



Données

Passer des documents à des données



Trace, enregistrement, donnée

La trace

- Une proposition de caractérisation :
 - Une trace est un élément dont le statut causal est en rupture avec son environnement.
- Exemple :
 - Trace comme résidu : des substances qui restent, mais elles n'ont rien à faire là ; elles appartiennent à une histoire qui n'est pas celle du contenu qui contient ces traces ;
 - Trace comme empreinte ou indice : la trace a été produite, causée, par un actant qui n'est plus là, actant qui introduit une causalité qui n'est pas celle de l'environnement (branche cassée par le passage d'un gibier, trace de pas, etc.).



Statut de la trace

- La trace est un « ce qui reste », un intrus qui manifeste par sa présence une réalité qui n'est plus.
- La trace est une contemporanéité, elle est là, qui révèle un non contemporain révolu.
- La trace est un passé qui ne passe pas, mais révèle un passé qui est passé, révolu.
- La trace est donc d'emblée temporelle, porteuse d'une mémoire et d'un passé.



La donnée

- Une proposition de caractérisation :
 - Une donnée est une proposition affirmant un fait, dont l'expression est formatée et l'interprétation formalisée.
- Exemple :
 - L'enregistrement élémentaire d'une base de données (!)
 - Un triplet RDF
 - Une cellule d'un tableau Excel
 - Etc.



Statut de la donnée

Est décrétée

- Est définie en vue d'un traitement d'information, ce que l'on se donne pour calculer: « les données du problème (à traiter) »
- n'est pas le point d'aboutissement ou point d'arrivée d'un processus l'ayant produit.

Est « donnée »

- Par la collecte de la donnée : la donnée est ce que l'on se donne, non ce que l'on reçoit ;
- la donnée n'est pas un don du réel, mais l'instauration d'un réel.

Est réduite à sa manipulabilité

- Elle est définie pour être manipulable dans un calcul : format, sémantique, opérationnalisation.
- Ne pas compter des choux et des carottes, car justement on peut le faire...

Remplace le réel dont elle tient lieu

- N'ayant pas d'origine, elle est le réel sur lequel opère le calcul. D'où la revendication d'empiricité de l'actuelle IA alors qu'on est en plein formalisme coupé du réel...

Une donnée est :

➤ Une assertion formatée et formalisée minimale :

➤ Assertion :

- C'est une affirmation à propos du monde, l'expression dans un langage d'un fait du monde ;

➤ Formatée :

- Elle respecte une syntaxe dont la machine sait faire quelque chose ;

➤ Formalisée :

- Elle respecte une syntaxe formelle lui donnant une sémantique, une signification associée ;

➤ Minimale :

- Ses parties ne sont pas des données.

➤ Exemples :

➤ un triplet RDF

- [La Joconde]-(a_pour_auteur)->[Léo]

➤ Une cellule d'un tableau Excel

➤ La ligne d'un fichier en CSV

➤ Une série de tokens (caractères, pixels, etc) dans un document :

- C'est l'affirmation de leur présence dans un contenu
- C'est le contenu comme fait, comme la somme des faits / tokens dans lesquels on le décompose et l'analyse.

Dit autrement

Description élémentaire pour l'humain

- Description : ce qui se rapporte à quelque chose pour en dire quelque chose. « La révolution française a eu lieu en 1789 » est une description, un dit donc, qui se rapporte à un événement du monde.
- Élémentaire : c'est la plus petite partie d'une description pour qu'elle soit signifiante et pourvue d'une valeur de vérité.

Description élémentaire pour la machine

- Description : suite de symboles conforme à un format de description
 - Suite de caractères, pixels, d'enregistrement d'une base de données, etc.
- Élémentaire : la plus petite partie dont la machine sait faire quelque chose quand elle connaît le format ;
 - Le caractère, le pixel, une cellule excel, etc.

Une donnée est une description élémentaire d'une réalité. (Wikipédia)

Formatée, interprétée, formalisée

➤ Description formatée :

- Suit un format permettant d'associer une suite de bits / d'octets à un format dont l'enjeu est de permettre de construire une expression perceptible pour l'humain
 - Format : unicode, jpeg, etc. : l'octet devient un caractère, une couleur, un pixel, etc.
 - « la révolution française a eu lieu en 1789 » : format = texte
 - [Revolution_Française]-(date)->[1789] : format ≈ RDF

➤ Description interprétée assertant une proposition :

- Structurée et labellisée de telle manière que l'on puisse associer une sémantique (signification) du type : $x = y$
- Le format est donc mobilisé dans :
 - une structure technique :
 - Cellule Excel; triplet RDF, ligne d'une base de données
 - Une convention de lecture :
 - E.g. : lire le premier mot comme un prénom, le deuxième comme un nom.
- L'objectif est de tirer de la donnée l'affirmation du type : Date(révolution française) = 1789, ou la date de la Révolution française est 1789.

➤ Description formalisée

- Idem mais la convention de lecture est une sémantique formelle associée à la syntaxe permettant à la machine, en manipulant des formats, de faire des inférences sur le contenu.
 - Ontologies et raisonneurs (cf 3e journée).

Remarque

➤ On peut avoir :

- Un fichier de format texte (un simple logiciel de texte, comme TextEdit peut l'ouvrir).
- Le contenu de ce fichier est structuré formellement selon un format tabulaire ;
 - La structure formelle n'est pas un format connu par la machine, mais une convention d'écriture.
 - Cela revient à un pseudo-format, ou un format sur-générique (le texte) dont la convention de lecture permet de le transformer en format contrôlé d'un Excel par exemple, ou une base de données.

➤ Exemple :

➤ Le format CSV (Coma Separated Values)

- Prenom, Nom, Email, Age, Ville
Robert, Lepingre, bobby@exemple.com, 41, Paris
Jeanne, Ducoux, jeanne@exemple.com, 32, Marseille
Pierre, Lenfant, pierre@exemple.com, 23, Rennes
- On a 5 données (description élémentaire) par ligne.

