

BUSINESS INTELLIGENCE

PROJET QLIKSENSE: BUSINESS INTELLIGENCE SUR LES PUBLICATIONS DBLP

December 8, 2018

Joseph Gesnouin: 21312478
Université Paris Descartes
Department of Computer Science: LIPADE

Contents

Traitement des données	3
Visualisation	7
Clustering	11
K-means	12
Modèle vectoriel	12
Spherical K-means	13
Choix du nombre de clusters	14
Résultats du clustering et visualisations	15
Bonus: Clustering sur les abstracts	19
Conclusion	23
Bibliographie	24
Code	24

INTRODUCTION

De nos jours, les entreprises doivent faire face à un amas de données provenant de tout types de sources. Arriver à extraire de l'information qui a de la valeur pour l'entreprise et transmettre les résultats sous une forme facilement compréhensible est devenu un problème conséquent dans le monde du travail actuel: Une entreprise capable d'avoir un recul suffisant sur ses données et capable d'en prendre compte lors de ses futures actions aura un avantage conséquent comparé à celles qui ne font pas usage de leur data.

Le rapport s'inscrit dans une démarche de visualisation et d'analyse des données fournies pour résoudre ces problèmes liés à la donnée en utilisant des ressources proposées par DBLP: un site web publiant un catalogue de bibliographies en informatique listant plus de 2,9 millions d'article et considéré comme l'un des moteurs de recherche les plus utilisés dans le monde de la recherche informatique. La masse d'informations utilisée pour le projet n'est pas facile à visualiser sans traiter ces données au préalable. De plus, l'exploitation de ces données peut s'avérer compliquée de part leur taille assez grande.

C'est pour résoudre ces problématiques que les data analysts ont mis en place des outils haut niveau de visualisation, faciles d'accès permettant de résumer et d'obtenir des visualisations facilement compréhensibles. Ces outils leur permettent de communiquer leur résultats aux autres entités de leurs entreprises et ainsi générer de l'information facilement compréhensible pour leur collègues.

Nous nous intéresserons principalement à l'utilisation de QlikSense durant ce rapport: Un outil gratuit de data discovery et de visualisation qui permet à chacun d'analyser facilement ses données pour prendre de meilleures décisions plus rapidement.

TRAITEMENT DES DONNÉES

Nous disposons d'un fichier texte contenant des informations sur des articles scientifiques parus dans des revues et conférences. Ces informations comprennent: le titre de l'article, le ou les auteurs, l'année de publication, le nom de la revue (ou de la conférence), les citations entre articles ou encore leur abstract.

```
##Information geometry of U-Boost and Bregman divergence
#@Noboru Murata,Takashi Takenouchi,Takafumi Kanamori,Shinto Eguchi
#2004
#Neural Computation
#index436405
#%94584
#%282290
#%605546
#%620759
#%564877
#%564235
#%594837
#%479177
#%586607
```

Figure 1: Exemple d'un article et son format de données

Avant toute méthode de visualisation, il aura fallu réorganiser ces informations sous forme de matrices, structures de données plus manipulables qu'un fichier texte où l'on accède aux informations lignes par lignes. À partir de ce fichier texte nous avons générés plusieurs matrices différentes pour nos analyses.

Création du dataframe pour les articles provenant de SIGIR ou de STOC

Le nombre d'articles étant conséquents, nous nous sommes seulement intéressés aux articles provenants de la revue STOC et de la revue SIGIR et avons laissé de coté les articles provenant des autres revues.

Figure 2: Nombres d'articles STOC et SIGIR sélectionnés

Count(Id)

5,828

Le fichier d'origine étant assez bien nomenclaturé, il aura été facile de récupérer les informations sous forme d'un dataframe: les premiers caractères de chaque ligne définissaient le type d'information de la ligne. L'utilisation d'un parser était nécessaire afin de remplir le dataframe des informations nécessaires: Titre, Auteurs, Année, Revue, Id, Liste des citations, Abstract, Nombre d'auteurs et Nombre de citations.

Title	Authors	Year	Venue	Id	ListCitation	Abstract	NbrAuthor	NbrCitation
1 Formal models for expert finding in enterprise corpora.	Krisztian Balog,Leif Azzopardi,Maarten de Rijke	2006	SIGIR	594377	595386 , 362694 , 772628 , 595551 , 26506 , 59477...	Searching an organization's document repositories fo...	3	11
2 Latent Semantic Indexing is an Optimal Special Case o...	Brian T. Bartell,Garrison W. Cottrell,Richard K. Belew	1992	SIGIR	594378	771904 , 2025	Latent Semantic Indexing (LSI) is a technique for repre...	3	2
3 Latent semantic-space: iterative scaling improves pre...	Rie Kubota Ando	2000	SIGIR	594379	937405 , 594378 , 243650 , 594808 , 594831 , 7717...	We present a novel algorithm that creates document v...	1	11
4 Automatic Combination of Multiple Ranked Retrieval S...	Brian T. Bartell,Garrison W. Cottrell,Richard K. Belew	1994	SIGIR	594380	1120095 , 772280 , 595299 , 936910 , 1120350 , 59...		3	6
5 Planning in an Expert System for Automated Informati...	Christine Barthes,Pierre Clize	1988	SIGIR	594381	595655 , 594818 , 594757 , 3446	Searching online databases requires an information r...	2	4
6 The Paraphrase Search Assistant: Terminological Feed...	Peter G. Anick,Suresh Tippineni	1999	SIGIR	594382	594409 , 936932 , 594969 , 594995 , 775344 , 164 ...		2	16
7 Discovering and structuring information flow among...	Joan C. Bartlett,Elaine G. Toms	2003	SIGIR	594383		In this poster, we present a model of the flow of infor...	2	0
8 Iterative Residual Rescaling: An Analysis and General...	Rie Kubota Ando,Lillian Lee	2001	SIGIR	594384			2	0
9 Adapting a Data Organization to the Structure of Stor...	M. B-Cu-00E4#rtsch,Hans-Peter Frei	1982	SIGIR	594385	594586 , 17846	A data organization for information retrieval (IR) syste...	2	2
10 A joint framework for collaborative and content filter...	Justin Basilio,Thomas Hofmann	2004	SIGIR	594386	121997 , 11788	This paper proposes a novel, unified, and systematic ...	2	2
11 Why spectral retrieval works.	Holger Bast,Debapriyo Majumdar	2005	SIGIR	594387	594379 , 594808 , 594833 , 595044 , 595834 , 7756...	We argue that the ability to identify pairs of related te...	2	8
12 Type less, find more: fast autocomplete search with...	Holger Bast,Ingnar Weber	2006	SIGIR	594388	209878 , 544779 , 157582 , 545016 , 545004 , 1120...	We consider the following full-text search autocompl...	2	19
13 Information Retrieval Using a Transportable Natural L...	Madeleine Bates,Robert J. Bobrow	1983	SIGIR	594389	797650 , 777157 , 41492 , 41506 , 108122 , 41709 ...	This paper describes work in progress to develop a fa...	2	7
14 Applying User Research Directly to Information Syst...	Marci J. Bates,Raya Fidel,Efthimis N. Efthimiadis,Ann...	1999	SIGIR	594390			4	0
15 A Probabilistic Model for Distributed Information Retri...	Christoph Baumgarten	1997	SIGIR	594391			1	0

Figure 3: Dataframe créé en fonction des informations du fichier texte

Création de la matrice document-termes pour les titres des articles

Une fois le dataframe précédent créé, il était plus facile de réaliser des traitements de text-mining sur les données structurées. Le premier traitement aura consisté en la création d'une matrice document termes sur les titres des articles, pour dans le futur, s'amuser à comparer les termes récurrents et voir si il était possible de classifier les articles en fonction des mots utilisés pour les définir.

La taille de la matrice étant conséquente et QlikSense n'acceptant pas des fichiers de taille supérieure à 50Mo, il m'aura fallu réduire cette matrice tout en essayant de conserver le plus d'information possible. La matrice étant dans un format articles/termes, comme le but était de réaliser un clustering sur les articles, j'ai préféré préserver toutes les lignes de ma matrice afin de conserver chaque article. La réduction s'est faite sur les colonnes des matrices, à savoir les termes. En utilisant une version de TF-IDF j'ai conservé les 1000 termes les plus porteurs d'informations dans la matrice en espérant que cela puisse conserver une signification suffisante.

