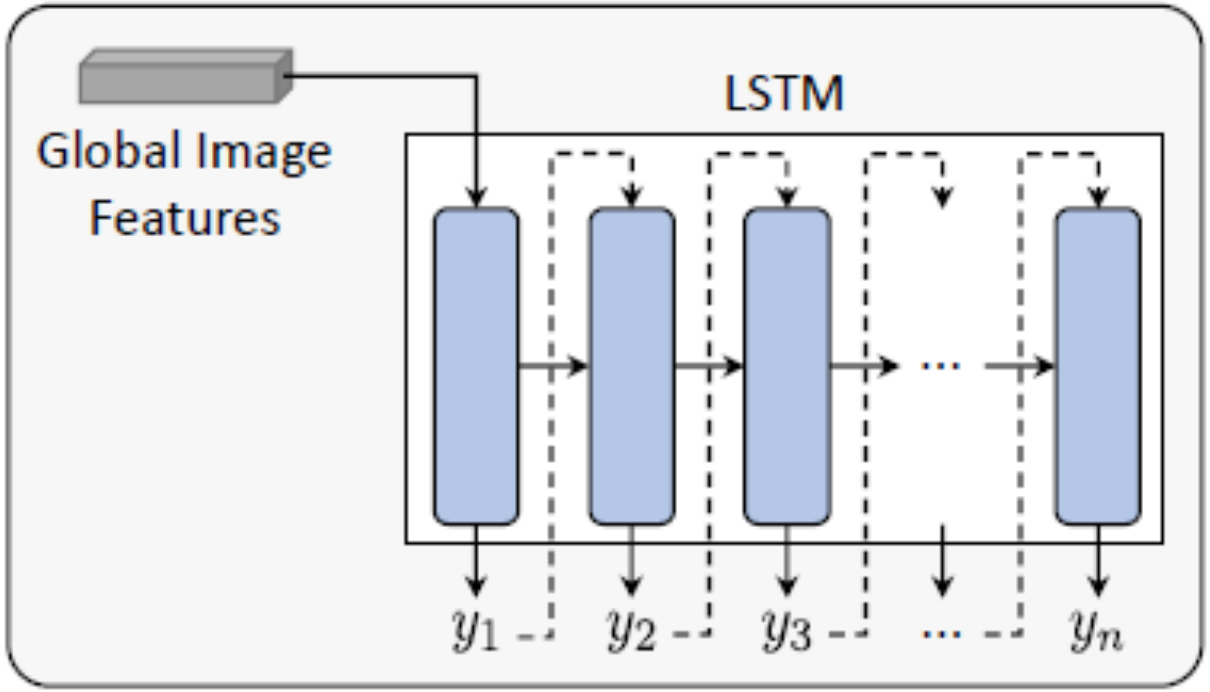
Language models

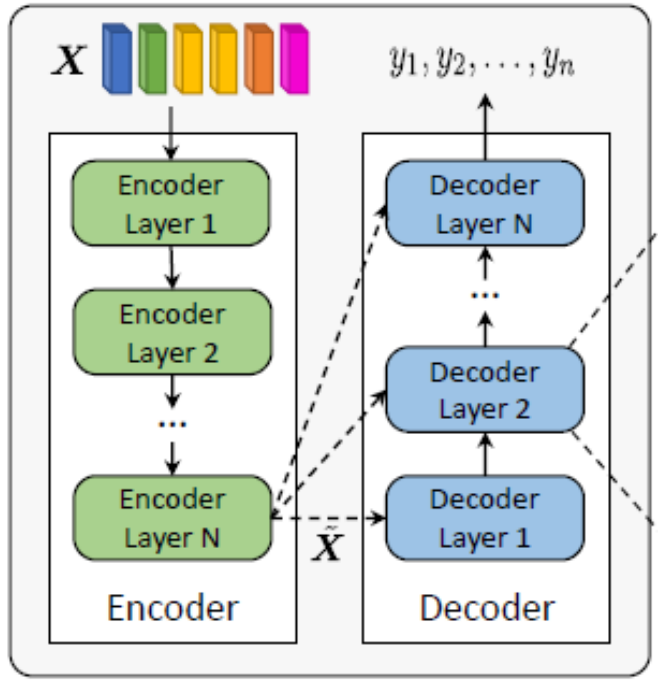
- Given the learned representation of the image, language models generate textual that is meaningful and syntactically correct.
- The goal of a language model is to predict the probability of a given sequence of words to occur in a sentence.
- The model makes incremental predictions where probability of i^{th} word in the sequence is conditioned on the preceding sequence.

$$P\left(y_1, y_2, \ldots y_n \mid \boldsymbol{X}\right) = \prod_{i=1}^{n} P\left(y_i \mid y_1, y_2, \ldots, y_{i-1}, \boldsymbol{X}\right)$$

Language models

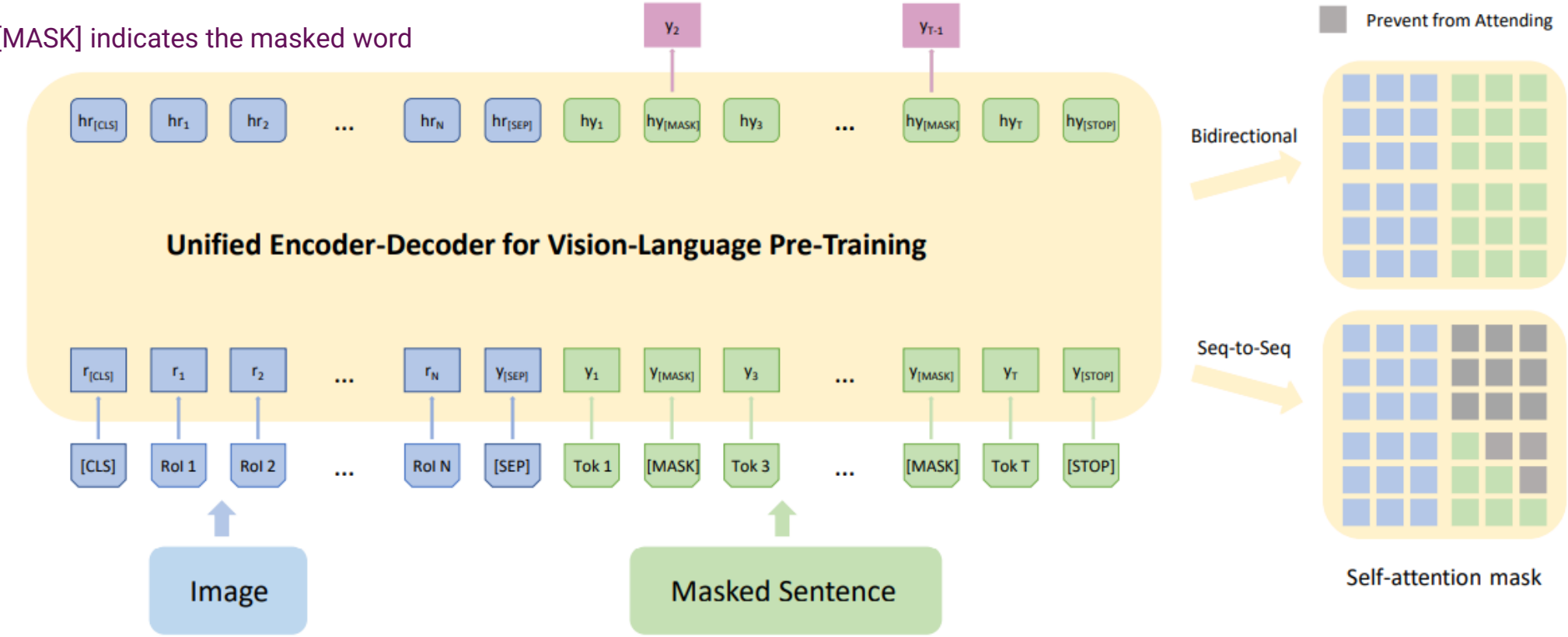
- The main language modeling strategies applied to image captioning are:
  1. LSTM based
  2. Transformer based fully attentive approaches
  3. Image-text early fusion (BERT-like)





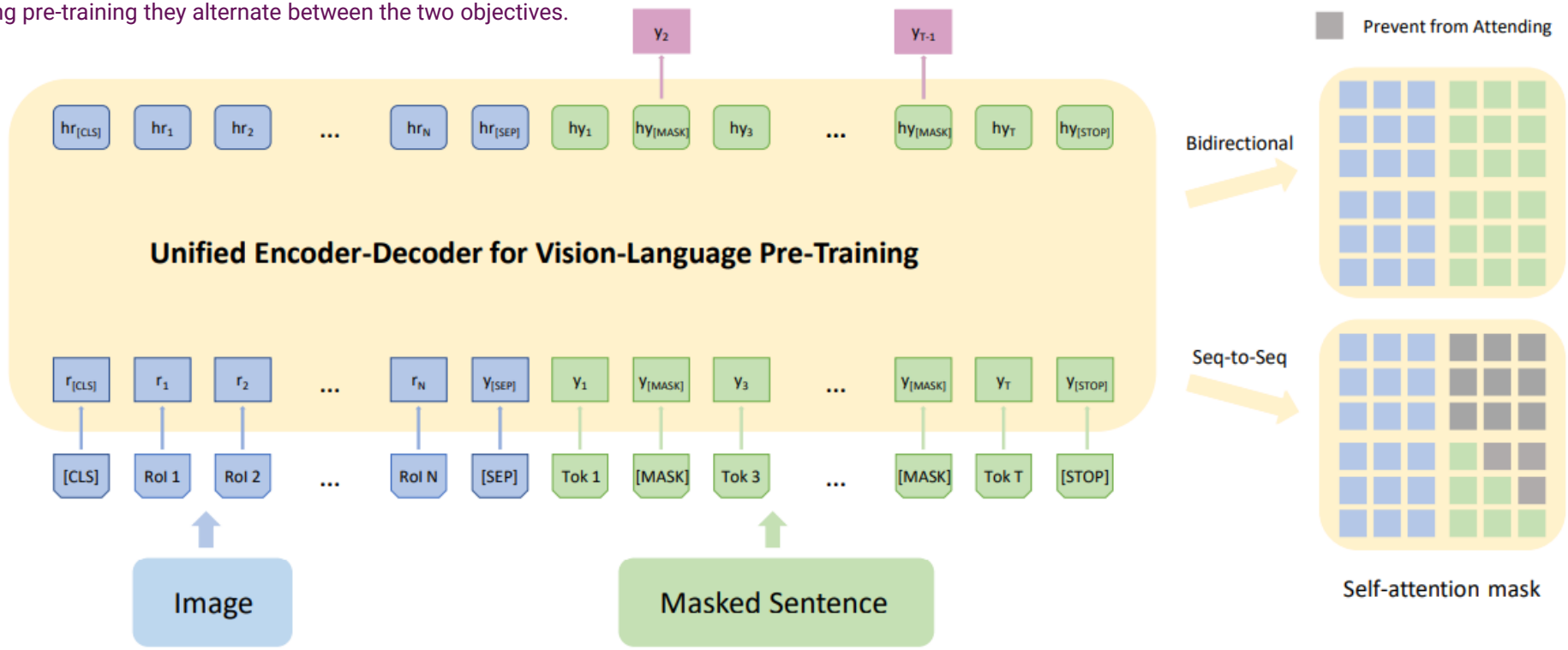
Vision-Language Transformer Model

- Unifies the transformer encoder and decoder into a single model.
- Model input consists of class-aware region embeddings, word embeddings and three special tokens [CLS], [SEP] and [STOP].
- [CLS] indicates the start of the visual input.
- [SEP] indicates the seperation betzeen visual and text input.
- [STOP] indicates end of sentence.
- [MASK] indicates the masked word



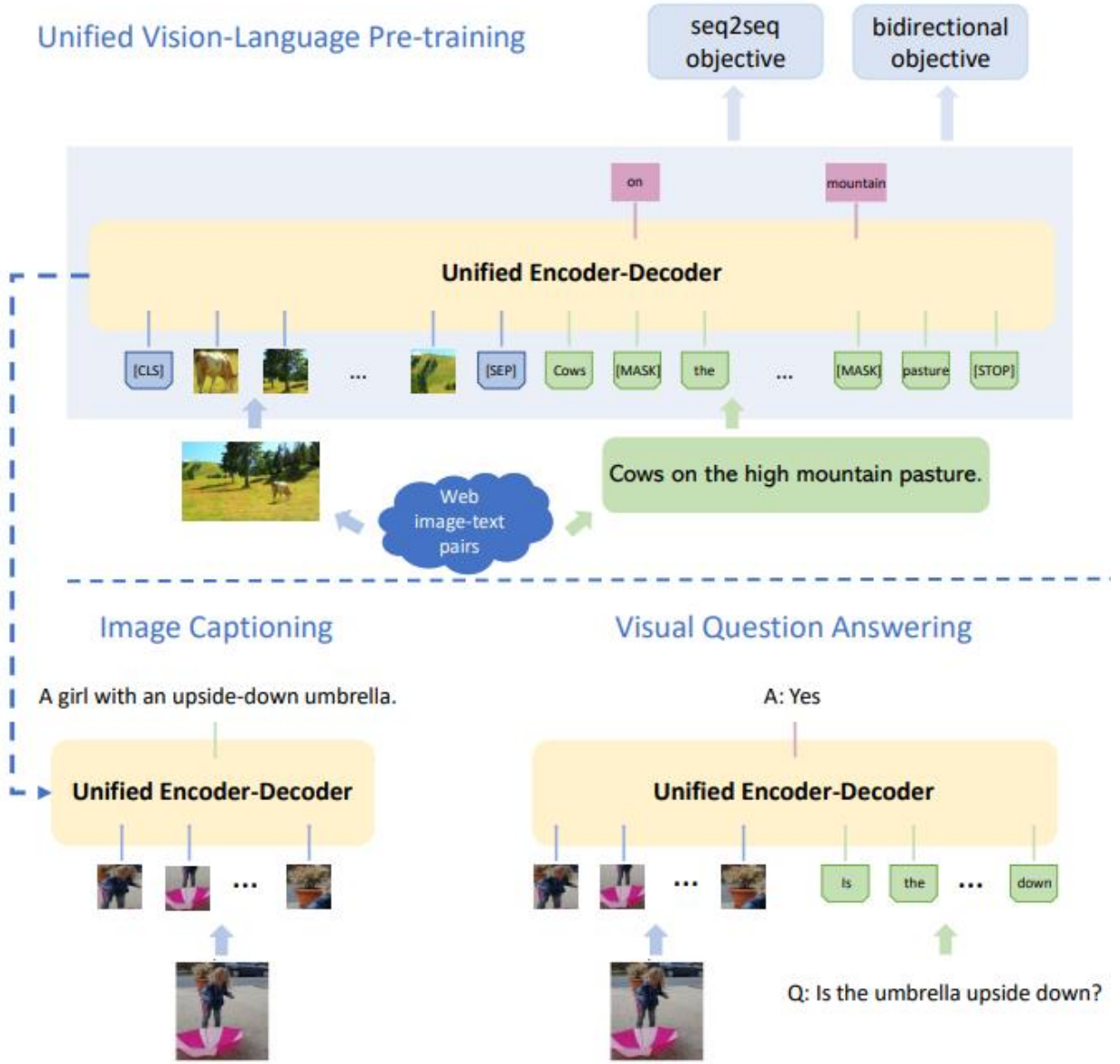
Vision-Language Transformer Model (pre-training)

- 15% of the text tokens are replaced by [MASK] token, random token or original token.
- The hidden state from the last transformer block is projected to word likelihoods where the masked token is predicted as classification problem.
- Through this reconstruction the model learns the dependencies in the context and forms a language model.
- Two objectives are considered within this model:
  - Bidirectional: every token can attend to every other token.
  - Seq-to-seq: tokens cannot attend to future tokens. It satisfies the auto-regressive property.
  - During pre-training they alternate between the two objectives.