Prenom	Nom	Email	Âge	Ville
Robert	Lepingre	bobby@exemple.com	41	Paris
Jeanne	Ducoux	jeanne@exemple.com	32	Marseille
Pierre	Lenfant	pierre@exemple.com	23	Rennes

Exemple : la tokenisation dans les LAg

➤ Soit le texte suivant :

- `text = ""`
- English and CAPITALIZATION
- 🎵 鸟
- `show_tokens False None elif == >=`
`else: two tabs:" " Three tabs: " "`
- `12.0*50=600`
- `""`

➤ Intérêts:

- Traiter tous les types de caractères / symboles
- Réduire la variabilité des formes en les réduisant à des formes élémentaires (token) que l'on peut combiner.
- 100 000 (GPT4), 50000 (GPT2)

➤ Sa tokenisation (GPT 4) :

English and CAPITAL IZATION

🎵 鸟

show tokens False None elif == >= else : two tabs : " "

Three tabs : " "

12 . 0 * 50 = 600

➤ Fonctionnement :

- Apprentissage
- Dépend des langues
- Paramétrages variés

Des traces aux données

- On s'empare des traces pour en faire des données :
 - Les traces sont issues d'un passé révolu ;
 - Les données sont ce qu'on se donne en vue d'un traitement d'information.
- Les données sont orientées vers le futur du traitement à entreprendre, les traces sont la rétention d'un passé révolu.
 - Les données sont l'amnésie de la trace.



L'enregistrement

- Ni une trace (car déjà une donnée)
- L'enregistrement n'est pas un reste, mais une transformation qui engendre un nouvel objet selon un format propre :
 - Signal électrique à partir de vibrations sonores ou lumineuses
 - Codes numériques à partir de cellules (natif) ou signaux (numérisation)
- Même produit de manière mécanique et automatique, l'enregistrement est un arbitraire imposé à une origine.
- Ni une donnée (car encore une trace)
- L'enregistrement n'est pas une pure convention ni un pur décret :
 - On se le donne à partir d'un donné, d'un fait.
- Même définie en fonction de la manipulabilité attendue et de la résolution espérée, la donnée issue d'un enregistrement emporte avec elle un résidu, une trace de son origine.

Exemple : l'archive du Web

➤ Démarche :

- Captation par crawling sur les sites où l'on simule l'activité d'un internaute :
 - On récupère les contenus via les requêtes HTTP déclenchées
- Conservation de cette captation dans son format d'enregistrement
- Restitution sous la forme d'une consultation Web avec transformation à la volée du contenu enregistré.

➤ Cette captation est partielle et partielle :

- Mise à jour du site pendant la captation ;
- Fréquence de la captation différente selon la profondeur des pages ;
- Format d'enregistrement propre à l'archive et différente de l'architecture du site (format DAFF par exemple à l'Ina).

Le statut d'une archive du web

➤ La trace d'un enregistrement, trace que l'on transforme en données

C'est-à-dire :

➤ La mise en donnée de la trace d'un enregistrement :

➤ Enregistrement :

➤ Captation d'un état du Web pendant une période de collecte

➤ Trace :

➤ Cet enregistrement reste dans le temps, il est la trace de la captation effectuée.

➤ Données :

➤ Cet enregistrement devient une donnée pour l'interprétation et pour construire une compréhension du monde du Web et du monde reflété par le Web.

L'archive du web: un web ré-inventé

- 3 objectivités à composer : données, trace, enregistrement ;
- 3 régimes d'objectivité
 - Pour l'enregistrement :
 - Conséquence physique (causalité physique, mesure et objectivité mécanique)
 - Rejouer mécaniquement l'archive
 - Pour la trace :
 - Ressemblance (analogie, archétype et objectivité comme vérité d'après nature)
 - Catégoriser sémiotiquement ce que l'on voit
 - Pour la donnée :
 - Conséquence formelle (codage, déduction et objectivité du jugement exercé)
 - Dédire le monde dont l'archive est l'archive.

DIKW : données, information, connaissance,
sagesse



Données et Information

Données

- La donnée est une inscription formelle formatée.
 - Du fait de son format, elle est manipulable mécaniquement ;
 - Du fait de son formalisme, elle peut recevoir une interprétation, plus ou moins riche.
 - Problème de l'expressivité du format et de son interprétation implicite ou formelle.

Information

- L'information est la donnée dans le contexte d'autres données formatées.
- Les formats peuvent être différents, pour représenter le même contenu :
 - Nécessité d'harmoniser les différents formats qui agissent ou portent sur le même contenu
 - Problème de l'interopérabilité

Information et connaissance

information

- L'information est la donnée dans le contexte d'autres données formatées.
- Les formats peuvent être différents, pour représenter le même contenu :
 - Nécessité d'harmoniser les différents formats qui agissent ou portent sur le même contenu
 - Problème de l'interopérabilité

Connaissance

- La connaissance est la donnée dans un contexte d'interprétation (humain) ;
- Les contextes peuvent être différents, et pas explicites.
- Seuls les êtres humains savent interpréter selon le contexte pertinent du moment.
 - Problème du sens

Connaissance et sagesse (ce qui échappe à la machine)

Connaissance

- L'humain interprète les données selon le contexte
- La machine n'accède pas à la connaissance, mais seulement à sa représentation sous forme de données formatées.