TF-IDF permet de donner un poids relatif à chaque mot présent dans un groupe de documents. Cette valeur numérique permet notamment d'évaluer l'importance de chacun des mots présents dans un corpus.

Il n'existe pas de version universelle de la formule de normalisation par TF-IDF. Pour autant, chacune des variantes disponibles dans la littérature a été développée dans un but précis : l'adaptabilité.

Cette méthode de classification de l'importance des mots peut se décomposer en deux parties :

- la normalisation en ligne

- la normalisation en colonne

La normalisation en colonne est réalisée par l'application de l'algorithme IDF. Dans une matrice documents - termes, IDF permet de mesurer à quel point un terme est important.

Il existe de nombreuses variantes de IDF ; certaines sont répertoriées dans le tableau ci-après.

Weighting scheme	IDF weight
unary	1
inverse document frequency	$\log N/n_t = -\log n_t/N$
inverse document frequency smooth	$\log(1 + N/n_t)$
inverse document frequency max	$\log(\max_{t' \in d} n_{t'})/(1 + n_t)$
probabilistic inverse document frequency	$\log((N - n_t)/n_t)$

Figure 4: Différentes variantes de IDF

C'est donc d'après ces valeurs que je me suis basé pour la pondération des mots afin de conserver les mots les plus importants et réduire ma matrice tout en conservant le plus d'information afin de pouvoir l'afficher sur QlikSensse.

Id	acquisition	automatic	methods	proof	the	formal	learning	philosophical	problems	somé	theory	with	abington	challenge	cost	for.	grand	low	mini	c
1	1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
2	2	0	0	0	0	1	1	1	1	1	1	1	0	0	0	0	0	0	0	
3	3	0	0	0	0	1	0	0	0	0	0	0	1	1	1	1	1	1	1	
4	4	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
5	5	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
6	6	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
7	7	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
8	8	0	0	0	0	1	0	0	0	0	0	0	0	0	0	0	0	0	0	
9	9	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
10	10	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
11	11	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	0	0	
13	13	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
14	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
15	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	

Figure 5: Extrait de la matrice Documents-Termes sur les titres des articles

Création de la matrice documents-documents pour les citations inter-articles

De la même manière, une matrice documents-documents pour lister les citations entre articles a été créée afin de nous permettre de visualiser les articles plus populaires par exemple. Une réduction de la taille de la matrice aura également été nécessaire afin de pouvoir uploader ce fichier sur QlikSense.

Id	X12184	X26506	X362694	X594777	X595386	X595551	X595671	X596024	X772628	X935966	X95047	X2025	X771904	X1120247	X243650	x
594377	1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
594378	0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
594379	0	0	0	0	0	0	0	0	0	0	0	1	0	1	1	1
594380	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594381	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594382	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594383	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594384	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594385	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594386	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594387	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594388	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594389	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594390	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594391	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
594392	0	0	0	0	0	0	0	0	0	0	0	1	0	0	0	0
594393	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Figure 6: Extrait de la matrice Documents-Documents pour les citations inter-articles

Création de la matrice Auteurs

Finalement, une dernière matrice aura été générée via les informations du dataframe précédent: la matrice des auteurs, attribuant à chaque auteur les articles auxquels il aura participé.

Iden	Author	Id	dfAuthersArticle
40	Adam J. Grove	40	3166
41	Adam Jatowt	41	5804,5805
42	Adam Kalai	42	2455,3367
43	Adam Klivans	43	3332,3514,3515,3516,5760,5772
44	Adam L. Berger	44	137,138,139
45	Adam L. Buchsbaum	45	2585,2586
46	Adam Meyerson	46	2275,3104,3171,3710,4367
47	Adam R. Klivans	47	4338
48	Adam Smith	48	5736
49	Adam Stepinski	49	1753
50	Adam Tauman Kalai	50	4351,4387,4431,4434,5780
51	Adenike M. Lam-Adesina	51	2476
52	Adi Akavia	52	2132,5724
53	Adi Shamir	53	4104,4106
54	Adish Singla	54	5545
55	Aditya Bhaskara	55	5769
56	Aditya Kumar Sehgal	56	1298
57	Adrian Popescu	57	5130
58	Adrian Vetta	58	4288
59	Adriana Budura	59	2075
60	Adriana Karagiozova	60	4290
61	Adriano Veloso	61	2076

Figure 7: Extrait de la matrice Auteurs

PREMIÈRES VISUALISATIONS

Une fois toutes ces matrices construites, il était intéressant d'avoir un premier jeu d'indicateurs vis à vis des articles sélectionnés: pour se faire une idée du dataset que nous manipulions.

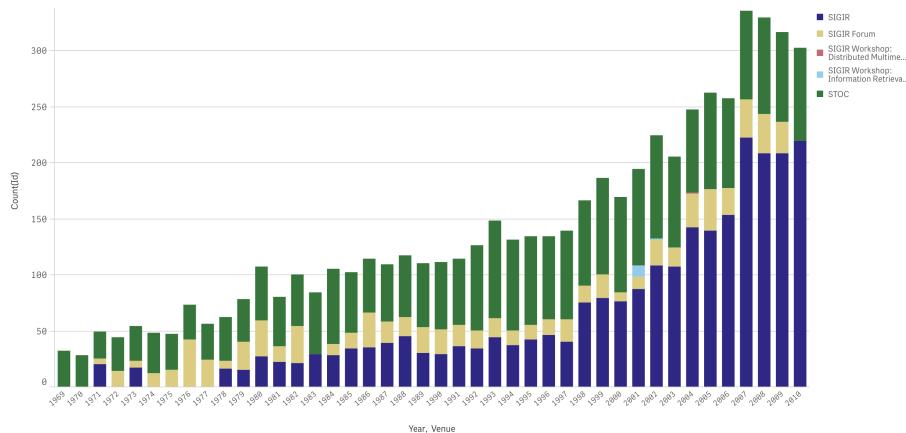


Figure 8: Évolution du nombre de publications par revues et par Années

Ce graphique nous permet d'avoir une idée de la répartition par année de la provenance des articles, mais également de la quantité d'articles produits chaque année. On peut par exemple appercevoir que 2007 a été l'année la plus prolifique toute revues confondues. Visuellement, il est également très clair que la production d'articles scientifiques provenant de ces deux revues ou de leur annexes (par exemple leur forums) croît de manière régulière chaque année.

Concernant la distribution du nombre d'articles par revues & conférences, les deux revues sont plus ou moins à part égales dans cette répartition.

Si l'on considère l'ensemble des publications produites par SIGIR, en incluant les forums et certain de ces Workshop, SIGIR représente environ 56% de la distribution, à contrario, STOC représentera à lui seul 43% des articles.

Figure 9: Distribution du nombre d'articles par revues/conférences

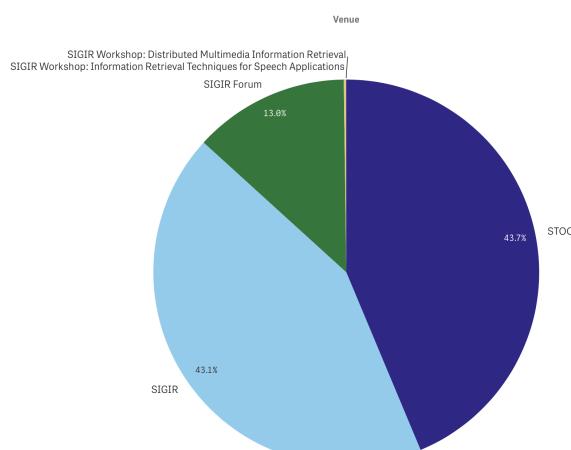
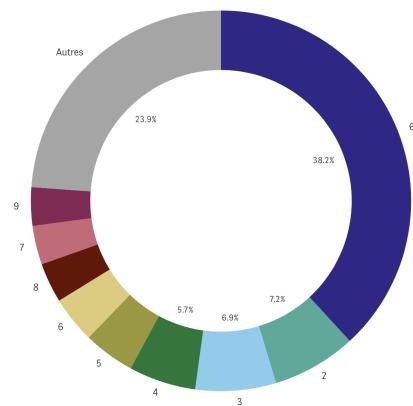


Figure 10: Distribution du nombre de citations



Un indicateur qui peut s'avérer porteur d'information est la distribution du nombre de citations par articles. Cela nous permet d'avoir une idée globale de l'apport de l'état de l'art dans les nouvelles parutions. Comme on peut le constater presque 40% des articles produits ne citent aucun articles de l'état de l'art ce qui semble être assez énorme. Pour expliquer ce résultat j'ai fait le choix d'approfondir ces résultats en affichant la moyenne du nombre de citations par années afin de voir si au fil des ans le nombre de citations augmentait avec la parution d'internet et la facilité d'accès aux articles.

