Apprentissage Automatique : représentation des données

L. Grobol (MoDyCo, Université Paris Nanterre)

M2 Plurital

Nanterre, France, 2024-12-04

« *kevrin* » : nom ou verbe ?

sardin
gwez
reizh
kentel

labourin
c’hoarzhec’h
soñjont
graet

« *kevrin* » : nom ou verbe ?

sardin	sardine
gwez	arbres
reizh	genre
kentel	leçon

labourin	travailler-fut-1sg
c’hoarzhec’h	rire-ipfv-2pl
soñjont	penser-pres-3pl
graet	aller-pst-ptcp

« *kevrin* » : nom ou verbe ?

sardin

kentel

gwez

reizh

labourin

c’hoarzhec’h

soñjont

graet

sardin

kentel

gwez

reizh

kevrin

labourin

c’hoarzhec’h

soñjont

graet

« *kevrin* » : nom ou verbe ?

sardin	sardine
gwez	arbres
reizh	genre
kentel	leçon
kevrin	secret

labourin	travailler-fut-1sg
c’hoarzhec’h	rire-ipfv-2pl
soñjont	penser-pres-3pl
graet	aller-pst-ptcp

Machine

Les algos d'apprentissage marchent plutôt sur des nombres ou souvent des séries de nombres (parce qu'on peut les voir comme des points dans un espace géométrique) On a donc un problème constant qui est :

Comment transformer nos données (texte, signal sonore, images, vidéo) en nombres ?

Méthodes immédiates

En soi on a toujours une solution :

Méthodes immédiates

En soi on a toujours une solution : de fait, si on a des données sous forme numérique (dans une machine), elles sont comme leur nom l'indique numérique (représentées par des nombres, vous suivez ?)

Ces représentations peuvent être bien (par exemple le codage numérique des images traduit une réalité physique/perceptive du coup, c'est pas mal) ou pas du tout (une suite de nombres représentant des points de code Unicode qui représentent des caractères, c'est assez éloigné de la façon dont on pense le langage).

Méthodes immédiates

En soi on a toujours une solution : de fait, si on a des données sous forme numérique (dans une machine), elles sont comme leur nom l'indique numérique (représentées par des nombres, vous suivez ?)

Ces représentations peuvent être bien (par exemple le codage numérique des images traduit une réalité physique/perceptive du coup, c'est pas mal) ou pas du tout (une suite de nombres représentant des points de code Unicode qui représentent des caractères, c'est assez éloigné de la façon dont on pense le langage).

Ça ne veut pas dire que c'est impossible de se servir de ça, mais plutôt que c'est pas toujours la meilleure solution. Ça dépend de beaucoup de facteurs.

Comment représenter des données linguistiques écrites ?

Représenter les mots

On peut représenter les mots comme des nombres en constituant un lexique et en affectant à chaque item un nombre entier : sa position dans l'ordre lexicographique.

Représenter les mots

On peut représenter les mots comme des nombres en constituant un lexique et en affectant à chaque item un nombre entier : sa position dans l'ordre lexicographique.

a	0
aa	1
abaca	2
abaissa	3
...	...

Représenter les mots

On peut représenter les mots comme des nombres en constituant un lexique et en affectant à chaque item un nombre entier : sa position dans l'ordre lexicographique.

a	0
aa	1
abaca	2
abaissa	3
...	...

Alors un mot, c'est juste un nombre et une suite de mots (phrase, tour de parole, document...) c'est juste une suite de nombres.

Est-ce que vous voyez des problèmes ?

Une affaire de topologie

Un problème assez problématique, c'est que ces représentations numériques sont purement orthographiques, ce qui n'est pas idéal.

Une affaire de topologie

Un problème assez problématique, c'est que ces représentations numériques sont purement orthographiques, ce qui n'est pas idéal.

Déjà, c'est un nid à problèmes liés à la standardisation.

Une affaire de topologie

Un problème assez problématique, c'est que ces représentations numériques sont purement orthographiques, ce qui n'est pas idéal.

Déjà, c'est un nid à problèmes liés à la standardisation.

Mais même si on en fait abstraction :

…	…
défiés	41 921
déflagration	41 922
…	…

Une affaire de topologie

Un problème assez problématique, c'est que ces représentations numériques sont purement orthographiques, ce qui n'est pas idéal.

Déjà, c'est un nid à problèmes liés à la standardisation.

Mais même si on en fait abstraction :

…	…
défiés	41 921
déflagration	41 922
…	…

On a des mots qui n'ont presque rien en commun, mais dont les représentations sont très proches.

Une affaire de topologie

Un problème assez problématique, c'est que ces représentations numériques sont purement orthographiques, ce qui n'est pas idéal.

Déjà, c'est un nid à problèmes liés à la standardisation.

Mais même si on en fait abstraction :

...	...
défiés	41 921
déflagration	41 922
...	...

On a des mots qui n'ont presque rien en commun, mais dont les représentations sont très proches.

