Université de Lille 1

En vue de l'obtention du

Label de recherche

Machine learning et visualisation des données

Auteur :

Marlot Maxime
Etudiant à Lille 1

Encadrant :

Tommasi Marc
Equipe MAGNET
Inria Lille - Nord Europe

Année 2016-2017

Résumé

Des techniques récentes en traitement de la langue cherchent à réaliser une représentation de mots (ou groupes de mots) sous la forme de points dans un espace de très grande dimension. Cette représentation vectorielle a souvent des propriétés intéressantes comme celle de rapprocher des mots dont le sens est proche. Elle permet ensuite d'effectuer des traitements de haut niveau comme de la classification de textes, de l'analyse,... . Malheureusement la visualisation de ces représentations pose problème car seulement deux ou trois dimensions peuvent être utilisées (représentations 2D ou 3D). Le sujet de mon stage a été d'étudier et d'implanter une méthode particulière de réduction de dimension appellée T-SNE adaptée à ce besoin de représentation.

Ce stage a été effectué dans l'équipe MAGNET qui s'intéresse au machine learning pour les vastes réseaux de données, essentiellement textuel. Ce stage est l'occasion de découvrir à la fois un domaine nouveau dans une équipe de pointe mais aussi de m'initier au monde de la recherche. Même si mon sujet d'étude traitait essentiellement de la visualisation de données, je me suis intéressé plus largement au sujet dans l'objectif de mieux comprendre ma partie.

Je vous souhaite une agréable lecture.

Remerciement

Avant de commencer ce document, j'aimerais remercier :
Marc Tommasi et son équipe pour leur accueil, leur aide et leurs réponses à mes innombrables questions durant ces quelques mois.

À l'équipe pédagogique du FIL qui nous a proposé de réaliser ce label de recherche en tant qu'option du S6.

Et enfin à Sylvain Salvati pour ses enseignements en TD qui m'ont apporté un regard différent et certainement plus critique sur le monde de la recherche.

Mots clés : *label recherche, machine learning, data visualisation, T-SNE*

Table des matières

1 Introduction des différentes entités

1.1 Présentation de l'Inria

L'INRIA (Institut National de Recherche en Informatique et Automatique) est un institut en recherche en mathématiques et informatique. L'institut a été fondé suite à la création du plan CALCUL décidé par le général De Gaulle en 1964. Ce plan avait pour objectif de prévenir de l'industrialisation informatique et ainsi de mettre au point des moyens d'assurer l'indépendance de la France en matière de superordinateur. Dans cet objectif de promouvoir l'informatique on notera aussi la création d'un vaste réseau de formations pour former des personnes compétentes dans ce domaine.

Aujourd'hui l'INRIA est représenté dans huit des plus grandes villes de France, s'appuie sur 178 équipes (soit 2600 collaborateurs) qui publient environ 4600 publications par an et a un budget d'approximativement 230 millions d'euros.

1.2 Présentation de l'équipe MAGNET

L'équipe MAGNET consiste en la définition de méthodes et modèles d'apprentissage automatique au sein de réseaux d'informations. Ces réseaux peuvent être représentés sous forme de graphes avec des liens souvent induits par une similarité. MAGNET se concentre sur les grands volumes de données, essentiellement textuel comme l'étude de corpus de documents ou de réseaux sociaux qui peuvent comporter plusieurs millions de mots à traiter. Pour ce faire l'équipe se base sur un apprentissage statistique dans un cadre semi supervisé, c'est-à-dire qu'on l'on possède des données déjà étiquetées (en faible nombre) et des données non étiquetées (en plus grand nombre). On utilise également des techniques d'apprentissage non supervisées où cette fois-ci c'est au programme de définir ses propres groupes (clustering) en fonction des similitudes des caractéristiques des données.

1.3 Le projet Magneto

Le projet magneto consiste à réaliser une plateforme à la fois expérimentale et opérationnelle pour la construction de représentation de textes ou parties de textes dans des espaces vectoriels de grande dimension (construction d'*embeddings*, voir 2.1). Le projet est développé par plusieurs membres de l'équipe dans le cadre d'une action de développement et de transfert. En effet, l'équipe compte l'utiliser en interne pour ses travaux de recherche mais aussi dans les différents partenariats avec des entreprises ayant le besoin de réaliser des tâches de traitement de la langue.