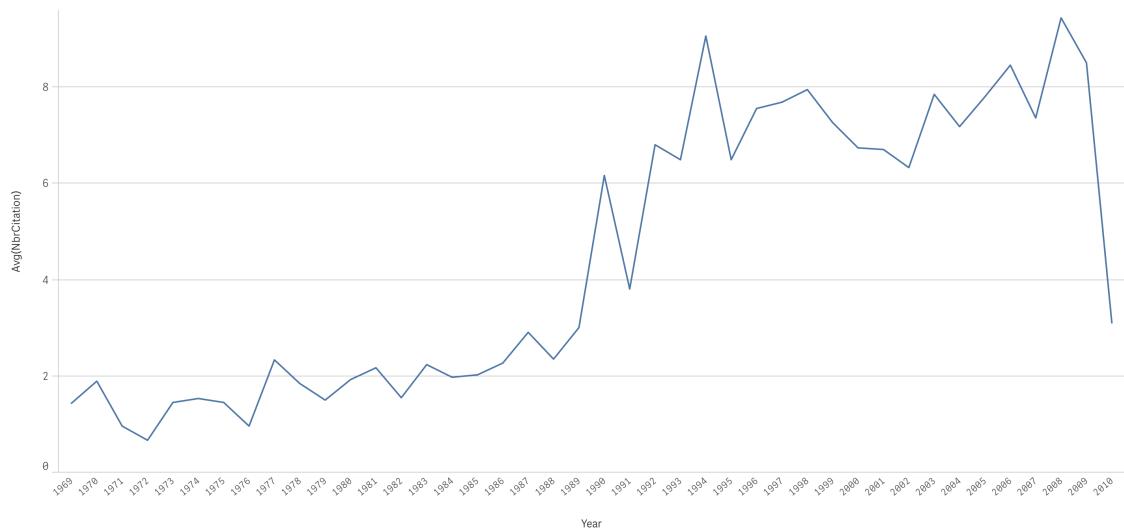


Figure 11: Nombre moyen de citations par années

On remarque qu'au même titre que le nombre de parution d'article croissait au fil des ans, le nombre de citations par article a également grandi au fur et à mesure des années. Ce qui peut s'expliquer par une explosion du monde de l'informatique et des maths appliquées dans les années 80 et donc inéluctablement un nombre croissant de chercheurs se focalisant sur ce domaine. Ce qui entraînera une émulsion du nombre d'articles et de citations, et par conséquent une meilleure connaissance du domaine.

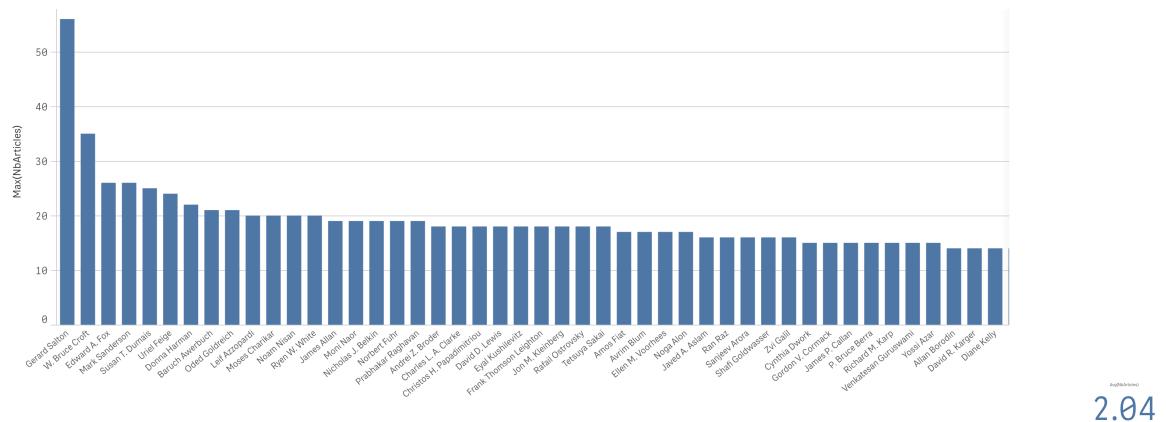
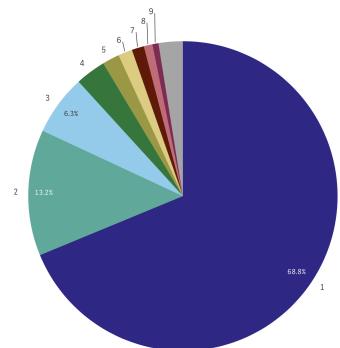


Figure 12: Les auteurs les plus productifs comparés au nombre moyen de production par auteurs

En moyenne, les auteurs repertoriés dans les articles sélectionnés ont écrit 2 articles. On remarque d'ailleurs qu'un énorme tiers de ces chercheurs n'ont en réalité écrit qu'un seul article pour le moment. Cependant, il existe un nombre non négligeable de chercheurs assez productifs: par exemple Gerard Salton, plus de 50 articles présentés à son actif ou bien encore W. Bruce Croft, plus d'une trentaine d'articles.

Petite parenthèse historique, Gérard Salton, un des chercheurs les plus productif dans notre jeu de données, est connu pour être un des inventeurs du modèle vectoriel, méthode algébrique de représentation d'un document visant à rendre compte de sémantique. J'utiliserai ce modèle pour réaliser un sphérical k-means dans la prochaine partie afin de conserver le maximum de sémantique possible lors du clustering de la matrice document-termes.

Figure 13: Distribution d'articles par auteurs



les termes les plus utilisés seront très utiles lorsqu'il s'agira de réaliser la classification des articles par leur noms. Ayant déjà traité la taille des matrices pour pouvoir l'envoyer sur QlikSense, les termes de la matrice sont déjà considérés comme tous porteurs d'information:

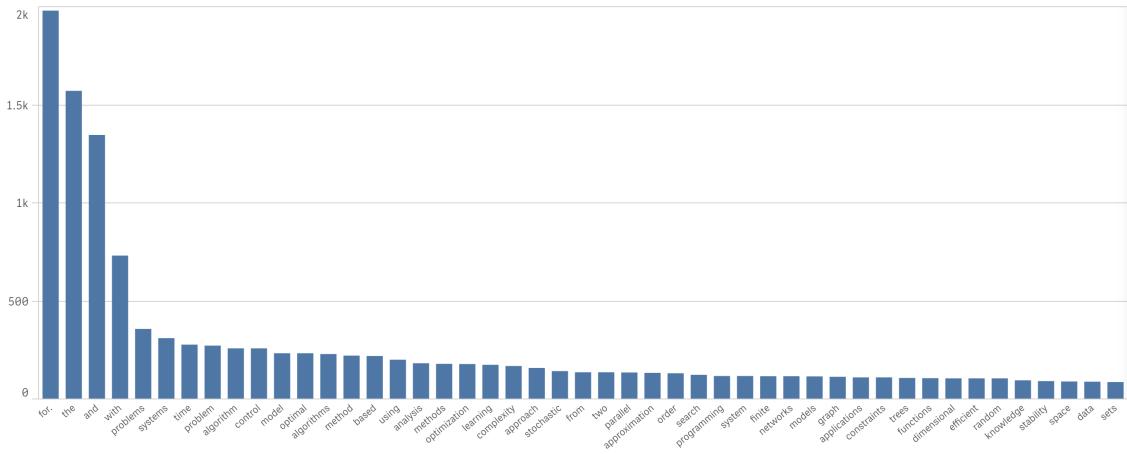


Figure 14: Barplot des termes les plus utilisés

Excepté les trois premiers termes, qui font plus partie du langage courant. On remarque que ce sont souvent des termes généraux relatifs au champ lexical des mathématiques appliquées à l'informatique: algorithme, méthodes, analyses, system, problem, time... Il sera donc nécessaire d'utiliser tf-idf sur cette matrice avant le clustering afin de ne pas tomber dans le piège de la classification des termes génériques.