VLP model for image-captioning

- For image captioning we fine-tune using seq2seq objective.
- During inference
  - Encode image region along with special tokens ([CLS] and [SEP] tokens).
  - We then start the generation by feeding in the [MASK] token and sampling a work from word likelihood output.
  - Replace the [MASK] token in the input sequence with sampled word and add new [MASK] token to the end of sequence.
  - Generation terminates when [STOP] token is chosen.



Training strategies

Cross-Entropy Loss

- Most used objective for image captioning.
- The aim is to minimize the negative log-likelihood of the current word given the previous ground-truth words.
- The loss works at word level and optimizes the probability of each word without considering long range dependencies.

$$L_{XE}(\theta) = -\sum_{i=1}^{n} \log\left(P\left(y_i \mid y_{1:i-1}, \boldsymbol{X}\right)\right)$$

Training strategies

Masked Language Model

- Idea is to randomly mask a subset of input tokens and train the model to predict these masked tokens based on remaining tokens, both previous and subsequent.
- The model relies more on context making it more robust.
- Training on these models is much slower since they only train in masked tokens and not entire sentence.

Data

	Domain	Nb. Images	Nb. Caps (per Image)	Vocab Size	Nb. Words (per Cap.)
COCO [128]	Generic	132K	5	27K (10K)	10.5
Flickr30K [129]	Generic	31K	5	18K (7K)	12.4
Flickr8K [19]	Generic	8K	5	8K (3K)	10.9
CC3M [130]	Generic	3.3M	1	48K (25K)	10.3
CC12M [131]	Generic	12.4M	1	523K (163K)	20.0
SBU Captions [4]	Generic	1M	1	238K (46K)	12.1
VizWiz [132]	Assistive	70K	5	20K (8K)	13.0
CUB-200 [133]	Birds	12K	10	6K (2K)	15.2
Oxford-102 [133]	Flowers	8K	10	5K (2K)	14.1
Fashion Cap. [134]	Fashion	130K	1	17K (16K)	21.0
BreakingNews [135]	News	115K	1	85K (10K)	28.1
GoodNews [136]	News	466K	1	192K (54K)	18.2
TextCaps [137]	OCR	28K	5/6	44K (13K)	12.4
Loc. Narratives [138]	Generic	849K	1/5	16K (7K)	41.8

Evaluation

- Evaluating quality of generated text is a tricky and subjective task.
- Image captions are further complicated since the caption cannot only be grammatical and fluent but also needs to properly refer to the image.
- The best way to evaluate this is still human evaluations
  - Costly
  - Not reproducible
- Automatic methods compare generated captions against human-produced references and are usually defined for other NLP tasks.

Evaluation

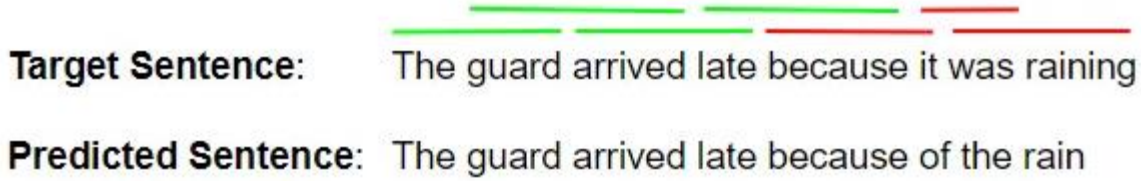
BLEU score

- Target sentence: The guard arrived late because it was raining
- Predicted sentence: The guard arrived late because of the rain

We first calculate precision scores for 1-gram through 4-grams.

$P_1 = 5/8$

$P_3 = 3/6$



$P_2 = 4/7$

$P_4 = 2/5$

Evaluation

BLEU score

- Brevity penalty: it penalizes sentences that are too short.

If the predicted sentence is just « the », the 1-gram precision is 1/1=1, indicating perfect score.

$$Brevity\ Penalty = \begin{cases} 1, & \text{if } c > r \\ e^{(1-r/c)}, & \text{if } c <= r \end{cases}$$

c = predicted sentence lenght

r = target sentence lenght

$$Bleu\ (N) = Brevity\ Penalty \cdot Geometric\ Average\ Precision\ Scores\ (N)$$

$$Geometric\ Average\ Precision\ (N) = (p_1)^{\frac{1}{4}} \cdot (p_2)^{\frac{1}{4}} \cdot (p_3)^{\frac{1}{4}} \cdot (p_4)^{\frac{1}{4}}$$

Evaluation

ROUGE

- Compared to BLEU score that focuses on precision, ROUGE focuses on recall.

$$\text{ROUGE-N} = \frac{\sum_{S \in \{ReferemceSummaries\}} \sum_{gram_n \in S} Count_{match}(gram_n)}{\sum_{S \in \{ReferenceSummaries\}} \sum_{gram_n \in S} Count(gram_n)}$$

TRAITEMENT AUTOMATIQUE DES LANGUES AVANCE

Master Informatique

2ème Année - 1er Semestre

Intervenants CM:

Gaël DIAS (gael.dias@unicaen.fr), Marc SPANIOL, Fabrice MAUREL

Navneet AGARWAL, Kirill MILINTSEVICH

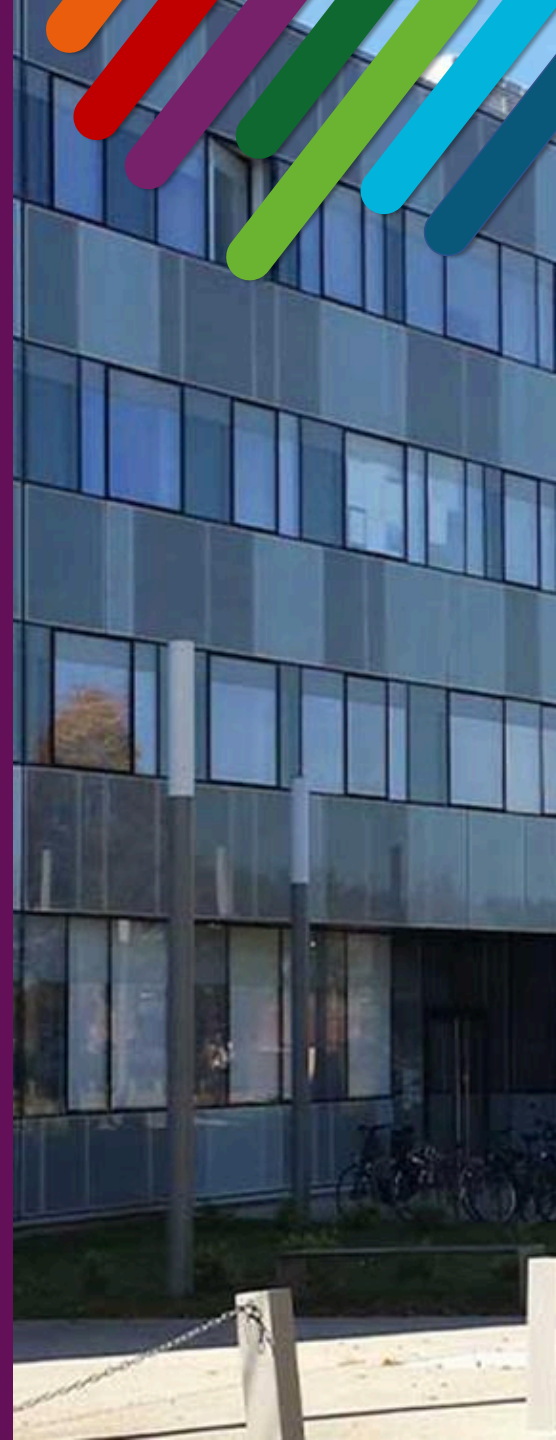
GREYC Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN ÉCOLE PUBLIQUE D'INGÉNIEURS CENTRE DE RECHERCHE

cnrs



ARTIFICIAL INTELLIGENCE AND PSYCHIATRY: A FOCUS ON DIAGNOSTIC AUTOMATION

Some preliminary work experiences

Gaël DIAS @ Sorbonne

joint work with
Navneet AGARWAL, Mohammed HASANUZZAMAN, Arbaaz QURESHI, Kirill MILINTSEVICH, Valentin RENIER, Soumaya SABRY, Sriparna SAHA, Kairit SIRTS, and more to come ;)

University of Caen Normandie - CNRS - GREYC UMR 6072 Research Laboratory
gael.dias@unicaen.fr

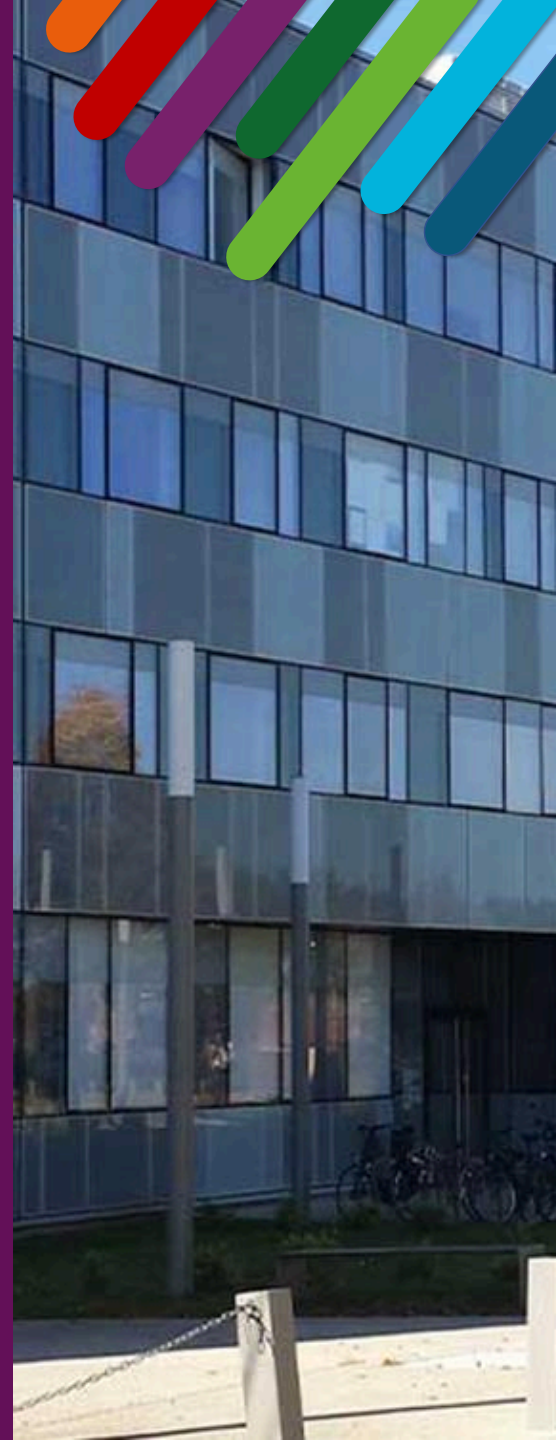
GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

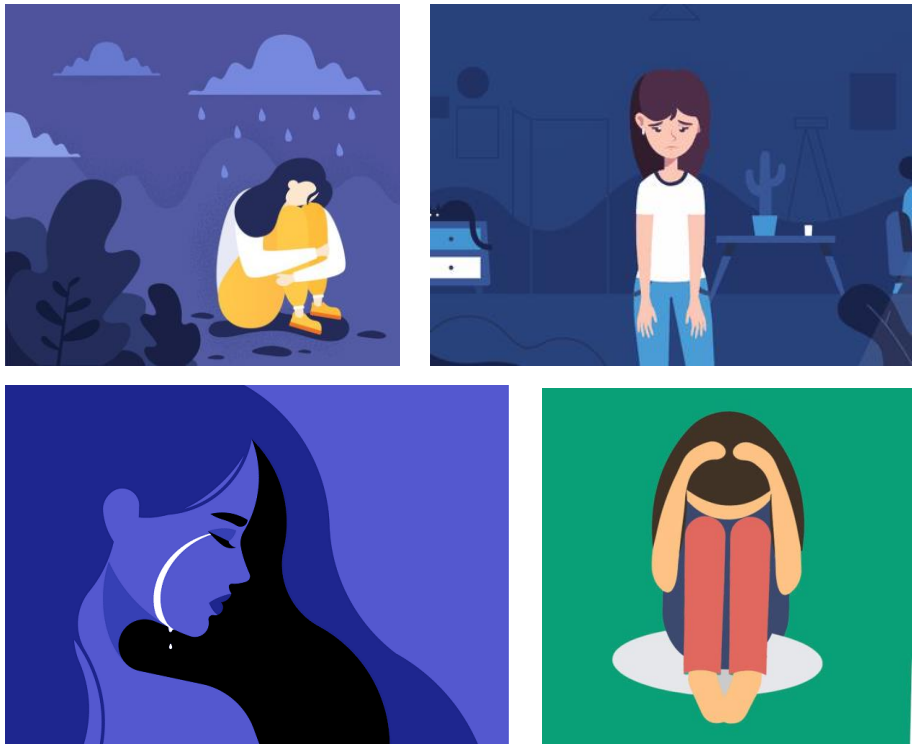
ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs



Outline

1. **Mental Health and Depression**
2. **6P Medicine**
3. **Interesting Iniatives**
4. **Computer-aided Diagnosis**
   i. *Multimodality*
   ii. *Emotionality*
   iii. *Gender-awareness*
   iv. *Dialogue structure*
   v. *Symptom-based diagnosis*
5. **A Favourable Research Environment**



Mental Health

- The world is experiencing a mental health crisis.
- It is estimated that 970 million people worldwide had a mental or substance use disorder in 2017, of which 284 million showed anxiety disorders and 264 million suffered from depression, mostly affecting females (source Forbes).
- What's next after/during the COVID-19? Unemployment, divorce, etc.
- The critical shortfall of psychiatrists and other mental health specialists to provide treatment exacerbates this crisis. In the Ain region in France, the supply of psychiatric care is half the national average, i.e. 9 psychiatrists for 100,000 inhabitants (source Le Progrès). This shortage of doctors results in less frequent appointments and practitioners who no longer take new patients.
- This crisis is even more exacerbated in France, where Psychiatry has been defined as the "parent pauvre de la médecine" (i.e. the poor relative of medicine) by the French Ministry Agnès Buzyn in 2018. In particular, she stated that "*Psychiatry is a discipline of the future, but the organization of mental health care and its place in the society are not up to the task […]. Prevention is insufficient, and diagnosis too late […]. I make it a health priority*". (Source Science et Avenir).

Mental Disorders

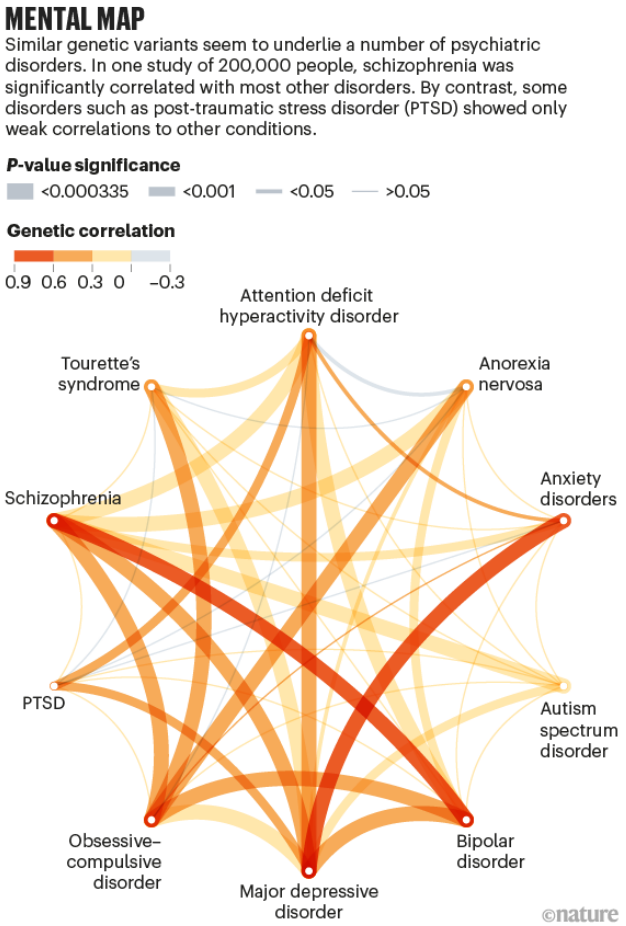
- Anxiety disorders,
- Bipolar and related disorders,
- Depressive disorders,
- Disruptive, Impulse-control, and Conduct disorders,
- Dissociative disorders,
- Feeding and eating disorders,
- Gender dysphoria,
- Obsessive-compulsive and related disorders,
- Personality disorders,
- Trauma and stressor-related disorders,
- Schizophrenia spectrum and other psychotic disorders,
- etc.

