

RAPPORT DE STAGE

M2 – TAL

ANNÉE UNIVERSITAIRE 2020-2021



Jean-Marc Boucher

Table des matières

1 Introduction.....	3
1.1 Le contexte.....	3
1.2 Travaux précédents.....	3
1.3 Mon projet.....	6
2 Organisation et méthode.....	7
3 Analyse de l'existant.....	8
3.1 Etat de l'art.....	8
3.2 Outils et données à ma disposition.....	10
4 Maquettes.....	15
5 Première version.....	16
5.1 La structure d'annotation.....	16
5.2 Représentation des données.....	17
5.3 Réalisation.....	18
5.4 Précisions sur les informations présentées.....	22
6 Évolutions fonctionnelles.....	25
6.1 Nouveaux graphiques, nouvelles fonctionnalités.....	26
6.2 Intégration de l'import.....	33
6.3 Derniers besoins.....	35
7 Processus de traitements.....	39
8 Déploiement.....	40
8.1 Livraison pour Python.....	40
8.2 Plus de serpent au Paradis.....	40
9 Partage et transmission.....	41
10 Remerciements.....	43
11 Index des figures.....	44
12 NOTES ET COMPLÉMENTS.....	46

1 Introduction

Ce rapport de stage est rédigé dans l'ordre chronologique de son déroulement. Ceci afin de partager les étapes du projet avec le lecteur, les connaissances acquises, les difficultés rencontrées, les solutions proposées, de communiquer sur les besoins exprimés et les décisions prises pour y répondre, enfin, donner un peu de relief à un exercice convenu.

1.1 Le contexte

Je rencontre Marie Chagnoux, ma responsable de stage, lors d'un séminaire organisé dans le cadre de mon cursus M2 TAL à l'université de Nanterre. Chercheuse en linguistique et informaticienne de formation, Madame Chagnoux, parmi les sujets qu'elle nous présente lors de son exposé de trois heures, fait référence au projet « Etude 1000 ». Il s'agit d'un travail qui allie l'analyse de la mémoire et des mécanismes de l'oubli et du souvenir et l'analyse du discours. Le principe est le suivant : sur un programme d'une durée de douze ans, mille personnes vont témoigner sur leur perception des attentats parisiens du 13 novembre 2015, leurs activités pendant cette journée et la façon dont ils ont vécus cette tragédie. Tous les trois ans environ (ce délai a dû être adapté en raison de la crise sanitaire), les même personnes sont interrogées de la même façon. Vivement attiré par le caractère humain et historique de cette étude, je demande à Marie Chagnoux si ma participation à ce programme dans le cadre d'un stage de fin d'étude en M2 l'intéresse. Elle me confirme la possibilité d'une collaboration et je commence à travailler à la maison de la recherche de l'université Paris 8, où elle officie.

1.2 Travaux précédents

Afin de comprendre le contenu de mon projet de stage, je prends connaissance de deux publications :

- « Informer sans s'engager : variations de prise en charge énonciative dans les sujets d'actualitéⁱ. », Chagnoux 2008.

- « Vers un outil de visualisation de la dynamique textuelle : l'exemple des phénomènes citationnels et modaux »ⁱⁱ, Chagnoux 2008.

Ces deux documents proposent des méthodes d'analyse de la modalisation de textes, d'identification des structures d'un discours et des représentation graphiques des formes discursives que l'on peut produire.

La modalisation d'un discours peut être définie comme le degrés de prise en charge des propositions du locuteur par lui-même. Ces études s'appuient principalement sur des articles de presse et montrent comment le journaliste, rapportant des informations trop rapidement ou simplement difficiles à vérifier, se transforme parfois en rédacteur dont la stratégie permet de communiquer sans risque d'erreur. Voici, deux extraits issus des publications de Madame Chagnoux qui résument cela :

« La possibilité de faire varier la prise en charge énonciative au sein d'un même discours permet aux auteurs de sujets d'actualité d'informer sans s'engager sur la véracité de leurs propos, voir même de faire figurer dans un même article des informations contradictoires. », extrait de la première publication.

« Nous proposons d'exposer ici une méthode d'investigation du phénomène des différentes prises en charge énonciatives (plus particulièrement, le cas des citations) et modales à l'œuvre dans les textes » , extrait de la seconde publication.

Chacun de ces documents de recherche démontrent qu'il est très difficile, par l'analyse textuelle automatique, d'identifier la modalité de chaque segment du discours. Il peut s'avérer parfois contre-productif de tenter une analyse algorithmique de la variation des propositions et de leur prise en charge par le rédacteur. Le conditionnel, les conjonctions (mais, si, or) sont identifiables mais les différentes ruptures du discours sont encore difficiles à être identifier sans risques de contresens.

Marie Chagnoux propose alors une méthode d'analyse humaine pour *« appréhender la dynamique interprétative d'un texte comme un cheminement qui s'opère entre différents niveaux de discours au fur et à mesure de la lecture syntagmatique »*.