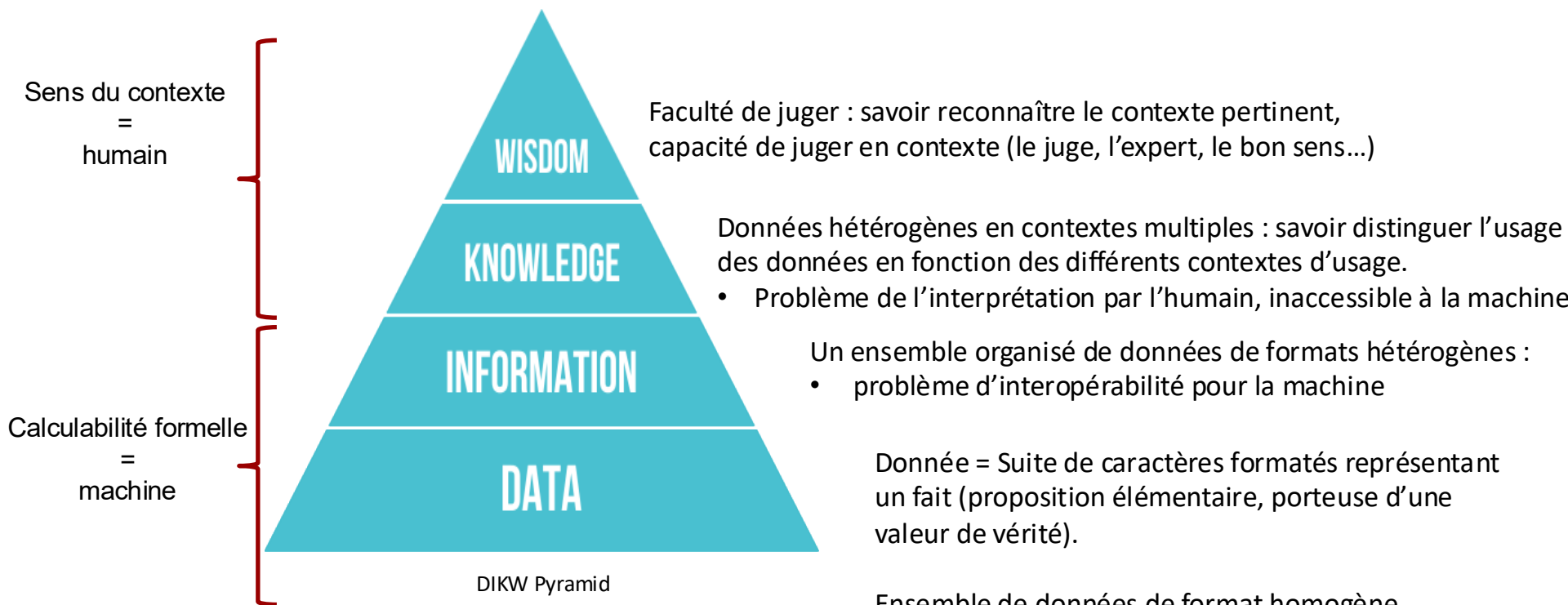
Sagesse

- Ce n'est pas un contenu ni une information, c'est le sens du contexte, savoir définir quel contexte retenir pour telle information :
 - Sens du jugement et du discernement.

Complexités

- Les données :
 - Format unique et cohérent
 - Complexité analytique de leur masse ;
 - Problème de la calculabilité
 - Traitement possible : Visualisation des données
- Les informations :
 - Formats hétérogènes et incohérents entre eux
 - Complexité synthétique de leur hétérogénéité
 - Portent sur des informations de même signification :
 - Problème de l'interopérabilité
 - Traitement possible : Ontologies comme interopérabilité sémantique
- Les connaissances :
 - Formats hétérogènes, relevant de contextes d'interprétation différents
 - Complexité herméneutique de leur interprétation
 - Problème du sens
 - Traitement possible : pas de solution technique, modélisation du système humain-technique en intégrant la place de l'humain et le rôle des contextes de son intervention. C'est l'ingénierie des connaissances qui rapporte les connaissances à des représentations informatiques manipulables par la machine, i.e. des connaissances à des informations.

DIKW: Données, Informations, Connaissances, Sagesse



Des contenus aux données : trois approches



Structurer, indexer, analyser

Enjeu

- On dispose de contenus qu'on veut pouvoir retrouver et transformer :
 - Ces contenus sont des expressions d'intentions éditoriales et autoriales
 - Maximiser la possibilité d'inscription et d'expression
 - Ces contenus doivent se prêter à la gestion documentaire :
 - Maximiser la conformité et leur formatage pour optimiser la manipulation
- Ces objectifs sont contradictoires :
 - Plus on autorise de modes d'expression, moins on sait optimiser leur manipulation ;
 - Plus on optimise la manipulation, plus on contraint et réduit l'expressivité.

Alors, on fait quoi ?

- Enrichir les documents :
 - On récupère les documents comme ils sont, et on se débrouille (on les indexe) pour les manipuler / retrouver à partir de ces index ;
 - Deux approches :
 - L'indexation
 - Le balisage documentaire (une forme d'indexation)
- Formaliser les documents en données :
 - On réduit les contenus à des données et on fait des requêtes formelles (comme une base de données) pour retrouver ce que l'on cherche ;
- Analyser les documents en tokens élémentaires :
 - On tokenise les contenus pour chercher des corrélations ou des visualisations pour faire émerger des propriétés globales.

Trois manières de faire -1 : formaliser l'information structurée

- Possibilité de ramener l'information à un format bien maîtrisé et facilement manipulable :
 - Formalisme pour exprimer les informations : schéma de base de données
 - Outils exploitant ce formalisme.
- On dépense de l'énergie au départ pour structurer l'information, mais la recherche est plus simple, rapide, efficace.
- Conditions et limites :
 - Le format possède l'expressivité nécessaire;
 - Le format permet une manipulation correspondant à la signification de ce qui est exprimé :
 - Ce qu'on manipule a du sens
 - Ce qui a du sens est manipulable
- Paradigme : base de données ;

Trois manières de faire - 2 : indexer l'information non structurée

- Information dans une forme complexe difficile à formater dans un formalisme logique
 - Texte, vidéo, son, etc.
- Information récupérée sous une forme non numérique ou numérisée, donc dans des formats non différenciés (des pixels, des caractères, etc.)
- On ne structure pas a priori, on indexe a posteriori:
- Ajouter des index associés au contenu, qui permettent de le retrouver et de le manipuler.
 - Peu coûteux au départ (on ne structure pas), coût élevé à la fin : indexation et recherche moins efficace.
- Conditions et limites :
 - S'applique à toutes ressources
 - Nécessite un enrichissement a posteriori reposant sur des index / balises / annotations formalisées et structurées.
- Paradigme : document XML, indexation documentaire, Google