Many Different Symptoms

- Apathy,
- Avoidance,
- Excessive fear or uneasiness,
- Feeling of disconnection,
- Increased sensitivity,
- Mood changes,
- Problems thinking,
- Significant tiredness,
- Sleep or appetite changes,
- Withdrawal,
- etc.

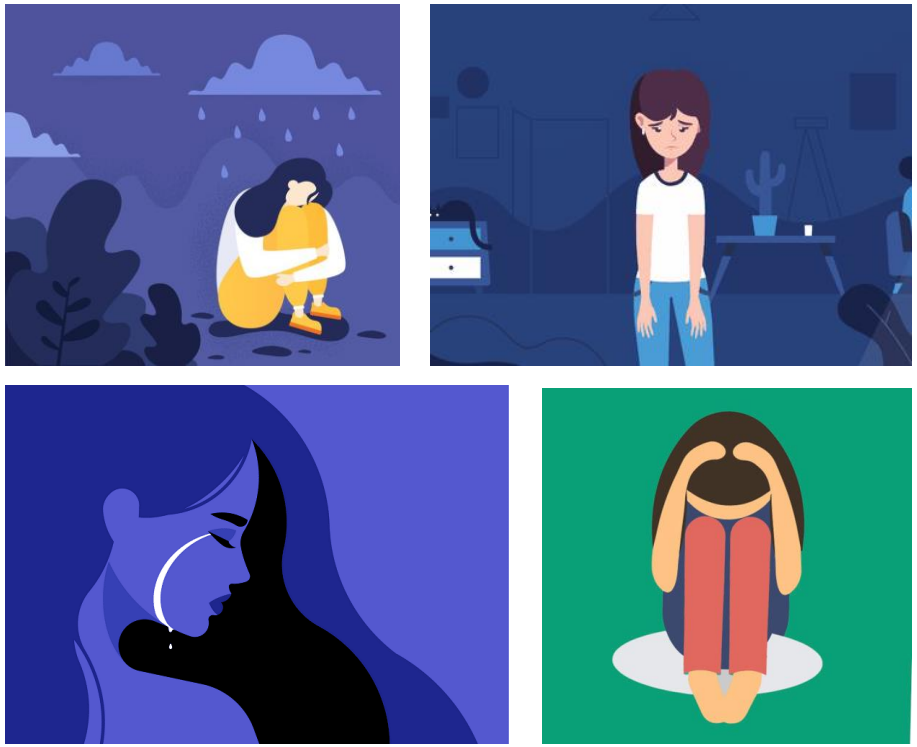
Depression

- Depression is characterized by chronic low mood, low self-esteem and loss of interest.
- Depression is a disabling condition that can impact family, school or work. In the most severe cases, depression is characterized by a high suicide rate.
- The causes of depression are multiple and not well understood: e.g. genetic predisposition, traumatic experiences, inability to cope with rejection or failure.
- The diagnosis of depression is based on the patient's personal feelings, the behavior perceived by those around and the results of psychological examination.
- The diagnosis of depression is complex due to:
  - the high rate of comorbidity,
  - the subjectivity of the examinations,
  - the non-regular therapeutic follow-up,
  - the patient coverage of symptoms.



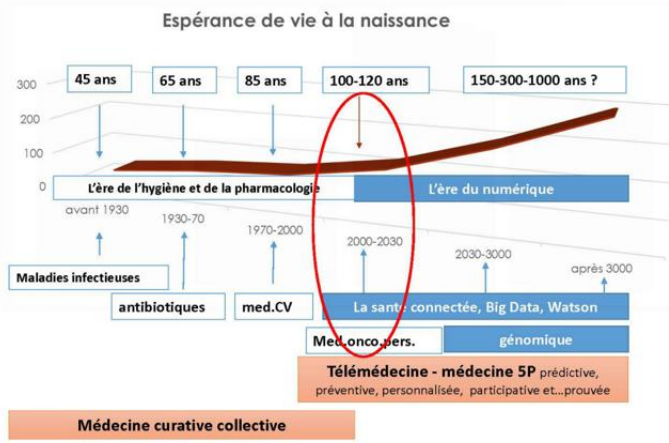
Outline

1. *Mental Health and Depression*
2. **6P Medicine**
3. **Interesting Iniatives**
4. **Computer-aided Diagnosis**
   i. *Multimodality*
   ii. *Emotionality*
   iii. *Gender-awareness*
   iv. *Dialogue structure*
   v. *Symptom-based diagnosis*
5. **A Favourable Research Environment**

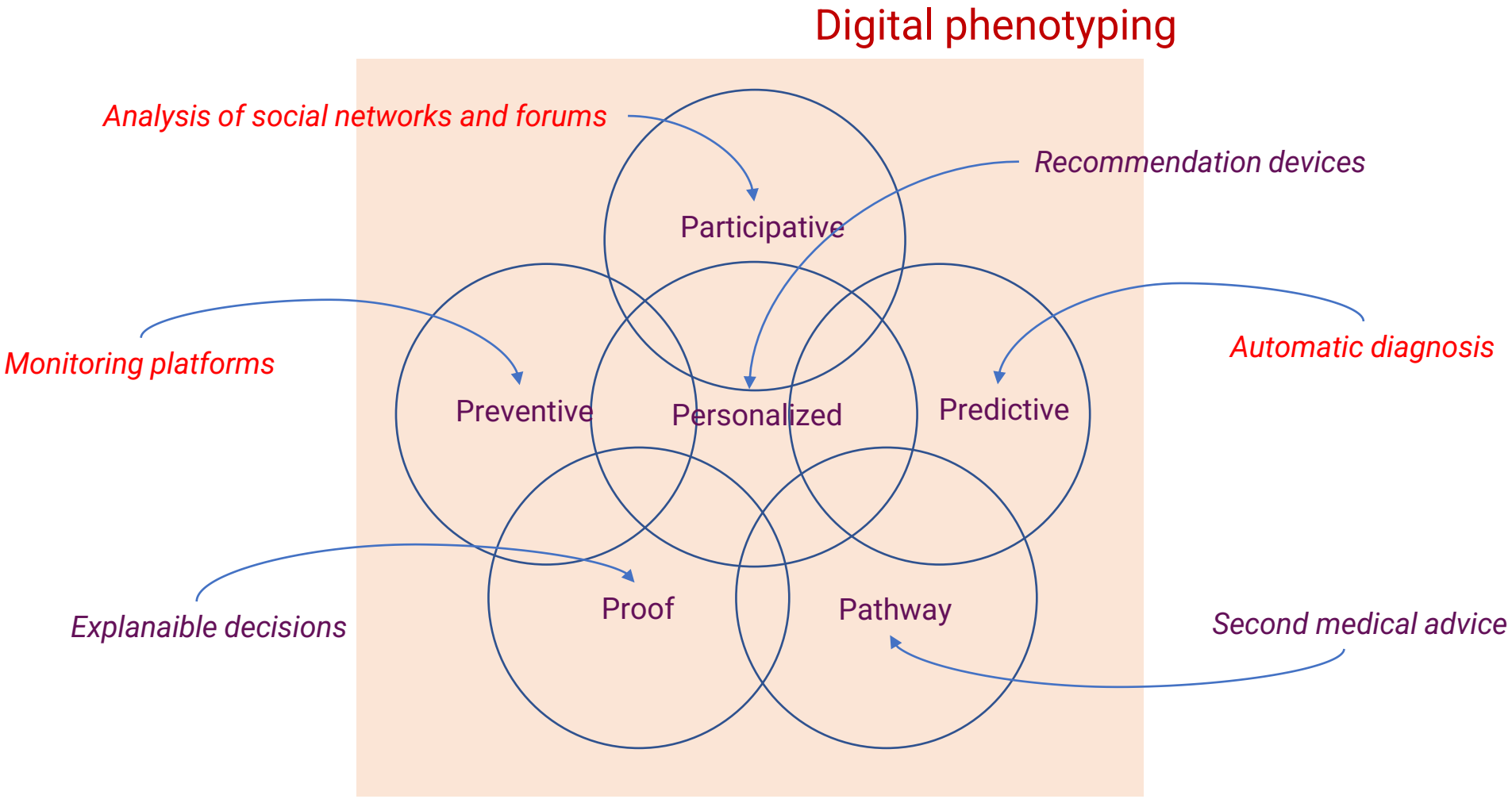


6P Medicine

- 1P - Personalized: Personalized medicine consists of adapting a medical treatment according to the individual characteristics of a patient.
- 2P - Preventive: Preventive medicine focuses on wellness, and consists of measures taken for disease prevention.
- 3P - Predictive: Predictive medicine is a branch of medicine that aims to identify patients at risk of developing a disease.
- 4P - Participative: Medicine should be participatory, leading patients to be more responsible for their health and care.
- 5P - Proof: Medicine must be based on evidence of medical service to patients, especially when it relies on connected health and telemedicine.
- 6P - Pathway: Coordinating multiple interventions (medical, social, occupational medicine, etc.) such that the healthcare pathway is progressively articulated, according to the pathology and its evolution.

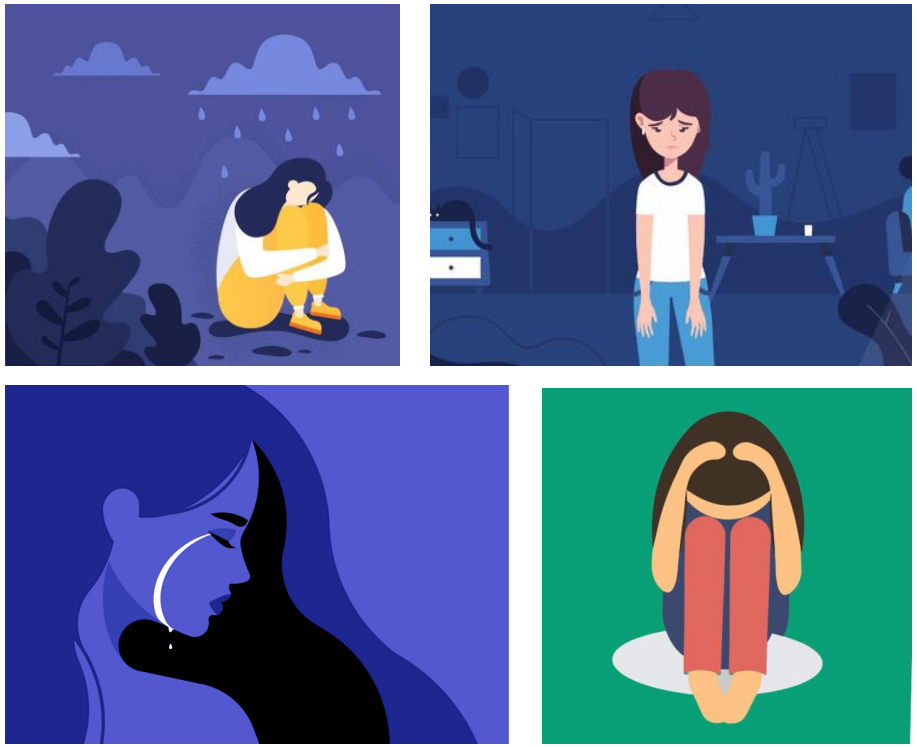


6P Medicine and AI

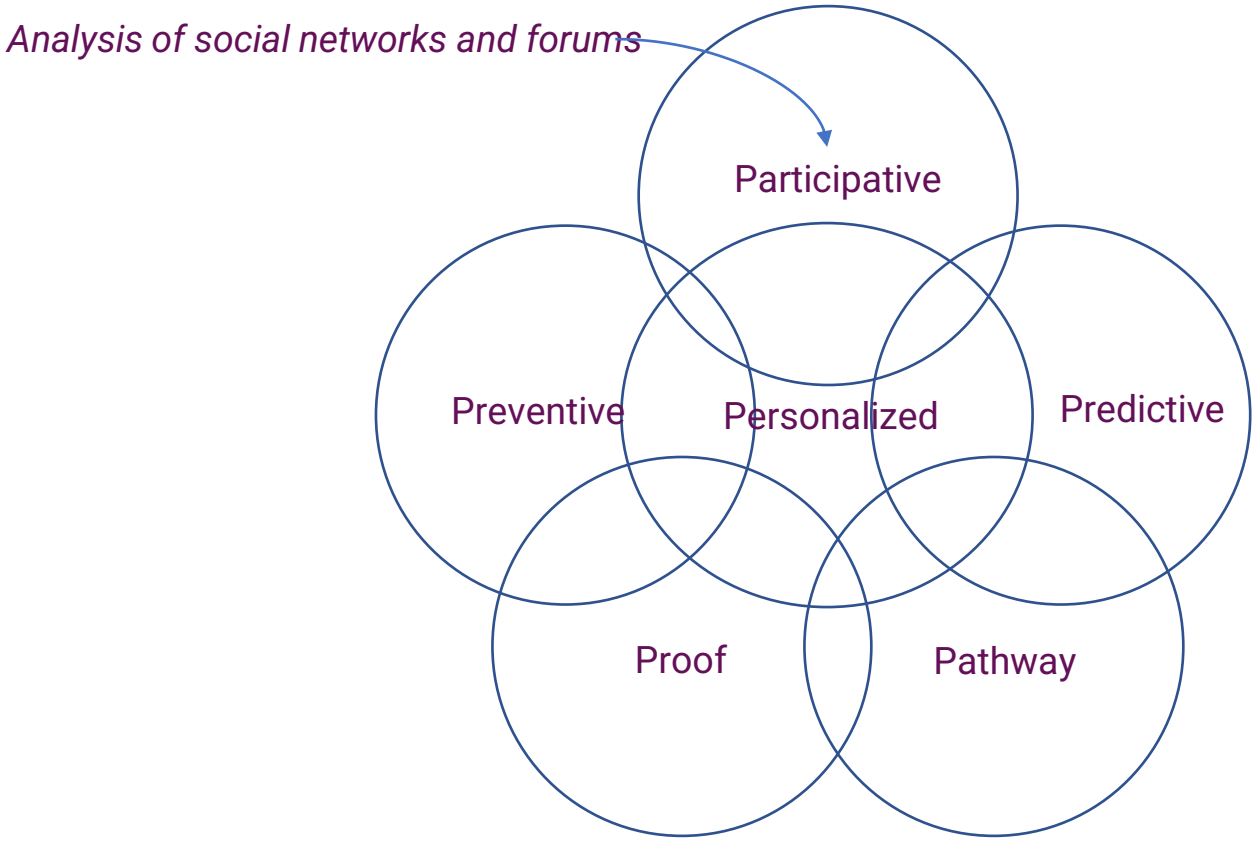


Outline

1. *Mental Health and Depression*
2. *6P Medicine*
3. **Interesting Iniatives**
4. **Computer-aided Diagnosis**
   - *i. Multimodality*
   - *ii. Emotionality*
   - *iii. Gender-awareness*
   - *iv. Dialogue structure*
   - *v. Symptom-based diagnosis*
5. **A Favourable Research Environment**

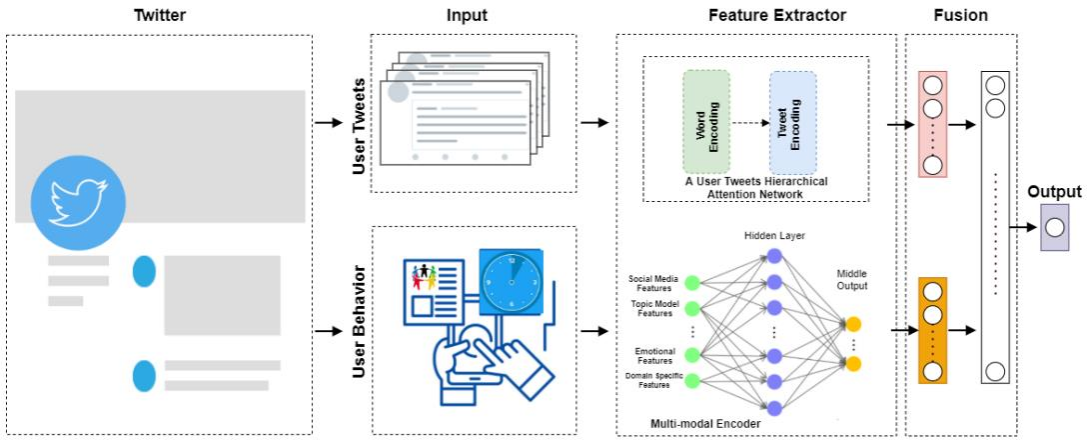


Computer-aided Diagnosis



Social Network Analysis

- Social networks are an important support for Participative medicine, which automatic analysis might allow Preventive/Predictive actions.
- It is common for people who suffer from mental health problems to often disclose their feelings and their daily struggles with mental health issues on social media as a way of relief.
- Twitter, Reddit, Doctissimo, to name but a few platforms have become an excellent resource to automatically discover people who are under depression.
- [Zogan et al., 2021] propose a depression detection framework by tackling textual, behavioral, temporal, and semantic modalities.