La seconde idée des publications de 2008 est que cette analyse peut alimenter une représentation graphique des résultats qui concourrait à mieux percevoir la construction des textes et la démarche du locuteur, notamment la modélisation, les ruptures énonciatives et la cohérence discursive.

Voici le type de schéma proposé :

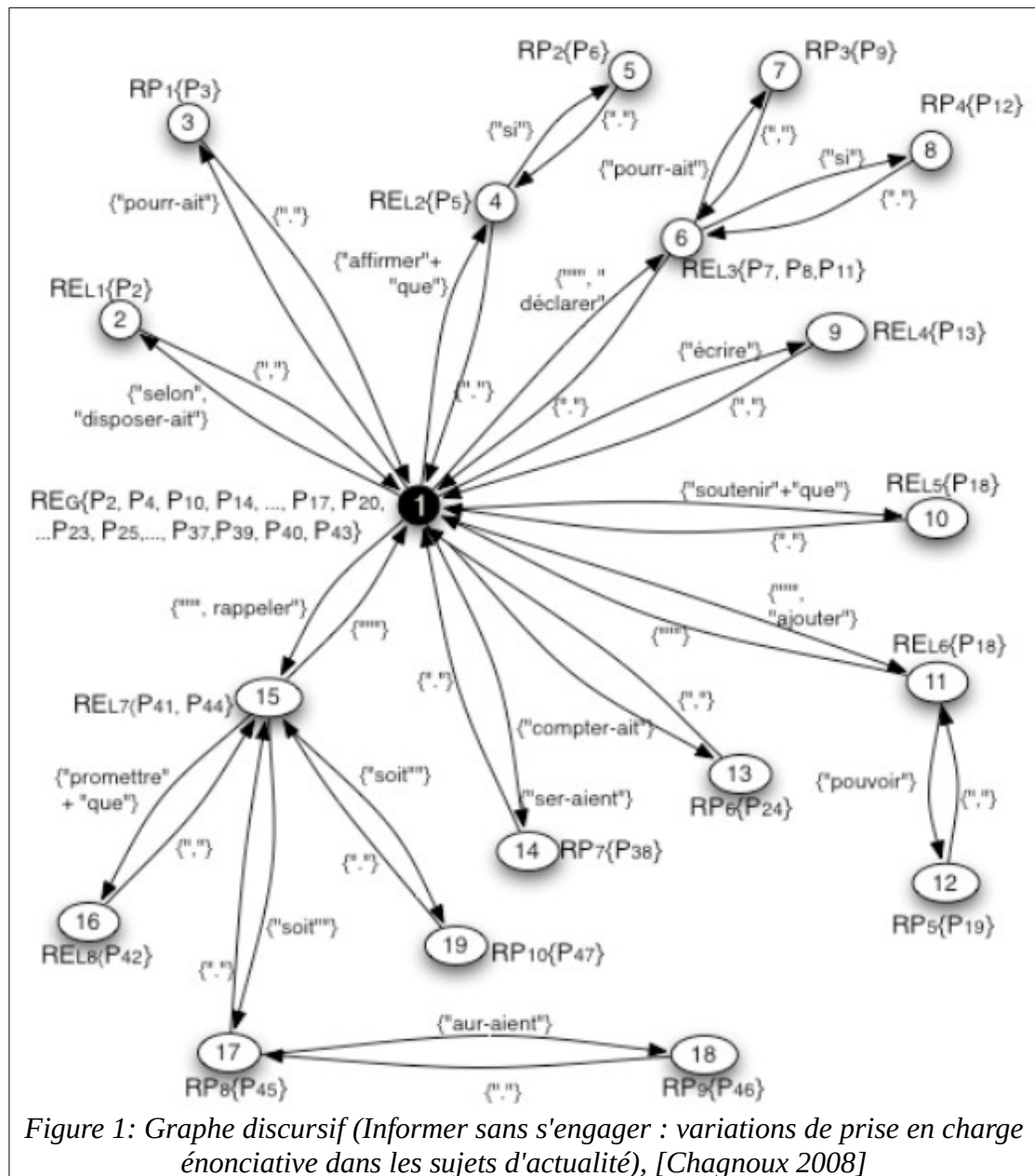


Figure 1: Graphe discursif (Informer sans s'engager : variations de prise en charge énonciative dans les sujets d'actualité), [Chagnoux 2008]

1.3 Mon projet

A partir de ce travail sur la rédaction journalistique la méthode peut être appliquée à tout type de texte. Ce qui intéresse la chercheuse aujourd'hui, notamment, c'est l'utilisation de cette méthode sur les témoignages de l'Etude 1000.

Par extension aux recherches sur la modélisation, sur la structure et sur le cheminement dans les articles de presse, la méthode proposée pourrait être appliquée aux interviews sur les événements du 13 novembre. Cette fois l'analyse permettrait de porter l'accent sur les mécanismes de la mémoire pour construire un discours cohérent dans un contexte d'intense émotion, dans un domaine social partagé et historique. Par ailleurs, en accumulant les traitements sur de grands ensembles de témoignages portant sur le même sujet, il pourrait être intéressant d'étudier, non seulement les différents mécanismes intellectuels mis en œuvre selon les individus, mais également de comparer les productions émanant des mêmes personnes à des époques différentes.

