

Overton

Le ML goût pomme

NMLM du 7 octobre 2019

Overview

Overview

Quand / Quoi / Où

Sortie

Début septembre

Conf

NeurIPS

But

Gestion du cycle software ML

Maturité

Production

Overview

Défis à relever

- Monitoring précis
- Pipelines complexes
- Feedback loop fonctionnelle

Overview

Choix architecturaux

- Apprentissage profond sans code
- Apprentissage multi-tâches
- Supervision faible

Apprentissage
sans code

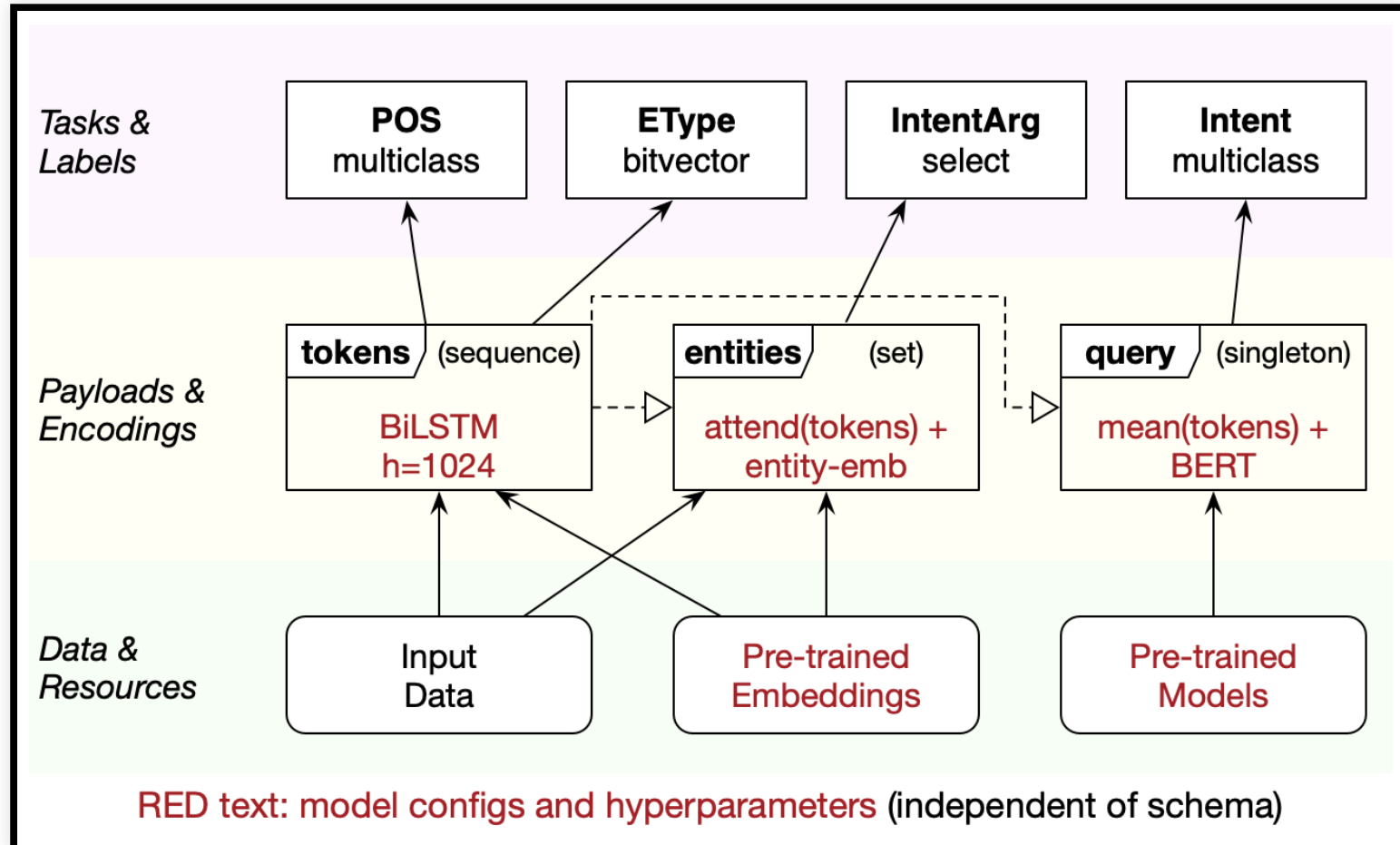
Apprentissage sans code

Principes

- Modèles & code d'entraînement = experts
- Utilisation « black box » par les ingénieurs