2 Les phases d'apprentissage

Durant ce court stage, j'ai eu l'occasion de toucher à un des domaines du machine learning qu'est la visualisation des données et de découvrir les divers problèmes qui y sont liés (bien plus nombreux que ce que j'aurai pu imaginer).
Il n'existe pas, comme toujours en informatique, un algorithme unique permettant d'arriver à nos fins, cependant nous avons décidé ici d'étudier T-SNE (t-distributed stochastic neighbor embedding) qui offre d'après son créateur une représentation plus intéressante et certaines options de paramétrages comparé à d'autres comme Isomap ou Sammon mapping. Nous y reviendrons juste après. Néanmoins pour bien comprendre notre algorithme et même si ce n'était pas mon travail il est intéressant de voir les données entrantes à traiter et notamment la notion d'embeddings qui y est liée.

2.1 Les embeddings

Lorsque que l'on étudie les données textuelles, dans notre cas des mots, on peut naturellement se demander comment leur donner du sens pour une machine. Une idée intéressante serait que chaque mot possède divers attributs qui pourraient ainsi le représenter de manière précise dans un espace vectoriel de grande dimension. On peut définir deux types d'attributs qui nous intéressent :

— **Les attributs quantitatifs** ***(attributs ordinaux)*** comme par exemple un réel représentant la longueur d'un animal et qui sont facilement représentables.

— **Les attributs qualitatifs** ***(attributs nominaux)*** comme par exemple un ensemble de couleurs pour représenter un animal { Rouge, Vert, Noir, ... } qui ne sont donc pas représentables immédiatement dans notre espace mais qui sont ceux que nous essayons d'étudier.

On cherche donc une fonction f où $f : X \mapsto Y$ avec X appartenant à notre lexique et Y une représentation de X sous forme d'embeddings à D dimensions où $D \in \mathbb{N}$ (généralement assez grand).

2.2 Word2Vec

Word2Vec est un algorithme développé par Tomas Mikolov pour Google et qui nous permet justement d'obtenir ce genre d'embeddings. Généralement chaque mot comporte quelques centaines de dimensions mais dans nos exemples plus bas, cinquante dimensions nous suffiront pour obtenir une représentation satisfaisante.

Cet algorithme se base sur l'analyse du contexte de chaque mot et cela itéré sur tout notre texte. Pour chacun des mots du corpus on utilise « une fenêtre » centré sur notre mot qui est composée en général de dix, voir vingt mots. Ainsi on peut capturer la sémantique de chaque mot et les relations entre eux. On parle aussi de sémantique distributionnelle qui permet de regrouper des mots coocurents. Ensuite on applique éventuellement une réduction de dimension, i.e. une réduction du nombre de composantes (pour généraliser on élimine les colonnes à faible variance). Une implémentation possible est de prendre notre mot et de chercher son contexte le plus probable, si notre modèle se trompe, alors on peut se servir de la matrice de sortie comme d'une matrice d'erreurs pour corriger notre modèle. Un papier a déjà été publié par Rémi Gilleron pour plus de détails [1].

À noter qu'il est impossible d'interpréter de manière concrète les valeurs obtenues dans nos embeddings mais qu'elles peuvent nous servir de distance entre nos mots (ou de similarité). Rappel des propriétés d'une distance d

$$d : E * E \mapsto R^+$$

Propriété 1. La symétrie - $\forall x1, \forall x2 \in X \; d(x1, x2) = d(x2, x1)$

Propriété 2. L'identité - $\forall x1, \forall x2 \in X \; d(x1, x2) = 0 \leftrightarrow x1 = x2$

Propriété 3. L'inégalité triangulaire - $\forall x1, \forall x2 \in X \; d(x1, x3) \leq d(x1, x2) + d(x2, x3)$

1. `http://www.grappa.univ-lille3.fr/~gilleron/WordToVector.html`

2.3 Visualisation des données

La visualisation des données et donc la réduction de dimension dans un espace interprétable pour l'homme comme R^2 pose un certain nombre de problèmes : on note l'amassement des points ou encore la perte d'information dûs à la réduction que nous verrons juste après.