Le papier de Marie Chagnoux intitulé Informer sans s'engager : variations de prise en charge énonciative dans les sujets d'actualité (Op. cit.) spécifiait, à partir de la segmentation réalisée manuellement ceci : « La possibilité de matérialiser la structure dans un espace restreint permet un affichage graphique qui modifie le rapport au texte puisqu'il est possible de visualiser directement la complexité du texte pour faire émerger d'autres caractéristiques de la structure discursive. »

C'est sur ce terreau que s'est développé mon stage. Précisément, mon objectif était de réaliser un outil qui permette une représentation automatisée des résultats de ces analyses de discours.

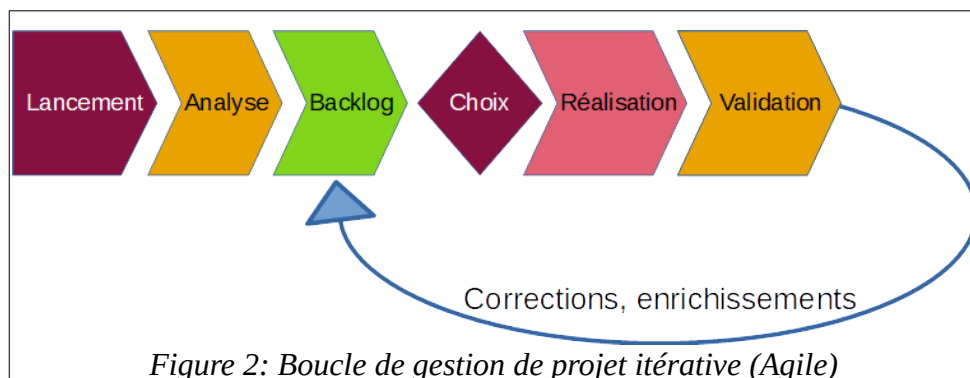
2 Organisation et méthode

Avant de vous présenter les étapes de mon travail, voici la méthode que j'ai déployée. Étant donné que je ne connais pas, d'un point de vue informatique, le domaine de la représentation graphique, ni celui de l'annotation linguistique, je devais passer par une étape de découverte en deux parties :

- La recherche d'outils du marché répondant, au moins partiellement, aux besoins fonctionnels collectés auprès de mon encadrante.
- L'apprentissage des outils utilisés dans l'Etude 1000.

Après avoir reformulé les besoins avec ma responsable de stage, j'envisage une démarche Agile à base de :

- Construction de maquettes à faire valider avant développement,
- Réalisation par approches successives : faire grossir le contenu fonctionnel par priorité (backlog), autour d'un noyau consolidé à chaque itération,
- Partage avec des experts pour s'appuyer sur leurs recommandations,
- En fin de projet, enrichissement de fonctions moins essentielles si le temps le permet.



Quant à la partie technique de mon projet, je décide de développer avec le langage Python. J'ai d'abord envisagé l'utilisation de Java mais étant donné le délai (c'est un stage de quatre mois) et les ambitions de l'outil, Python me semble plus adapté. Cette décision a l'avantage de me permettre de générer un code source compatible avec tous les systèmes d'exploitation mais aussi un défaut : elle implique l'installation d'un interpréteur Python et des bibliothèques utilisées sur la machine d'exécution. Ce défaut sera supprimé en fin de projet, nous le verrons.

3 Analyse de l'existant

Deux actions me permettent de démarrer mon projet :

- l'analyse de l'état de l'art,
- celle des données et des outils à ma disposition.

3.1 État de l'art

Ma première réaction est d'interviewer Madame Chagnoux sur les outils existants dont je pourrais m'inspirer ou qu'elle aurait pu juger intéressant. Sa réponse est sans équivoque, elle n'a pas trouvé de solutions à retenir jusqu'à présent. Je réalise donc une recherche sur internet des outils autour des mots clés « analyse textuelle », « représentation graphique de textes », « textes et graphes », etc. (en français et en anglais).

Voici quelques résultats intéressants :

