Identification de partenariats entre acteurs économiques

Azim Makboulhoussen

*OpenClassrooms*

*Projet 8 – Parcours Data Scientist*

*05 Août 2018*



**Table des matières**

[I. Introduction 3](#_Toc521413321)

[II. Présentation de Snorkel 3](#_Toc521413322)

[1. Quel est l’objectif de Snorkel ? 3](#_Toc521413323)

[2. Comment fonctionne Snorkel ? 4](#_Toc521413324)

[3. Comment utiliser Snorkel ? 7](#_Toc521413325)

[III. Mise en application de Snorkel 7](#_Toc521413326)

[1. Présentation du cas testé 7](#_Toc521413327)

[2. Les différentes étapes de l’implémentation 8](#_Toc521413328)

[3. Évaluation des résultats 10](#_Toc521413329)

[IV. Conclusion 11](#_Toc521413330)

# Introduction

Les algorithmes de Machine Learning connaissent un succès indéniable. Néanmoins pour être efficaces, ces algorithmes ont besoin d’apprendre sur des quantité très importante de données.

Le volume de données et d’information ne cesse d’augmenter de nos jours cependant une grande partie de ces donnée n’est pas exploitable directement par les machines car non structurée. Pour appliquer l’apprentissage machine, il faut alors réaliser un travail minutieux et demandant souvent de l’expertise du domaine pour préparer ces données dit d’entrainement. Ce travail complexe et long est souvent un frein à l’utilisation de l’apprentissage machine.

Afin de répondre à cette contrainte, une équipe de recherche de l’université de Stanford a développé la solution Snorkel, un système de génération de données d’entrainement.

Dans le cadre du projet 8 du parcours Data Scientist, nous avons choisi de faire une veille thématique sur cette solution.

Dans ce document, nous commencerons par expliquer le principe de Snorkel puis nous le mettront en pratique pour prédire des relations de partenariat économique d’entreprise depuis des articles de presse.

# Présentation de Snorkel

## Quel est l’objectif de Snorkel ?

De plus en plus de données sont collectés mais près de 80% de celles-ci ne sont pas exploitées aujourd’hui car elles sont stockées dans des formats difficilement lisibles par les machines. Ces données sont appelés Dark Data.

C’est particulièrement le cas dans le domaine biomédical et scientifique avec les imageries, les rapports scientifiques et les notes qui sont plus adaptés à la lecture humaine que des ordinateurs.

Pour extraire ces données précieuses, il faut mettre en place des solutions complexes afin de les structurer pour ensuite pouvoir les analyser.

Partant de ce constat, une équipe de chercheurs de Stanford a développé le solution Snorkel. Snorkel a pour objectif de faciliter la mise en œuvre de solutions prédictives ainsi que l’exploitation des Dark Data. Elle s’attaque au principal goulet d’étranglement dans l’apprentissage machine en facilitant la création des jeux de données d’entrainement.

Prenons l’exemple de scientifiques souhaitant utiliser l’apprentissage machine pour prédire s’il existe ou pas un lien entre un produit chimique et une maladie à partir de la littérature scientifique.



Dans un projet classique de Machine Learning, les étapes sont les suivantes :

1. Extraire des couples Produit Chimique / Maladie à partir de chaque phrase de chacun des documents
2. Création d’un jeu d’entrainement. Pour cela la contribution d’un expert est nécessaire pour indiquer si oui ou non le lien entre le couple existe dans le texte ou pas.
3. Ensuite il faut créer des features pour donner à la machine beaucoup de petites informations qu’elle utilisera pour faire ses prédictions. Par exemple quels sont les mots que nous trouvons entre les 2 entités, …
4. Ensuite nous pouvons appliquer un algorithme de classification pour faire l’apprentissage pour ensuite être capable de faire la prédiction.

Il existe aujourd’hui de nombreuses librairies facilitant la mise en œuvre de la dernière étape et l’application d’algorithmes. De plus avec l’avènement du Deep Learning, l’extraction des features se fait directement par la machine. Mais en contrepartie, les algorithmes de Deep Learning nécessite un volume de données d’entrainement conséquent.

C’est donc l’étape de construction de données d’entrainement et notamment d’étiquetage qui est la plus coûteuse en terme de temps mais elle demande également une certaine expertise pour être réalisée correctement. C’est la partie la plus fastidieuse à produire dans les projets d’apprentissage machine de nos jours.