Rappelons que l'objectif est d'obtenir une représentation dans notre espace réduit tout en conservant la structure local des données.

Si dans R^{50}, les mots « chat » et « chaton » sont proche alors ils doivent impérativement l'être dans R^2. L'inverse s'applique aussi pour les mots éloignés même si on verra qu'il peut être intéressant de gonfler (légèrement) artificiellement cette distance pour obtenir des clusters plus distincts.

2.3.1 SNE

T-SNE se base sur un autre algorithme du nom de SNE (Stochastic Neighbor Embedding). Cet algorithme va utiliser deux espaces pour ses calculs, on notera X, notre espace à grande dimension qui est celui de départ et que nous avons obtenu grâce à nos embeddings ainsi que Y notre projection, qui est notre espace réduit. On cherche à convertir la distance Euclidienne entre deux points en probabilité conditionnelle. Si X_i et X_j sont deux points voisins alors $P_{i|j}$ sera relativement élevé alors qui s'ils sont éloignés $P_{i|j}$ sera infinitésimal. Pour avoir cette aspect de voisinage on utilise la densité d'une gaussienne centrée en X_i qui nous permet de donner plus d'importance aux voisins proches.

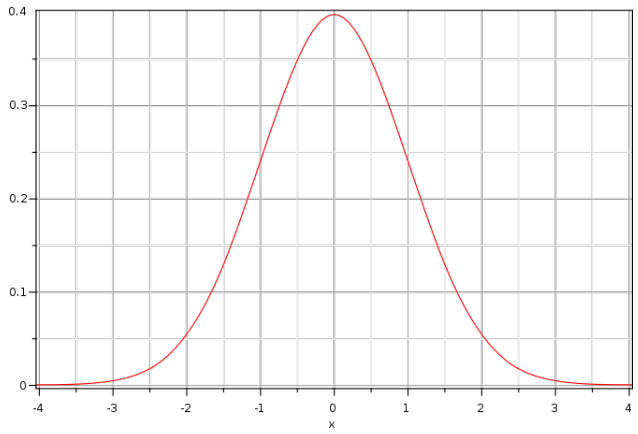

FIGURE 1 – Fonction de densité de la loi normale centrée réduite

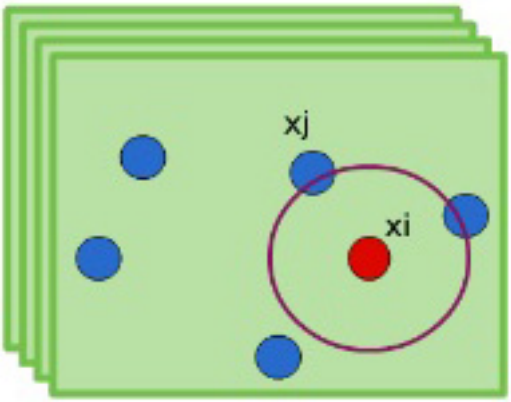

FIGURE 2 – Voisinage du point X_i

$$p_{i|j} = \frac{\exp(\frac{-\|x_i - x_j\|^2)}{2\sigma i^2})}{\sum_{k \neq i} \exp(\frac{-\|x_i - x_k\|^2}{2\sigma i^2})} \tag{1}$$

Par la même occasion on normalise notre résultat afin d'obtenir notre propre densité de probabilité qui nous sera utile juste après. Pour avoir une projection de Y fidèle à X, l'idée est de projeter aléatoirement des points dans l'espace d'arrivée et de calculer également leurs probabilités conditionnelles telle que :

$$q_{i|j} = \frac{\exp(-\|y_i - y_j\|^2)}{\sum_{k \neq i} \exp(-\|y_i - y_k\|^2)} \tag{2}$$

Grâce à cela, nous pouvons utiliser une mesure de dissimilarité entre nos deux lois qui nous servira de coût C, ici via la divergence de Kullback-Leibler. En d'autres termes nous pouvons ainsi déterminer si nos mots ont conservé les mêmes voisins d'un point de vue local. Grâce à la méthode de descente de gradient on pourra alors chercher à minimiser C.

$$C = \sum_i \mathrm{KL}(P_i \| Q_i) = \sum_i \sum_j p_{j|i} \log \frac{p_{j|i}}{q_{j|i}} \tag{3}$$