Finalement, pour conclure ces premières visualisations rapides, il est intéressant de voir quels sont les articles les plus populaires:

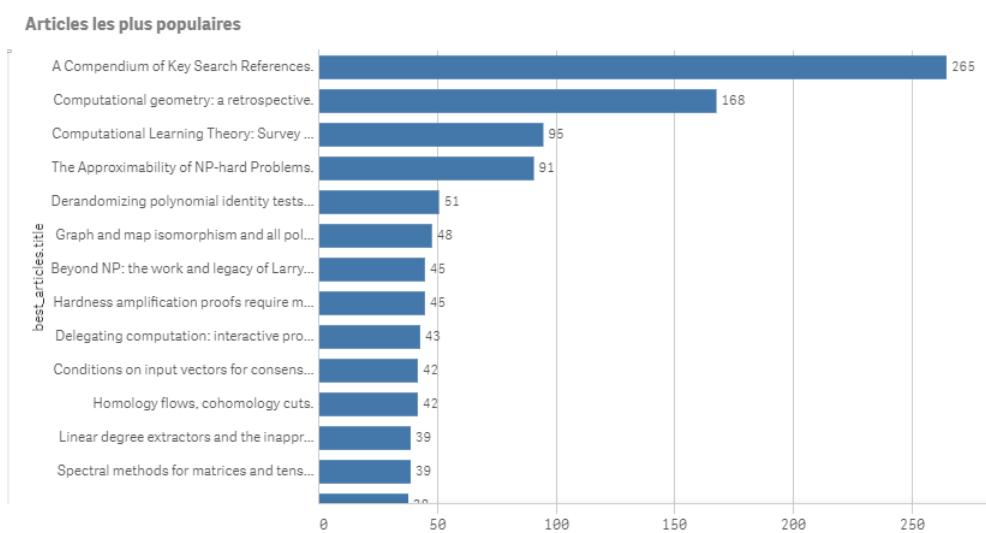
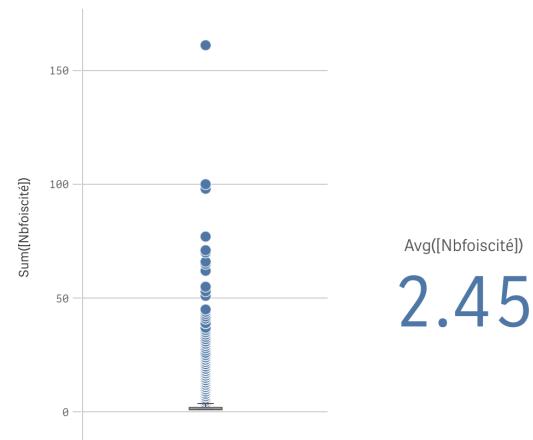


Figure 15: Barplot des articles les plus cités

En moyenne, on remarque qu'un article est cité 2.45 fois. La boxplot correspondante est assez explicite, à part quelques cas extrêmes, la majorité des articles sont peu cités, voire très peu. Il en découle une petite quantité d'articles beaucoup cités qui font figure de repère au niveau de l'état de l'art, et toute une panoplie d'articles s'inspirant de ces travaux préalables en les citant. Ces articles sont peu cités à leur tour mais, dans certains domaines, un papier peut faire figure d'autorité scientifique pendant des décénies. C'est ce qui explique pourquoi certains de ces articles sont énormément cités.

À titre d'exemple, un des articles assez cité, dans les données STOC et SIGIR est "A Language Modeling Approach to Information Retrieval." Après une brève recherche sur Google Scholar il s'avère que ce papier a été cité des milliers de fois toutes revues confondues. Cet article porte sur l'indexation de document et la récupération d'informations provenant de cette indexation. Problème assez commun à tout type de système d'information et c'est pour ça qu'il est autant cité: c'est un problème générique, qui pose problème à une multitude de personnes manipulant les systèmes d'informations. Si l'on regarde le diagramme en barres précédents, les titres des articles figurant dans les articles les plus cités sont souvent des articles portant sur des sujets vastes comme le un des problèmes du millénaire et lié à la théorie de la complexité le calcul de complexité d'un problème NP-hard.

Figure 16: Boxplot du nombre de citation par articles et moyenne



CLUSTERING

La seconde partie du rapport consiste en un clustering des articles grâce à la matrice documents-termes générée. J'ai décidé de ne pas utiliser que l'algorithme du K-means mais une variante conservant plus la sémantique des données textuelles: Spherical K-means.

Dans cette section, sont présentés ces deux algorithmes d'apprentissage automatique assez populaires dans la littérature scientifique pour le clustering. Ces deux méthodes se basent sur le même principe afin de partitionner en groupes homogènes des objets décrits par un ensemble de propriétés : le calcul de centroïdes.

K-means

Soit un ensemble de points $S = (x_1, x_2, x_3, \dots, x_n)$. On cherche à partitionner ces n points en k sous-ensembles tels que $S = (s_1, s_2, s_3, \dots, s_k)$ (avec $k < n$), de sorte à minimiser la distance entre les points à l'intérieur de chaque partition.

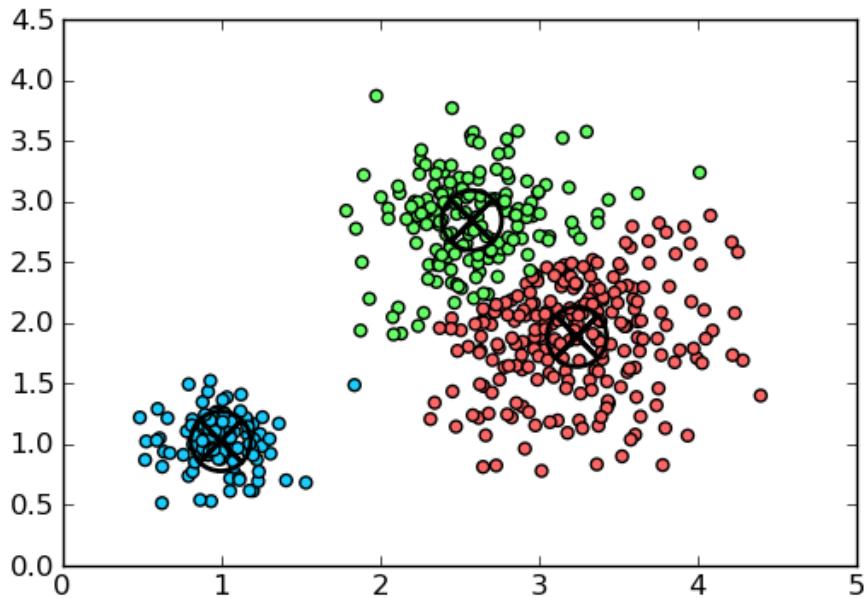


Figure 17: Exemple de clustering avec kmeans

L'algorithme est itératif : à chaque itération les centroïdes se déplacent de façon à minimiser la distance euclidienne intra-classe et à maximiser la distance inter-classe.

On définit la convergence de l'algorithme au moment où les centroïdes ne se déplacent plus et restent immobiles : on obtient alors la position des centroïdes ainsi que l'affectation de tous les objets à leur ensemble respectif.