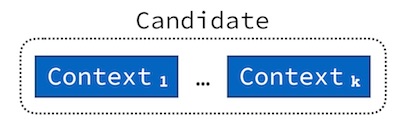
Snorkel est un projet open source qui aide les utilisateurs à créer de large quantité de données d’entrainement simplement de manière programmatique. Elle évite ainsi le travail pénible de devoir labéliser à la main chacune des données d’entrainement.

## Comment fonctionne Snorkel ?

L’idée derrière le système Snorkel est qu’au lieu de créer des données labelisées manuellement nous pouvons créer des modèles performants en nous basant sur des données labelisées de manière programmatique. En modélisant notre processus de création de jeu de données d’entrainement, nous pouvons utiliser des sources de faible qualité pour entrainer des modèles de haute qualité.

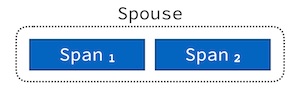
Snorkel se base sur de la Data Programming qui consiste à écrire des scripts qui permettent d’étiqueter des données.

### Candidat et Contexte

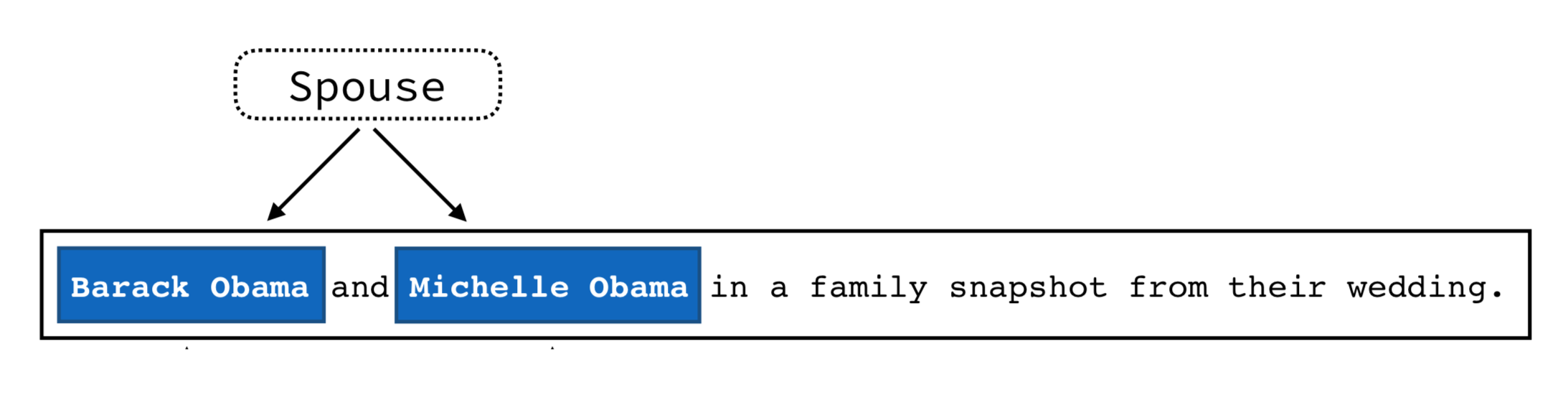


Snorkel utilise un ensemble d’objets de base pour modéliser nos données.

Il faut commencer par définir un Candidat qui représente les mentions potentielles que nous voulons extraire de nos données. Un candidat est défini sur la base d’un ou plusieurs objets Contexte. Un contexte est simplement une unité textuelle comme un mot ou une phrase.



Par exemple, si on veut construire une base de connaissance de couples mariés en réalisant une extraction des relations de mariage à partir d’article de presse, nos candidats seront toutes les paires de noms de personnes trouvé dans nos documents.



### Les fonctions de labélisation

Snorkel se base donc sur les fonctions de labélisation. Les LFs sont simplement des règles métiers programmés par l’utilisateur et qui vont permettre de labéliser un sous ensemble de nos données.

Dans le cas d’une classification binaire, la fonction va retourner 1 pour un label positif, -1 pour un label négatif et 0 si elle s’abstient de labéliser.