Un des avantages de SNE (et T-SNE) est son paramétrage et notamment son paramètre de « perplexity » qui est défini comme étant le nombre de voisins locaux à un point X_i. Pour ce faire, on influe donc sur σi qui est, je le rappelle centré sur X_i et permet de donner plus d'importance aux voisins proches. Néanmoins il n'existe pas de valeur de σi optimal. En effet certaines zones sont plus denses que d'autres, on favorisera alors un σi plus petit car plus σi augmentera, plus l'entropie augmentera. La chose qui rend ce paramètre intéressant est que son choix peut grandement faire varier notre projection.

Deux représentations d'un même dataset à 50 points avec une perplexity différente sous T-SNE

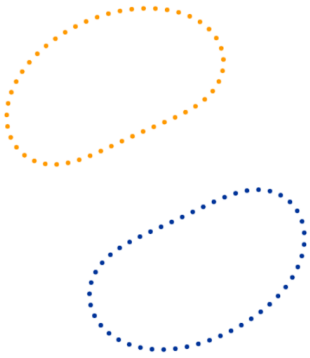

FIGURE 3 – Exemple avec une perplexity de 5

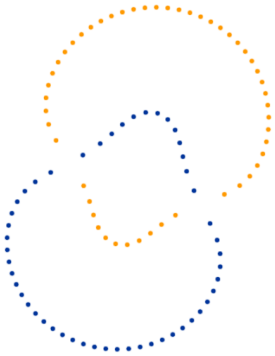

FIGURE 4 – Exemple avec une perplexity de 45

2.3.2 T-SNE

T-SNE s'inspire donc fortement de SNE et propose deux modifications, tout d'abord rendre la fonction de coût symétrique, ainsi on obtient un gradient plus simple à minimiser et surtout plus rapide. On a donc :

Propriété 1. Pii et $Qii = 0$

Propriété 2. $\forall i, \forall j \ Pij = Pji$ et $Qij = Qji$

Une autre lacune de SNE est souvent sa représentation qui est difficilement interprétable dûe à l'amassement des points au centre de notre repère. Ce problème est appelé «The crowding problem» et affecte d'autres algorithmes de visualisation comme Samon mapping.

2.3.3 The crowding problem

A mon sens, ce problème se base sur deux aspects : tout d'abord la «perte d'informations» lors de la réduction, qu'on peut comprendre par un exemple simple. Imaginons que nous avons 3 points A, B et C dans un espace en 2 dimensions et qu'on peut comparer dans notre cas à X comme sur la figure 5.

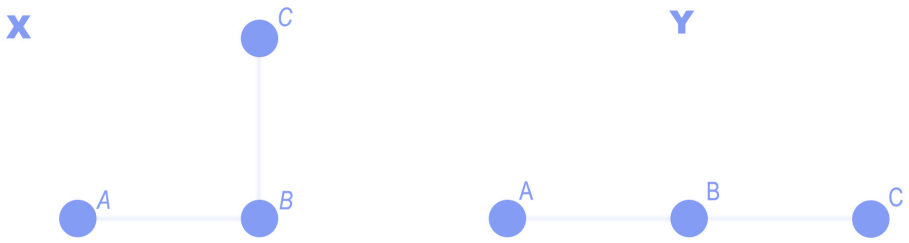

FIGURE 5 – Passage de 2 dimensions à 1 dimension

Si on considère que A-B et B-C sont équidistants et que l'on peut les considérer comme proches, on souhaite donc conserver au maximum leur aspect local dans notre espace de petite dimension, ici représenté par Y.
On peut très bien le faire comme sur la figure 5 qui convertit un espace en deux dimensions à un espace unidimensionnel et qui a bien gardé les distances A-B et B-C. Néanmoins, on s'aperçoit que la distance entre A-C a augmenté de manière significative, ce qui fait que notre représentation ne sera pas vraiment exacte. Ici on le montre sur trois points mais l'effet est bien évidemment plus important lorsque l'on traite des milliers de points.
Un autre aspect, et qui est la continuité de notre problème, c'est qu'on peut représenter chaque point comme possédant «une force» par rapport aux autres et étant donné le fait que nous avons moins d'espace dans Y que dans X pour représenter nos points de manière satisfaisante, l'algorithme va essayer de réduire un petit peu la distance entre A-B et B-C. Une fois encore sur un dataset de trois points le problème peut paraître minime mais sur des milliers de données, on va avoir cet amassement de points au centre car un point va attirer ses voisins, qui eux même vont à leurs tours attirer leurs voisins et ceci a pour effet la suppression des écarts entre les clusters, ce qui rend la représentation bien plus difficilement exploitable pour un humain.