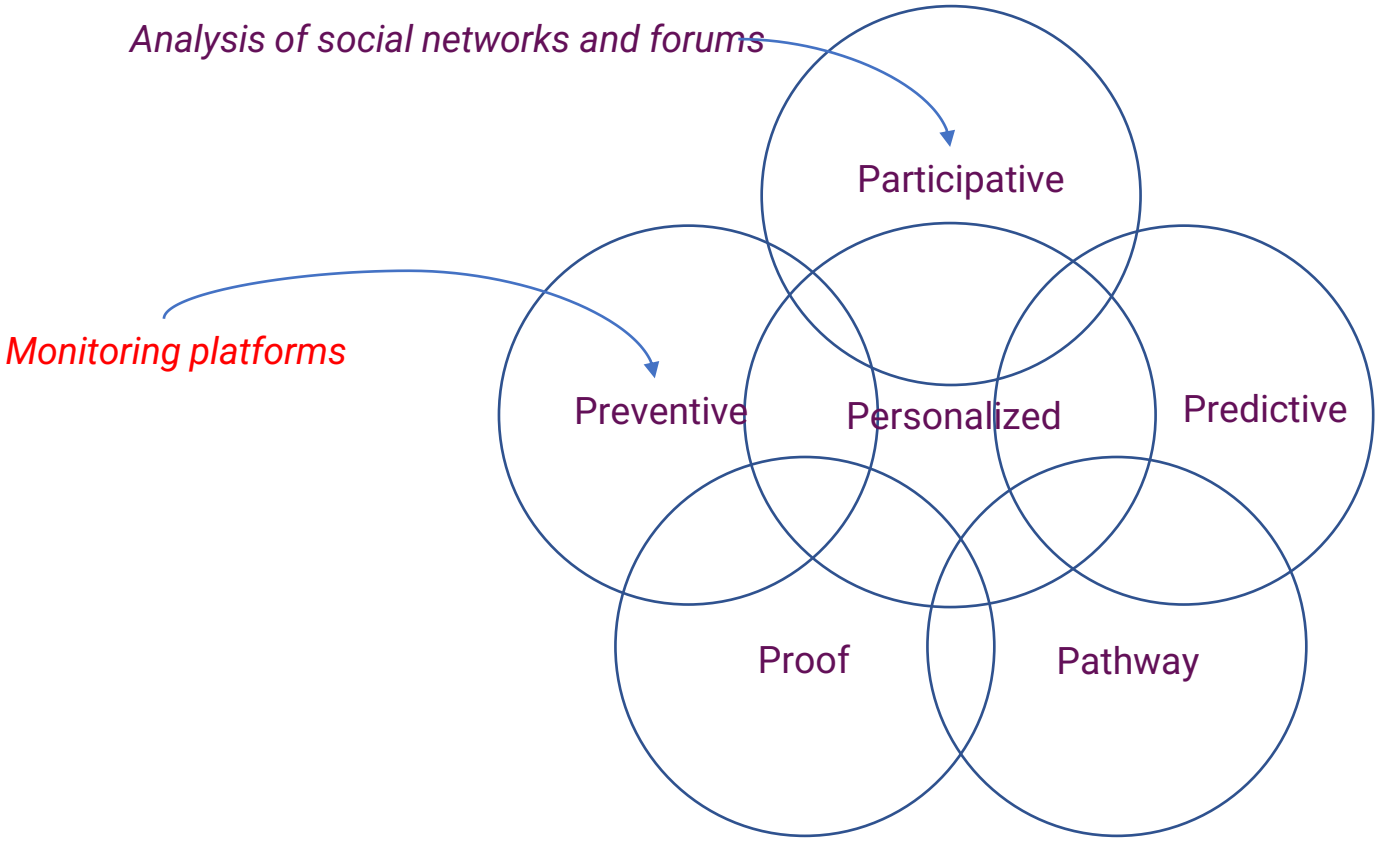
Social Network Analysis

- [Losada & Gamallo, 2018] propose to analyze and improve current language resources for identifying signs of depression on Reddit.
- Other lexicons: Pedesis (2012) obtained from the web, Choudhury (2013) based on Twitter analysis, Schwartz (2014) focused on Facebook posts, etc.
- They propose to expand existing lexicons with selected terms following distributional and paradigmatic-based models, and thesaurus-based models.
- Their Rocchio based experiments show that the resulting lexica are effective at identifying signs of depression in a non-supervised way.

accelerate adsorb affect alleviate anger ask avoid beat bestow blotched bruise cancel capture carry cause cdot characterise characterize clinch collapse colour confront conquer convert convince cry decline defeat define delay denote depopulate derive destroy detect devastate devote diminish disappear disappoint divide elongate emit encircle enclose encourage enlarge erode evaporate evoke evolve exacerbate exclude exercise extract facilitate fade fill finish flank flatten fleck focus foil forward grab grieve halt hamper hawthorn heal hinder hope impede imply impress induce infuse inject innervate invade ionize isolate kill leach metabolize minimize opt orange-red outflank outrage overhang owe oxidise oxidize pacify peasantry penetrate pertain plan postpone pray prepare present prevent protrude ravage react refer relate remove repel repulse reschedule respond revere reward satisfy schedule seedling seep send separate sharpen shock shower slate soothe speckle stop streak strive subdue subjugate submit surprise surround swell taper tell thwart ting transform traverse treat tremble turn urinate vaporize venerate vine vomit wait wane wield win wish worship yearn

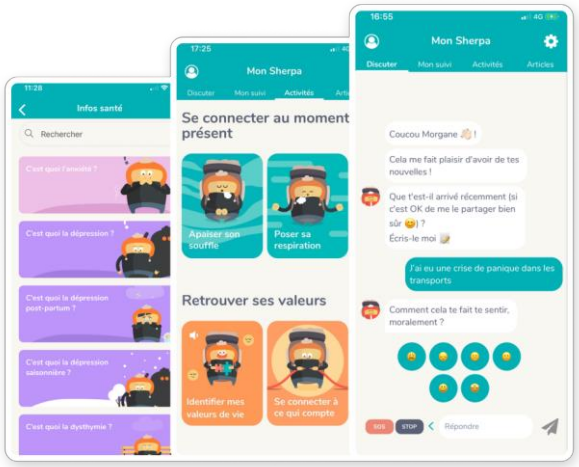
Table 7 New words included in the Pedesis lexicon by the DE expansion method

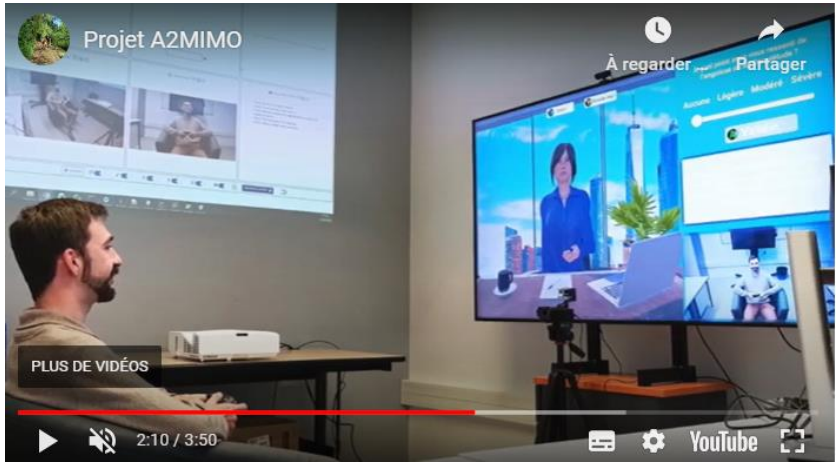
Computer-aided Diagnosis



Monitoring Platforms and (Embodied) Chatbots

- A chatbot is a system that is able to converse and interact with human users using spoken, written, and visual languages (embodied).
- Chatbots can be useful preventive tools for individuals who are reluctant to seek mental health advice due to stigmatization.
- [Abd-alrazaq et al., 2019] studied 41 different embodied and non-embodied chatbots. Most tackle depression and autism.
- Among other scientific issues, therapeutic alliance is the key factor for the success of chatbots and ECAs.

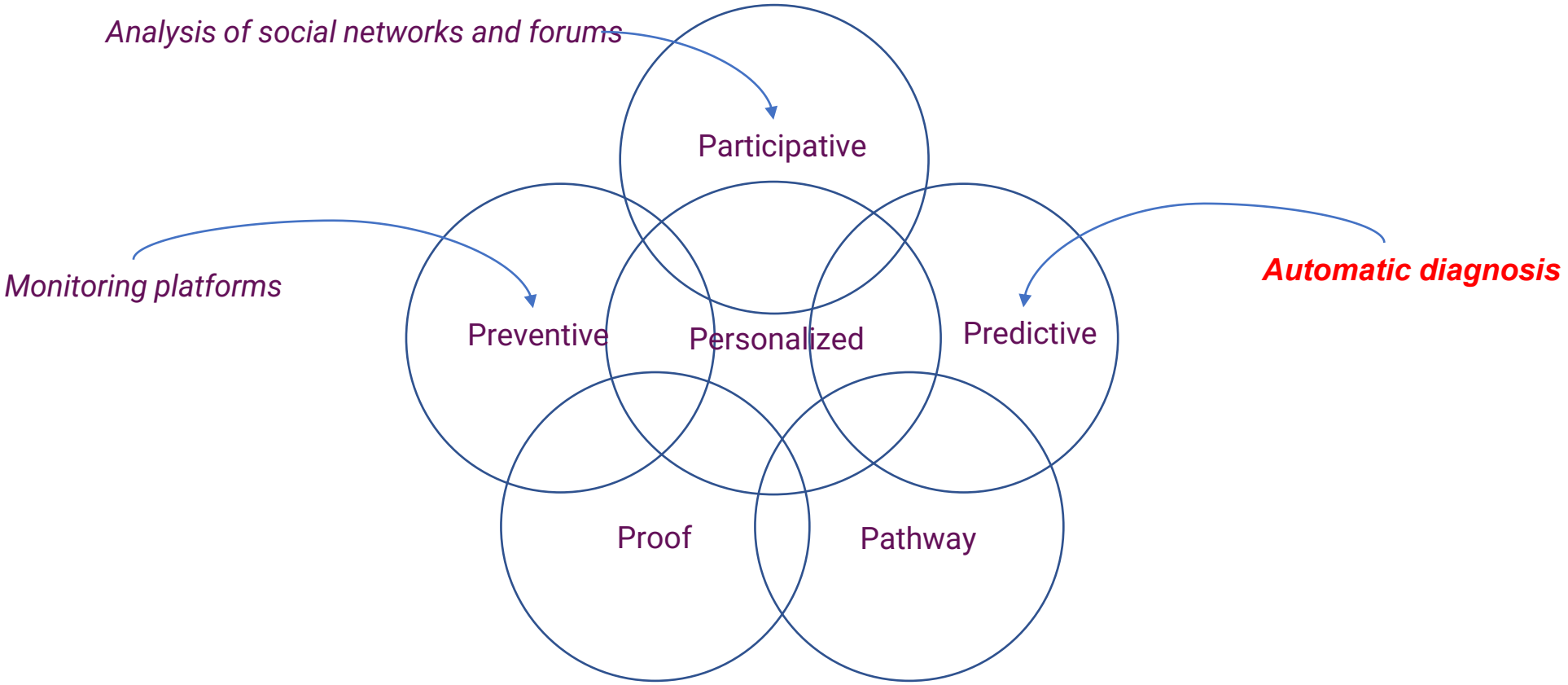




Data Sets and Related Events

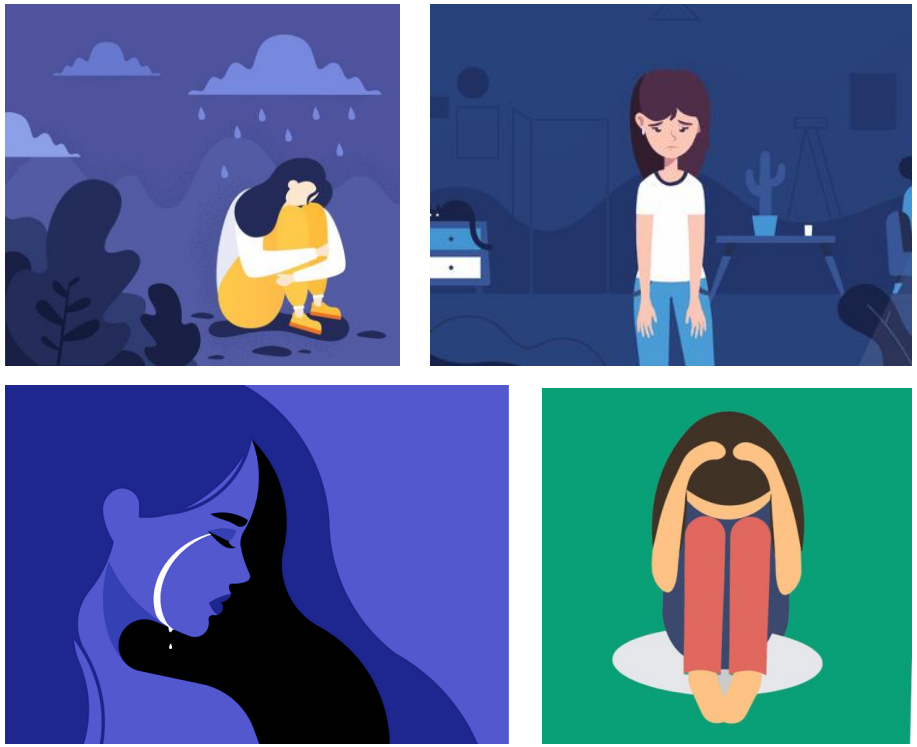
- The major issue with mental health applications is the availability of datasets. Most datasets are not available for reproducibility.
- Some very few exceptions for clinical interviews:
  - DAIC-WOZ [Gratch et al. 2014].
  - General Psychotherapy Corpus [Alexander Street Press?].
  - Audio-visual Depressive Language Corpus [AVEC 2013].
  - Bipolar Disorder Corpus [AVEC 2018] – Turkish language / Bipolarity.
- More exist which are based on social networks :
  - Research on Depression in Social Media [Rissola et al. 2020].
  - Early Detection of Depression [eRisk 2017].
  - CLPsych dataset [Milne et al., 2016] – Risk, Red, Amber, Green / Depression and PTSD.
  - Early Detection of Signs of Anorexia [eRisk 2018].
  - Suicide Watch [Shing et al. 2018].
  - And certainly many others …

Computer-aided Diagnosis



Outline

1. *Mental Health and Depression*
2. *6P Medicine*
3. *Interesting Iniatives*
4. **Computer-aided Diagnosis**
   - *i. Multimodality*
   - *ii. Emotionality*
   - *iii. Gender-awareness*
   - *iv. Dialogue structure*
   - *v. Symptom-based diagnosis*
5. **A Favourable Research Environment**



DAIC-WOZ Dataset

- The DAIC-WOZ dataset [DeVault et al. 2014] includes Wizard-of-Oz interviews, conducted by an animated virtual interviewer called Ellie, controlled by a human interviewer in another room.
- This share includes 189 sessions of interactions ranging between 7-33min (average of 16min).
- Each session is combined with a PHQ-8 questionnaire.