Modèle vectoriel

Il est possible d'introduire la notion d'espace vectoriel sur l'espace des documents en langage naturel. Mathématiquement parlant, il est donc possible d'arriver à quantifier la proximité sémantique entre chaque document, en introduisant des mesures de comparaisons plausibles, principalement via la distance angulaire.

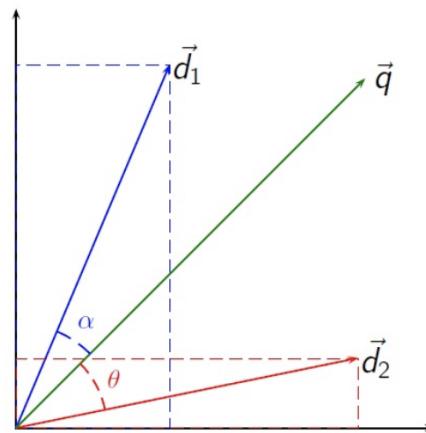


Figure 18: Représentation de deux documents et d'une requête

Le modèle vectoriel permet de modéliser la proximité d'une requête q vis à vis des documents d_1 et d_2 via les angles α et θ . Cette méthode de représentation des données permet à la fois de regrouper les documents similaires sémantiquement parlant et ce, de manière non supervisée.

Bien que relativement simple à implémenter, il est nécessaire de garder à l'esprit les limites d'un tel modèle:

- la qualité du modèle généré dépend principalement de la qualité des données initiales: vocabulaire et pondération des termes présents dans les documents.
- Le modèle est extrêmement sensible à l'utilisation des synonymes: deux textes traitant de la même chose avec des termes différents ne se verront pas classés dans le même sous groupe.
- Les termes sont considérés comme étant statistiquement indépendants ce qui est rarement le cas dans le langage courant
- L'ordre d'apparition des termes dans les documents n'est pas conservé

Spherical K-means

Cette version, dérivée de l'algorithme de base K-means, est beaucoup plus optimisée pour le traitement et l'analyse des données textuelles. Spherical K-means nécessite l'utilisation d'un vector space model communément appelé un sac de mots : le corpus. Celui-ci est représenté sous forme de matrice documents termes.

Contrairement à l'algorithme K-means, la fonction de minimisation de Spherical K-means ne prend pas en compte la distance euclidienne mais la distance angulaire, grâce au cosinus:

$$\sum_i (1 - \cos(x_{ij} p_{c(i)}))$$

En choisissant de représenter des valeurs textuelles sous le format vectoriel, il faut inévitablement choisir une distance angulaire afin de les comparer. Il s'avère que cette distance propose de bien meilleurs résultats que les autres distances puisqu'elle conserve généralement mieux la valeur sémantique des textes et n'est pas affectée par leurs longueurs. En effet, pour classifier des documents textuels, il est plus raisonnable de comparer les mots qui les composent plutôt que leurs tailles.

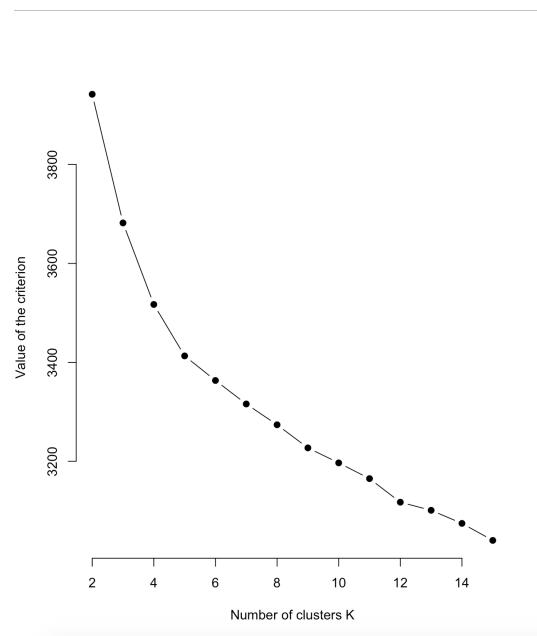
En prenant en compte toutes ces limites et principalement celles du modèle vectoriel, je ne m'attend à avoir un clustering excellent vis à vis de la matrice documents-termes générée. Pour autant, il est indéniable que le résultat obtenu sera meilleur qu'avec l'utilisation d'un simple k-means.

Choix du nombre de clusters

Afin de déterminer le nombre de clusters à garder sur les valeurs, je me suis inspiré de la méthode d'Elbow: il s'agit de tracer la courbe entre le nombre de clusters et la fonction de coût afin de déterminer un "coude". J'ai donc du réaliser des spherical K-means pour des rangs allant de 2 à 15 afin de savoir à quelle valeur de k nous allions nous intéresser. Cette méthode regarde le pourcentage de variance qu'apporte chaque cluster.

En règle générale, on détermine le nombre de clusters à choisir au moment où rajouter un autre cluster n'apporte pas réellement d'information pour modéliser les données. Les premiers clusters apporteront énormément d'information car ils expliquent beaucoup de variance.

Figure 19: Résultat de la méthode d'Elbow sur spherical k-means : nombre de cluster variant de 2 à 15



À un certain moment, le gain marginal devient tellement faible qu'il en est négligeable, cela se caractérise par une virage en forme "d'angle" au graphe. Le nombre de clusters est déterminé par ce point de "coupure". Dans ce cas précis, le coude n'était pas flagrant, j'ai donc eu recours à un calcul pour trouver le rang k : en considérant que ce rang était défini comme le point avec la plus grande distance orthogonale depuis les extrémités. Avec cette méthode, un nombre de clusters de 4 avec la méthode Spherical K-means sera utilisé pour les visualisations à venir.

Résultats du clustering et visualisations

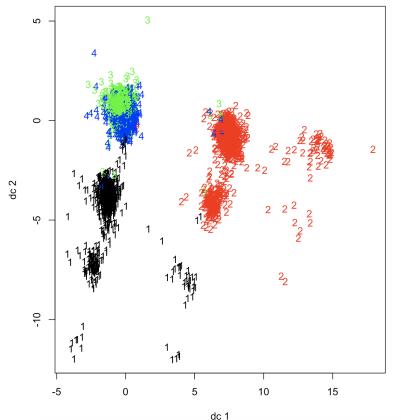


Figure 20: Spherical Kmeans: 4 clusters

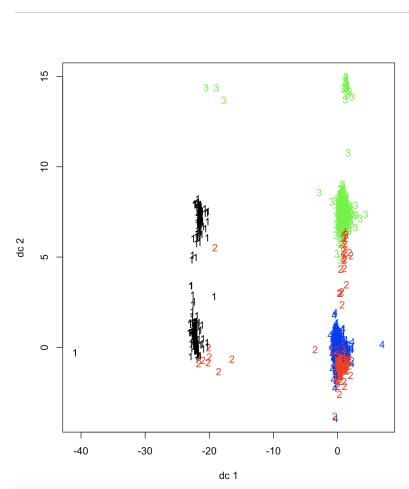


Figure 21: Kmeans: 4 clusters

Ci-dessus, la visualisation du mapping dans un espace vectoriel de dimension réduite des points de la matrice documents-termes ainsi que le groupe qui leur a été attribué en fonction des deux algorithmes de clusterisation utilisés: Kmeans et Spherical.

Figure 22: Tableau comparatif des résultats du clustering, kmeans et Spherical kmeans

```
+ table(k$cluster,y$cluster)
```

	1	2	3	4
1	42	29	334	760
2	59	835	324	10
3	83	0	1140	35
4	109	2	4	2063

Ci-contre, une comparaison des deux résultats pour chacun des points afin de voir si les deux algorithmes distinguent les mêmes points pour leur cluster: on peut facilement remarquer que certains groupes font office de groupe "majoritaires" probablement avec les termes génériques tels que "for", "the" et "and".

En ce qui concerne la répartition des revues en terme de clusters, la répartition est assez uniforme et représentative de la distribution que nous avions trouvée dans la première partie: Aucun des clusters ne se distinguent énormément en ce qui concerne leur distribution par revues.