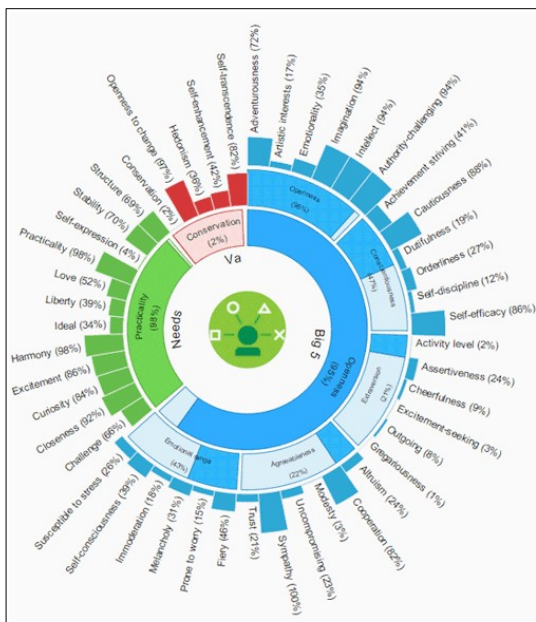


Figure 4: Eric Delcroix, IBM Watson Personality

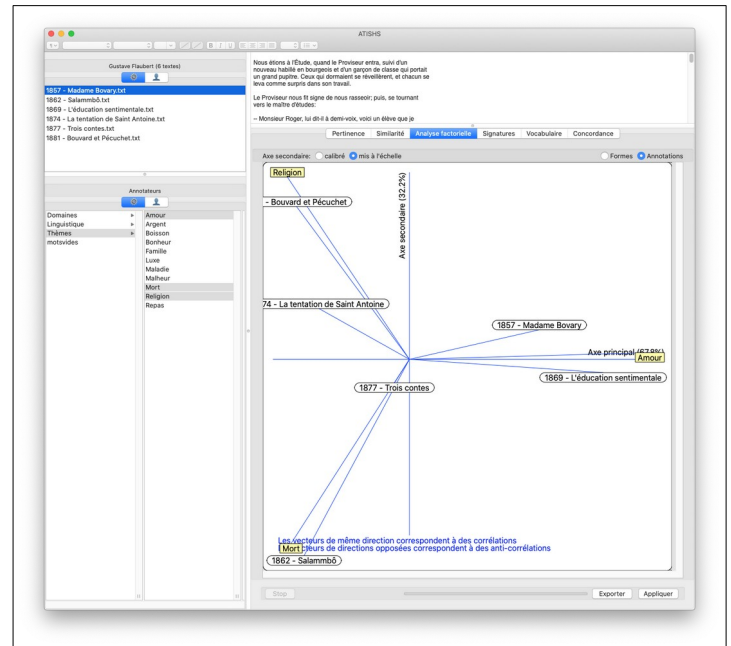


Figure 3: ATISHS Analyseur de Textes Innovant pour les Sciences de l'Homme et de la Société

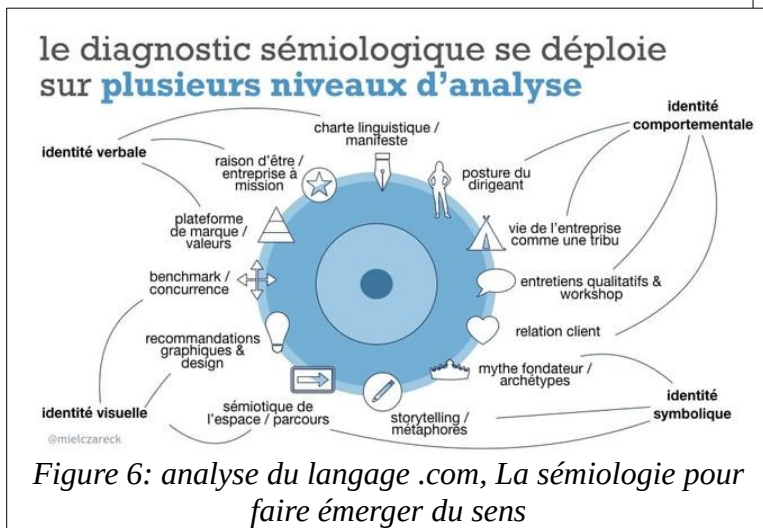


Figure 6: analyse du langage .com, La sémiologie pour faire émerger du sens

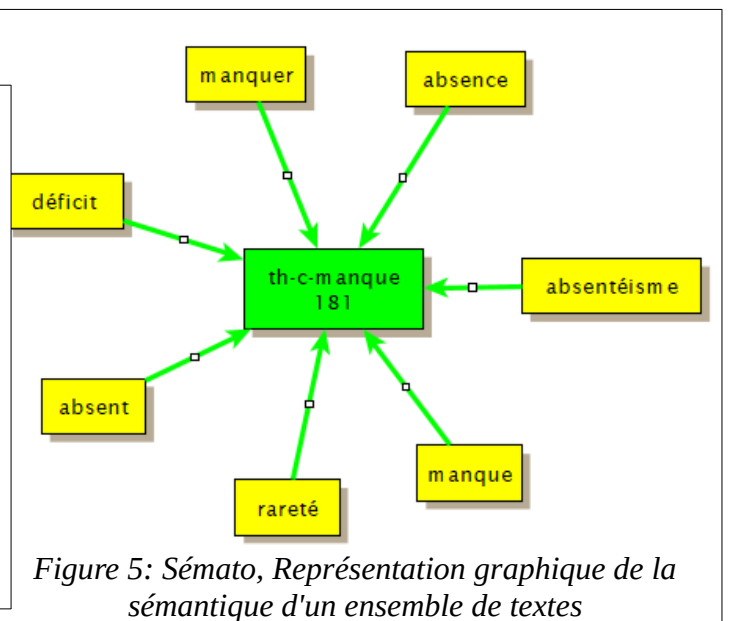


Figure 5: Sémato, Représentation graphique de la sémantique d'un ensemble de textes