Ça sera un problème pour beaucoup des algorithmes d'apprentissage.

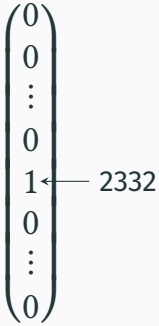
Une solution de mathématicien·ne

Il y a une façon de représenter un lexique à peine plus compliquée qui évite ce problème : un encodage « *one-hot* » : on représente chaque mot par un vecteur avec des zéros partout, sauf à sa position dans le lexique où on met un 1.

Une solution de mathématicien·ne

Il y a une façon de représenter un lexique à peine plus compliquée qui évite ce problème : un encodage « *one-hot* » : on représente chaque mot par un vecteur avec des zéros partout, sauf à sa position dans le lexique où on met un 1.

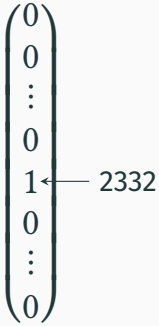
Par exemple « chat » est en position 2332, on va donc le représenter par



Une solution de mathématicien·ne

Il y a une façon de représenter un lexique à peine plus compliquée qui évite ce problème : un encodage « *one-hot* » : on représente chaque mot par un vecteur avec des zéros partout, sauf à sa position dans le lexique où on met un 1.

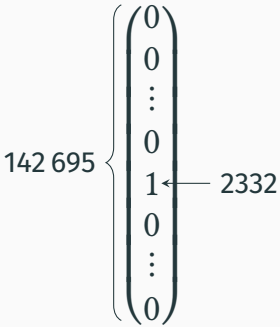
Par exemple « chat » est en position 2332, on va donc le représenter par



Une solution de mathématicien·ne

Il y a une façon de représenter un lexique à peine plus compliquée qui évite ce problème : un encodage « *one-hot* » : on représente chaque mot par un vecteur avec des zéros partout, sauf à sa position dans le lexique où on met un 1.

Par exemple « chat » est en position 2332, on va donc le représenter par



Une solution de mathématicien·ne

Dans cette configuration, quelle est la distance (euclidienne) entre deux mots ?

Une solution de mathématicien·ne

Dans cette configuration, quelle est la distance (euclidienne) entre deux mots ?

Est-ce que ça résout le problème précédent ?

Une solution de mathématicien·ne

Dans cette configuration, quelle est la distance (euclidienne) entre deux mots ?

Est-ce que ça résout le problème précédent ?

Est-ce que vous voyez d'autres problèmes avec cette représentation ?

Des idées pour faire mieux ?

Une solution de linguiste

Si on dispose d'un lexique annoté (au hasard JeuxDeMots (Lafourcade et Le Brun 2020)), on peut aller dans la direction carrément opposée : représenter chaque mot par une structure de traits linguistiques.

Une solution de linguiste

Si on dispose d'un lexique annoté (au hasard JeuxDeMots (Lafourcade et Le Brun 2020)), on peut aller dans la direction carrément opposée : représenter chaque mot par une structure de traits linguistiques.

$$\begin{pmatrix} \text{NOUN} \\ \text{singular} \\ \text{animated} \\ \vdots \end{pmatrix}$$

Une solution de linguiste

Si on dispose d'un lexique annoté (au hasard JeuxDeMots (Lafourcade et Le Brun 2020)), on peut aller dans la direction carrément opposée : représenter chaque mot par une structure de traits linguistiques.

$$\begin{pmatrix} \text{NOUN} \\ \text{singular} \\ \text{animated} \\ \vdots \end{pmatrix}$$

Est-ce que ça résout nos problèmes précédents ?

Une solution de linguiste

Si on dispose d'un lexique annoté (au hasard JeuxDeMots (Lafourcade et Le Brun 2020)), on peut aller dans la direction carrément opposée : représenter chaque mot par une structure de traits linguistiques.

$$\begin{pmatrix} \text{NOUN} \\ \text{singular} \\ \text{animated} \\ \vdots \end{pmatrix}$$

Est-ce que ça résout nos problèmes précédents ?

Est-ce que ça en créé d'autres ?

Une solution de linguiste *de corpus*

« *You shall know a word by the company it keeps.* »

(Firth 1957)

Une solution de linguiste *de corpus*

« *You shall know a word by the company it keeps.* »

(Firth 1957)

Voir par exemple Brunila et LaViolette (2022) pour une mise en contexte plutôt nécessaire.

Une solution de langouste

Bien entendu on va s'empresser d'oublier toute nuance sur cette citation et d'en tirer une idée pas forcément parfaite mais pratique.

Une solution de langouste

Bien entendu on va s'empresser d'oublier toute nuance sur cette citation et d'en tirer une idée pas forcément parfaite mais pratique.

On va faire l'hypothèse sauvage que des mots qui apparaissent dans des contextes similaires ont des propriétés similaires.