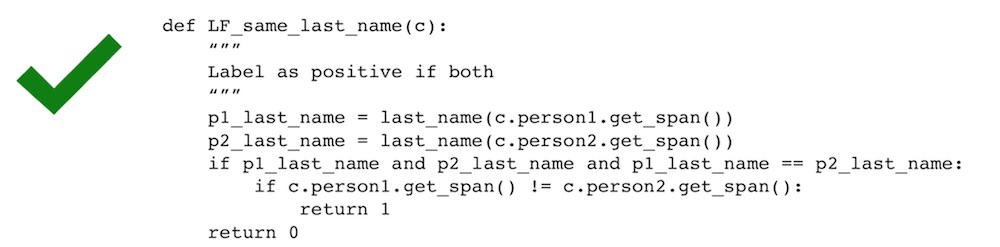
Les fonctions de labélisation s’appliquent sur des Candidats. Un Candidat représente une relation entre une ou plusieurs entités dans nos données. Par exemple, si on veut construire une base de connaissance de couples mariés en réalisant une extraction des relations de mariage à partir d’article de presse, nos candidats seront toutes les paires de noms de personnes trouvé dans chacune des phrases de nos données comme l’illustre la figure ci-dessous.

Les candidats se composent à la fois d’instances positives et négatives. Les fonctions de labélisation sont appliqués à l’ensemble des candidats et doivent prédire à la fois les labels positifs et négatifs.

Snorkel permet d’écrire deux types de Labeling Functions :

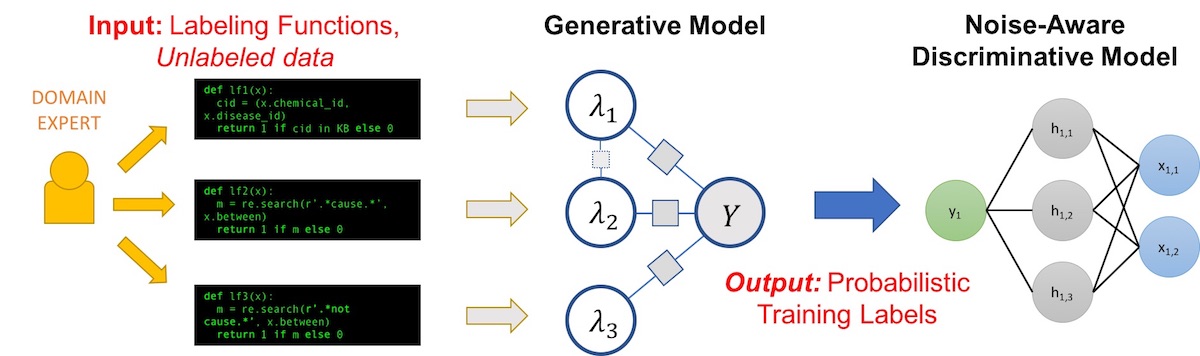
* Celles basés sur des patterns. Ils sont implémentés à l’aide d’expression régulières et d’autres méthodes d’analyse de texte.
* Celles basées sur d’autres sources d’informations comme des bases de connaissances externes (ex : DBPedia, WikiData, …) pour labéliser les données.

Exemple de LF :



Comme les LF sont juste des bouts de codes, il est alors facile de les modifier, les réutiliser et les adapter à d’autres problématiques.

### Le processus Snorkel



Snorkel va donc utiliser un ensemble de fonctions de labélisation pour le modèle d’apprentissage machine. Les sorties de ces fonctions contiendront probablement du bruit, des chevauchement ou des conflits rendant le résultat sur les labels d’entrainement imparfait. Cependant, la puissance de Snorkel est justement d’apprendre sur ces imperfections.

**L’utilisateur :**

1. Charge les données non labélisées
2. Écrit les fonctions de labélisation (LFs)
3. Choisit un modèle discriminatif pour sa prédiction

**Snorkel :**

1. Les LFs sont appliqués aux données non labellisés.
2. Un modèle génératif est utilisé pour apprendre la précision et les corrélations des fonctions de labellisation et un poids est alors donné à chacune.
3. Le résultat du modèle génératif est un ensemble probabiliste de labels d’entrainement. Il est alors utilisé pour entrainer un modèle discriminatif pour faire les prédictions finales.

Dans le modèle génératif, Snorkel va comparer les points d’accords et de désaccords des fonctions de labélisation pour apprendre.

Un modèle génératif modélise la façon dont les données ont été générés. Il apprend la distribution conjointe de x et y. Un modèle discriminatif, ne se soucie pas de la façon dont les données ont été générés. Il apprend la probabilité de y sachant x.

Un petit jeu de données annotés manuellement est nécessaire afin d’évaluer la qualité des fonctions de labélisations. Il est également utilisé dans la phase d’apprentissage de la modélisation discriminative.