Images présentées :

Figure 2 : Eric Delcroix, page Pinterestⁱⁱⁱ

Figure 3 : Université de Franche Comté. ATISHS Analyseur de Textes Innovant pour les Sciences de l'Homme et de la Société.^{iv}

Figure 4 : analyse du langage .com^v

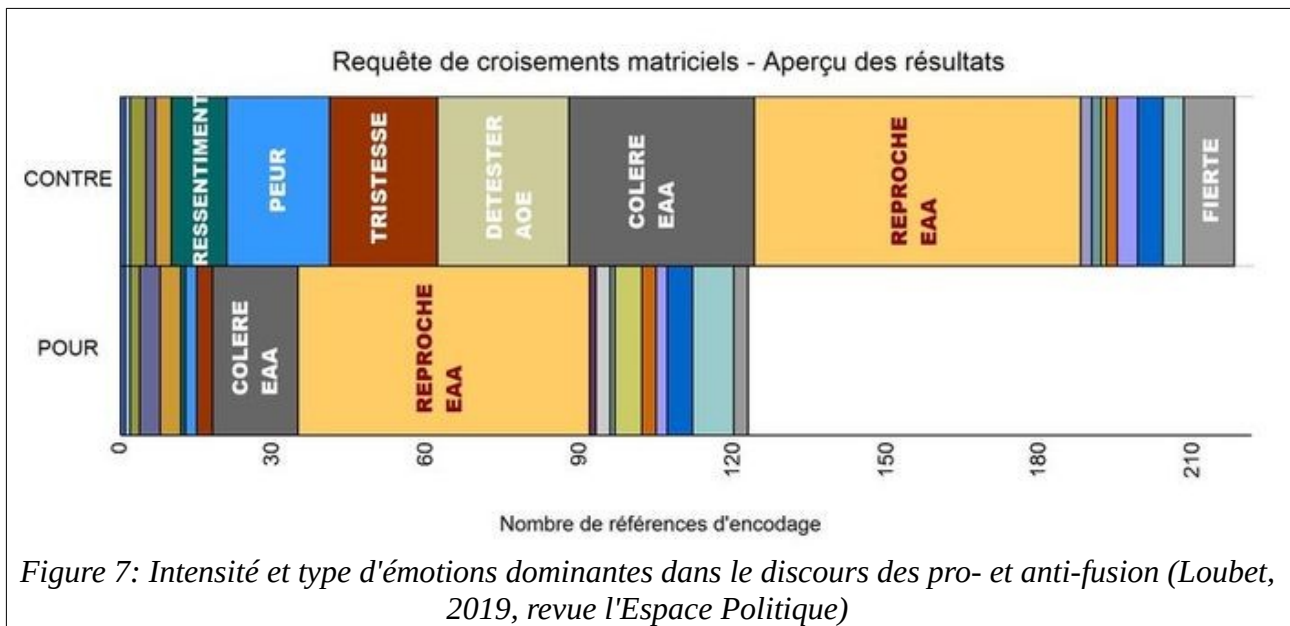
Figure 5 : Sémato, Représentation graphique de la sémantique d'un ensemble de textes^{vi}.

Ces quelques ressources dénichées sur le web m'inspirent cette première analyse :

- Sur le fond, la plupart des solutions proposant une mise en forme graphique de la structure de textes sont focalisées à la fois sur la linguistique et sur les sciences sociales. Il s'agit, non seulement de comprendre comment s'articule l'expression orale ou écrite, mais aussi (et peut-être surtout), d'identifier des structures et modes de pensées.
- Sur la forme, le cercle semble la référence. Tous ces graphiques représentent les données dans une figure circonscrite. Le premier graphique de Marie Chagnoux respecte d'ailleurs cette mise en forme. Par ailleurs, dans certains cas, le cercle permet d'ajouter l'information concernant la chronologie des événements rapportés.

J'y vois plusieurs avantages pour notre projet : les sciences sociales sont aussi une discipline de notre sujet, la forme périmétrée permet d'envisager une représentation maîtrisée, visuellement pour l'utilisateur et techniquement pour l'écriture du code; elle ouvre la voie vers une vision des différents niveaux de segmentations des textes annotés par imbrication de cercles; enfin, elle enrichit le contenu grâce à la mise en forme de la chronologie du discours (dans le sens des aiguilles d'une montre).

Ce n'est qu'une première approche, je trouve par la suite d'autres propositions de représentations graphiques, notamment les histogrammes et les frises temporelles, voici un dernier exemple^{vii} :



3.2 Outils et données à ma disposition

Les outils du linguiste

Avant d'envisager des premières maquettes, je dois prendre connaissance des outils avec lesquels Marie Chagnoux travaille, ceux utilisés par l'Etude 1000 ainsi que les données que mon projet devra manipuler.