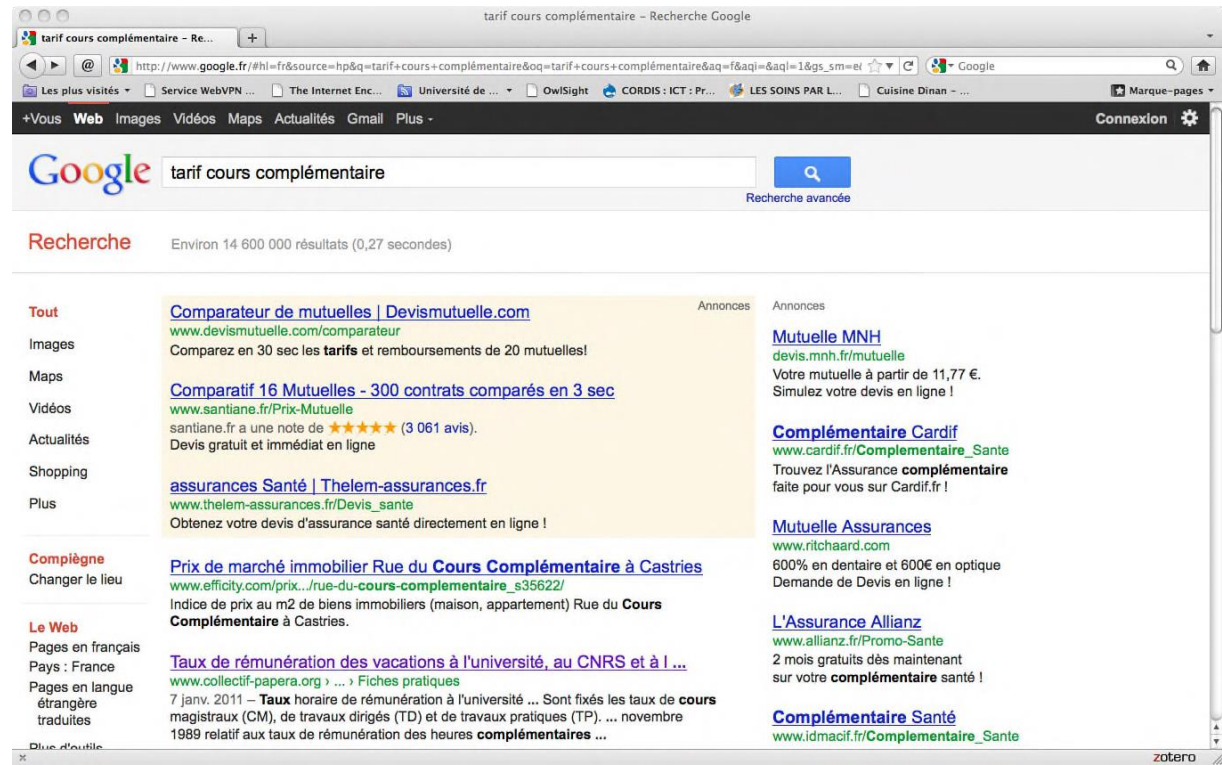
Trois manières de faire – 3 : analyser l'information formatée

- Le contenu est une ressource numérique dont le format permet l'analyse :
 - L'analyse des contenus permet de passer des flux de pixels, de caractères, etc. à des structures reconnues dans le contenu : c'est une locution, c'est un visage, ce visage est telle personne, c'est un plateau télé, etc.
 - C'est là où les modèles d'IA (machine et deep learning) sont utiles et intéressants.
- Les données secondaires créées peuvent être reliées à d'autres données issues d'autres sources (e.g. Wikidata) ;
- L'analyse peut aussi porter sur les données extraites des ressources (les locuteurs d'une vidéo) et les données formatées dès l'origine (Wikidata).
- Objectif :
 - Tout doit être réduit à des données formatées et manipulables
 - L'analyse établit des corrélations ;
- Paradigme : Big data, Cultural analytics, IA générative.



Enrichir le document : indexation

Approche indexée ou indexation

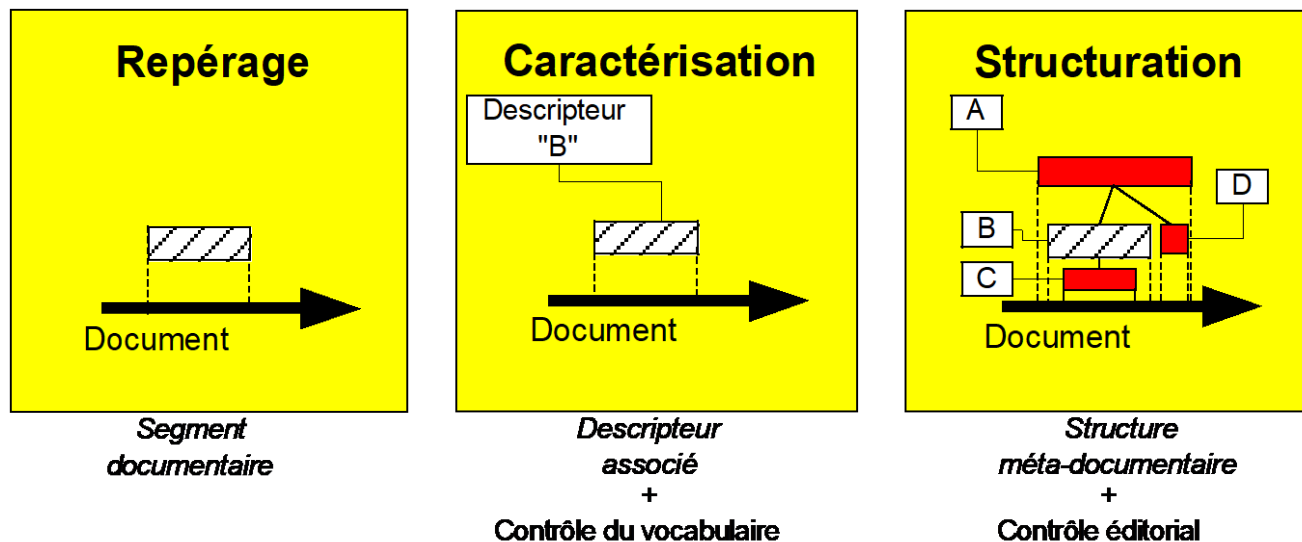


3 étapes fondamentales

- Comment représenter un document ?
 - Indexation ;
- Comment exprimer une requête ou le besoin d'information ?
 - Formulation de la requête ;
- Comment apparier la requête avec les représentations des documents ?
 - Recherche ;

Les 3 étapes de l'indexation

- Paraphraser le document en langue en se basant sur une stratégie de description formalisée.
- L'indexation sémiotise le document et y repère des données.
- Trois étapes



La localisation

- Le problème est de définir ce qui fait signe et fait sens dans un document :
 - Un signe est une unité proposant une forme se dégageant d'un fond.
 - Un signe est une unité *signifiante*, i.e. qui signifie un signifié.
- Le problème est donc de repérer les unités signifiantes :
 - Il n'existe pas a priori d'unités signifiantes
 - Les unités signifiantes résultent d'une interprétation.