Une solution de langouste

Bien entendu on va s'empresser d'oublier toute nuance sur cette citation et d'en tirer une idée pas forcément parfaite mais pratique.

On va faire l'hypothèse sauvage que des mots qui apparaissent dans des contextes similaires ont des propriétés similaires.

Ça nous donne une solution opérante pour représenter des mots : on va les représenter par les contextes dans lesquels ils apparaissent.

Une solution de langouste

Bien entendu on va s'empresser d'oublier toute nuance sur cette citation et d'en tirer une idée pas forcément parfaite mais pratique.

On va faire l'hypothèse sauvage que des mots qui apparaissent dans des contextes similaires ont des propriétés similaires.

Ça nous donne une solution opérante pour représenter des mots : on va les représenter par les contextes dans lesquels ils apparaissent.

En général : leurs fréquences de cooccurrence avec chacun des mots d'un lexique dans un grand corpus.

Cooccurrences

Représenter un mot par des fréquences de cooccurrences, ça donne toujours des vecteurs de la taille du lexique, mais :

Cooccurrences

Représenter un mot par des fréquences de cooccurrences, ça donne toujours des vecteurs de la taille du lexique, mais :

- Beaucoup moins creux (moins de zéros).

Cooccurrences

Représenter un mot par des fréquences de cooccurrences, ça donne toujours des vecteurs de la taille du lexique, mais :

- Beaucoup moins creux (moins de zéros).
- Dont la topologie, l'organisation spatiale, a plus d'intérêt.

Cooccurrences

Représenter un mot par des fréquences de cooccurrences, ça donne toujours des vecteurs de la taille du lexique, mais :

- Beaucoup moins creux (moins de zéros).
- Dont la topologie, l'organisation spatiale, a plus d'intérêt.

Cooccurrences

Représenter un mot par des fréquences de cooccurrences, ça donne toujours des vecteurs de la taille du lexique, mais :

- Beaucoup moins creux (moins de zéros).
- Dont la topologie, l'organisation spatiale, a plus d'intérêt.

La grande dimension est toujours un problème, mais il y a des techniques pour la réduire, compresser les vecteurs en gardant, voire améliorant l'intérêt de leur géométrie.

Quel rapport avec l’apprentissage automatique ?

Apprendre des représentations

On a ici appris des représentations : on a exploité des régularités repérées dans un échantillon de données.

Apprendre des représentations

On a ici appris des représentations : on a exploité des régularités repérées dans un échantillon de données.

On l'a fait ici avec un mécanisme très *ad hoc*, mais il y a d'autres solutions.

Apprendre des représentations

On a ici appris des représentations : on a exploité des régularités repérées dans un échantillon de données.

On l'a fait ici avec un mécanisme très *ad hoc*, mais il y a d'autres solutions.

Notamment la plus populaire et celle qui va nous préoccuper maintenant : utiliser des représentations apprises pour une tâche auxiliaire pour laquelle les données abondent.

En fait, ce qui se passe dans beaucoup de modèles d'apprentissage, ça ressemble à

En fait, ce qui se passe dans beaucoup de modèles d'apprentissage, ça ressemble à

Le modèle apprend à la fois une transformation des entrées en des représentations intermédiaires pertinentes pour lui et à déterminer une sortie à partir de ces représentations.

En fait, ce qui se passe dans beaucoup de modèles d'apprentissage, ça ressemble à

Le modèle apprend à la fois une transformation des entrées en des représentations intermédiaires pertinentes pour lui et à déterminer une sortie à partir de ces représentations.

Si on applique un tel modèle à une tâche qui demande des représentations intéressantes et pour laquelle on a facilement des données, on pourra alors extraire ces représentations et les utiliser pour d'autres tâches.

⚠️

Évidemment ce que vous obtenez dépend très fortement de vos données et de la tâche.

⚠️

Évidemment ce que vous obtenez dépend très fortement de vos données et de la tâche.

Les données faciles à obtenir c'est bien pratique, mais c'est aussi dangereux (on en a déjà parlé, on en reparlera etc).

Est-ce que le serpent ne se mord pas un peu la queue ?

Est-ce que le serpent ne se mord pas un peu la queue ?

Pour cette tâche auxiliaire, on prend quoi comme représentation de départ ?

Est-ce que le serpent ne se mord pas un peu la queue ?

Pour cette tâche auxiliaire, on prend quoi comme représentation de départ ?

En fait on peut prendre un peu ce qu'on veut tant qu'on perd pas d'information : on est de toute façon pas très intéressé·e par le modèle de la tâche auxiliaire en soi. Tout ce qu'on veut, c'est qu'elle apprenne des bonnes représentations intermédiaires.

Est-ce que le serpent ne se mord pas un peu la queue ?

Pour cette tâche auxiliaire, on prend quoi comme représentation de départ ?