Les outils informatiques, que ce soit en ligne ou à installer sur un ordinateur personnel, sont très nombreux. Chacun a sa spécialité, voici quelques exemples :

- les étiqueteurs (ou tagger) qui analysent automatiquement un texte et produisent plusieurs types d'analyse de surface et profonde, morphologique, morpho-syntaxique ou syntaxique :

- SDU : <http://visl.sdu.dk/visl/fr/parsing/automatic/complex.php>
- CENTAL : <https://cental.uclouvain.be/treetagger/>
- MultiTAL : <http://multital.inalco.fr/>

- ceux qui identifient les entités nommées :

- EXPLOSION : <https://explosion.ai/demos/displacy-ent>
- FNER : <https://cloud.gate.ac.uk/shopfront/displayItem/french-named-entity-recognizer>
- SEM : <http://apps.lattice.cnrs.fr/sem/>

Je passe trois jours à installer TXM puis à m'auto-former avec ces vidéos (l'installation standard contient un corpus de textes de démonstration).

Par ailleurs, mon encadrante me donne les coordonnées du responsable technique de TXM afin d'entrer en contact et de lui expliquer le contexte de mon stage. J'échangerai avec lui tout au long du projet afin d'avoir quelques informations techniques sur le format des données à exploiter et les méthodes d'annotation, j'en reparlerai plus loin.



Le second logiciel qui semble important pour mon projet est Nvivo^{ix}. C'est aussi une solution à installer sur un poste mais dont il faut acquérir la licence d'utilisation (environ 800€). Elle est développée par la société QSR International. Une version d'essai est téléchargeable, utilisable pendant 14 jours. Je la téléchargerai le moment venu afin de faire des tests d'interfaçage avec mon outil. Il est utile pour le travail de ma responsable de stage dans le cadre de ses recherches et de ses publications. Je devrai en tenir compte comme interface d'acquisition des données.

Données

Avant de parler de données, il faut présenter l'Etude 1000. Le 16 avril je suis invité à un séminaire de cette étude avec l'ensemble des chercheurs du groupe de Metz pour un moment de partage des travaux.

J'en ai expliqué plus haut l'ambition générale : recueillir auprès de 1000 personnes, au niveau national, leurs souvenirs de la journée du 13 novembre 2015. Cette démarche sur l'ensemble du territoire est représentée ainsi :

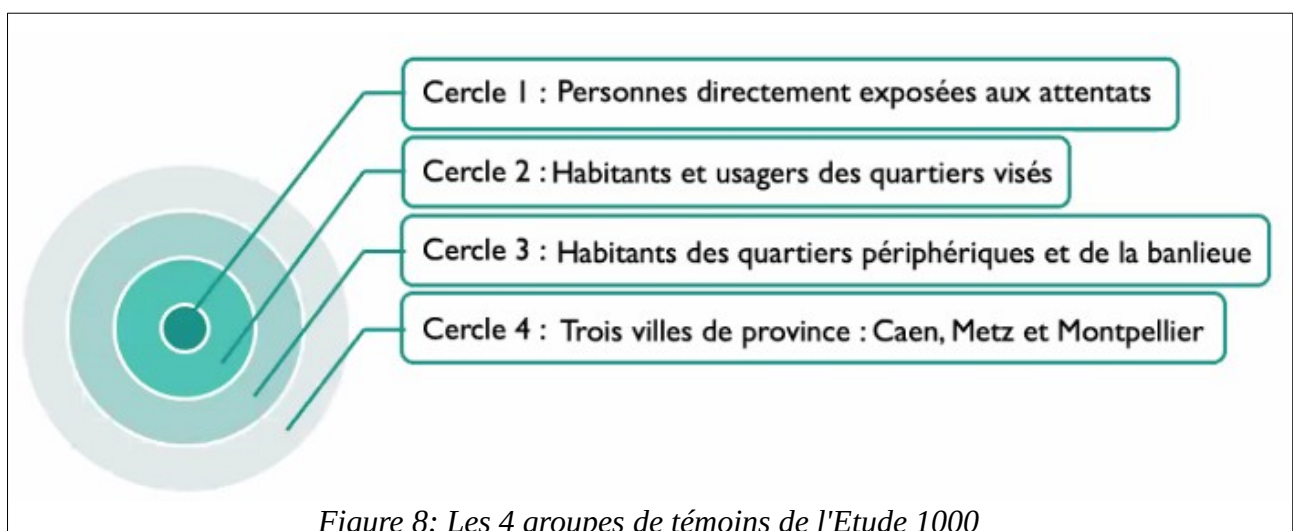


Figure 8: Les 4 groupes de témoins de l'Etude 1000

Les 1000 personnes interrogées sont donc choisies sur deux critères : leur distance physique avec les lieux des attentats et leur distance humaine et sociale avec les victimes. Il n'y a pas de sélection, leur participation est volontaire suite à une campagne de « recrutement » dans les media locaux ou par leur proximité avec les organisateurs.

L'enrôlement est d'ailleurs un souci pour l'ensemble du projet car le panel est finalement peu représentatif de la population nationale. Les témoins sont généralement d'une classe sociale supérieure à la moyenne en termes de niveau d'études et de revenu. C'est un biais identifié.