## Comment utiliser Snorkel ?

La mise en œuvre est plutôt simple. Il existe un grand nombre de documentations, d’articles de publication de recherche, de tutoriaux disponible sur le site de Snorkel :

<https://hazyresearch.github.io/snorkel/>

L’installation est assez simple et il faut simplement suivre la procédure décrite ici : <https://github.com/HazyResearch/snorkel#installation>

Une fois Snorkel installé, pour comprendre son fonctionnement, il est conseillé de suivre le tutorial d’introduction qui décrit bien les 3 étapes principales de l’utilisation de la solution Snorkel :

1. Le prétraitement et l’analyse syntaxique des documents avec l’instanciation des objets Candidates.
2. L’écriture des fonctions de labélisations et l’application du modèle génératif
3. L’application d’un modèle discriminitatif pour la prédiction finale.

# Mise en application de Snorkel

## Présentation du cas testé

Pour tester et évaluer la solution Snorkel, nous avons choisi d’extraire les relations de partenariat entre acteurs économiques à partir d’article de presse.

Pour cela, nous sommes partis d’extraits d’articles de presse fournis par la société GeoTrend qui développe une plateforme de Business Discovery. Nos données étaient composée de phrases anglaises annotées avec les relations entre acteurs économiques sous le format suivant :

* actor1; actor2; type; sentence
* actor2; actor1; type; sentence

Exemple :

ALTB;Boeing;CAPITALISTIC;"Boeing, Northrop Grumman, and Lockheed are developing the ALTB for the Missile Defense Agency which calls the airborne laser program "a pathfinder" for the nation\'s directed energy program and for missile defense technology, Boeing is the prime contractor on the project, providing the aircraft and the battle management systems, Lockheed developed the beam control/fire control system which sits at the front of the aircraft, and Northrop is in charge of the megawatt-class chemical, oxygen, and iodine laser (COIL)."

Notre corpus est composés de 645 phrases avec plus de 450 noms d’entreprises.

Les types de relation enregistrés dans nos données : Partenariat, Concurrence, Clients, Alias, Procès, Filiale et Non Défini

Par souci de montée en compétence progressive, dans notre évaluation, nous avons choisi d’implémenter une classification binaire indiquant un partenariat ou pas.

## Les différentes étapes de l’implémentation

### Préparation des données

Dans un premier temps il a fallu mettre en forme nos données pour pouvoir les utiliser dans les notebook Snorkel.

Pour cela nous avons :

1. Créé un nouveau fichier de données contenant uniquement les phrases et rattaché à un identifiant unique de document.
2. Générer nos jeux d’entrainement, de développement et de test. Pour les jeux de développement et test nous avons dû faire un traitement particulier pour y ajouter les annotations de relations
3. Générer un dictionnaire des noms d’entreprise

### Définition du schéma et extraction des candidats

Nous avons modéliser un candidat comme une relation binaire entre 2 noms d’entreprises.

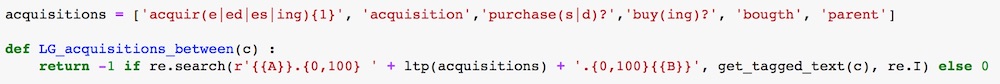


Puis nous avons utilisé les fonctionnalités de Snorkel pour extraire l’ensemble de nos candidats de notre corpus. Pour cela nous nous sommes basés sur notre dictionnaire contenant les noms d’entreprises.

### Fonctions de labellisation et modèle génératif

Nous avons ensuite analyser un certain nombre d’articles et d’identification de candidats pour nous aider à écrire les fonctions de labellisation. Nous en avons écrit plus d’une douzaine en nous basant notamment sur les expressions régulières.

Exemple de LF :



* Chaque fonction de labellisation est appliqué aux candidats de nos données d’entrainement afin d’obtenir une matrice de labels.
* Cette matrice est alors utilisée pour entrainer le modèle génératif de Snorkel. On obtient en sortie de ce modèle une distribution des probabilités de représenter un partenariat pour les candidats de notre jeu d’entrainement (une valeur proche de 1 indique une forte probabilité de partenariat et 0 le contraire). Ce vecteur est appelé marginals.

*Métrique d’évaluation*

Pour déterminer la qualité et tuner nos fonction de labellisation, nous nous sommes basés sur les métriques suivantes :

* Accuracy  : le pourcentage de candidats que la fonction labellise correctement.
* Coverage : Le pourcentage de tous les candidats qui ont été labélisé par une ou plusieurs LF.
* Conflit : le pourcentage de candidats avec au moins 2 LFs donnant des résultats en désaccord.

Nous avons utilisé notre jeu de développement contenant les annotations pour avoir ces métriques. Idéalement il faut une forte couverture (coverage) et une bonne précision (accuracy) pour nos LFs. Les conflits sont plutôt une bonne chose car ils permettent à l’algorithme d’apprendre.