Le signe est le résultat du parcours interprétatif et non sa condition

La qualification

- Il s 'agit de définir la signification associée aux unités signifiantes localisées:
 - Pour qu'il y ait sens, il faut qu'il y ait un système de signes où ceux ci se définissent les uns par rapport aux autres ;
 - Les unités signifiantes doivent par conséquent constituer un système pour qu'il y ait une qualification possible.

- La qualification est la condition sous laquelle on est capable d 'attribuer une valeur, une utilité à un document; elle justifie son exploitation.

La structuration

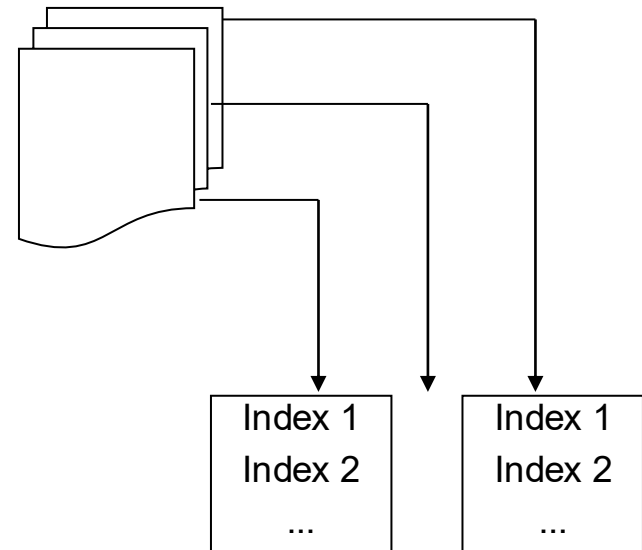
- Il s'agit de rassembler les différentes qualifications effectuées en une documentation:
 - Notice documentaire ;
 - Métadonnées ;
- L'enjeu est d'articuler la structure des qualifications sur la structure du document décrit.

Typologies de l'indexation

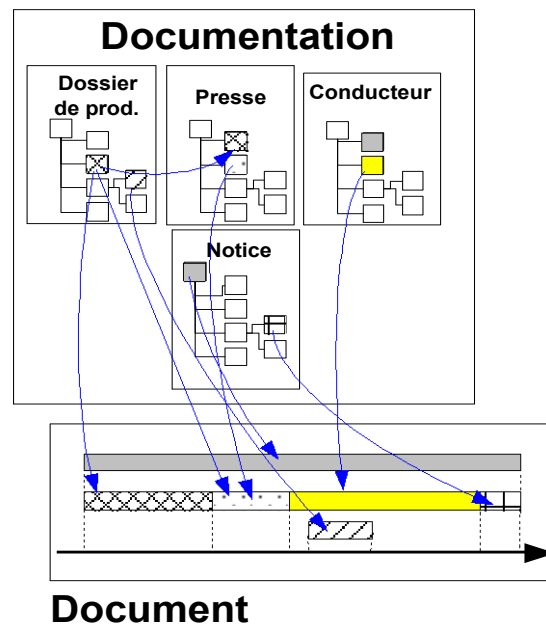
	Libre	Contrôlée
Assignment	Indexation sauvage	Indexation documentaire
Extraction	Extraction terminologique text mining Google	Word spotting

Indexation documentaire : document non numérique

- Contexte : support non numérique
- Les UTM sont les unités documentaires physiques : livres, cassettes, documents.
- Inutile de dégager des USI plus fines, car elles ne sont pas manipulables.
- UTM=USI=document
- L'indexation reste globale



Indexation documentaire : document numérique



Dans le cas de Google...

➤ Représentation des contenus :

- Chaque mot des documents constitue un index : il représente un aspect du contenu et on veut pouvoir retrouver le document grâce à ce mot, l'idée étant que le document doit en parler puisqu'il le contient
 - Techniquement, le mot est repéré comme une chaîne de caractères non blancs séparée par des caractères blancs
 - Certains mots sont repérés comme non pertinents, d'autres sont agrégés (expression), etc.

➤ Formulation de la requête

- Tout terme qu'un internaute saurait écrire

➤ Recherche d'information

- Retrouver les textes dont les index (les mots) correspondent aux requêtes (termes de la requête).



Enrichir le document: le balisage textuel

Pour commencer...

- Déjà, quand on veut manipuler des données, on a donc :
 - Le format: c'est ce que la machine appréhende ;
 - La forme sémantique : c'est l'interprétation qu'on lui associe.
- Le problème est que la machine ne peut aborder la forme sémantique que via le format :
 - Si la forme est une simple convention de lecture, on l'a dans le... baba (non mais..., un peu de décence quand même...)
 - Il faut donc s'assurer que la manipulation de la donnée par la machine respecte le sens que l'on donne à la donnée : il faut articuler format et forme, format syntaxique et forme sémantique.

Donc trouver un compromis

... entre ce que l'on veut dire, ce que l'on dit à la machine et ce qu'elle doit faire.

Ainsi faut-il distinguer ce que l'on dit au chien...

« Sors ta truffe des poubelles !

Tu entends Toby !?!

Sors ta truffe des poubelles ! "



De ce que le chien comprend...

« Bla bla bla bla bla bla bla !

Bla bla bla Toby !?!

Bla bla bla bla bla bla bla ! »



De même ...

Distinguer ce que l'on écrit sur le web...

« il existe un article intitulé 'le web sémantique n'est pas antisocial' »



De ce que l'ordinateur comprend...

➤ « bla bla bla bla bla bla bla ' bla bla bla bla bla' »



Enrichir ce que comprend la machine

- Définir des termes que la machine connaît, sait reconnaître et manipuler.
 - Lister et organiser ces termes et définir des balises à partir de ces derniers ; pour chaque terme ainsi déterminé, on aura :
 - Balise ouvrante : < terme>
 - Balise fermante : </terme>
- Baliser le contenu incompréhensible par la machine par des termes compréhensibles pour la machine
 - Intégrer les balises dans le contenu pour encadrer le contenu par ces balises.
 - <terme> bla bla bla </terme>

Paradigme documentaire-1

- La machine ne comprend qu'une suite de caractères.
- Pour qu'il en soit autrement, on va introduire les balises :
 - Un terme connu par la machine, signalé par des « < » et des « > »,
 - Une balise peut être ouvrante ou fermante.
 - Ce terme donne la signification à associer à ce qui est entre les balises ouvrantes et fermantes.
 - Exemple :
 - <nom> Bachimont </nom>
 - Ainsi enrichit-on le format textuel par des balises.