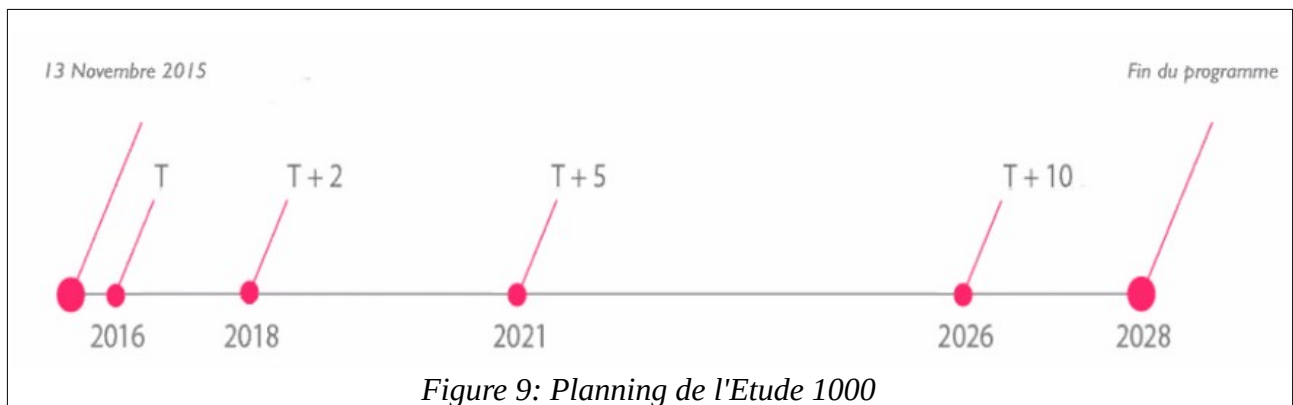


Figure 9: Planning de l'Etude 1000

Par ailleurs, l'étude s'étale sur 12 ans en 4 campagnes d'entretiens :

La pluridisciplinarité est le principal moteur de ces travaux. Grâce à des interviews construites pour répondre aux questions de plusieurs domaines de recherche (entretiens segmentés en parties non directives et semi-directives), la collecte d'information socio-démographiques sur les participants, l'alimentation d'une base de données consultable (par les personnes accréditées) dans le temps et l'objectivation des résultats à termes (budgets conditionnés au passage de grands jalons), ce travail doit permettre de conserver une trace substantielle de notre Histoire qui pourrait être réutilisée dans différentes disciplines.

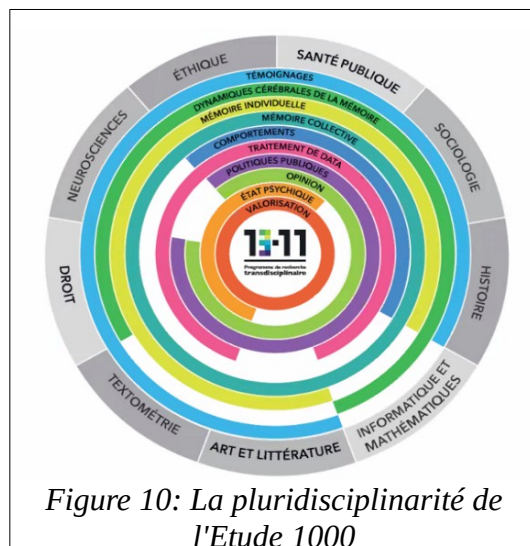


Figure 10: La pluridisciplinarité de l'Etude 1000

Le protocole de recueil des témoignages suit les étapes suivantes : la signature d'un contrat d'autorisation de droit à filmer, un entretien hors caméra pour recueillir les données sociologiques et biographiques, un entretien devant caméra en deux parties, la première (Question I) est une question ouverte sur le vécu individuel, une seconde, directive, sur la mémoire émotionnelle, enfin un questionnaire hors caméra sur la mémoire de l'évènement.

Ces informations, on le comprend, sont extrêmement confidentielles. Elles sont donc, non seulement anonymisées pour les chercheurs travaillant sur les recueils, mais leur accès est restreint. C'est, pour moi, une difficulté avec laquelle je vais devoir composer.