En fait on peut prendre un peu ce qu'on veut tant qu'on perd pas d'information : on est de toute façon pas très intéressé·e par le modèle de la tâche auxiliaire en soi. Tout ce qu'on veut, c'est qu'elle apprenne des bonnes représentations intermédiaires.

En pratique : souvent des vecteurs *one-hot* pour les mots. On pourrait faire du bootstrap, mais à ma connaissance ça ne se fait pas vraiment en pratique.

Tâches auxiliaires

Quelles tâches ?

Tâches auxiliaires

Quelles tâches ?

- Suffisamment complexe.

Tâches auxiliaires

Quelles tâches ?

- Suffisamment complexe.
- Facilement beaucoup de données.

Tâches auxiliaires

Quelles tâches ?

- Suffisamment complexe.
- Facilement beaucoup de données.

Tâches auxiliaires

Quelles tâches ?

- Suffisamment complexe.
- Facilement beaucoup de données.

Le plus populaire en TAL, ce sont des formes de « modèles de langue ».

Tâches auxiliaires

Quelles tâches ?

- Suffisamment complexe.
- Facilement beaucoup de données.

Le plus populaire en TAL, ce sont des formes de « modèles de langue ».

Par exemple génératif : « Le petit chat est ___ » ← quelle est la proba pour chacun des mots du vocabulaire de continuer la phrase ?

Tâches auxiliaires

Quelles tâches ?

- Suffisamment complexe.
- Facilement beaucoup de données.

Le plus populaire en TAL, ce sont des formes de « modèles de langue ».

Par exemple génératif : « Le petit chat est ___ » ← quelle est la proba pour chacun des mots du vocabulaire de continuer la phrase ?

C'est historiquement ce que fait Bengio et al. (2006).

Tâches auxiliaires

Quelles tâches ?

- Suffisamment complexe.
- Facilement beaucoup de données.

Le plus populaire en TAL, ce sont des formes de « modèles de langue ».

Par exemple génératif : « Le petit chat est ___ » ← quelle est la proba pour chacun des mots du vocabulaire de continuer la phrase ?

C'est historiquement ce que fait Bengio et al. (2006).

C'est ce que fait Mikolov et al. (2013) avec plusieurs modèles de vraisemblance partielles (en regardant en biais).

Ces fameux *embeddings*

Ce que Bengio et al. (2006), Mikolov et al. (2013) et les autres obtiennent ce sont des représentations vectorielles de mots.

Ces fameux *embeddings*

Ce que Bengio et al. (2006), Mikolov et al. (2013) et les autres obtiennent ce sont des représentations vectorielles de ~~mots~~ formes orthographiques.

Ces fameux *embeddings*

Ce que Bengio et al. (2006), Mikolov et al. (2013) et les autres obtiennent ce sont des représentations vectorielles de ~~mots~~ formes orthographiques.

Fondamentalement, on a un dictionnaire qui associe à chaque forme orthographique un vecteur qui contient des ~~trucs~~ informations sur son usage observé dans un corpus.

Ces fameux *embeddings*

Ce que Bengio et al. (2006), Mikolov et al. (2013) et les autres obtiennent ce sont des représentations vectorielles de ~~mots~~ formes orthographiques.

Fondamentalement, on a un dictionnaire qui associe à chaque forme orthographique un vecteur qui contient des ~~trucs~~ informations sur son usage observé dans un corpus.

On les appelle embeddings (ou *plongements*) pour des raisons. Ce sont des vecteurs denses (peu de zéros), de dimensions très inférieures à la taille du lexique. Tout ce que les algos d'apprentissage (en particulier les réseaux de neurones) aiment bien.

Ces fameux *embeddings*

Ce que Bengio et al. (2006), Mikolov et al. (2013) et les autres obtiennent ce sont des représentations vectorielles de ~~mots~~ formes orthographiques.

Fondamentalement, on a un dictionnaire qui associe à chaque forme orthographique un vecteur qui contient des ~~trucs~~ informations sur son usage observé dans un corpus.

On les appelle embeddings (ou *plongements*) pour des raisons. Ce sont des vecteurs denses (peu de zéros), de dimensions très inférieures à la taille du lexique. Tout ce que les algos d'apprentissage (en particulier les réseaux de neurones) aiment bien.

C'est difficile de les évaluer intrinsèquement (même si Mikolov et al. propose des idées sur les liens géométrie-sémantique), mais on observe en pratique que pré-entraîner ainsi des représentations aide pour des tâches en aval.

La tuyauterie des données

Si on revient à notre objectif initial

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

Une solution très orthodoxe : un problème d'apprentissage ?