Nous avons également utilisé les métriques classiques de précision, rappel et F1-score.

Nous nous sommes appuyés sur la distribution des marginals pour évaluer la qualité de notre modèle génératif. Idéalement on doit obtenir une distribution bi-modale avec une franche séparation entre les 0 et les 1.

### Application d’un modèle discriminatif

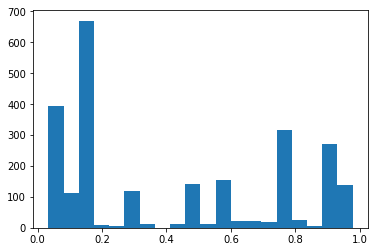
Nous avons ensuite utilisé le résultat du modèle génératif pour entrainement notre modèle final. Cette dernière modélisation permet de mieux généraliser la prédiction et donc d’améliorer la qualité de prédiction sur des données nouvelles.

Nous avons utilisé l’algorithme LSTM (Long Short-Term Memory) qui est un type de réseau de neurones récurrents. Ils sont particulièrement efficace pour les tâches de classification de texte.

Nous avons ensuite utilisé notre jeu de test pour évaluer le résultat.

## Évaluation des résultats

Notre distribution des marginals :



Nous avons une couverture négative plus importante (gauche). Il faudrait écrire plus de LFs positives pour mieux répartir les probabilités.

*Jeu de développement :*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **Précision** | **Rappel** | **F1-Score** |
| Modèle génératif | 37,9 % | 58,1 % | 45,9 % |
| LSTM | 39,7% | 62,8% | 48,6% |

*Jeu de test :*

|  |  |  |  |
| --- | --- | --- | --- |
|  |  |  |  |
|  | **Précision** | **Rappel** | **F1-Score** |
| Modèle génératif | 44,8 % | 61,5 % | 51,9 % |
| LSTM | 38,6 % | 83,5% | 52,8 % |

Nous constatons que les scores sont améliorés après l’application de la dernière phase de modélisation. Le modèle discriminatifs augmente à chaque fois le F1-score de près de 3 points pour les données de développement et 1 point pour les données de tests.

Ces résultats devraient pouvoir être améliorés notamment en utilisant beaucoup plus de données car les algorithmes d’apprentissage profond (LSTM) requiert une masse de données importante pour être efficaces. De plus les fonctions de labellisation devraient pouvoir être améliorer et plus précise pour distinguer plus efficacement les relation de partenariat et il serait également judicieux d’utilser une base de connaissance externe et écrire de nouvelles fonctions de labélisaiton basée sur cette nouvelle source. Enfin il faudrait écrire encore plus de LFs.

Dans notre cas, nous n’avions malheureusement pas de baseline à comparer avec les scores données par Snorkel. Cependant Snorkel a clairement faciliter la génération de données d’entrainement. Il aurait facilement fallu deux jours de travail à temps plein pour annoter l’ensemble des articles alors que nous avions qu’un petit jeu de données. L’écriture des LF n’a pas pris plus d’une journée. Nous avons donc clairement gagner en temps et c’est avant tout l’objectif de la solution Snorkel.

# Conclusion

Les algorithmes de machine learning et particulièrement de Deep Learning s’appuient sur des quantités importantes de données d’entrainement annotés manuellement. Il est souvent difficile d’obtenir de telles données d’entrainement.

Snorkel permet de remédier à cette problématique. A la place d’annoter manuellement chacune des données d’entrainement, Snorkel va générer un large volume de données en se basant sur des fonctions de labellisation programmés par l’utilisateur. Snorkel est particulièrement adapté dans le développement d’application d’extraction d’information dans des domaines dans lesquels il est difficile d’obtenir les données d’entrainement.

Snorkel demande un certain effort pour maîtriser correctement les concepts et l’objectif de chaque étape dans le pipeline de construction du programme. Cependant il est ensuite assez simple à mettre en œuvre et il dispose de nombreux tutoriaux explicatifs.

Nous avons pu l’appliquer sur un cas concret d’extraction de relation. Les scores obtenus ne sont pas très élevés mais ils doivent être améliorables en travaillant les fonctions de labellisation, en augmentant la quantité de données et en optimisant l’algorithme discriminatif. Dans notre cas, nous avons surtout gagner un temps considérable par rapport à une solution avec annotation manuelle.

Snorkel est toujours en cours de recherche et il intègre régulièrement des améliorations. C’est donc un système que nous recommandons très vivement pour tous les projets dans lesquels les données ne serait pas annotés.