Apprentissage sans code

Modularité



Apprentissage sans code

Configuration

Schema

```
{
  "payloads": {
    "tokens": {
      "type": "sequence",
      "max_length": 16
    },
    "query": {
      "type": "singleton",
      "base": ["tokens"]
    },
    "entities": {
      "type": "set",
      "range": "tokens"
    }
  },
  "tasks": {
    "POS": {
      "payload": "tokens",
      "type": "multiclass"
    },
    "EntityType": {
      "payload": "tokens",
      "type": "bitvector"
    },
    "Intent": {
      "payload": "query",
      "type": "multiclass"
    },
    "IntentArg": {
      "payload": "entities",
      "type": "select"
    }
  }
}
```

Example Data Record

```
{
  "payloads": {
    "tokens": ["How", "tall", ...],
    "query": "How tall is the president of the united states",
    "entities": {
      0: {"id": "President_(title)", range: [4, 5]},
      1: {"id": "United_States", range: [6, 9]},
      2: {"id": "U.S._state", range: [8, 9]},
      ...
    }
  },
  "tasks": {
    "POS": {
      "spacy": ["ADV", "ADJ", "VERB", ...]
    },
    "EntityType": {
      "eproj": [[], ..., ["location", "country"]]
    },
    "Intent": {
      "weak1": "President",
      "weak2": "Height",
      "crowd": "Height"
    },
    "IntentArg": {
      "weak1": 2,
      "weak2": 0,
      "crowd": 1
    }
  }
}
```



Model Tuning

```
{
  "tokens": {
    "embedding": [
      "GLOV-300",
      "BERT",
      "XLNet"
    ],
    "encoder": [
      "LSTM",
      "BERT",
      "XLNet"
    ],
    "size": [
      256, 768, 1024
    ]
  },
  "query": {
    "agg": [
      "max", "mean"
    ]
  },
  "entities": {
    "embedding": [
      "wiki-256",
      "combo-512",
    ],
    "attention": [
      "128x4", "256x8"
    ]
  }
}
```



Granularité fine
et multi-tâches

Granularité fine et multi-tâches

Problèmes

- Beaucoup de sous-tâches (implicites et explicites)
- Besoin de les surveiller
- Besoin de les améliorer

Granularité fine et multi-tâches

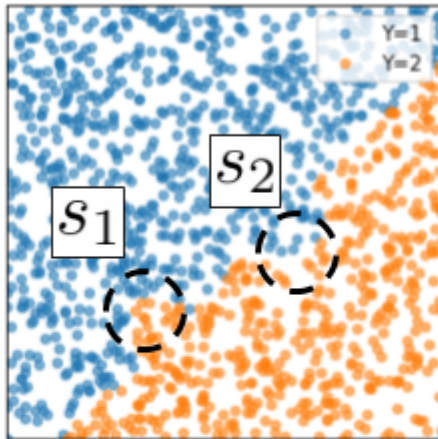
Approche

Slice-based learning

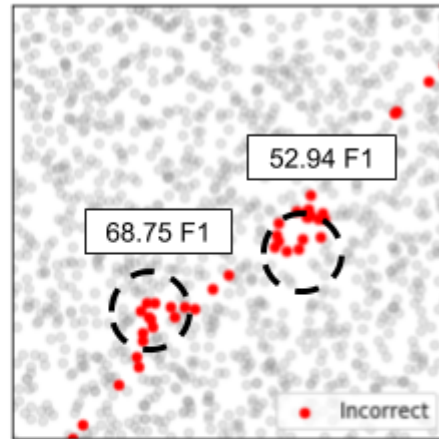
- Définition de sous-ensembles de données
- Augmentation de la capacité du modèle
- Métriques dédiées