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

Une solution très orthodoxe : un problème d'apprentissage ? Résolvons-le avec deux fois plus d'apprentissage :

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

Une solution très orthodoxe : un problème d'apprentissage ? Résolvons-le avec deux fois plus d'apprentissage :

- Une première phase exploite des régularités sur des données qui n'ont pas été annotées pour cette tâche. Nombreuses, « bon marché » et peu spécifiques

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

Une solution très orthodoxe : un problème d'apprentissage ? Résolvons-le avec deux fois plus d'apprentissage :

- Une première phase exploite des régularités sur des données qui n'ont pas été annotées pour cette tâche. Nombreuses, « bon marché » et peu spécifiques

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

Une solution très orthodoxe : un problème d'apprentissage ? Résolvons-le avec deux fois plus d'apprentissage :

- Une première phase exploite des régularités sur des données qui n'ont pas été annotées pour cette tâche. Nombreuses, « bon marché » et peu spécifiques : par exemple un corpus de texte brut genre OSCAR (Ortiz Suárez et al. 2019) et en apprend ce faisant des représentations.
- La deuxième apprend des corrélations entre ces représentations et des informations contenues dans des données annotées spécifiquement

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

Une solution très orthodoxe : un problème d'apprentissage ? Résolvons-le avec deux fois plus d'apprentissage :

- Une première phase exploite des régularités sur des données qui n'ont pas été annotées pour cette tâche. Nombreuses, « bon marché » et peu spécifiques : par exemple un corpus de texte brut genre OSCAR (Ortiz Suárez et al. 2019) et en apprend ce faisant des représentations.
- La deuxième apprend des corrélations entre ces représentations et des informations contenues dans des données annotées spécifiquement

La tuyauterie des données

Si on revient à notre objectif initial qui était ?

Quelles représentations pour nos données dans un algo d'apprentissage pour une tâche précise ?

Une solution très orthodoxe : un problème d'apprentissage ? Résolvons-le avec deux fois plus d'apprentissage :

- Une première phase exploite des régularités sur des données qui n'ont pas été annotées pour cette tâche. Nombreuses, « bon marché » et peu spécifiques : par exemple un corpus de texte brut genre OSCAR (Ortiz Suárez et al. 2019) et en apprend ce faisant des représentations.
- La deuxième apprend des corrélations entre ces représentations et des informations contenues dans des données annotées spécifiquement : par exemple un treebank comme Sequoia (Candito et Seddah 2012).

Opinions?

Une note

Si pour plusieurs tâches aval on a l'intention d'utiliser le même pré-entraînement de représentations, on est pas obligé de recommencer à chaque fois !

Une note

Si pour plusieurs tâches aval on a l'intention d'utiliser le même pré-entraînement de représentations, on est pas obligé de recommencer à chaque fois !

On a donc potentiellement des représentations polyvalentes.

Une note

Si pour plusieurs tâches aval on a l'intention d'utiliser le même pré-entraînement de représentations, on est pas obligé de recommencer à chaque fois !

On a donc potentiellement des représentations polyvalentes.

Et ça permet de faire des économies assez importantes.

Encore des problèmes

Ces représentations de formes, ça marche ~~bien~~ pas si pire pour beaucoup de choses, mais elles ont défaut majeur.

Encore des problèmes

Ces représentations de formes, ça marche ~~bien~~ pas si pire pour beaucoup de choses, mais elles ont défaut majeur.

Ce sont des représentations de formes.

Encore des problèmes

Ces représentations de formes, ça marche ~~bien~~ pas si pire pour beaucoup de choses, mais elles ont défaut majeur.

Ce sont des représentations de formes.

Les homo(graphes|nymes) ça existe.

Encore des problèmes

Ces représentations de formes, ça marche ~~bien~~ pas si pire pour beaucoup de choses, mais elles ont défaut majeur.

Ce sont des représentations de formes.

Les homo(graphes|nymes) ça existe.

Comment faire pour que deux formes identiques de lexèmes différents, ou issues d'un syncrétisme morphologique d'un même lexème aient des représentations différentes ?

Désambigüiser

Peut-être on pourrait au préalable désambigüiser ?

Désambigüiser

Peut-être on pourrait au préalable désambigüiser ?

Comment ?

Désambigüiser

Peut-être on pourrait au préalable désambigüiser ?

Comment ?

Avec un modèle appris ? 👉👈

Désambigüiser

Le plan ça pourrait être :

1. On apprend des représentations de formes

Désambigüiser

Le plan ça pourrait être :

1. On apprend des représentations de formes
2. On les utilise comme entrées dans un modèle qui saurait désambigüiser des formes

Désambigüiser

Le plan ça pourrait être :

1. On apprend des représentations de formes
2. On les utilise comme entrées dans un modèle qui saurait désambigüiser des formes
3. Le modèle qui réalise notre tâche cible finale a donc accès à des représentations des formes, sans les ambigüités

Désambigüiser

Le plan ça pourrait être :

1. On apprend des représentations de formes
2. On les utilise comme entrées dans un modèle qui saurait désambigüiser des formes
3. Le modèle qui réalise notre tâche cible finale a donc accès à des représentations des formes, sans les ambigüités

Désambigüiser

Le plan ça pourrait être :

1. On apprend des représentations de formes
2. On les utilise comme entrées dans un modèle qui saurait désambigüiser des formes
3. Le modèle qui réalise notre tâche cible finale a donc accès à des représentations des formes, sans les ambigüités

Est-ce que ça vous donner une idée lumineuse ?