Jean Dupont

3, avenue des Tilleuls

99620 Nulle Part

Le dossier 32 doit être traité
par M. Durand pour dans 15
jours.

Paradigme documentaire-2

- La machine comprend qu'il y a deux suites de caractères qui constituent chacune un paragraphe
 - On peut demander à la machine de lister tous les paragraphes, à l'envers, etc.
 - Mais on ne peut pas lui demander de lister les noms propres (Dupont et Durand)

```
➤ <p>
➤ Jean Dupont
➤ 3, avenue des Tilleuls
➤ 99620 Nulle Part </p>

➤ <p>
➤ Le dossier 32 doit être traité par M.
  Durand pour dans 15 jours.
➤ </p>
```

Paradigme documentaire-3

- Toutes les suites de caractères pertinentes sont séparées et encadrées par des balises correspondant à des types reconnus par la machine :
 - Elle peut lister les noms, les prénoms
 - Retrouver le numéro de dossier, en faisant la différence entre un nombre comme 99620 qui est un CP et 32 qui est un n° de dossier.
 - Pour les deux, ce ne sont que des suites de caractères pour la machine si on ne les précise pas via les balises.

```

<p>
  <prenom>Jean</prenom>
  <nom>Dupont</nom>
  <numero>3</numero>,
  <typeVoie>avenue</typeVoie> des
  Tilleuls
  <cp>99620</cp> <ville>Nulle
  Part</ville> </p>
<p>
  Le dossier
  <nbDossier>32</nbDossier> doit
  être traité par M.
  <nom>Durand</nom> pour dans
  <delais>15</delais>
  <uniteDuree>jours</uniteDuree>.
</p>

```

Paradigme « données »

- La vision est poussée à la limite :
 - Il n'y a plus que les balises
 - Il n'y a plus de textes encadrés par les balises.
- Le document est totalement structuré :
 - Possible si on peut formaliser totalement l'information sous la forme de champ (le type, comme un nom), et des valeurs associées (comme une chaîne de caractère qui commence par une majuscule).
 - En général, impossible, car il faut laisser toutes les variations possibles pour exprimer une même information, et donc utiliser une balise qui ne varie pas pour encadrer un contenu qui peut varier de manière non prévisible ou indéfinie.
- Le document n'est plus une ressource, mais est devenu un ensemble de données.

```
<personne prenom="Jean"  
          nom="Dupont"/>  
<adresse numero="3"  
          typeVoie="avenue"  
          nomVoie="rue des Tilleuls"  
          cp="99620"  
          ville="Nulle Part" />  
<tache  
          dossier="32"  
          responsable="Durand"  
          delais="15"  
          uniteDuree="jours" />
```

Paradigme formel ou « logique »

contenu

Jean Dupont

3, avenue des Tilleuls

99620 Nulle Part

Le dossier 32 doit être traité
par M. Dupont pour dans 15
jours.

Reformulation logique pour le traitement du contenu

```
[Personne:#1] -  
  (a-pour-nom) -> [Nom: Dupont];  
  (a-pour-prenom) -> [Prenom: Jean];  
  (a-pour-adresse) ->  
    [adresse:#2] -  
      (numero) -> [integer: 3];  
      (nomrue) -> [string: "Nulle Part"];  
      (typeVoie) -> [token: avenue];  
      (cp) -> [CodePostal: 99960].
```

Bon, ça paraît simple...

- On a :
 - Ce que l'humain veut/peut comprendre : du texte incompréhensible pour la machine, qui le traite comme la donnée la plus élémentaire qui soit, un flux de caractères (ce n'est déjà pas si mal) ou un flux binaire.
 - Ce que l'on précise à la machine pour guider son traitement : les balises qu'on ajoute.
- Pour cela déterminer :
 - Quelles balises on choisit ;
 - Comment elles sont définies pour la machine.
- Problème :
 - Il faut se mettre d'accord !
 - Et là, ça redevient compliqué...

Interopérabilité : Ce que l'on peut voir...

WWW2002

The eleventh international world wide web conference

Sheraton waikiki hotel

Honolulu, hawaii, USA

7-11 may 2002

1 location 5 days learn interact

Registered participants coming from

australia, canada, chile denmark, france, germany, ghana, hong kong, india, ireland, italy, japan, malta, new zealand, the netherlands, norway, singapore, switzerland, the united kingdom, the united states, vietnam, zaire

Register now

On the 7th May Honolulu will provide the backdrop of the eleventh international world wide web conference. This prestigious event ...