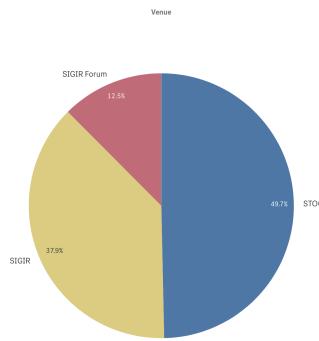


Figure 23: Distribution du nombre d'articles par revues: cluster 1

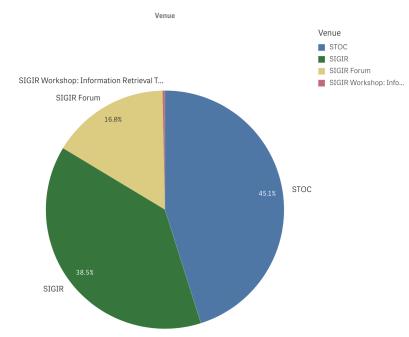


Figure 24: Distribution du nombre d'articles par revues: cluster 2

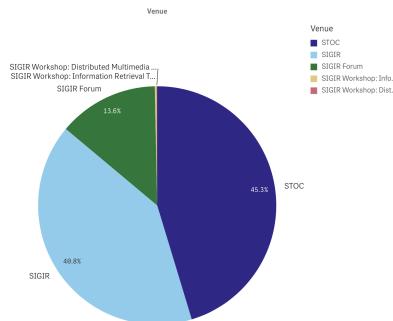


Figure 25: Distribution du nombre d'articles par revues: cluster 3

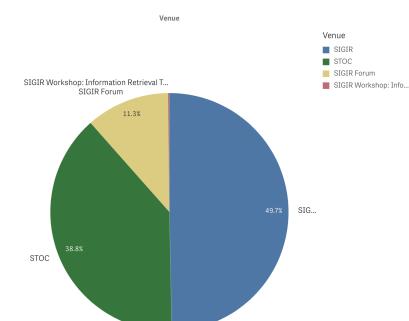


Figure 26: Distribution du nombre d'articles par revues: cluster 4

On en conclue donc que les deux revues SIGIR et STOC partagent sur les mêmes sujets et qu'il semble compliqué de déterminer si un article provient d'une des deux revues avec pour seule information son titre. SIGIR correspond à l'anagramme de Special Interest Group on Information Retrieval et respectivement, STOC correspond à Symposium on Theory of Computing. Ce sont donc deux revues portant sur des sujets des systèmes d'informations assez vaste et cela explique donc cette difficulté à regrouper les articles par leur revue.

Lorsque nous avions regardé plus en détail les termes les plus fréquents sur la totalité des articles sélectionnés, nous avions surtout retrouver des termes génériques appartenant au champ lexical de la computer science. Ci-dessous, quatres graphiques représentant respectivement, les termes les plus fréquents de chacun des clusters réalisés par le spherical k-means afin de voir quels sont les termes récurrents qui ont servi à la classification non supervisée de tout ces articles.

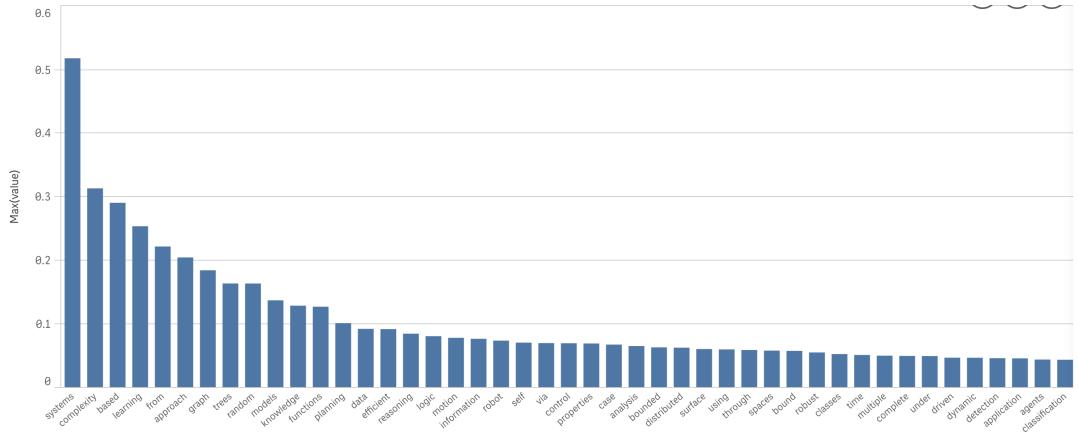


Figure 27: Barplot des termes les plus fréquents cluster 1

Il semblerait que les articles construisant le premier cluster correspondent à un groupe d'articles portant sur la gestion de l'information et la structuration de celle-ci: ainsi on y retrouve des termes tels que graph, trees, knowledge, data, systems.

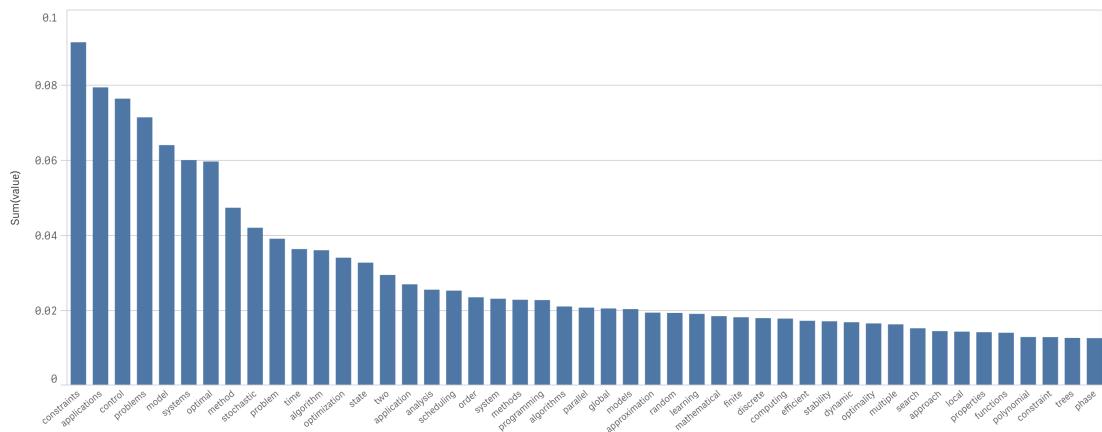


Figure 28: Barplot des termes les plus fréquents cluster 2

Le second cluster correspondrait à un groupe d'algorithme pour lequel l'optimisation et le temps de calcul est important, donc des articles traitant sur la complexité. Ainsi on y

retrouve des termes tels que: optimal, time, optimization, parallel, computing, dynamic, polynomial... Sans approfondir l'analyse, il est impossible de déterminer exactement à quoi chacun des clusters correspond, mais à première vue, celui-ci semble se focaliser sur le temps et la compléxité plus que les trois autres.

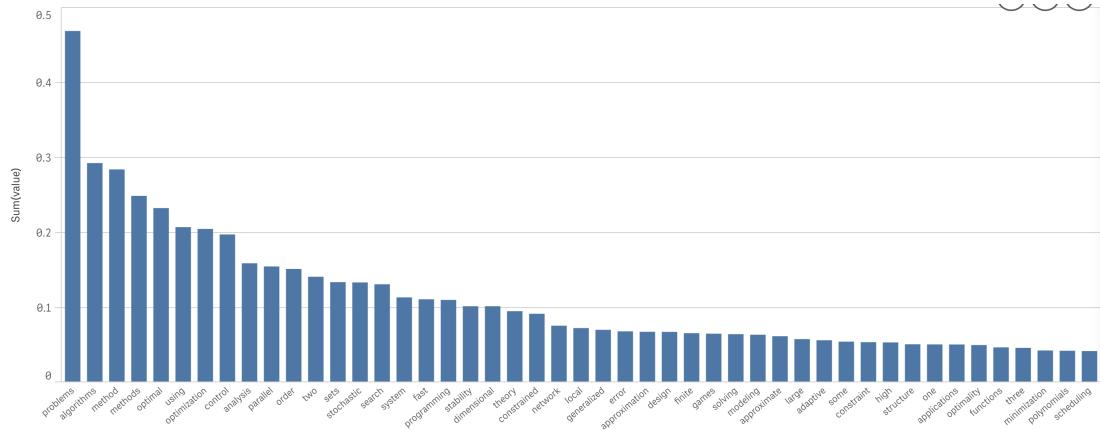


Figure 29: Barplot des termes les plus fréquents cluster 3

Le troisième cluster semble utiliser des termes assez génériques tels que problem, algorithms, method, methods, optimal, using. À eux seuls, ces mots représentent plus de 80% de la totalité de la répartition de l'importance des mots pour le cluster. Il semblerait que ce cluster soit une sorte de cluster "poubelle". Regroupant tous les titres d'articles dont le nom est très générique et pas du tout explicite.