Wavelength collapse

Un modèle de désambigüisation va désambigüiser en utilisant des représentations des formes en contexte.

Wavelength collapse

Un modèle de désambigüisation va désambigüiser en utilisant des représentations des formes en contexte.

Il va donc y avoir dedans des représentations de chacune des formes dans son contexte.

Wavelength collapse

Un modèle de désambigüisation va désambigüiser en utilisant des représentations des formes en contexte.

Il va donc y avoir dedans des représentations de chacune des formes dans son contexte.

C'est encore mieux qu'une désambigüisation arbitraire : si on a de la chance ça pourrait capturer des nuances qu'on a pas annoté explicitement.

Wavelength collapse

Un modèle de désambigüisation va désambigüiser en utilisant des représentations des formes en contexte.

Il va donc y avoir dedans des représentations de chacune des formes dans son contexte.

C'est encore mieux qu'une désambigüisation arbitraire : si on a de la chance ça pourrait capturer des nuances qu'on a pas annoté explicitement.

Le plan ça pourrait donc être de prendre directement ces représentations : quand on nous donne une entrée, on la passe au modèle de désambigüisation, on intercepte ses représentations contextuelles internes et on se sert de ça comme entrée pour notre tâche cible.

Wavelength collapse

Un modèle de désambigüisation va désambigüiser en utilisant des représentations des formes en contexte.

Il va donc y avoir dedans des représentations de chacune des formes dans son contexte.

C'est encore mieux qu'une désambigüisation arbitraire : si on a de la chance ça pourrait capturer des nuances qu'on a pas annoté explicitement.

Le plan ça pourrait donc être de prendre directement ces représentations : quand on nous donne une entrée, on la passe au modèle de désambigüisation, on intercepte ses représentations contextuelles internes et on se sert de ça comme entrée pour notre tâche cible.

Et qu'est-ce que vous pensez de ce qui se passe dans un modèle de langue ?

Wavelength collapse

Un très bon modèle de langue devrait contenir aussi des représentations en contexte

Wavelength collapse

Un très bon modèle de langue devrait contenir aussi des représentations en contexte

Et il doit probablement faire une forme de désambigüisation.

Wavelength collapse

Un très bon modèle de langue devrait contenir aussi des représentations en contexte

Et il doit probablement faire une forme de désambigüisation.

Alors pourquoi ne pas shunter la désambigüisation et utiliser directement ces représentations contextuelles-là ?

Wavelength collapse

Un très bon modèle de langue devrait contenir aussi des représentations en contexte

Et il doit probablement faire une forme de désambigüisation.

Alors pourquoi ne pas shunter la désambigüisation et utiliser directement ces représentations contextuelles-là ?

C'est exactement ce qui s'est produit avec quasi-simultanément Devlin et al. (2019), Howard et Ruder (2018), Peters et al. (2018) et Radford et al. (2018)

(BERT, ULMFit, ELMo, GPT : diverses formes d'entraînement de représentations à partir de formes de modèles de langue)

Killing the game

Les représentations issues desdits modèles ont pris d'assaut le TAL, la linguistique informatique, les esprits et la société.

Killing the game

Les représentations issues desdits modèles ont pris d'assaut le TAL, la linguistique informatique, les esprits et la société.

Il faut dire qu'elles apportent des gains impressionnants de performances (lesquelles ?)

Killing the game

Les représentations issues desdits modèles ont pris d'assaut le TAL, la linguistique informatique, les esprits et la société.

Il faut dire qu'elles apportent des gains impressionnants de performances (lesquelles ?) à un coût computationnel défiant toute concurrence

Killing the game

Les représentations issues desdits modèles ont pris d'assaut le TAL, la linguistique informatique, les esprits et la société.

Il faut dire qu'elles apportent des gains impressionnants de performances (lesquelles ?) à un coût computationnel défiant toute concurrence (monstrueux).

Killing the game

Les représentations issues desdits modèles ont pris d'assaut le TAL, la linguistique informatique, les esprits et la société.

Il faut dire qu'elles apportent des gains impressionnants de performances (lesquelles ?) à un coût computationnel défiant toute concurrence (monstrueux). Mais peut-être acceptable (???) selon ce qu'on veut faire.

Killing the game

Les représentations issues desdits modèles ont pris d'assaut le TAL, la linguistique informatique, les esprits et la société.

Il faut dire qu'elles apportent des gains impressionnants de performances (lesquelles ?) à un coût computationnel défiant toute concurrence (monstrueux). Mais peut-être acceptable (???) selon ce qu'on veut faire.

De fait, ce qui se passe : quand on veut une représentation vectorielle d'un mot (dans un contexte) on a plus seulement à aller la chercher dans un dictionnaire, mais on doit à chaque fois faire passer tout le contexte dans un modèle qui peut être très lourd.

