Anonymisation d'emails

Antoine Lafouasse

28 décembre 2015

Introduction

L'anonymisation de données est une tâche de pré-traitement, qui consiste à effacer d'une donnée textuelle toute mention ou élément pouvant servir à identifier une entité du monde réel ; elle sert de fait à sécuriser une donnée au sens où il devient plus difficile — idéalement impossible — de remonter aux origines des données.

Elle joue ainsi un rôle essentiel en TAL, dans la mesure où des données anonymisées peuvent être plus facilement exploitées en tant que corpus sans lever de problèmes d'éthique : des données brutes, sans anonymisation, posent le problème de l'utilisation de données potentiellement sensibles et personnelles de personnes physiques ou morales, sans autorisation de leur part. L'anonymisation de données en TAL est ainsi portée par une motivation autant éthique que juridique.

Cette étude se propose de présenter une solution logicielle visant à anonymiser un corpus écrit, consistant en données issues d'emails.

1 Présentation des données

Les données sur lesquelles nous travaillons proviennent entièrement du corpus Enron, qui sont un ensemble de communications entre employés d'une entreprise sous forme d'emails en langue anglaise.

Ces données nous sont présentées entièrement sous forme de texte brut, ce qui veut dire que les métadonnées propres aux emails et qui dépassent le cadre du corps du message sont encodées avec une représentation textuelle

spécifique, avec chaque champ défini sur une ligne en en-tête du message, comportant le nom du champ et sa valuer séparés par un point-virgule (:).

```
Message-ID: <25013650.1075846656636.JavaMail.evans@thyme>
Date: Tue, 14 Dec 1999 10:19:00 -0800 (PST)
From: susan.scott@enron.com
To: brycoop@gte.net
Subject: Re: Hi
Mime-Version: 1.0
Content-Type: text/plain; charset=us-ascii
Content-Transfer-Encoding: 7bit
X-From: Susan Scott
X-To: brycoop@gte.net
X-cc:
X-bcc:
X-Folder: \Susan_Scott_Dec2000_June2001_1\Notes Folders\All documents
X-Origin: SCOTT-S
X-FileName: sscott3.nsf

Bryan,

Are you still having a great time?  So what is Bali like?
The only images I can summon are Gauguin's paintings, but
weren't those of Tahiti?

Things are pretty good here.  Lots of Christmas parties,
lots of Christmas cheer, etc.  Not a whole lot going on at
the office, for a change. Really, though, I think I would
much rather be in Bali.

Take care.

S.
```

FIGURE 1 – Exemple d'email issu du corpus Enron

L'exemple 1 nous montre cet encodage des métadonnées propres aux emails, et nous donne par là-même l'occasion d'aborder les mentions et traces d'identification susceptibles d'apparaître dans notre corpus ; nous tâcherons d'en établir une liste, bien que non-exhaustive.

Noms de personnes La façon la plus simple et évidente d'identifier une personne est d'utiliser son nom. Ces données sont particulièrement courantes dans les emails dans la mesure où les noms de l'expéditeur et des destinataires sont généralement générés automatiquement par l'annuaire ou le serveur mail des entreprises.

Les noms peuvent également survenir sous forme de mention au sens courant du terme, lorsqu'une personne en évoque une autre dans le corps du message. Ce type de mention est généralement plus approximatif que les noms rapatriés automatiquement, en ce qu'ils peuvent présenter une variation typographique, des contractions, des surnoms ou mêmes des fautes d'orthographe — ce qui ajoute à la difficulté d'identification des entités nommées par la mention à laquelle nous nous intéressons.

Adresses email Ces mentions peuvent servir d'identification plus formelle que les noms, puisque les adresses email doivent être respectées exactement — au détail près qu'elles sont insensibles à la casse : les fautes d'orthographe, contractions et alias sont ainsi beaucoup plus rares sur les adresses email que les noms de personnes. De plus, ces adresses suivent un formatage précis qui les rendent faciles à détecter, et dans un cadre professionnel elles comportent bien souvent le nom et le prénom du propriétaire directement dans le texte de l'adresse. Ainsi, les adresses email sont un élément d'identification particulièrement puissant dans le corpus Enron qui consiste de communications professionnelles.

Noms d'organisations Les noms d'organisation sont un élément d'identification majeur au sens où ils permettent une identification au même titre que les noms de personnes, mais ils peuvent se révéler plus difficiles à identifier, pour des raisons diverses.

En effet, les noms d'organisation peuvent être des noms inventés de toutes pièces, ou être pris d'un vocabulaire soit de langue étrangère, soit spécifique à un domaine — il devient ainsi difficile d'identifier une organisation avec le simple critère d'un mot inconnu, puisqu'il est presque impossible de le distinguer des autres néologismes.

