Rapport du projet Anonymisation

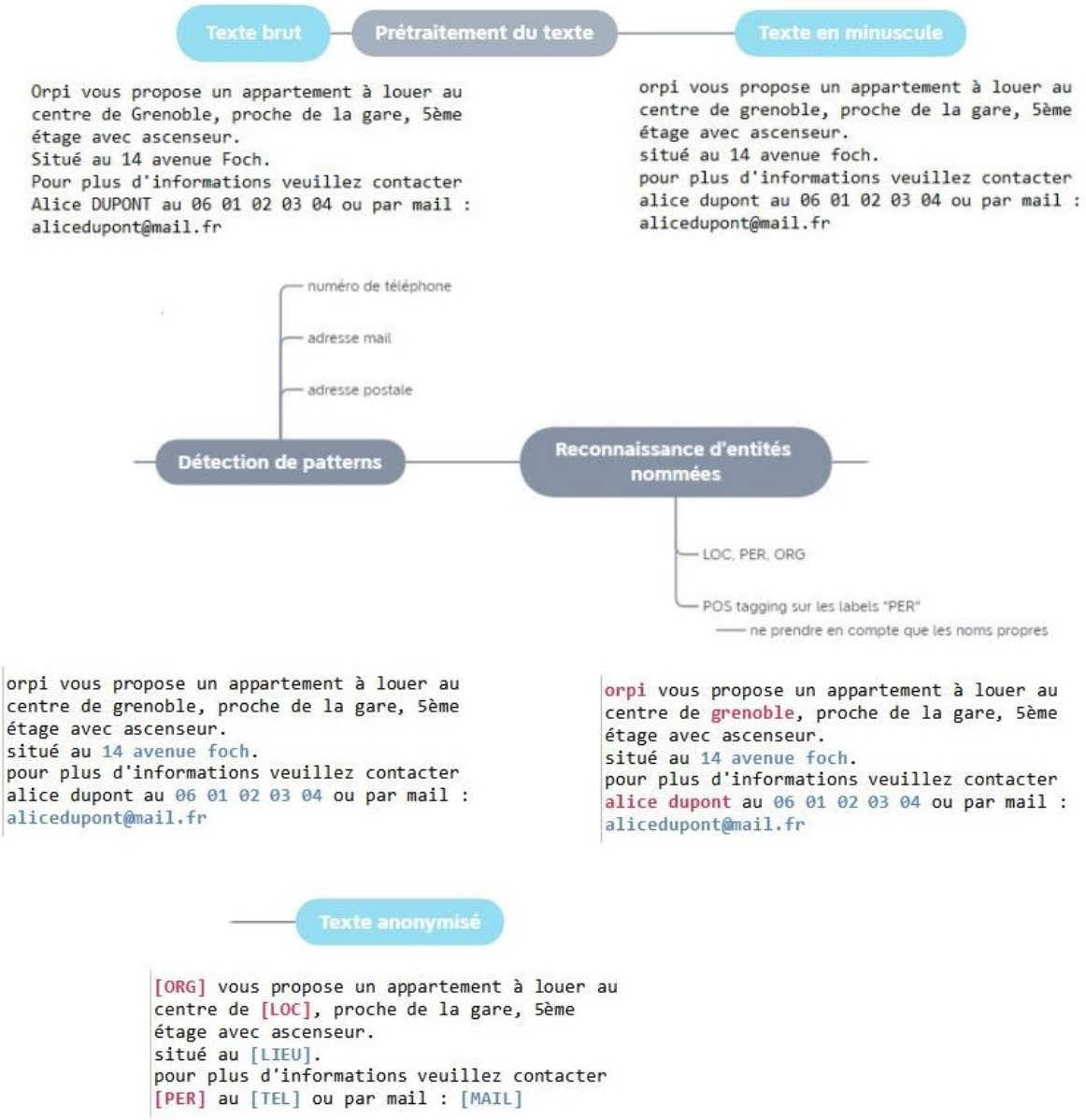
Description du projet

Notre projet consiste à anonymiser une quarantaine d'annonces immobilières sur le site Leboncoin (https://www.leboncoin.fr/_immobilier_/offres), de manière à ne pas identifier la personne qui a publié l'annonce et ne pas localiser le bien. Pour ce projet, nous avons donc jugé pertinent d'anonymiser :

- les données personnelles (téléphone, email, nom)
- la localisation de l'annonce (ville, nom des lieux)
- le nom de l'entreprise

Pour le scraping, nous avons utilisé requests_html. Pour l'installer : `pip install requests-html` .

Chaîne de traitement



- Prétraitement du texte

Tout d'abord nous effectuons un prétraitement sur le texte en le mettant intégralement en minuscules. En effet, l'algorithme de reconnaissance des entités nommées est relativement sensible à la casse. Les textes sur Leboncoin ne sont pas écrits de manière standard (noms sans déterminants, certains mots écrits en majuscules pour les mettre en valeur), les retranscrire en minuscules permet moins de confusions pour détecter les entités nommées.

- Définition de patterns

Nous avons ensuite défini des patterns pour repérer les informations suivantes :

- adresse email
- numéro de téléphone
- adresse postale

Ces données ont chacune un format unique qui est simple à identifier en utilisant des patterns.