Transfert

Si on fait un pas en arrière sur ce dont on vient de parler, du point de vue de l'apprentissage :

Transfert

Si on fait un pas en arrière sur ce dont on vient de parler, du point de vue de l'apprentissage :

- On a appris un modèle pour une certaine tâche.

Transfert

Si on fait un pas en arrière sur ce dont on vient de parler, du point de vue de l'apprentissage :

- On a appris un modèle pour une certaine tâche.
- Puis on prend ce modèle, on lui ajoute une surcouche et on entraîne l'ensemble sur une nouvelle tâche.

Transfert

Si on fait un pas en arrière sur ce dont on vient de parler, du point de vue de l'apprentissage :

- On a appris un modèle pour une certaine tâche.
- Puis on prend ce modèle, on lui ajoute une surcouche et on entraîne l'ensemble sur une nouvelle tâche.
- D'une certaine façon, on a transféré des connaissances d'une tâche vers une autre.

Cette idée est parallèle à celle de l'apprentissage par transfert en général.

Transfer learning

Cette idée est parallèle à celle de l'apprentissage par transfert en général.

Pour le dire vite : si on a un modèle qui marche bien pour un contexte X, c'est peut-être moins coûteux et plus efficace de l'« adapter » pour un contexte Y que de faire un modèle pour Y à partir de zéro.

Transfer learning

Cette idée est parallèle à celle de l'apprentissage par transfert en général.

Pour le dire vite : si on a un modèle qui marche bien pour un contexte X, c'est peut-être moins coûteux et plus efficace de l'« adapter » pour un contexte Y que de faire un modèle pour Y à partir de zéro.

Ça a été essayé pour plein de combinaisons de X et Y (Ruder (2019) pour un bon historique).

- D'une langue à une autre.
- Entre domaines de spécialité.
- Entre tâches proches (POS/parsing/...)

Adaptation

Adapter un modèle concrètement ça peut être

Adaptation

Adapter un modèle concrètement ça peut être

- Juste prendre le modèle de base et lui donner de nouvelles données.

Adaptation

Adapter un modèle concrètement ça peut être

- Juste prendre le modèle de base et lui donner de nouvelles données.
  → Faisable aussi pour le modèle de représentation !

Adaptation

Adapter un modèle concrètement ça peut être

- Juste prendre le modèle de base et lui donner de nouvelles données.
  → Faisable aussi pour le modèle de représentation !
- Ajouter des sur-/sous-/inter-couches dans le modèle qui gèrent l'adaptation :

Adaptation

Adapter un modèle concrètement ça peut être

- Juste prendre le modèle de base et lui donner de nouvelles données.
  → Faisable aussi pour le modèle de représentation !
- Ajouter des sur-/sous-/inter-couches dans le modèle qui gèrent l'adaptation :
  → En continuant à entraîner les paramètres du modèle original (« *fine-tuning* »).

Adaptation

Adapter un modèle concrètement ça peut être

- Juste prendre le modèle de base et lui donner de nouvelles données.
  - → Faisable aussi pour le modèle de représentation !
- Ajouter des sur-/sous-/inter-couches dans le modèle qui gèrent l'adaptation :
  - → En continuant à entraîner les paramètres du modèle original (« *fine-tuning* »).
  - → En fixant les paramètres du modèle original et juste entraîner les nouvelles parties.

Adaptation

Adapter un modèle concrètement ça peut être

- Juste prendre le modèle de base et lui donner de nouvelles données.
  - → Faisable aussi pour le modèle de représentation !
- Ajouter des sur-/sous-/inter-couches dans le modèle qui gèrent l'adaptation :
  - → En continuant à entraîner les paramètres du modèle original (« *fine-tuning* »).
  - → En fixant les paramètres du modèle original et juste entraîner les nouvelles parties.

Adaptation

Adapter un modèle concrètement ça peut être

- Juste prendre le modèle de base et lui donner de nouvelles données.
  - → Faisable aussi pour le modèle de représentation !
- Ajouter des sur-/sous-/inter-couches dans le modèle qui gèrent l'adaptation :
  - → En continuant à entraîner les paramètres du modèle original (« *fine-tuning* »).
  - → En fixant les paramètres du modèle original et juste entraîner les nouvelles parties.

On peut aussi envisager plusieurs entraînements simultanés, en multi-tâches (en partageant des représentations), c'est d'ailleurs déjà ce que faisait Bengio et al. (2006).