Speakers confirmed

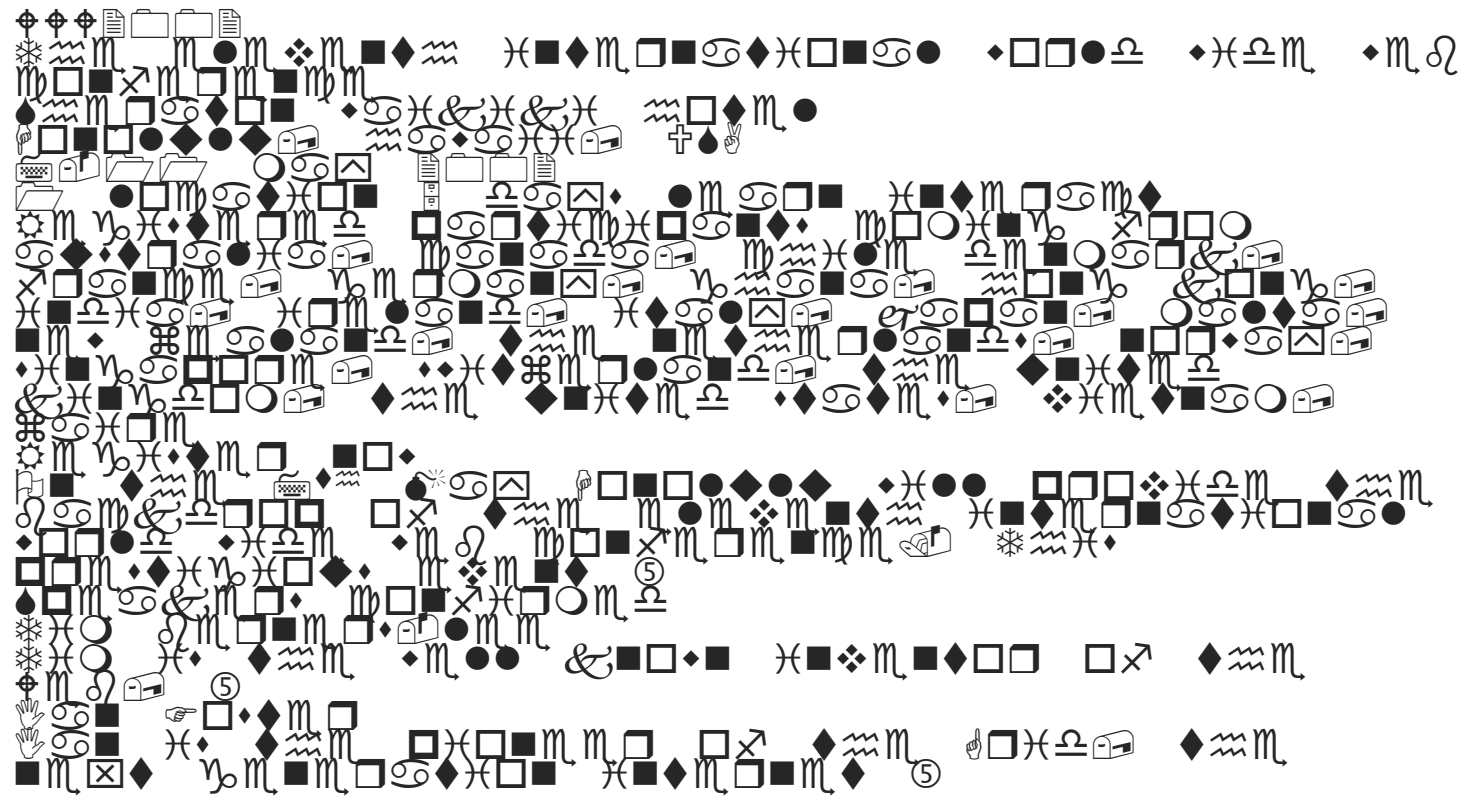
Tim berners-lee

Tim is the well known inventor of the Web, ...

Ian Foster

Ian is the pioneer of the Grid, the next generation internet ...

Interopérabilité: ce que la machine comprend...



© 2013 Pearson Education, Inc. or its affiliate(s). All rights reserved. This publication is protected by copyright. Any unauthorized distribution or reproduction of this work is illegal. All other rights reserved. Printed in the United States of America. 10 9 8 7 6 5 4 3 2 1

Bruno Bachimont - La transformation numérique des documents

Mais que dire de...

<conf>⌘⌘⌘⌘⌘⌘
 ✱⌘⌘⌘ ⌘●⌘✱⌘■◆⌘ ✱■◆⌘□⌘◆✱□⌘⌘◆ ●□□●◆ ✱✱◆⌘ ◆⌘⌘⌘□⌘◆</conf>
 <place>⌘⌘⌘⌘⌘⌘⌘ ◆✱✱✱✱✱✱ ✱□◆⌘●
 ♢□□□◆◆◆◆◆ ✱◆◆✱✱◆ ♣♣</place>
 <date>📅📅📅📅 📅📅📅📅</date>
 <slogan>📁 ●□⌘◆✱□■ 📅 ◆◆◆● ⌘⌘⌘□⌘⌘◆</slogan>
 <participants>☼⌘⌘✱◆⌘□⌘◆ □⌘◆✱⌘⌘□⌘◆◆ ⌘□○✱⌘⌘ ✎□□○
 ◆◆◆□◆◆✱◆◆ ⌘◆◆◆◆◆◆ ⌘⌘⌘◆●⌘ ◆⌘□○◆◆◆◆ ✎□◆◆⌘⌘◆
 ⌘⌘□○◆◆◆◆ ⌘⌘◆◆◆◆ ✱□⌘⌘ &□⌘⌘◆ ✱◆◆◆◆◆ ✱□⌘●◆◆◆◆
 ✱◆◆◆◆◆ ◆◆◆◆◆ ○◆◆◆◆◆ ■⌘◆ ✱⌘◆◆◆◆◆ ◆⌘⌘
 ■⌘◆⌘⌘□●◆◆◆◆◆ ■□□◆◆◆◆ ◆✱⌘⌘◆□□□⌘◆◆ ◆✱◆✱⌘□●◆◆◆◆ ◆⌘⌘
 ◆◆◆◆⌘◆ &✱⌘⌘◆◆□○◆◆ ◆⌘⌘ ◆◆◆◆◆◆ ◆◆◆◆◆◆ ◆✱⌘◆◆◆◆◆
 ✱◆◆□⌘◆</participants>
 <introduction>☼⌘⌘✱◆⌘□ ■□◆
 📅 ◆⌘⌘ 📅◆◆◆ ♣□□□◆◆◆◆ ◆✱●● □□□◆✱◆⌘ ◆⌘⌘ ⌘◆◆◆◆◆◆
 □✎ ◆⌘⌘ ⌘●⌘✱⌘■◆⌘ ✱■◆⌘□⌘◆✱□⌘◆◆ ◆□□●◆ ◆✱◆⌘ ◆⌘⌘
 ⌘□■✎⌘□⌘⌘⌘◆ ✱⌘⌘◆ □□⌘◆✱⌘□◆◆ ⌘◆⌘■◆ ⑤
 ◆□⌘◆◆⌘◆◆ ⌘□□✎✱□○⌘◆◆</introduction>
 <speaker>✱✱○ ⌘⌘□⌘⌘◆◆●⌘⌘</speaker>
 <bio>✱✱○ ✱◆◆⌘ ◆⌘●● &■□◆◆ ✱◆◆⌘■◆□□ □✎ ◆⌘⌘ ♣⌘⌘◆...

Problèmes des labels

Synonymie (Abréviation, Orthographe)

Homonymie

Amplifié dans un contexte **multilingue**

Voiture



Car



Solution

- Définir les termes utilisés dans les balises de manière normée et explicite :
 - Modéliser des « ontologies » qui spécifient:
 - Un lexique des concepts et relations qu'on peut utiliser ;
 - La signification précise de ces concepts et relations
 - Les combinaisons que l'on peut faire à partir des concepts et des relations pour construire des balises.
 - Ces concepts et relations deviennent la référence pour tous les termes qu'on peut utiliser de manière variée pour s'adapter aux utilisateurs :
 - *Place* et *location* seront définis comme ayant la même référence : le concept <Venue> d'une ontologie des conférences internationales.
 - On peut aussi directement utiliser le concept de l'ontologie : <dc:author>Orson Wells</dc:author> signifie que l'on considère Orson Wells comme un auteur au sens de la Dublin Core.