Un nom d'organisation peut également être un mot très courant, qui devient encore plus difficile à identifier lorsque le nom de l'organisation n'est souvent que partiellement mentionné — par exemple, *Apple Inc.* n'est souvent mentionnée que par le nom *Apple*, ce qui rend la mention difficile à identifier

puisque faisant partie du vocabulaire courant. Cette remarque s'appliquent aussi pour les organisations dont le nom forme un acronyme, qui lui fait partie du vocabulaire courant — souvent intentionnellement.

2 Description de la procédure utilisée

2.1 Démarche générale

L'approche que nous avons retenue pour anonymiser ces emails est de retenir en mémoire les mentions que nous rencontrons dans les données, de sorte à opérer un remplacement mot-à-mot de ces mentions de manière cohérente : en gardant une table d'équivalence entre les mentions et leur remplacement, nous sommes certains de la stabilité de notre anonymisation. Ainsi, nous préservons l'exploitabilité des données résultantes pour des tâches s'appuyant sur les entités nommées, comme la résolution de coréférence.

Ainsi, notre démarche d'anonymisation s'articule en trois étapes :

— Segmentation de l'entrée (tokenisation)
— Reconnaissance des entités nommées
— Remplacement des entités nommées

2.2 Choix de technologies

La conception du projet a principalement été conduite par une volonté d'utiliser au maximum des outils existants, de sorte à limiter nos développements à l'articulation de ces outils ensemble — c'est particulièrement le cas pour la reconnaissance d'entités nommées, dont l'implémentation n'est pas triviale.

Notre choix d'environnement s'est porté sur Java, qui nous garantissait non seulement une portabilité du code, mais nous permettait aussi de mettre rapidement en place un environnement complet avec une intégration continue et un package managing efficace et largement utilisé — en l'occurrence, Apache Maven.

Notre choix pour la reconnaissance d'entités nommées s'est ainsi porté sur le module NER de Stanford, qui fournissait non seulement des outils en interface graphique et en ligne de commande pour l'expérimenter et en observer les résultats, mais aussi directement la bibliothèque compilée dans une archive JAR et un exemple d'utilisation du code ; ainsi, la documentation

fournie du code nous a séduits et conforté notre sélection. Le seul point noir que nous retenons à cette solution est que comme l'université de Stanford ne semblait pas avoir de serveur Nexus, nous avons du créer à la main un artefact contenant le module de NER et l'intégrer dans notre projet Maven. Ce détail est préjudiciable en termes de légèreté du dépôt GiT, mais n'a pas altéré la flexibilité de l'architecture du projet autour de Maven.

Nous avons ainsi pu nous appuyer très largement sur la puissance de ce module — qui est un classifieur supervisé utilisant les CRF — en n'ayant à ajouter qu'une simple heuristique pour reconnaître les adresses email en plus des entités nommées — adresses dont la fréquence est dûe à l'aspect épistolaire de notre corpus. Pour la tokenisation nous avons opté pour une simplification du processus en laissant la NER de Stanford opérer sa propre tokenisation et travailler à partir de ses résultats.

3 Analyse des résultats et limites

```
Message-ID :
Date : Tue , 14 Dec 1999 10:19:00 -0800 -LRB- PST -RRB-
From : EMAIL_0
To : EMAIL_1
Subject : Re : Hi
Mime-Version : 1.0
Content-Type : text/plain ; charset = us-ascii
Content-Transfer-Encoding : 7bit
X-From : PERSON_2 PERSON_3
X-To : EMAIL_1
X-cc :
X-bcc :
X-Folder : \ Susan_Scott_Dec2000_June2001_1 \ Notes Folders \ All documents
X-Origin : SCOTT-S
X-FileName : sscott3.nsf

PERSON_4 ,

Are you still having a great time ?  So what is LOCATION_5 like ?
The only images I can summon are PERSON_6 's paintings , but
were n't those of LOCATION_7 ?

Things are pretty good here .  Lots of Christmas parties , lots
of Christmas cheer , etc. .  Not a whole lot going on at the
office , for a change . Really , though , I think I would much
rather be in LOCATION_5 .

Take care .

S.
```

FIGURE 2 – Exemple d'email issu du corpus Enron

L'exemple 2 est l'exemple 1 anonymisé par notre outil. Il nous montre que les cas simples ont été correctement traités, à savoir les adresses email, les noms de personnes et les noms de lieux qui ont été correctement remplacés En revanche, la signature contractée (*"S."*) et le nom dissimulé dans un chemin

de dossier ont été laissés tels quels.