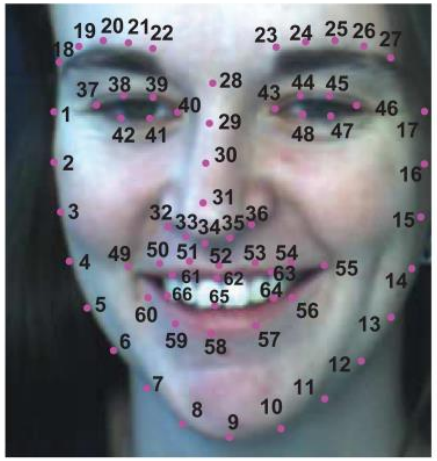
	Opening Rapport Building Phase
Ellie	*What are some things you really like about LA?* (top level question)
User	*I love the weather, I love the palm trees, I love the beaches, there's a lot to do here.*

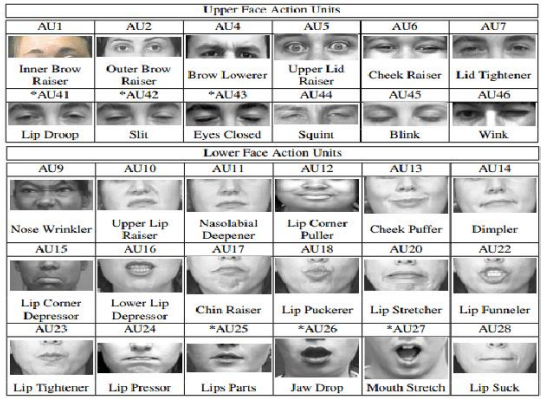
	Diagnostic Phase
Ellie	*Have you noticed any changes in your behavior or thoughts lately?* (top level question)
User	*Yes.*
Ellie	*Can you tell me about that?* (continuation prompt)
User	*I'm having a lot more nightmares now uh can't sleep have haven't really been eating uh trying to eat… I have to force down food um just feeling like an emotional wreck.*
Ellie	*I'm sorry to hear that.* (empathy response)
Ellie	*What are you like when you don't sleep well?* (follow-up question)
User	*Irritable, emotional, it just adds to my overall stress um [long pause]*
Ellie	*What…* (Ellie speaks after the participant's long pause)
User	*Can't concentrate uh I uh…* (the participant starts speaking while Ellie is speaking)
Ellie	*I'm sorry please continue.* (Ellie realizes that she has interrupted the participant and apologizes)

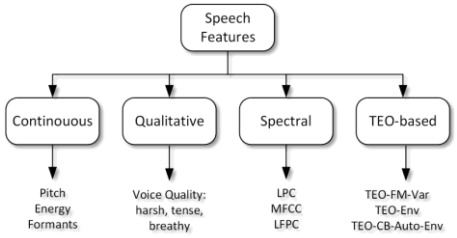
Questionnaire	**0-1 day** Not at all	**2-6 days** Several days	**7-11 days** More than half days	**12-14 days** Nearly everyday
1. Limited interest in doing work	15%	20%	40%	25%
2. Subjects with feeling of depression, hopelessness	20%	35%	15%	30%
3. Difficulty in sleeping or long sleep	40%	13%	25%	22%
4. Tiredness	20%	17%	35%	28%
5. Anorexia or excessive eating	14%	15%	37%	34%
6. Self bad feeling	10%	10%	30%	50%
7. Difficulty in concentration in work	30%	15%	15%	40%
8. Speaking or moving so slowly	23%	20%	25%	32%

Multimodal Estimation of PHQ-8

- In a patient-therapist interview, different signals shoud be combined for a correct diagnosis.
- Within the DAIC-WOZ dataset, the following signals are available:
  - Visual signals : expression of sadness, gaze escape, etc.
    - Facial Landmarks (FL), Head Pose (HP), Eye Gaze (EG), Action Unit (AU).
  - Speech signals : veiled voice, monotonous tone, etc.
    - Formant (FMT), COVAREP (COV).
  - Language signals : negative vocabulary, lack of perspective, etc.
    - Universal Sentence Encoder (TR).

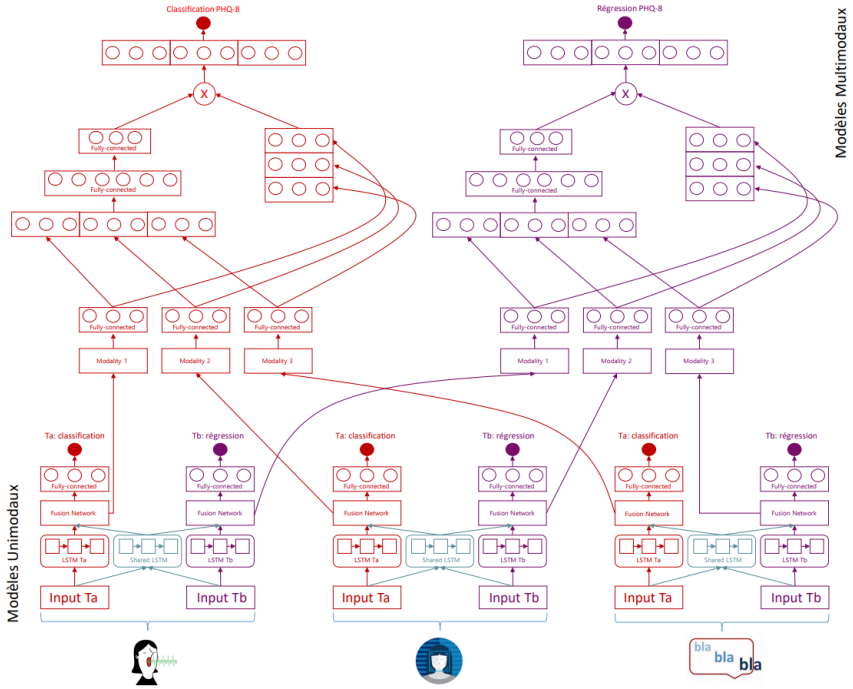


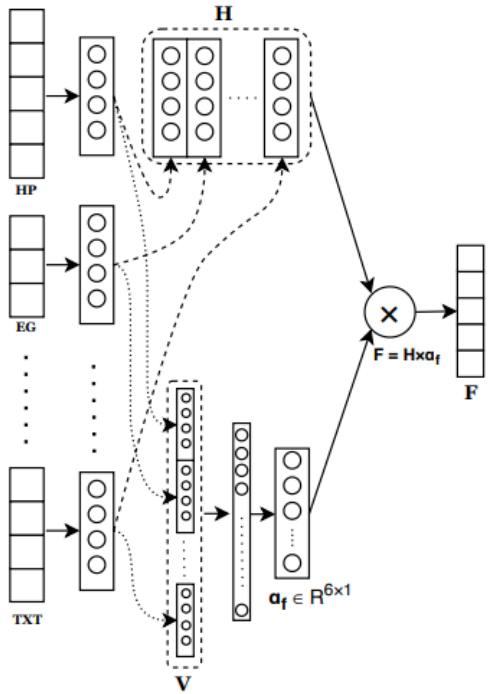


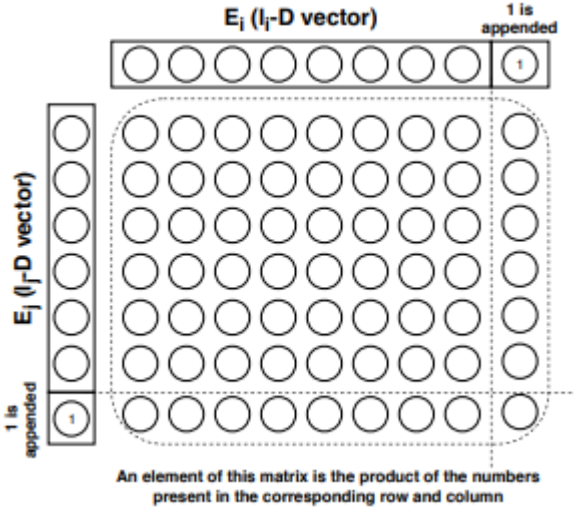


Multimodal Estimation of PHQ-8

- Combining classification and regression of depression estimators.
- An attention fusion network is used to combine inputs.
- Intra-modality inputs signals are combined with tensors.







Multimodal Estimation of PHQ-8

Table 1. Overall results. ST: Single Task, MT: Multitask, FS: Fully Shared, SP: Shared Private, DLC: Depression Level Classification, DLR: Depression Level Regression, HP: Head Pose, EG: Eye Gaze, AU: Action Units, COV: COVAREP, FMT: Formant, TXT: Text.

	Architectures	RMSE	MAE	Acc (%)	F-score
Unimodal	ST-DLR-HP	6.89	5.67	-	-
	ST-DLC-HP	-	-	54.54	0.41
	FS-MT-HP	6.75	**5.48**	**60.60**	**0.43**
	SP-MT-HP	**6.65**	5.53	54.54	0.42
	ST-DLR-EG	6.67	4.72	-	-
	ST-DLC-EG	-	-	54.54	0.37
	FS-MT-EG	**6.50**	**4.60**	**57.57**	**0.41**
	SP-MT-EG	6.59	5.16	57.57	0.39
	ST-DLR-AU	6.49	5.55	-	-
	ST-DLC-AU	-	-	54.54	0.42
	FS-MT-AU	**6.28**	**5.03**	54.54	0.44
	SP-MT-AU	6.46	5.42	**57.57**	**0.45**
	ST-DLR-COV	6.64	5.72	-	-
	ST-DLC-COV	-	-	51.51	0.36
	FS-MT-COV	**6.55**	**5.67**	**54.54**	**0.40**
	SP-MT-COV	6.59	5.71	54.54	0.37
	ST-DLR-FMT	6.91	5.89	-	-
	ST-DLC-FMT	-	-	51.51	0.34
	FS-MT-FMT	6.72	**5.77**	**54.54**	**0.36**
	SP-MT-FMT	**6.69**	5.79	51.51	0.34
	ST-DLR-TXT	4.90	3.99	-	-
	ST-DLC-TXT	-	-	60.60	0.45
	FS-MT-TXT	4.96	3.90	**66.66**	**0.53**
	SP-MT-TXT	**4.70**	**3.81**	60.61	0.42
Multimodal	ST-DLR-CombAtt	4.42	3.46	-	-
	MT-DLR-CombAtt	**4.24**	**3.29**	-	-
	ST-DLC-CombAtt	-	-	57.57	0.46
	MT-DLC-CombAtt	-	-	**60.61**	**0.48**
SOTA	$VFSC_{sem}$	4.46	3.34	-	-
	AW_{bhv}	5.54	4.73	-	-
	MMD	4.65	3.98	-	-

Results are still far from satisfactory

Combining regression and classification is beneficial

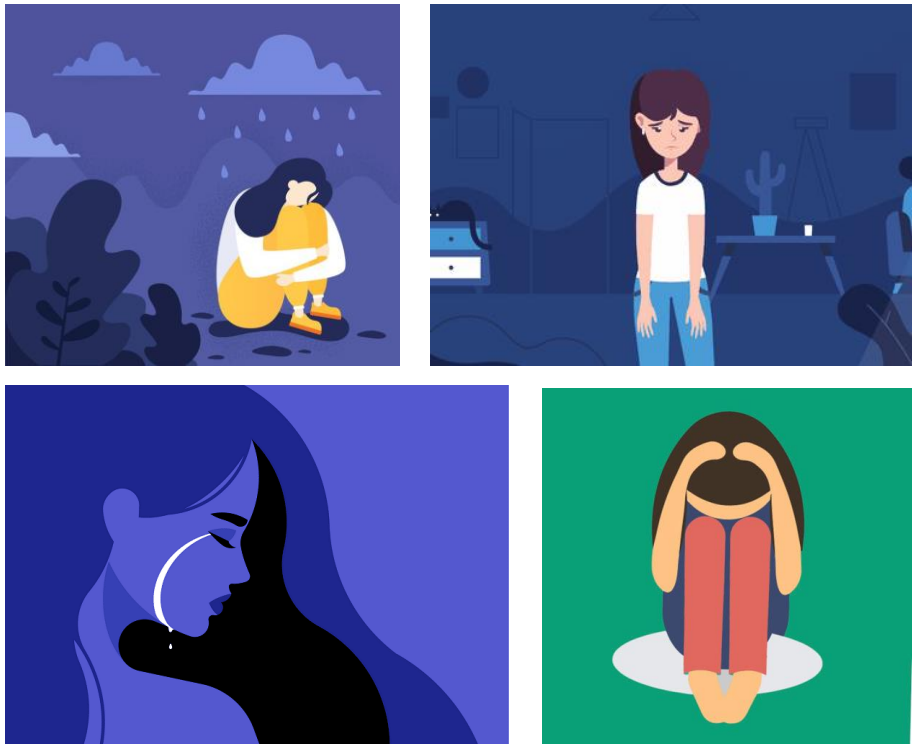
Language signal is strong

Multimodality is beneficial …

… But not for classification

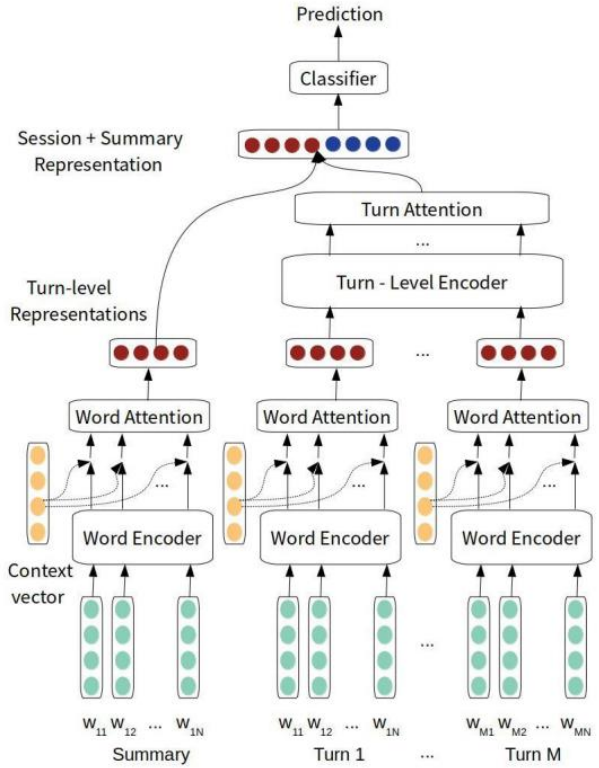
Outline

1. *Mental Health and Depression*
2. *6P Medicine*
3. *Interesting Iniatives*
4. **Computer-aided Diagnosis**
   - i. *Multimodality*
   - ii. *Emotionality*
   - iii. *Gender-awareness*
   - iv. *Dialogue structure*
   - v. *Symptom-based diagnosis*
5. **A Favourable Research Environment**