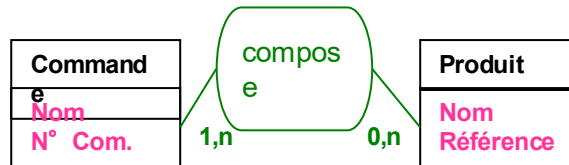
Et on interprétera les labels pour les traduire par exemple dans la langue de l'internaute : Lieu pour <ConferenceOntolgy:Venue>, etc.



Formaliser le document en base de données

Approche structurée ou formalisation

Exemple Produit - Commande:



Commande		Composition			Produit	
Nom	N° Co	Id com	Id prod	Quant.	Nom	Référence
Toto	001	001	15025	1	Brosse	15025
Tutu	002	001	96523	3	Tong	96523
...	...	002	96523	2	...	...
...	...	...	...	...	...	...

© P. Flard

Approche structurée : recherche

Commande		Composition			Produit	
Nom	N° Co	Id com	Id prod	Quant.	Nom	Référence
Toto	001	001	15025	1	Brosse	15025
Tutu	002	001	96523	3	Tong	96523
		002	96523	2		
...	...	...	...	...	...	...

Lecture d'une table:

- **SELECT** NOM **FROM** Produit;

Lecture d'une table avec filtrage:

- **SELECT** * **FROM** Produit **WHERE**
Référence=' 001 ';

© P. Flard



Analyser le document en token

Principe

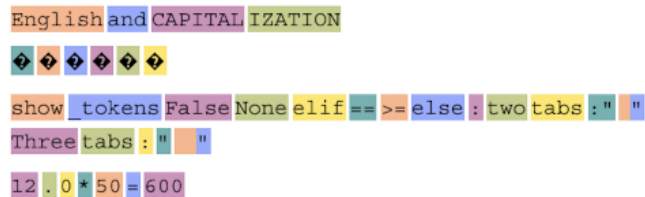
- Le document est décomposé en signaux ou composants élémentaires :
 - Des segments arbitraires de contenus : des tokens ;
 - Des valeurs mesurées et associées aux documents ou à leurs parties

Tokenisation

➤ Soit le texte suivant :

- `text = ""`
- English and CAPITALIZATION
- 🎵 鸟
- `show_tokens False None elif == >= else: two tabs:" " Three tabs: " "`
- `12.0*50=600`
- `""`

➤ Sa tokenisation (GPT 4) :



English and CAPITAL IZATION

🎵 鸟

show_tokens False None elif == >= else: two tabs:" " Three tabs: " "

12.0*50=600

➤ Intérêts:

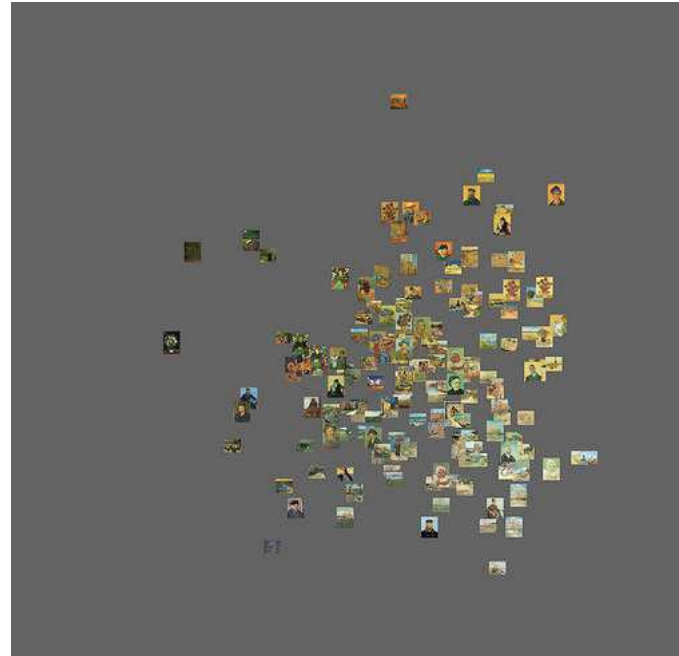
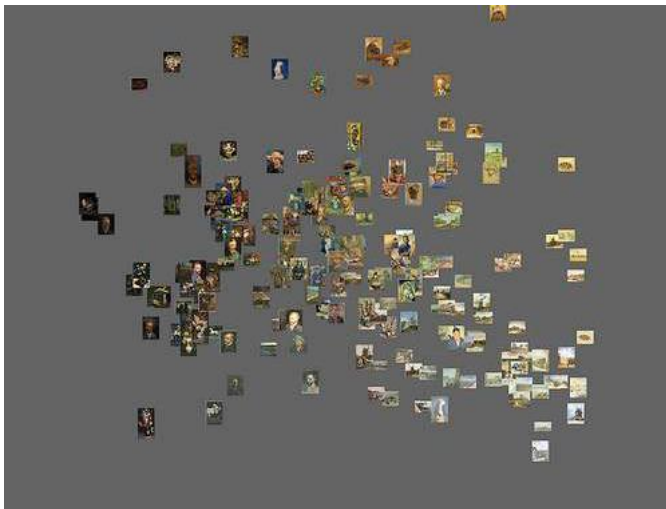
- Traiter tous les types de caractères / symboles
- Réduire la variabilité des formes en les réduisant à des formes élémentaires (token) que l'on peut combiner.
- 100 000 (GPT4), 50000 (GPT2)

➤ Fonctionnement :

- Apprentissage
- Dépend des langues
- Paramétrages variés

Analyse : cultural analytics (Manovich)

Peintures de Van Gogh, Paris (gauche) et Arles (droite) selon lumière (x) et saturation (y)



Conclusion

- Tension entre l'expression que permet le document et la manipulabilité contrôlée que permet la donnée ;
- Plusieurs manières de concilier ces objectifs et de résoudre la tension :
 - L'enrichissement documentaire
 - L'indexation
 - La structuration documentaire par balisage
 - La structuration en données formelles
 - L'analyse des contenus comme ensemble de données élémentaires (tokens).