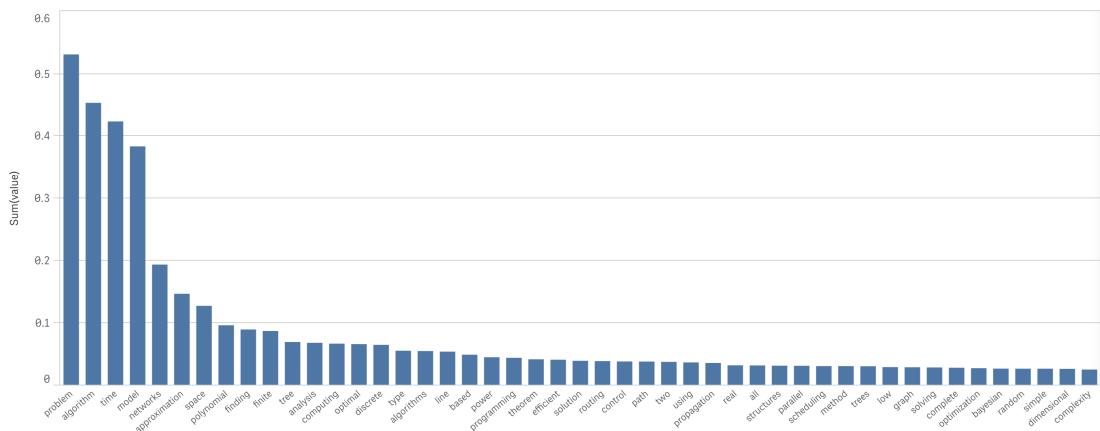


Figure 30: Barplot des termes les plus fréquents cluster 4

Le dernier cluster semble être assez similaire au troisième cluster: celui-ci est défini par

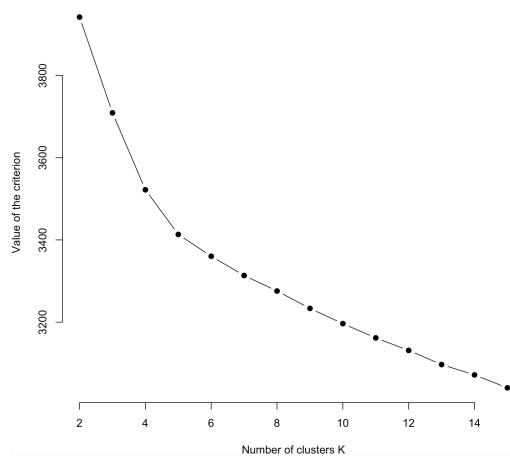
une majorité de termes génériques tels que problems, algorithm...

Tandis que le cluster 3 est majoritairement catégorisé par l'utilisation de "problem", celui ci l'est par "problems". Ceci est l'incarnation d'un des inconvenients de l'espace vectoriel: les synonymes sont mal gérés par ce genre de structures de données et le nombre de synonymes en IT pour catégoriser les mêmes choses est énorme. De ce fait, lorsque l'on réalise le spherical kmeans, celui-ci traite ces deux mots comme fondamentalement et sémantiquement différents alors qu'ils portent sur le même sujet. Au final certains de ces groupes auraient pu être réunis, mais ces algorithmes étant basés sur des calculs de distances angulaires et considérant chacun des mots comme indépendants, de l'information se perd.

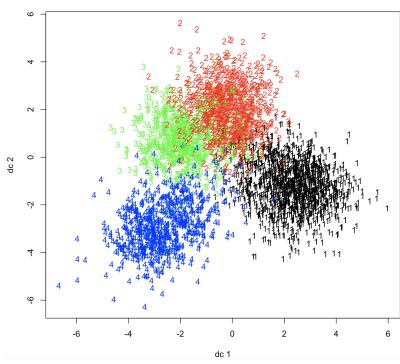
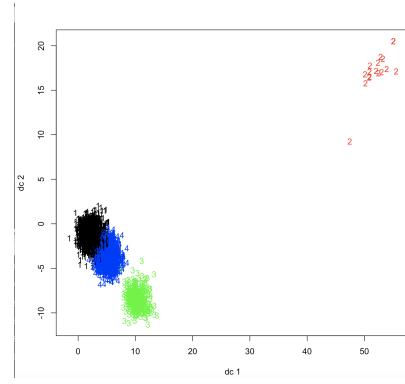
BONUS: CLUSTERING SUR LES ABSTRACTS

Cette dernière partie consiste en une comparaison du clustering généré précédemment grâce aux titres des articles et d'un autre clustering réalisé sur les abstracts des mêmes articles afin de voir lequel était le plus précis. De part leur taille, les abstracts comparés à la taille des noms d'articles pour le premier clustering, nous pouvons déjà supposer que ce clustering sera plus efficace que celui sur les titres: les abstracts étant plus longs et plus détaillés envers le sujet de l'article que seulement son titre.

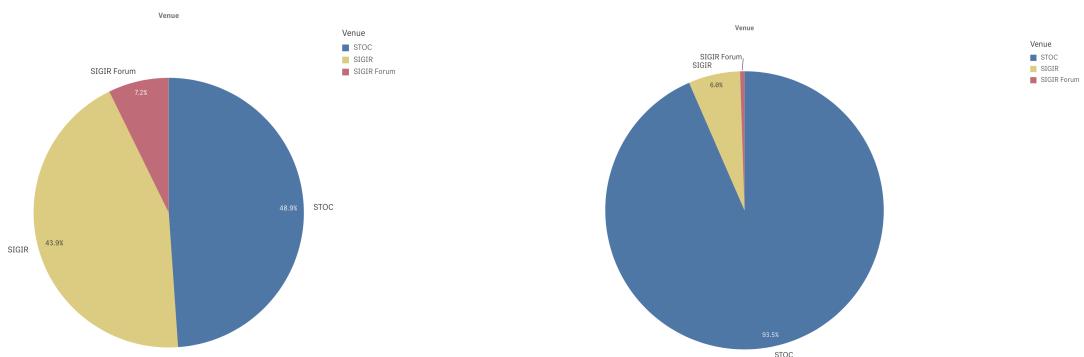
Figure 31: Résultat de la méthode d'Elbow sur spherical k-means : nombre de cluster variant de 2 à 15



De la même manière que pour le clustering sur les titres des articles, je me suis inspiré de la méthode d'Elbow afin de trouver le nombre de cluster le plus adéquat à mon problème avec Spherical K-means. On remarque que l'on retrouve le même nombre de clusters sur cette matrice documents-termes portant sur les abstracts que sur les titres des articles. Ce qui semble rassurant. Les abstracts représentant des courts textes souvent réutilisant les mots clés utilisés dans les titres de chaque article pour présenter le but de l'article plus en profondeur.

**Figure 32:** Spherical Kmeans: 4 clusters**Figure 33:** Kmeans: 4 clusters

De même sorte, la projection du spherical K-mean semble encourageante. Les groupes semblent être suffisamment distincts, et l'on peut visualiser assez facilement à l'oeil nu la distinction de chacune des classes potentielles. En ce qui concerne le K-means, il semble qu'en terme de distances euclidiennes, le groupe 2 est très éloigné des autres et aurait peut-être du être considéré comme un groupe d'outliers, à ceci faut-il raison garder et peut-être considérer ce groupe comme un groupe avec les abstracts les plus petits: K-means réalisant des calculs sur des distances euclidiennes, cela revient simplement à calculer la taille des documents et donc, la seule information dont on peut être assurés à la vue de cette projection dans un espace de dimension réduit est que certains des articles ont un abstract fondamentalement plus petits que les autres.