- Reconnaissance d'entités nommées

On utilise ensuite le module de reconnaissance d'entités nommées de SpaCy. Nous avons décidé de prendre en compte les entités suivantes : LOC, PER et ORG. Celles-ci correspondent aux informations que nous avions décidé d'anonymiser. Cependant on retrouve tout de même quelques erreurs où certaines données sont anonymisées alors qu'elles ne correspondent pas au label associé. Nous effectuons alors une analyse morphosyntaxique sur les éléments labellisés pour ne retenir que les noms propres. Cette analyse s'effectue seulement sur les labels PER, car certains noms de ville ne sont pas étiquetés comme nom propre par SpaCy.

- Traitement des erreurs

Pour le traitement des erreurs, nous avons fait le choix de minimiser au plus les erreurs, tout en gardant toutes les données citées précédemment (nom, localisation, téléphone, etc) anonymisées. C'est-à-dire que l'on jugeait préférable d'anonymiser des données non pertinentes plutôt que de rendre visible des données qui puissent permettre d'identifier la personne ou le bien posté.

Principales structures de données

Chaque annonce extraite du site est enregistrée dans un fichier texte dans un même dossier. Le contenu des annonces est traité comme une chaine de caractères. Après anonymisation, les textes sont également enregistrés dans des fichiers, dans un dossier à part.

Bogues constatés

Le script « scraper.py » n'est pas toujours fonctionnel. Il est possible d'être interdit par le site LeBonCoins en raison d'une interrogation de robot. Dans ce cas là, le script signalera une erreur de « NoneType ».

Évaluation

Pour effectuer une évaluation de notre système, nous avons choisi une dizaine de textes aléatoirement dans le corpus. Après une annotation manuelle, ces textes (le

dossier « Corpus_manuel ») servent de référence. Et puis, la comparaison entre les textes de référence et ceux sortis du système (le dossier « Corpus_anony ») a été réalisée. Nous collectons tous les cas d'oubli et d'erreur, y compris les erreurs de mal annoter le texte et les erreurs du label. Le résultat se présente ci-dessous :

Texte	Référence	Sortie	Oublie	Erreur d'annoter	Erreur du label
1879185093	6	2	4	0	0
1886389408	6	13	3	10	1
1887791336	13	25	5	17	0
1896597676	2	5	1	4	0
1902920369	7	7	0	0	0
1902920544	9	13	1	4	8
1902920665	1	3	0	2	0
1902920878	1	4	1	4	0
1902921255	2	0	2	0	0
1902921276	9	9	2	2	0
1902921711	4	3	2	2	0
Total	60	84	21	45	9

Nous pouvons obtenir une précision de 35,71% et un rappel de 65%. Le valeur de rappel est beaucoup élevée de celle de précision. Ce résultat convient de notre décision de ce projet : au lieu de laisser des données non anonymisées, nous essayons de trouver tous les éléments importants et gardons des données anonymisées de manière impertinente, ce qui réduira le taux de précision dans une certaine mesure.

Les raisons pour ce résultat peuvent être différentes :

1. Le corpus que nous avons choisi est composé des annonces sur le site LeBonCoins, celles qui ne font pas partie des textes standards. Le style est plutôt oral et les phrases sont généralement incomplètes. Pour mettre en évidence des informations importantes, il contient souvent des expressions, des abréviations et les mots écrits en majuscules. De plus, il existe de nombreuses fautes d'orthographe. Tous ces éléments augmentent la difficulté de bien détecter les informations importantes.

2. Notre projet se base principalement sur les règles en utilisant les expressions régulières comme l'anonymisation des numéros de téléphone, des adresses mail et lieux. Bien que cette méthode puisse garantir la précision, ce qu'il peut détecter est très limité. Par exemple, pour le cas de numéro de téléphone, nous arrivons à établir différents modules. Mais quand il rencontre le format comme 06123 12345, il n'arrive pas à le reconnaître. De plus, nous n'avons pas pris en compte l'influence du contexte. Les mots précédents ou suivants peuvent donner des informations nécessaires et servent à élever la précision.

Amélioration

Afin de vérifier la performance de différents outils sur la détection des entités nommées, nous avons testé les 2 outils : SpaCy et NLTK. Selon le résultat, NLTK est relativement moins performant que SpaCy et le projet est donc réalisé en utilisant SpaCy.

Après avoir exclu l'impact des outils, les améliorations des performances se situent principalement dans le corpus et l'algorithme.

Au niveau du corpus :

1. Il est important d'effectuer un prétraitement sur le corpus avec le but d'enlever les erreurs et les difficultés possibles pour les traitements ultérieurs.
2. Si c'est possible, il vaut mieux travailler sur un corpus de textes standards. Nous pouvons aussi utiliser le modèle de chunk en profitant des syntagmes significatifs.
3. Pour un corpus qui contient des textes non-standards, est ce qu'il est possible d'avoir un corpus annoté pour que nous puissions entraîner notre système et adapter des outils d'analyse ?

Au niveau de l'algorithme :

1. Comme nous l'avons dit précédemment, le contexte peut être utile à la détection. Il est possible d'être intéressant d'intégrer N-grammes et l'analyse de dépendance dans l'algorithme.
2. Pour l'instant, notre projet se limite à la construction des règles et la performance des entités nommées de SpaCy sur la langue française n'est pas très satisfaisante. Est ce que nous pouvons, dans le futur, entraîner les modules à travers Machine Learning en fournissant un grand corpus annoté ou non ?

Toutes ces idées d'amélioration nous attendent à explorer et à réaliser.