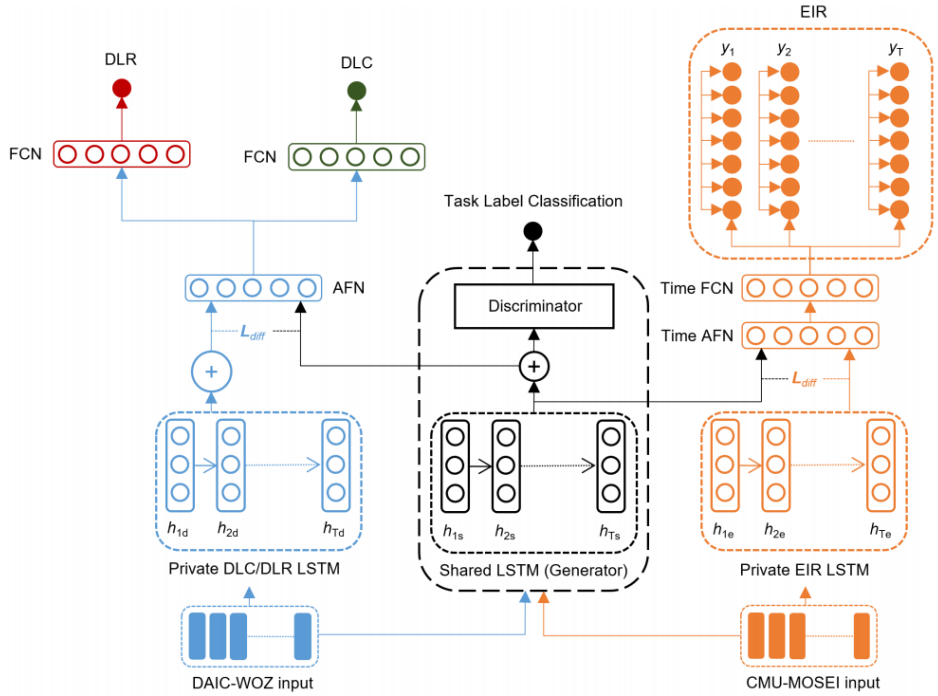
Emotional Language for the Estimation of PHQ-8

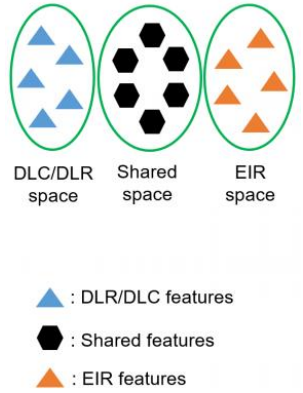
- Studies show that depression is a disorder of impaired emotion regulation.
- In particular, patients with major depression are often unable to control their emotional responses to negative situations, and overuse emotional expressions of sadness, disgust or fear.
- Emotion intensity can be evaluated on a Likert [0,3] scale for the six emotions of Ekman: happiness, sadness, anger, fear, disgust and surprise. But other models exist such as arousal and valence.
- In [Xezonaki et al., 2020], emotions are appended as an external context vector built from affective lexica (emotion, sentiment, valence).



Emotional Language for the Estimation of PHQ-8

- In [Qureshi et al., 2020], we hypothesize that the estimation of depression level can benefit from the concurrent learning of emotion intensity.
- The CMU-MOSEI dataset comprises 3,228 videos from 1,000 different speakers over 250 topics. Videos were gathered from an online video platform, where users emit their opinions in the form of monologues.





Emotional Language for the Estimation of PHQ-8

Not satisfactory for classification

Models	Evaluation Metrics													
	DLC							DLR						EIR
	Acc.	F1	MCC	RMSE	MAE	Ov.	Un.	RMSE	MAE	R^2	SM.	$\overline{Ov.}$	$\overline{Un.}$	$\overline{MSE}$
Baselines without Emotion Intensity Regression														
ST. DLC	60.61	0.54	0.38	1.31	0.75	3.03	36.36	-	-	-	-	-		-
ST. DLR	-		-	-	-	-	-	4.90	3.99	0.46	0.97	3.21	5.18	-
ST. EIR	-		-	-	-	-	-	-	-	-	-	-	-	7.15
FS MT. DLC+DLR	**66.66**	**0.62**	**0.49**	1.23	0.66	3.03	30.31	4.96	3.89	0.44	0.98	2.81	5.19	-
SP MT. DLC+DLR	60.61	0.51	0.39	1.26	0.72	0.00	39.39	4.70	3.81	0.50	0.99	3.39	4.32	-
Multi-task Results with Emotion Intensity Regression														
FS MT. DLC+EIR	60.61	0.51	0.42	1.58	0.90	0.00	39.39	-	-	-		-	-	6.98
SP MT. DLC+EIR	57.57	0.50	0.35	1.27	0.76	6.07	36.36	-	-	-	-	-	-	7.05
ASP MT. DLC+EIR	60.61	0.54	0.38	1.26	0.73	9.09	30.30	-	-	-	-	-	-	7.19
FS MT. DLR+EIR	-	-	-	-	-	-	-	4.60	3.74	0.52	0.99	3.16	4.63	6.88
SP MT. DLR+EIR	-	-	-	-	-	-	-	**4.51**	3.89	**0.54**	**0.94**	3.91	3.85	**6.82**
ASP MT. DLR+EIR	-	-	-	-	-	-	-	4.72	3.96	0.50	**0.94**	3.80	4.15	7.08
FS MT. DLC+DLR+EIR	57.57	0.46	0.38	1.36	0.82	3.04	39.39	4.83	4.03	0.47	0.97	3.13	5.11	6.96
SP MT. DLC+DLR+EIR	**63.64**	0.58	**0.48**	**0.94**	**0.51**	24.24	12.12	4.56	3.79	0.53	0.97	3.20	4.59	7.02
ASP MT. DLC+DLR+EIR	60.61	**0.60**	0.42	1.14	0.64	12.12	27.27	4.61	**3.69**	0.52	0.95	2.87	4.81	7.11

Interesting results for regression, although with small improvements

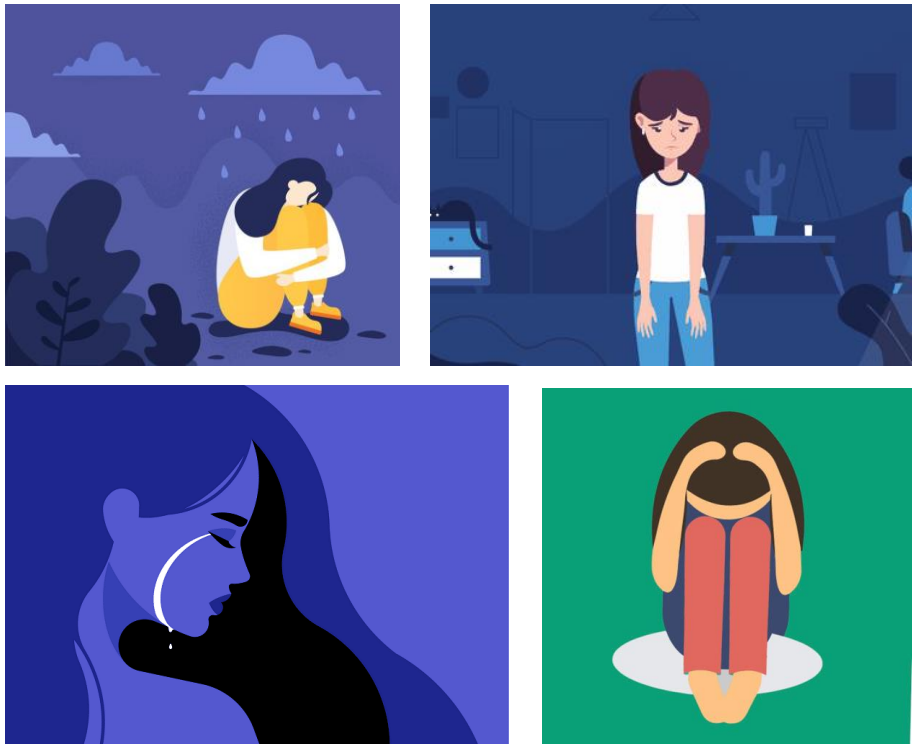
High standard deviation per class

Models	Evaluation Metrics								
	DLC					DLR			
	Acc.	RMSE	MAE	Ov.	Un.	RMSE	MAE	$\overline{Ov.}$	$\overline{Un.}$
	Best for DLC without EIR: FS MT. DLC+DLR					**Best for DLR without EIR: SP MT. DLC+DLR**			
None-minimal	100	0.00	0.00	0	-	3.97	3.22	3.51	1.14
Mild	40	1.10	0.80	20.00	40	3.80	3.11	3.82	2.05
Moderate	40	1.34	1.00	0.00	60	4.04	3.50	0.00	3.50
Moderately severe	33.33	2.27	1.83	0.00	66.67	6.78	5.75	0.47	6.81
Severe	0	2.00	2.00	-	100	6.81	6.81	0.00	6.81
	Best for DLC+EIR: ASP MT. DLC+EIR					**Best for DLR+EIR: SP MT. DLR+EIR**			
None-minimal	100	0.00	0.00	0	-	4.28	3.85	4.05	0.74
Mild	20	1.18	1.00	40	40	3.51	3.07	3.56	2.32
Moderate	20	1.61	1.40	20	60	2.94	2.60	0.00	2.60
Moderately severe	33.33	2.16	1.67	0	66.67	6.70	6.05	2.77	6.71
Severe	0	2.00	2.00	-	100	2.03	2.03	0.00	2.03
	Best for DLC+DLR+EIR: SP MT. DLC+DLR+EIR								
None-minimal	93.75	0.50	0.13	6.25	-	3.42	2.89	2.97	1.79
Mild	0	1.00	1.00	60	40	3.78	3.49	3.89	2.88
Moderate	80	0.89	0.40	0	20	3.84	3.37	0.00	3.37
Moderately severe	33.33	1.41	1.00	0	66.67	7.54	6.78	4.67	7.21
Severe	0	2.00	2.00	-	100	3.85	3.85	0.00	3.85

Strong under-evaluation for the moderately severe class

Outline

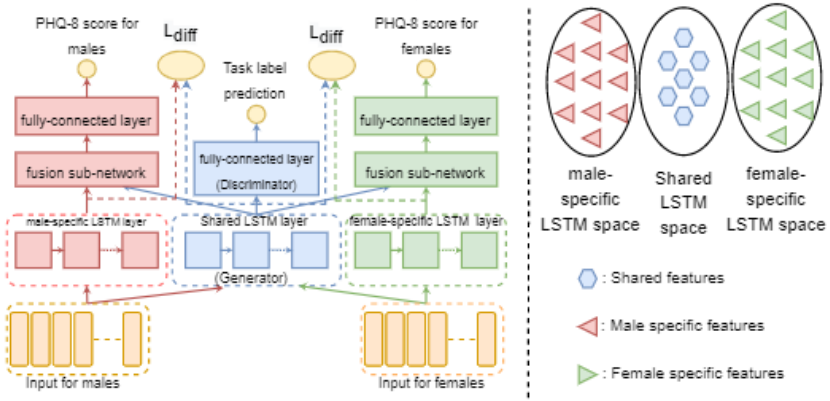
1. *Mental Health and Depression*
2. *6P Medicine*
3. *Interesting Iniatives*
4. **Computer-aided Diagnosis**
   - i. *Multimodality*
   - ii. *Emotionality*
   - iii. *Gender-awareness*
   - iv. *Dialogue structure*
   - v. *Symptom-based diagnosis*
5. **A Favourable Research Environment**



Gender-awareness for the Estimation of PHQ-8

- [Joan and Kaite, 2015] reviewed several works in psychological research on the difference in gender in depression.
  - They state that by the middle of adolescence, females are about twice as likely to be diagnosed with depression and exhibit twice as many depressive symptoms as males, and this trend may continue till they are at least 55 years old.
- However, very few works have been proposed on how depression is dependent on gender.
- In [Qureshi et al., 2021], we propose to study gender-aware models in multimodal settings.

- Depression estimation without gender information (Gen_{less})
- Depression estimation with concatenated gender information (Gen_{concat})
- Multitask prediction of depression level and gender (Gen_{pred})
- Multitask prediction of depression level in males and females separately, using shared-private multitask network [35] (Gen_{SP})
- Multitask prediction of depression level in males and females separately, using adversarial shared-private multitask network [35] (Gen_{ASP})



Gender-awareness for the Estimation of PHQ-8

Models	Evaluation Metrics									
	Gen_{less}		Gen_{concat}		Gen_{pred}		Gen_{SP}		Gen_{ASP}	
	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE	MSE	MAE
COVAREP	44.98	5.32	44.59	5.16	**43.05**	5.14	43.70	**5.11**	44.13	5.14
Formant	43.11	5.48	42.21	5.50	42.29	5.54	42.53	5.56	**41.96**	**5.19**
Facial action units	42.32	5.51	41.97	**5.13**	**41.06**	5.47	41.90	5.15	41.97	5.21
Eye gaze	47.26	5.57	47.04	5.62	46.05	5.75	48.01	5.72	**44.41**	**5.23**
Facial landmarks	52.82	6.21	50.72	6.06	52.45	5.93	47.16	5.87	**45.13**	**5.51**
Head pose	48.99	5.78	47.31	5.74	46.92	5.76	46.56	5.54	**44.29**	**5.40**
Text	23.82	**3.78**	23.28	3.87	**23.12**	3.87	24.12	4.10	24.02	4.09
Multimodal	24.12	3.74	**20.06**	3.50	20.56	3.50	21.01	3.51	22.25	**3.49**

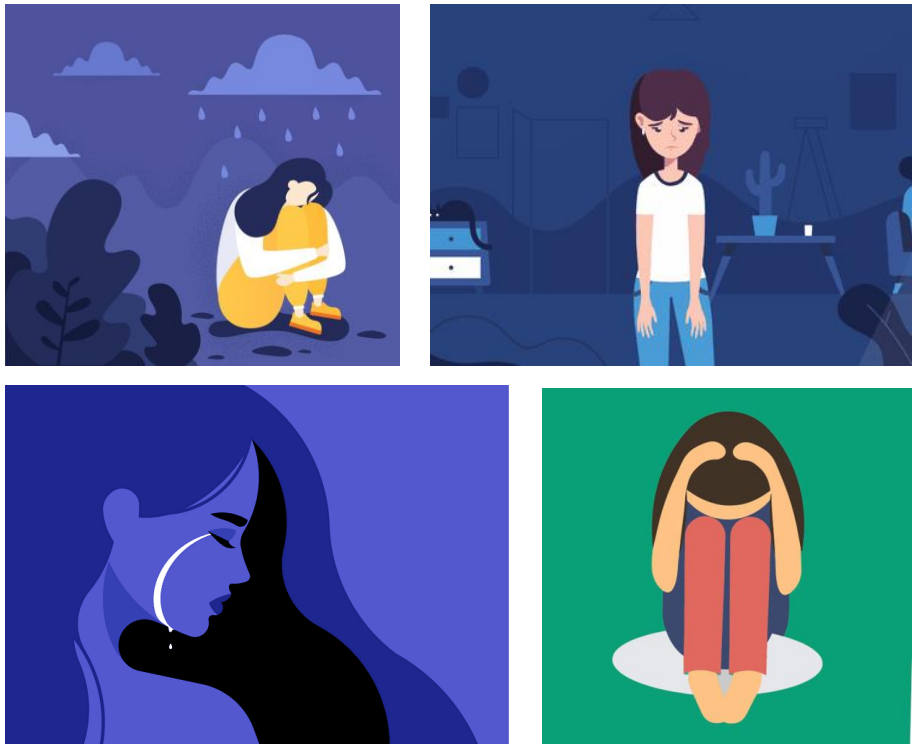
Strong indicator for the visual signal

Text is not so sensitive to gender!