Granularité fine et multi-tâches

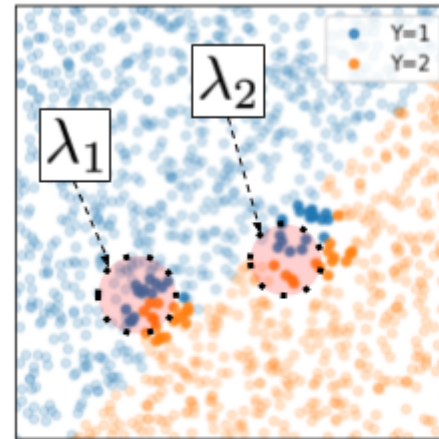
Slice-based learning



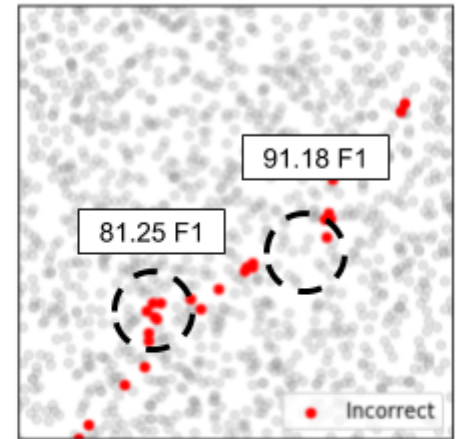
(a) Synthetic data with critical slices (dashed)



(b) Vanilla model errors



(c) User heuristically targets slices SFs (dotted)



(d) Slice-aware model errors

Granularité fine et multi-tâches

Définition de sous-ensembles de données critiques

Par des « slice functions » :

```
def sf_bike(x):  
    return "bike" in object_detector(x)  
  
def sf_night(x):  
    return avg(X.pixels.intensity) < 0.3
```

Granularité fine et multi-tâches

Experts de slice

Pour chaque slice, un expert :

- Capacité additionnelle du modèle
- Doit savoir quand s'activer
- A des métriques dédiées