**Figure 34:** Distribution du nombre d'articles par revues: cluster 1**Figure 35:** Distribution du nombre d'articles par revues: cluster 2

Contrairement au clustering sur les titres des articles, les clusters ici semblent assez bien regrouper les articles par leur revues, ainsi les groupes 3 et 4 semblent regrouper la majorité

des articles provenants de la revue SIGIR. Le cluster 2 est majoritairement composé d'articles provenants de la revue STOC. Et pour finir le cluster 1, semble ne pas réellement récupérer des articles provenant d'une certaine revue. Pour autant, le spherical k-means sur les abstract nous aura permis de classifier de manière non supervisée de manière convaincante les articles et leur provenance comparé au précédent spherical k-means sur les titres des articles.

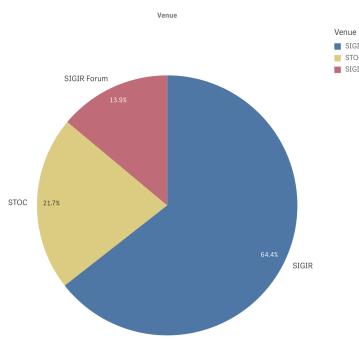


Figure 36: Distribution du nombre d'articles par revues: cluster 3

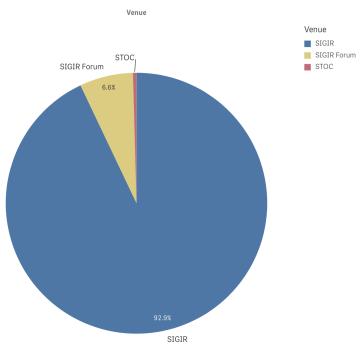


Figure 37: Distribution du nombre d'articles par revues: cluster 4

Comme pour le clustering réalisé sur les titres des articles, le problème de la représentation choisie est telle, que les synonymes et les mots relatifs au domaine de l'IT y sont mal gérés. Ainsi le cluster 1 semble regrouper des termes génériques de computer science tel algorithm, results, based, number, data. Ce cluster fait figure de cluster de stockage "poubelle". Le cluster 2 quand à lui semble être assez spécialisé avec des termes tels que: information, retrieval, search, paper, text, query. Dans ce groupe, on retrouve vraiment le champ lexical de l'accès aux données et au stockage de l'information.

Le cluster 3 semble être relativement lié au champ lexical du temps, de manière similaire à l'un des clusters que nous avions trouvé pour les titres: times,log,complexity. Finalement, le cluster 4 regroupe le champ lexical de la performance et des résultats: effectiveness, performance. Ce sont sans doute des articles basés sur des études comparatives ou le calcul de performance comparé à d'autres algorithmes y est omniprésent.

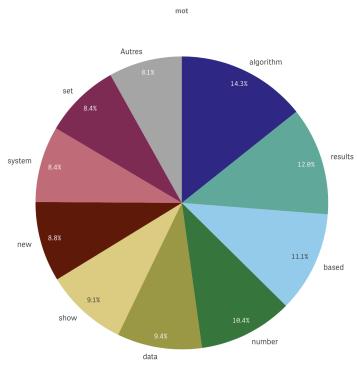


Figure 38: Distribution des termes les plus fréquents: cluster 1

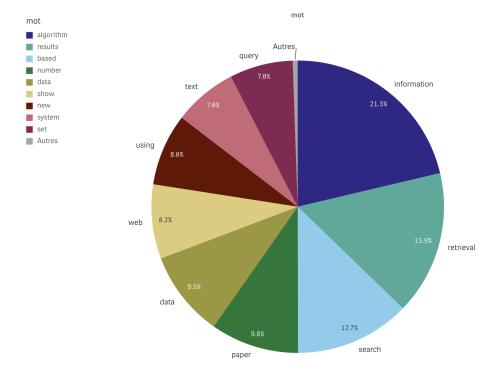


Figure 39: Distribution des termes les plus fréquents: cluster 2

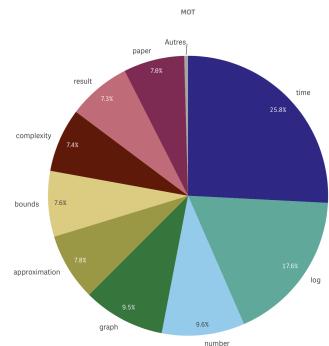


Figure 40: Distribution des termes les plus fréquents: cluster 3

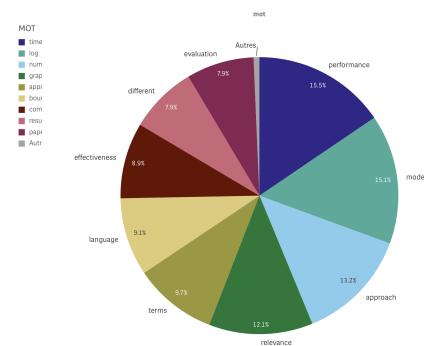


Figure 41: Distribution des termes les plus fréquents: cluster 4

CONCLUSION

Durant ce projet, j'ai eu l'occasion de me familiariser avec l'outil QlikSense, afin de réaliser des visualisations faciles d'accès et résumant bien le sujet que nous devions traiter. La nature de ce projet m'a permis de traiter la donnée presque de bout en bout: DBLP ayant déjà extrait les données, j'ai eu l'occasion de l'aggrégner, de la préparer et finalement: la visualiser.

Une fois ces étapes fondamentales réalisées, j'ai eu l'occasion de réaliser des algorithmes de classification non-supervisés sur deux des matrices précédemment générée afin d'essayer d'en tirer des conclusions vis-à-vis des articles grâce à leur titre et leur abstract.

Malheureusement, le vocabulaire informatique étant rempli de synonymes, les algorithmes que j'ai choisi pour mon clustering n'ont pas réussi à capturer la proximité de certains mots tels que problem/problems, optimal/optimum... Le modèle vectoriel est un outil puissant, malheureusement celui-ci dépend trop de la qualité des données initiales et est extrêmement sensible à l'utilisation de synonymes. Mes clusters ne sont donc pas trop significatifs mais ont le mérite de m'avoir aidé à comprendre le fonctionnement de QlikSense, un outil de visualisation utile pour partager les résultats de data mining à des gens non initiés au monde de la donnée. En tant qu'apprenti travaillant sur les séries temporelles, les technologies que nous utilisons chez Orange sont InfluxDB et Grafana, mais avoir l'occasion de manipuler d'autres outils est toujours formateur.

Si j'avais eu plus de temps, j'aurai bien aimé approfondir ce travail de text-mining en utilisant des algorithmes conservant l'ordre d'apparition des termes et capables de détecter les synonymes. De telle sorte que les clusters portent plus d'informations que ceux présentés dans le rapport.

Il aurait donc été intéressant de rajouter aux deux méthodes K-mean et Spherical K-mean des algorithmes tels que des mesures statistiques d'association de mots comme la PMI ou encore LSA / PLSA afin d'obtenir des groupes plus représentatifs.

Bibliography

- [1] Kurt Hornik, Ingo Feinerer, Martin Kober, Christian Buchta *Spherical k-Means Clustering*.
- [2] Jiming Peng, Jiaping Zhu. *Refining Spherical K-Means for Clustering Documents*(2006).
- [3] Yohei Seki *Sentence Extraction by tf/idf and Position Weighting from Newspaper Articles*(2002).
- [4] Akiko N. Aizawa *An information-theoretic perspective of tf-idf measures*(2003).
- [5] R Core Team,R Foundation for Statistical Computing,2014 *R: A Language and Environment for Statistical Computing*.

CODE

<https://github.com/JosephGesnouin/BIM2/blob/master/BIprojet.R>