Gender-awareness is important

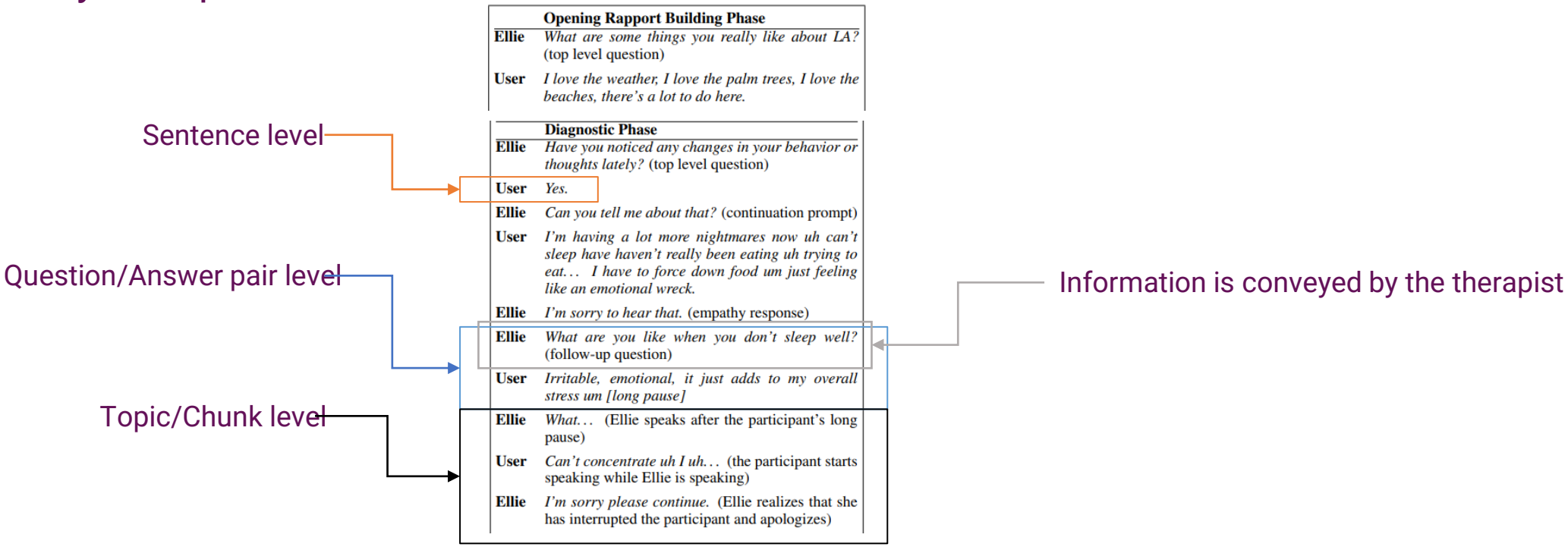
Outline

1. *Mental Health and Depression*
2. *6P Medicine*
3. *Interesting Iniatives*
4. **Computer-aided Diagnosis**
   i. *Multimodality*
   ii. *Emotionality*
   iii. *Gender-awareness*
   iv. *Dialogue structure*
   v. *Symptom-based diagnosis*
5. **A Favourable Research Environment**



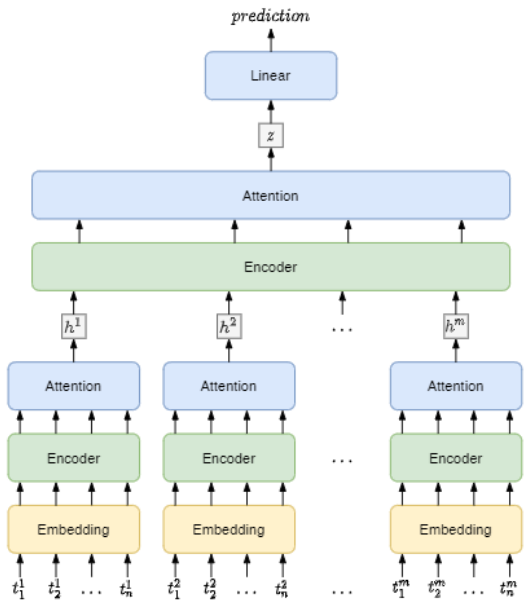
Analysis of Structured Interviews

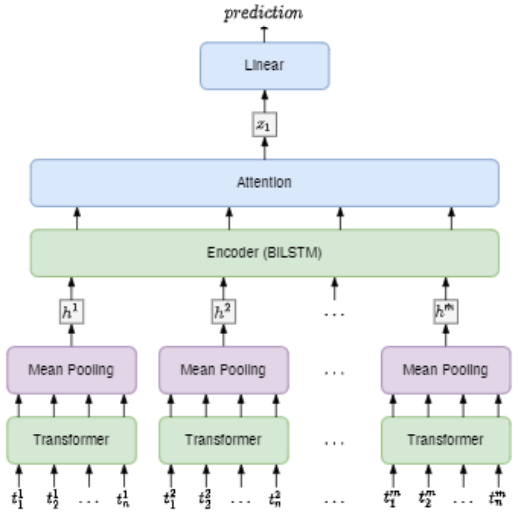
- First observation: most of the related works have been dealing with the interview on a line by line basis; the hypothesis being that sentence representation is the correct one.
- Second observation: some of the related works only deal with the patient information; the hypothesis being that only the patient information is important for the diagnosis.
- Our hypothesis is that better diagnosis can be established if the correct level of language analysis is performed.



Segmentation Level Analysis

- We propose to segment interviews into three different linguistic levels: sentences, question/answer pairs and semantic chunks.
- For that purpose, the DAIC-WOZ has been manually annotated at chunk level.
- To verify our hypothesis, we implement two different learning models: one on non-contextualized text embeddings [Xezonaki et al., 2020], and one with contextualized text embeddings.



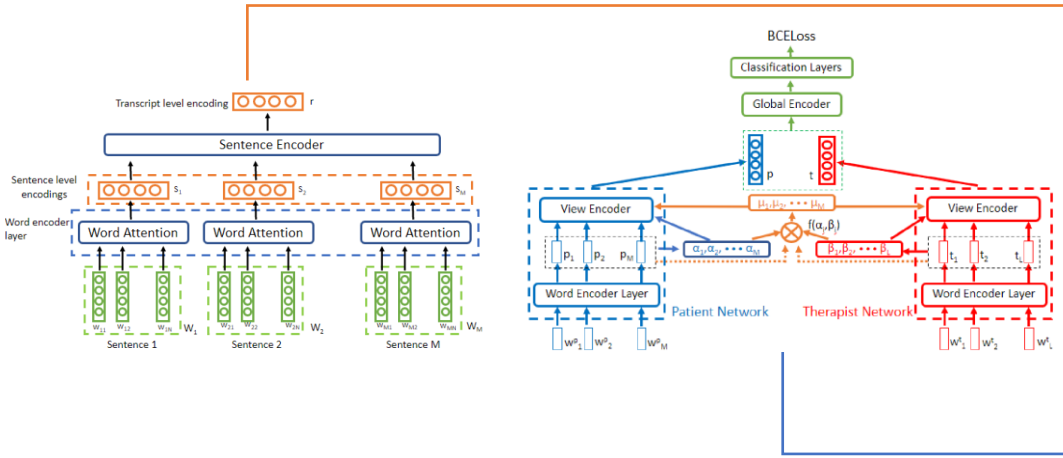


Model	Segmentation	F1	F1 (max)	MAE
GloVe+LSTM+LSTM	Lines	60.95 ± 8.02	72.63	5.606 ± 0.406
	Pairs	63.04 ± 7.84	79.55	5.678 ± 0.489
	Semantic	74.18 ± 9.05	87.59	5.302 ± 0.719
BERT+LSTM	Lines	91.38 ± 3.95	94.04	5.805 ± 0.440
	Pairs	87.71 ± 7.82	94.04	5.749 ± 0.638
	Semantic	74.69 ± 7.28	88.19	5.394 ± 0.507

Table: Results on the validation set. Each model was run 20 times with different random initializations. **F1** represents a macro-average F1-score and is given as a mean $\pm$ standard deviation; **F1 (max)** shows the maximum F1-score among 20 runs.

Patient vs. Therapist Information

- [Xenozaki, 2020] showed that both patient and therapist information convey information, but do not take advantage of this fact.
- So, we propose a multiview model that tackles both patient and therapist texts individually and then fuses the information to get a single prediction.
- Three different attention levels are proposed: local attention (patient OR therapist), cross attention (patient -> therapist and therapist -> patient), global attention (patient AND therapist).

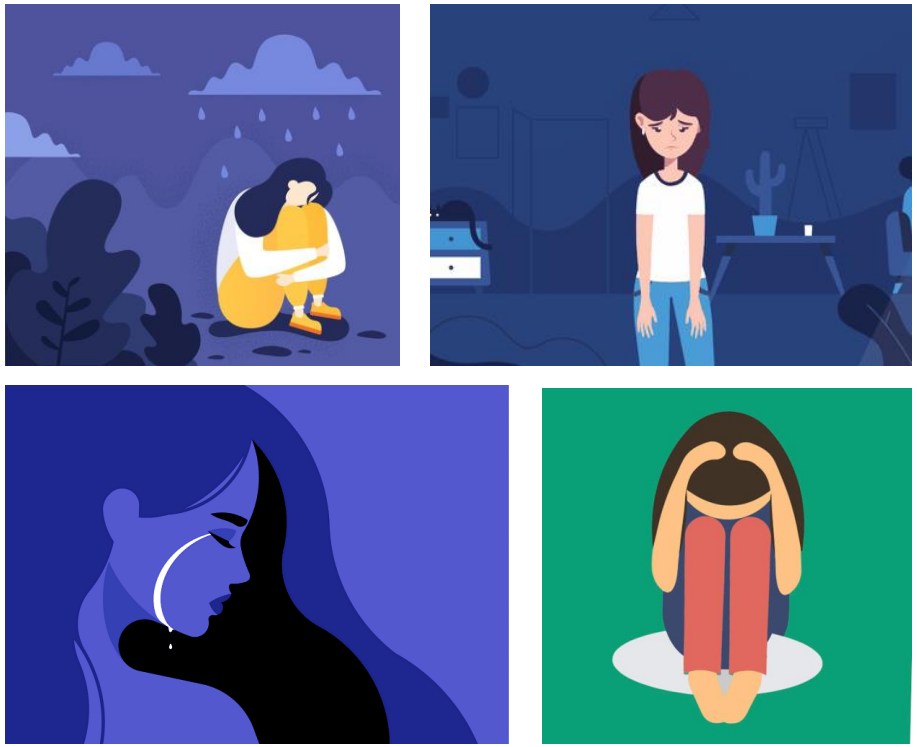


Architectures		macro F1 (Dev)	macro F1 Test	UAR (Dev)	UAR Test	Accuracy (Dev)	Accuracy Test	macro Precision (Dev)	macro Precision Test
Baseline	Patient	(0.6413)	0.6429	(0.6369)	0.6361	(0.6969)	**0.7608**	(0.6725)	0.6584
	Therapist	(0.8253)	0.5818	(0.8095)	0.5803	(0.8484)	0.6521	(0.8611)	0.6184
	Patient + Therapist	(0.7555)	0.6053	(0.7440)	0.6004	(0.7878)	0.6739	(0.7847)	0.6250
MV-Intra-Attention	View-Global Attention	(0.6944)	0.6811	(0.6845)	0.6674	(0.7575)	0.7391	(0.7870)	0.7252
	Global Attention	(0.6857)	0.7116	(0.6785)	0.7075	(0.7272)	0.7173	(0.7083)	0.6887
	View Attention	(0.6944)	0.6919	(0.6845)	0.6919	(0.7575)	0.6739	(0.7870)	0.6919
MV-Inter-Attention	Mean	(0.6857)	**0.7319**	(0.6785)	0.7232	(0.7272)	0.7173	(0.7083)	0.7450
	Learnable	(0.6434)	0.6043	(0.6428)	0.6093	(0.7272)	0.4782	(0.7571)	0.6020
	Max	(0.6616)	0.5801	(0.6845)	0.5982	(0.6666)	0.6304	(0.6709)	0.5846
	Patient	(0.5460)	0.5719	(0.5476)	0.5736	(0.6060)	0.6956	(0.5555)	0.5709
	Therapist	(0.7664)	0.5710	(0.7619)	0.5691	(0.7878)	0.6304	(0.7727)	0.5759

Table 2: Overall results over the DAIC-WOZ dataset. UAR stands for Unweighted Average Recall. We provide results for both development and test sets. The best model is chosen based on macro F1 over the development set.

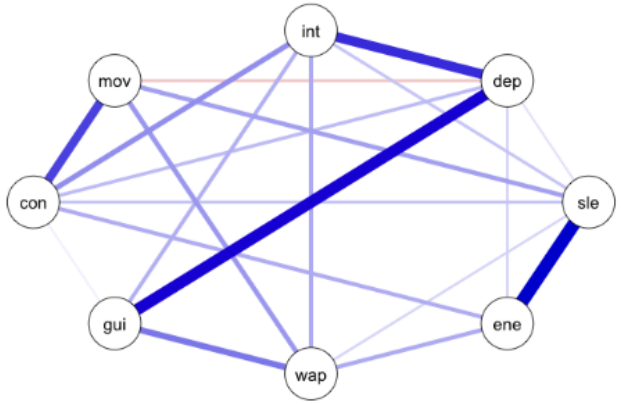
Outline

1. *Mental Health and Depression*
2. *6P Medicine*
3. *Interesting Iniatives*
4. **Computer-aided Diagnosis**
   i. *Multimodality*
   ii. *Emotionality*
   iii. *Gender-awareness*
   iv. *Dialogue structure*
   v. *Symptom-based diagnosis*
5. **A Favourable Research Environment**



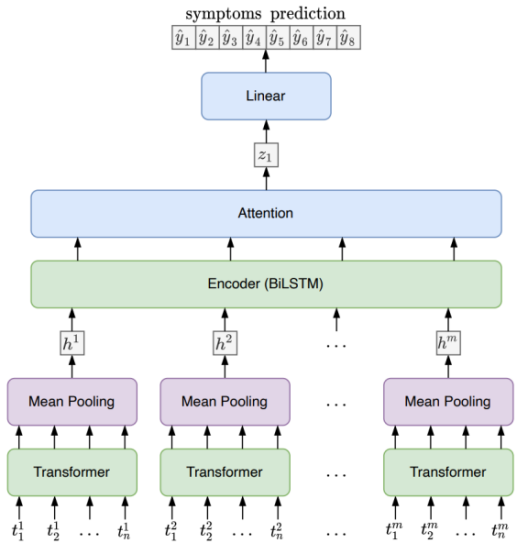
Symptom-based Analysis

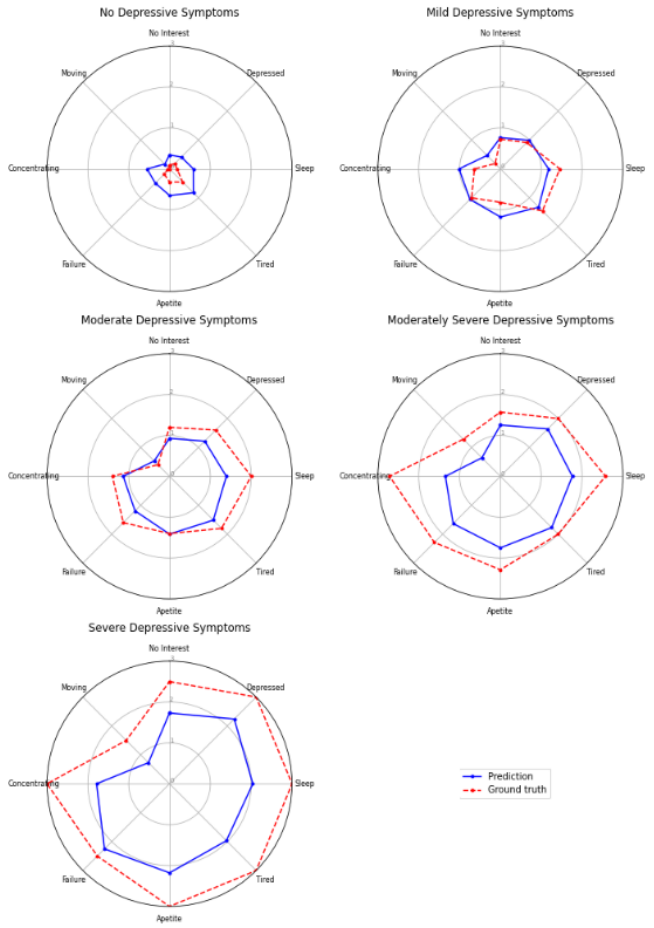
- Most related works have been tackling depression level estimation as a simple task (depressed or non depressed). More advanced models have been trying to predict the PHQ-8 score (between 0 and 24) directly or propose to solve the intermediate 5-class problem (none-minimal, mild, moderate, moderately severe, severe depression).
- In Psychiatry, there is a shift towards richer representations of psychiatric syndromes that can take into account the dimensional and heterogeneous nature of the clinical pictures of the same psychiatric diagnosis. One particular approach that is gaining attention concerns symptom network analysis.
- We develop similar models as previously to acknowledge if they can handle the prediction of individual symptom values, where each of the 8 symptoms is a value between 0 and 3.