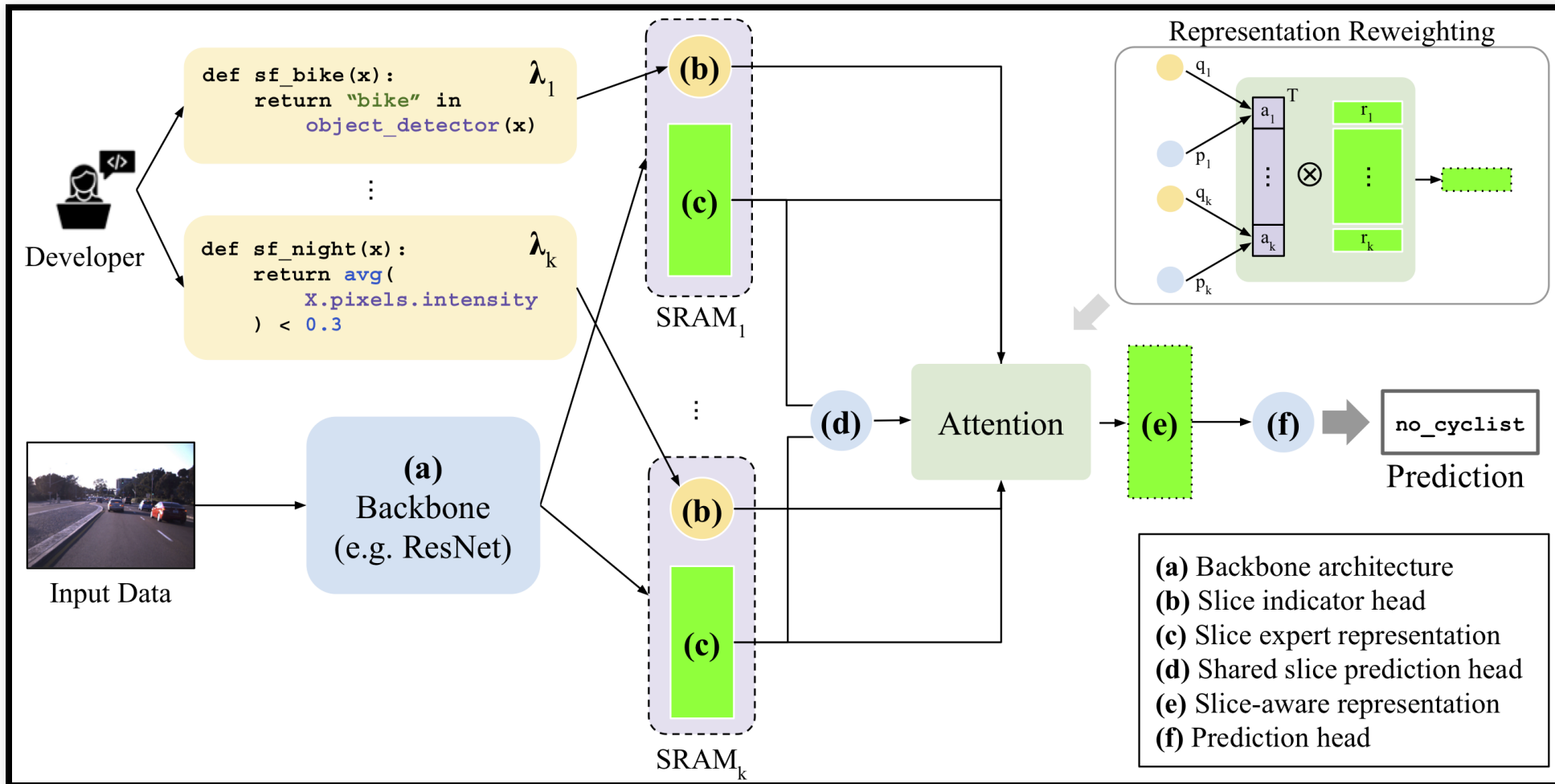
Granularité fine et multi-tâches

Points périlleux

- Bruit : les slices sont définies par des heuristiques
- Scale 1 : quand le nombre de slices augmente, performance du modèle ?
- Scale 2 : quand le nombre de slices augmente, qualité du modèle ?

Granularité fine et multi-tâches

Solution proposée



Ressources

- Blog snorkel « Slice-based Learning »
- Papier NeurIPS « Slice-based Learning: A Programming Model for Residual Learning in Critical Data Slices »

Supervision
faible

Supervision faible

Problématique

Besoin de données, mais :

- Coûte cher, pas dispo, yadda yadda
- Surtout pour certains cas rares
- Données bruitées

Supervision faible

Solution

Utiliser des heuristiques

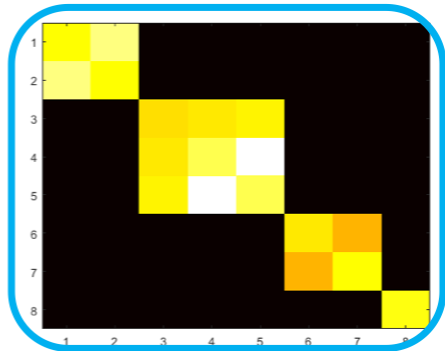
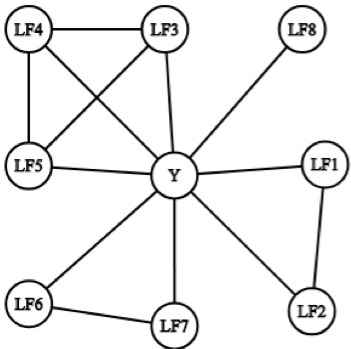
```
@labeling_function()  
def lf_regex_check_out(x):  
    """Spam comments say 'check out my video', 'check it out', et  
    return SPAM if re.search(r"check.*out", x.text, flags=re.I) e
```

Supervision faible

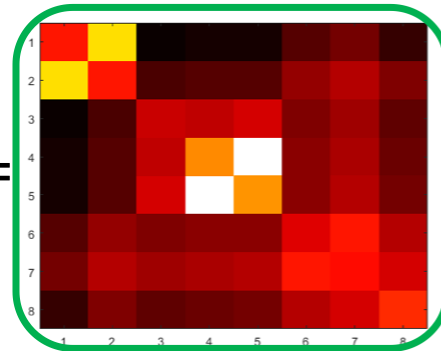
Solution

Puis corriger leurs corrélations dans la loss

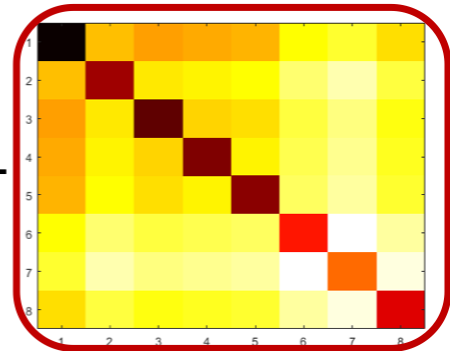
$$(\Sigma^{-1})_o = \Sigma_o^{-1} + zz^T$$



=



+



Granularité fine et multi-tâches

Implémentation

Overton utilise une variante du « Label Model » de Snorkel.

Ressources

- Blog snorkel « Introducing the New Snorkel »
- Papier ICML « Learning Dependency Structures for Weak Supervision Models »

Conclusion

- Modularité (comme AllenNLP, tensor2tensor)
- Métriques spécialisées et multi-tâches
- Correction des sous-ensembles importants
- Données artificielles corrigées

Questions / Discussion