2.3.4 La solution

Ce qu'on aimerait faire c'est d'augmenter de manière significative la distance de nos points éloignés mais aussi de ceux à une distance modérée. Ainsi on limitera fortement l'attraction entre eux et ce problème d'amassement. Pour ce faire, nous pouvons changer de distribution pour Y. Au lieu d'utiliser une gaussienne, il est préférable d'utiliser une loi dite à queue lourde comme celle de Student. La loi de Student (S) possède un degré de liberté k, pour rappel si $S \underset{k\to+\infty}{\longrightarrow}$ on obtient une distribution Gaussienne et c'est la raison pour laquelle nous pouvons nous permettre de l'utiliser. Ainsi un point conservera son voisinage local alors que son voisinage plus éloigné sera repoussé et favorisera l'apparition de clusters.

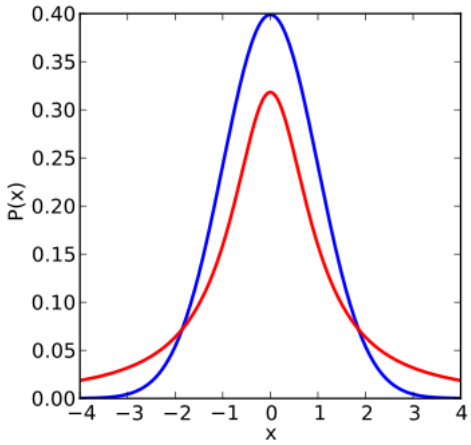

FIGURE 6 – En bleue, une Gaussienne et en rouge une Student où k = 1

Suite à toutes ces modifications, on obtient des formules un peu diffèrentes qu'avec SNE mais étant donné que ce rapport doit rester court et concis, je vous invite à les lire directement sur le papier de T-SNE. [2]

3 Application

Maintenant que nous avons introduit brièvement la phase d'apprentissage d'un texte, ainsi que le fonctionnement de l'algorithme de visualisation et les avantages de son choix qui est dans notre cas l'obtention, on l'espère, d'une projection lisible, nous pouvons désormais passer au travail que j'ai effectué.

3.1 Quelques mots sur la recherche

Tout d'abord, étant donné que ce stage était très court, je n'ai malheureusement pas réalisé autant d'applications que je l'aurai voulu. J'ai en effet passé beaucoup de temps à essayer de comprendre le papier de T-SNE, peut être parfois un peu trop en détails et pas nécessaire pour les utilisations que j'allais en faire. Après coup, et même si dans un premier temps je pensais que c'était une erreur, je me dis que le tableau n'est pas tout noir car ceci m'a vraiment fait prendre conscience, et ce en relation avec les enseignements de TD mais aussi avec les discussions avec l'équipe qu'il n'est pas nécessaire/envisageable de comprendre tout un papier car c'est souvent un travail de plusieurs mois, voire années qu'il y a derrière.
Cette prise de conscience a également été possible grâce à la lecture de deux papiers durant ce stage donné par le responsable de l'équipe (Marc Tommasi), l'un de l'université de Waterloo [3] et l'autre de l'université d'Harvard [4] qui nous expliquent «Comment doit-on lire un papier ? » et les différentes lectures à réaliser pour la meilleure compréhension possible.

2. http://www.cs.toronto.edu/~hinton/absps/tsne.pdf
3. http://blizzard.cs.uwaterloo.ca/keshav/home/Papers/data/07/paper-reading.pdf
4. http://www.eecs.harvard.edu/~michaelm/postscripts/ReadPaper.pdf

3.2 L'application

Durant ce stage j'ai également réalisé deux petits scripts, un dans l'objectif de mieux comprendre l'algorithme de descente de gradient qui est utilisé pour notre visualisation mais aussi un autre script qui adapte nos embeddings à T-SNE et qui nous donne une projection de notre document dans un espace en deux dimensions.