Cas pratique : détection des chaînes de coréférences

System	MUC	B^3	$CEAF_e$	CoNLL	BLANC
No pretraining	62.15	81.24	81.29	74.89	69.50
frELMo	67.06	82.53	83.56	77.72	71.74
mBERT	64.02	81.67	82.40	76.03	70.42
CamemBERT	67.32	82.53	83.63	77.83	71.96
frELMo+str	**72.50**	84.24	86.21	**80.98**	**74.15**
CamemBERT+str	72.34	**84.42**	**86.37**	**81.04**	**74.24**

Cas pratique : analyse syntaxique de l'ancien français

Model	UPOS	UAS	LAS
SotA	96.26	91.83	86.75
mBERT	96.19	92.03	87.52
BERTrade	96.60	92.20	87.95
mBERT+OF	97.11	93.86	90.37
FlauBERT+OF	97.15	93.96	90.57

Cas pratique : traduction Breton→Français

Model	BLEU	ChrF++	TER
Apertium	24.15	50.23	63.93
m2m100-418M	0.58	11.85	114.49
+OPUS	30.01	50.16	55.37
+ARBRES	37.68	56.99	48.65

Appendix

References i

Bengio, Yoshua, Holger Schwenk, Jean-Sébastien Senécal, Fréderic Morin et Jean-Luc Gauvain (2006). **« Neural Probabilistic Language Models ».**
In : *Innovations in Machine Learning : Theory and Applications*.
Berlin, Heidelberg : Springer Berlin Heidelberg.
URL : https://doi.org/10.1007/3-540-33486-6_6.

Brunila, Mikael et Jack LaViolette (juill. 2022). **« What Company Do Words Keep ? Revisiting the Distributional Semantics of J.R. Firth & Zellig Harris ».**
In : *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*.
NAACL-HLT 2022. Seattle, United States : Association for Computational Linguistics.
URL : https://aclanthology.org/2022.naacl-main.327.

References ii

Candito, Marie et Djamé Seddah (juin 2012). **« Le corpus Sequoia : annotation syntaxique et exploitation pour l'adaptation d'analyseur par pont lexical ».**
In : *Actes de la conférence conjointe JEP-TALN-RECITAL 2012.*
TALN 2012 (Grenoble, France). T. 2.
Association pour le Traitement Automatique des Langues.
URL : https://hal.inria.fr/hal-00698938.

Devlin, Jacob, Ming-Wei Chang, Kenton Lee et Kristina Toutanova (juin 2019). **« BERT : Pre-training of Deep Bidirectional Transformers for Language Understanding ».**
In : *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies.*
NAACL-HLT 2019 (Minneapolis, Minnesota, USA).
Association for Computational Linguistics.
URL : https://www.aclweb.org/anthology/N19-1423.

References iii

Firth, John R. (1957). **« A Synopsis of Linguistic Theory, 1930-1955 ».**
In : *Studies in Linguistic Analysis*. Blackwell.
URL : /paper/A-Synopsis-of-Linguistic-Theory%2C-1930-1955-Firth/88b3959b6f5333e5358eac43970a5fa29b54642c.

Howard, Jeremy et Sebastian Ruder (juill. 2018).
« Universal Language Model Fine-tuning for Text Classification ».
In : *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*. ACL 2018 (Melbourne, Australia).
Association for Computational Linguistics.
URL : https://www.aclweb.org/anthology/P18-1031.

Lafourcade, Mathieu et Nathalie Le Brun (1er déc. 2020). **« JeuxDeMots : Un réseau lexico-sémantique pour le français, issu de jeux et d'inférences ».**
In : *Lexique. Revue en Sciences du Langage* 27.
URL : http://www.peren-revues.fr/lexique/773.

References iv

Mikolov, Tomáš, Kai Chen, Greg Corrado et Jeffrey Dean (2 mai 2013).
« Efficient Estimation of Word Representations in Vector Space ». In : 1st International Conference on Learning Representations. Scottsdale, Arizona, USA.
URL : http://arxiv.org/abs/1301.3781.

Ortiz Suárez, Pedro Javier, Benoît Sagot et Laurent Romary (juill. 2019).
« Asynchronous Pipeline for Processing Huge Corpora on Medium to Low Resource Infrastructures ».
In : *7th Workshop on the Challenges in the Management of Large Corpora*. CMLCè7 (Caerdydd, Cymru, Deyrnas Unedig). Leibniz-Institut für Deutsche Sprache.
URL : https://hal.inria.fr/hal-02148693.

References

Peters, Matthew et al. (juin 2018).
« Deep Contextualized Word Representations ».
In : *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics : Human Language Technologies*.
NAACL : HLT 2018 (New Orleans, Louisiana, USA). T. 1.
Association for Computational Linguistics.
URL : http://aclweb.org/anthology/N18-1202.

Radford, Alec, Karthik Narasimhan, Tim Salimans et Ilya Sutskever (11 juin 2018).
***Improving Language Understanding by Generative Pre-Training*.**
Technical report. OpenAI.
URL : https://openai.com/blog/language-unsupervised/.

Ruder, Sebastian (2019).
« Neural Transfer Learning for Natural Language Processing ».
Thèse de doct. Gaillimh, Éire : Ollscoil na hÉireann. URL : http://ruder.io/thesis/.

Licence