Cet exemple nous donne l'occasion d'évoquer plusieurs cas de figure qui ne sont pas correctement reconnus par notre modèle d'anonymisation :

Adresses email dissimulées Ce type d'adresses est peu courant dans une communication intra-entreprise, mais le devient beaucoup plus lorsqu'on touche un message ou une publication qui est accessible sur Internet : pour éviter l'acquisition d'une adresse email par des robots dont le rôle est de crawler internet à la recherche d'adresses, il est occasionnel de les voir sous une forme qui n'est (idéalement) pas détectable par une machine, mais tout de même lisible par un être humain, par exemple : `antoine [point] lafouasse [at] student [point] 42 [point] fr`. On voit facilement qu'avec une adresse formatée ainsi, une expression rationnelle visant une adresse mail sera incapable de reconnaître, et donc d'anonymiser ce type d'adresse. On peut alors imaginer un ensemble d'heuristiques qui permettraient de reconnaître ces dissimulations, mais comme celles-ci ne sont pas normalisées elles sont généralement laissées à l'imagination de l'expéditeur, leur forme peut varier de manière difficilement prédictible.

Contractions de noms, surnoms Ces deux formes peuvent être résumées avec le même problème d'une forme alternative qui réfère à une entité déjà connue, qui plus est avec une expression qui ne ressemble pas forcément à une entité nommée ; nous faisons ainsi face à deux facettes d'un problème.

Premièrement, ces formes sont susceptibles de mettre en échec le module de reconnaissance d'entités nommées, qui pour notre cas n'a pas été spécifiquement entraîné sur un corpus épistolaire. Stanford ayant seulement fourni un jeu de modèles entraînés par leurs soins mais pas de solution permettant d'entraîner nos propres modèles, cela nous contraindrait soit à abandonner cet outil entièrement, soit penser un mécanisme complémentaire à la NER de Stanford, soit un deuxième modèle — partiel — de reconnaissance d'entités nommées. Bien entendu, l'un autant que l'autre est très coûteux en moyens et en temps.

Deuxièmement, même si on arrive à reconnaître ces expressions comme des entités nommées, nous nous retrouvons avec plusieurs formes pouvant référer à une même entité. Si nous opérons le remplacement tel que nous l'avons implémenté, forme par forme, on se retrouve avec une perte d'information importante puisque les formes des mentions elles-mêmes sont perdues et ne

peuvent pas être utilisées en résolution d'anaphore — et ce, parfois pour des contractions simples comme des surnoms courants, des diminutifs, ou des paraphes, qui sont des cas élémentaires en résolution de coréférence. Nous pourrions ainsi implémenter nous-mêmes un mécanisme de résolution de coréférence, ce qui a également un coût élevé puisqu'il s'agit d'une tâche de sémantique computationnelle à part entière.

Noms dissimulés dans les tokens Ce cas est particulièrement visible dans l'exemple 2, avec le nom Susan Scott dissimulé dans un chemin de dossier et qui, de fait, n'est pas relevé par le module de NER. Nous avons de toute évidence affaire à une lacune de tokenisation, dont la résolution est cependant très risquée puisque non seulement la proportion de bruit dans le token est très importante, mais la segmentation elle-même de l'entitée nommée à l'intérieur peutêtre très variable.

Une solution pourrait être d'écrire un module complémentaire spécifique aux chemins de dossiers, sachant que leur segmentation est libre — obtenir un résultat satisfaisant sera donc vraisemblablement difficile, et à vrai dire il s'agit peut-être de la plus grande difficulté à laquelle notre anonymiseur fait face.

Conclusion

L'approche que nous avons choisie dès le départ, de nous appuyer au maximum sur des outils existants, nous a permis d'arriver assez vite à une baseline fonctionnant sur les cas élémentaires de manière satisfaisante. De plus, le choix de Java et de Maven nous a permis de nous ouvrir à un package managing puissant et simple, et aussi à une architecture objet qui nous a donné l'occasion de mettre en place une base de code simple à étendre.

Cette baseline est cependant loin de l'état de l'art, et ce particulièrement sur des cas propres au format d'emails de notre corpus : nous pouvons par exemple citer les adresses email obfusquées, ou les noms dissimulés dans des chemins de dossiers ou autres tokens complexes. Ces cas, pour la plupart, nécessitent une refonte soit du tokeniseur soit du module de reconnaissance d'entités nommées, ce qui représente un temps de travail considérable, plus particulièrement quand on considère que les cas manqués sont très spécifiques.

Table des matières