Exemple d'un des mots du dataset sous forme d'embedding : [5]

```
From 0.23231 -0.126145 -0.218487 -0.088644 0.281589 -0.134168 0.151128 0.130896 0.119331
0.238036 -0.3703 0.376256 0.0820332 -0.395836 0.188524 -0.0228476 -0.233165 ...
```

Grâce à la puissance de Python et ici de numpy, il n'a pas été très difficile de parser le fichier et de récupérer le label du mot associé. Le script possède quelques options de paramétrage :

```
applyTsne <fileName> <nbLines> <perplexity> <option -a >
```

- **fileName :** nom du fichier
- **nbLines :** nombre de mots qu'on veut extraire de notre fichier (très pratique si on veut juste un échantillon et que le dataset contient plusieurs milliers de mots)
- **perplexity :** La «perplexity»
- **-a** : si on souhaite avoir à côté de chaque point l'étiquette (le mot qui y est lié). Ce qui est intéressant si on souhaite voir les mots proches d'un point de vue visuel.

3.3 Résultats et interprétation

Quelques exemples de projections obtenues sur les dataset de MAGNET

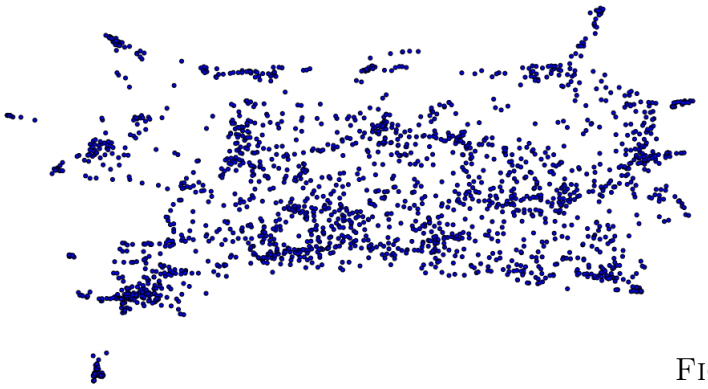

FIGURE 7 – 2000 mots, perplexity = 10

FIGURE 8 – Zoom sur un cluster représentant les mois

5. fourni par MAGNET

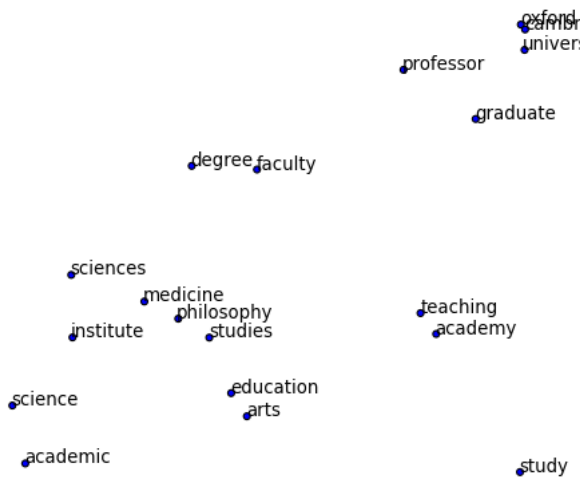

FIGURE 9 – Zoom sur un cluster en rapport avec l'éduction

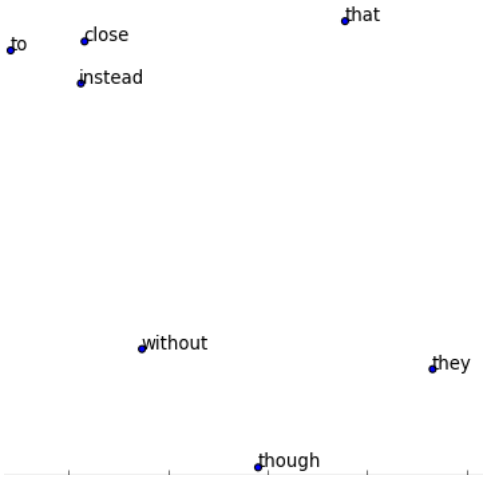

FIGURE 10 – Zoom sur une zone de mots au centre de notre espace

Nous pouvons faire un certain nombre de remarques et d'hypothéses sur ces résultats.
Tout d'abord, on peut noter que la représentation "a du sens". On retrouve des ensembles de mots qui partagent des relations. Par exemple à la figure 8 on peut très bien voir que presque tous les mois de l'année sont dans la même zone. On peut également voir sur la figure 9 que nos mots partagent une même sémantique : education et academic, teaching et professor ou encore graduate et master ... . On peut donc dire que nous avons réussi à obtenir une réduction de dimension qui a conservé la structure locale des données.