Symptom-based Analysis

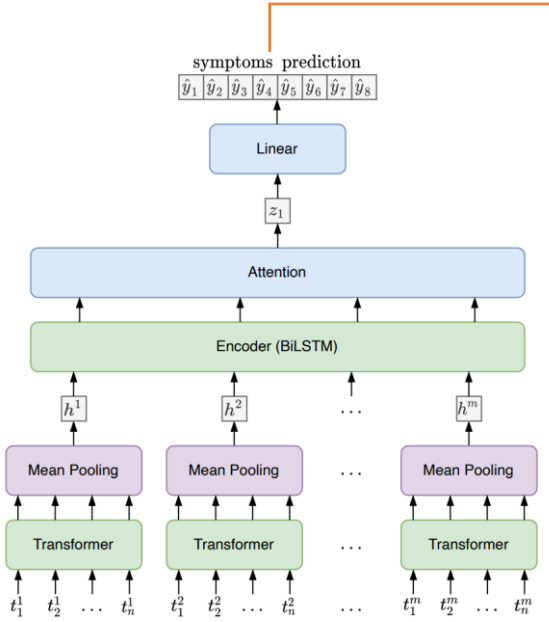
- In order to better understand results, we present a radar plot analysis that shows that adequate behavior of the model is obtained.





Symptom-based Analysis

- We evaluate the impact of categorical diagnosis based on symptoms prediction.

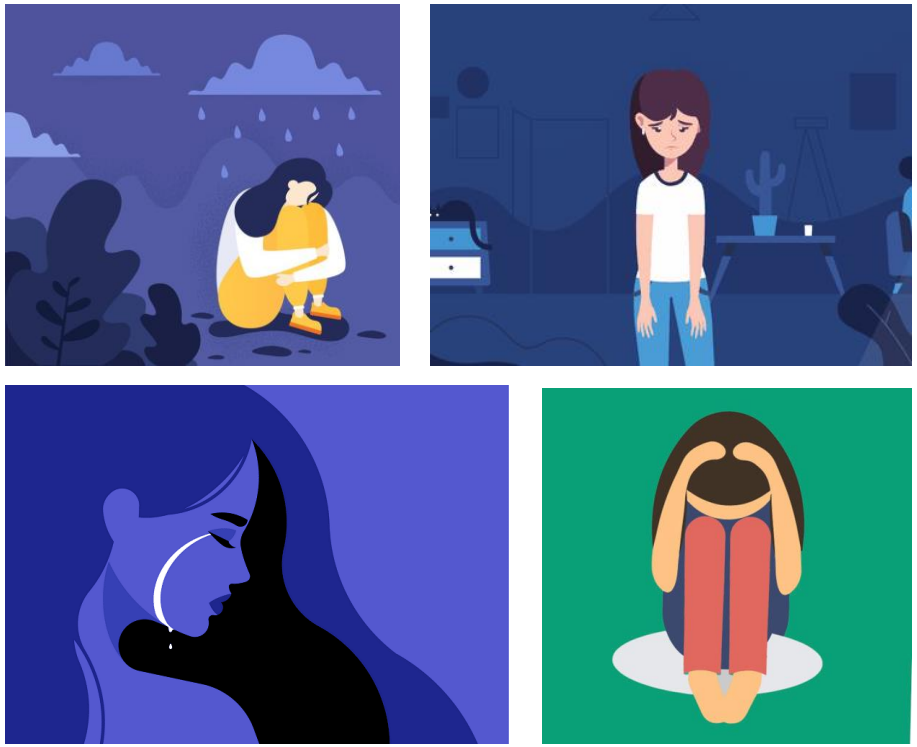


Model	$miF_1 \pm\sigma$	$maF_1 \pm\sigma$	miMAE $\pm\sigma$	maMAE $\pm\sigma$	$miF1$-5c $\pm\sigma$	$maF1$-5c $\pm\sigma$
Binary Diagnosis (BD)	71.91 ± 1.59	70.08 ± 1.03	-	-	-	-
5-Class Diagnosis (5CD)	71.06 ± 2.55	68.30 ± 2.39	-	-	**46.81** ± 2.33	**27.03** ± 2.46
PHQ-8 Score Diagnosis (PSD)	68.09 ± 1.90	58.44 ± 2.39	5.03 ± 0.09	5.69 ± 0.12	28.94 ± 2.89	13.49 ± 1.43
Symptom-based Diagnosis (SD)	**76.60** ± 2.33	**73.87** ± 2.48	**3.78** ± 0.13	**4.19** ± 0.13	42.55 ± 1.35	27.00 ± 1.93
SOTA results based on the text modality only						
HCAN [7] (2019)	-	†63.00	-	-	-	-
HAN+L [8] (2020)	-	†70.00	-	-	-	-
ASP MT. DLC+DLR+EIR [25] (2020)	-	-	3.69	-	60.00	-
HCAG-T [23] (2021)	-	†77.00	†3.73	-	-	-
SGNN [27] (2022)	-	-	†3.76	-	-	-
SOTA results based on the multiple modalities						
SVM:m-M&S [9] (2021)	-	67.00	3.98	-	-	-
Gen$_{ASP}$ [26] (2021)	-	-	3.49	-	-	-
MFCC-AU [31] (2021)	-	66.50	-	-	-	-
HCAG-A+T [23] (2021)	-	†92.00	†2.94	-	-	-
BLSTM [28] (2022)	-	-	-	-	‡95.80	-

Table 2 Experimental and state-of-the-art results over the test set of the DAIC-WOZ. Models are run five times with different seed values for BD, 5CD, PSD and SD, so that average values with standard deviation are presented. Note that $miF1$-5c (resp. $maF1$-5c) stand for the 5-class micro-averaged F1-score (resp. macro-averaged F1-score). Note that "†" indicates that results are given for the best configuration and not based on average performance. Note also that "‡" indicates that results are given for a balanced test set and not the original test set provided with the DAIC-WOZ.

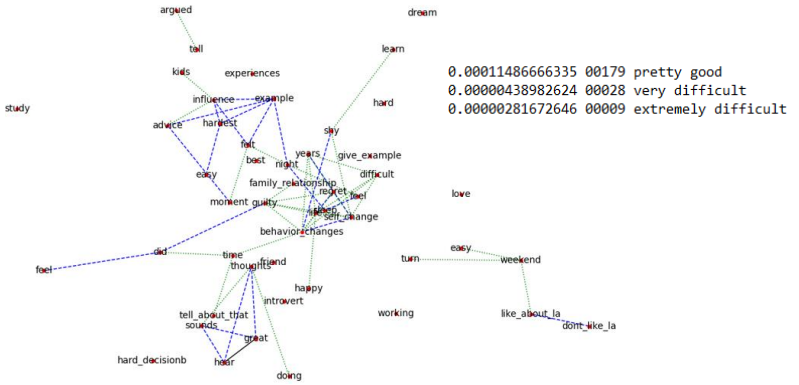
Outline

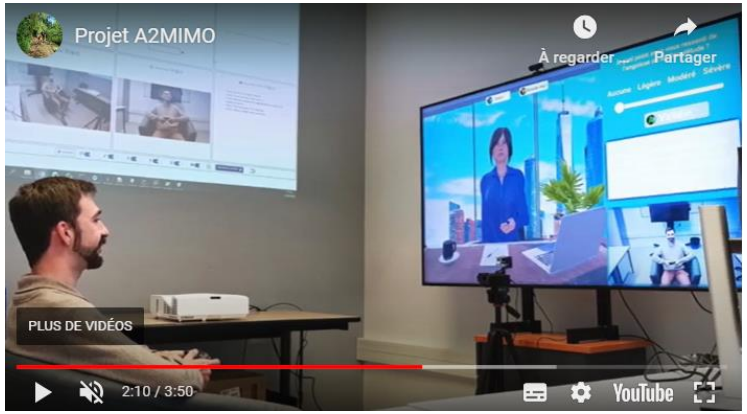
1. *Mental Health and Depression*
2. *6P Medicine*
3. *Interesting Iniatives*
4. *Computer-aided Diagnosis*
   i. *Multimodality*
   ii. *Emotionality*
   iii. *Gender-awareness*
   iv. *Dialogue structure*
   v. *Symptom-based diagnosis*
5. **A Favourable Research Environment**



On-going Research @ CAEN

- Navneet Agarwal
  - Graph-based representation and learning.
  - Psychiatrists into the loop.
- Kirill Milintsevich
  - External knowledge introduction.
  - Dataset quality assessment.
- Soumaya Sabry
  - Embodied conversational agents for early detection.
  - Therapeutic alliance.





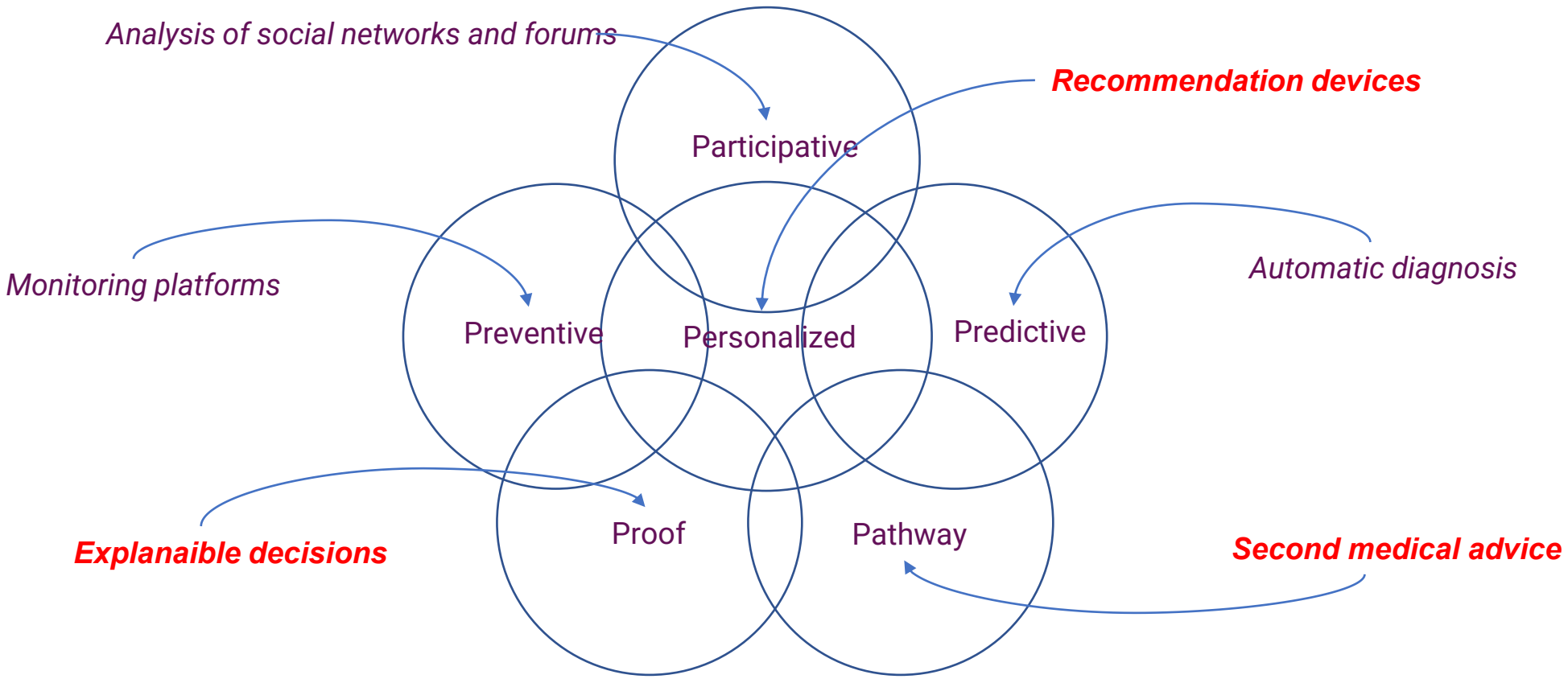
Other Running Projects

- Prediction of suicidal recidivism from phone conversations.
  - Pr. Françoise CHASTANG and Dr. Pierre GERARD.
  - Project VigilanS.
  - CHU Estran.
- Automatic level estimation of schizophrenia from emergency interviews.
  - Pr. Christophe LEMEY and Pr. Sonia DOLLFUS.
  - Project ASESID.
  - CHU Brest.

Structuring Mental Health in Normandy

- FHU A2M2P - Improving the prognosis of addictive and mental disorders through personalized medicine (Améliorer le pronostic des troubles Addictifs et Mentaux par une Médecine Personnalisée).
  - 5-year project including 11 research laboratories (CNRS, INSERM, EA), 4 hospitals (Amiens, Caen, Rouen), patient and family associations, public health services (ARS).
  - Jointly studying mental disorders and drug addiction.
- Department of Mental Health and Digital Sciences at BB@C (Blood and Brain GIS).
  - Gathering worldwide specialists in AI and Mental Health inside the same research structure.
  - Initiative of the Pole TES competitiveness cluster and the agglomeration of Caen.
- Second Workshop on Mental Health and Artificial Intelligence @ Caen
  - January 29th -30th, 2024.
  - https://mentalai.ubi.pt/symposium

Un(less)explored Areas



THANK YOU FOR YOUR ATTENTION

Freel free to ask caring questions ;)

Gaël DIAS @ Sorbonne

joint work with
Navneet AGARWAL, Mohammed HASANUZZAMAN, Arbaaz QURESHI, Kirill MILINTSEVICH, Valentin RENIER, Soumaya SABRY, Sriparna SAHA, Kairit SIRTS, and more to come ;)

University of Caen Normandie - CNRS - GREYC UMR 6072 Research Laboratory
gael.dias@unicaen.fr

GREYC
Electronics and Computer Science Laboratory

Normandie Université

UNICAEN

ENSI CAEN
ÉCOLE PUBLIQUE D'INGÉNIEURS
CENTRE DE RECHERCHE

cnrs