Cependant on peut également voir sur la figure 7 que même si nous avons certaines zones qui sont plus denses que d'autres, nous avons tout de même du mal à avoir des clusters vraiment distincts. On dirait qu'il y a un grand nombre de mots qui sont à mi-distance entre deux clusters et c'est particulièrement visible au centre de la représentation. On peut peut être relier cela au fait que certains mots sont utilisés de manière plus fréquente dans nos structures de phrase comme les verbes ou les mots de liaisons et de ce fait qu'ils sont attirés un petit peu par tous les autres clusters et ainsi qu'ils servent de "liant" dans la construction de nos phrases.
Si on s'intéresse d'un peu plus près à la structure de notre projection, on s'aperçoit effectivement que les ensembles de mots plutôt au centre ou bien "perdus" entre deux zones denses sont généralement composés de verbes ou de mots de liaison comme sur la figure 10. Alors que les groupes en bordures sont souvent des noms communs.

4 In fine

4.1 Conclusion du projet et perspectives

Nous avons donc pu observer les résultats de nos embeddings et essayer de voir si un ensemble proche de mots partageait une sémantique ou une relation commune. D'un point de vue IHM, on peut très bien imaginer une amélioration de l'interface pour la rendre plus intuitive comme représenter les données dans espace en trois dimensions et avoir ainsi la possibilité de «naviguer» à travers nos données.

D'un point de vue plus technique, notre représentation peut varier énormément juste en modifiant notre paramètre de «perplexity». On peut donc naturellement se demander quels résultats obtiendront nous en changeant d'algorithme de visualisation. En d'autres termes, est-ce qu'il existe un autre algorithme qui serait plus adapté pour représenter nos données ?

Un dernier axe qu'on pourrait peut-être exploiter c'est qu'on se sert de T-SNE comme d'un outil destiné à la réduction de dimension pour obtenir une projection intuitive pour l'homme. Cependant on a vu que cette réduction fait apparaître des clusters distincts donc est-ce qu'il n'est pas possible d'utiliser cette réduction comme d'un outil pour ensuite classifier de nouvelles données ?

4.2 Conclusion personnelle

Bien que très court, ce stage de découverte a été particulièrement instructif. Il m'a permis de découvrir à la fois le monde de la recherche, aussi bien en éveillant ma critique sur certains aspects mais aussi la découverte de certaines méthodes de travail très souvent utilisées en recherche (comme la lecture d'un papier) mais que je pourrai surement réappliquer dans d'autres types de lecture. A cela s'ajoute la découverte d'un domaine entièrement nouveau et pourtant particulièrement important en ce moment qu'est le machine learning.

Bien que je n'ai fait qu'effleurer le concept et que ça n'a pas toujours été facile de comprendre certaines notions, ne serait-ce que le vocabulaire propre à l'apprentissage automatique et même certains exemples qui demandent de s'imaginer des représentations dans des espaces de très grandes dimensions, je pense avoir énormément appris sur le sujet et je suis sûr que ces connaissances me seront utiles lors de ma poursuite en master.

Références

[1] Arnaud Bailly. Conférence vidéo : Comprendre word2vec - 2016.

[2] Rémi Gilleron. Word to vector. 2016.

[3] Srinivasan Keshav. How to read a paper. *University of Waterloo*, February 2016.

[4] Michael Mitzenmacher. How to read a research paper. *University of Harvard*.

[5] Andrew Ng. Mooc sur l'apprentissage automatique - stanford university - 2016.

[6] Laurens van der Maaten. Conférence vidéo sur t-sne - uc san diego - 2016.

[7] Laurens van der Maaten et Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 2008.