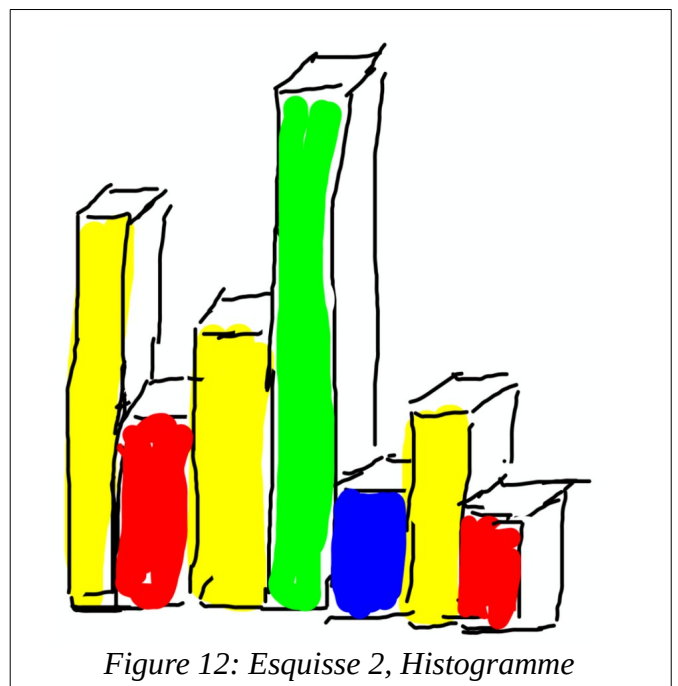
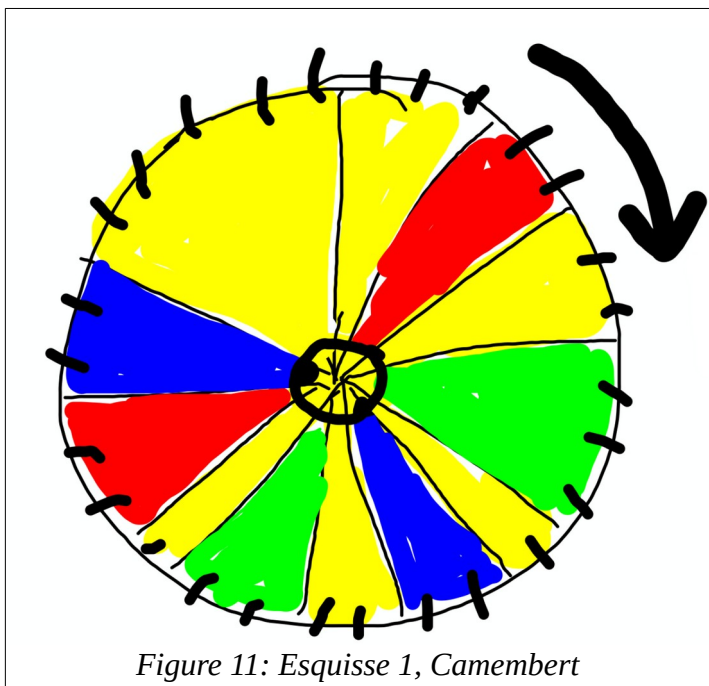
Dès le début, nous percevons, mon encadrante et moi, que nous allons devoir travailler sur des textes annexes (ce que nous appellerons des « fakes ») afin de ne pas me trouver en situation de blocage dans le cas où les demandes d'accès envoyées pour que je puisse utiliser ces témoignages seraient refusées ou trop longues à obtenir.

Précisément, c'est la Question I qui nous intéresse. Elle offre l'avantage de laisser les personnes s'exprimer librement sur leur vécu, sans aucune intervention de l'interviewer, ce qui permet l'analyse discursive que veut réaliser Madame Chagnoux. Cette question est « Pouvez-vous me raconter le 13 novembre 2015 ? ». Cette partie devra être annotée manuellement afin d'identifier chaque segment et sous-segment du discours. C'est sur ces annotations sur la réponse du témoin que je dois proposer une solution de mise en forme automatique.

4 Maquettes

Comme je l'ai expliqué dans la partie <Organisation et méthode du projet>, ma démarche est itérative. En mode Agile, je vais construire un premier noyau que je vais soumettre à validation, recueillir les retours de l'utilisatrice, corriger/adapter/enrichir le contenu fonctionnel et boucler sur ce principe.

Je commence donc, non pas par cibler la mise en forme de la figure 1, mais par proposer des schémas qui me permettront d'appréhender les données et de présenter des premiers résultats rapidement. J'utilise la simple paire crayon-support (numériques) pour dessiner ces esquisses :



Sur ces deux schémas les couleurs représentent les thématiques annotées, ou segments, et apparaissent dans l'ordre chronologique du discours.

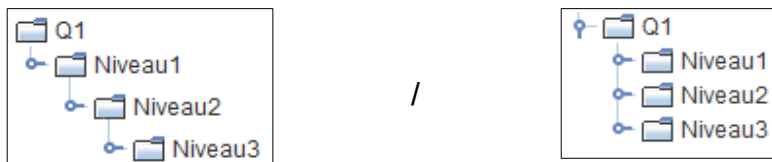
Marie Chagnoux est une scientifique curieuse... et c'est une bonne chose pour construire une solution pas à pas, mieux encore : pour entrer dans un processus créatif vertueux. Elle valide ces propositions en m'expliquant qu'elle n'avait pas envisagé ce type de graphiques mais qu'ils peuvent apporter de nouvelles informations. C'est une ouverture qui, selon elle, peut aussi nous permettre de découvrir des possibilités.

5 Première version

Les premières mises en forme étant décidées, je dois revenir un peu en arrière sur l'analyse des données issues de TXM. Ma responsable de stage me donne quelques textes « fakes » que je dois importer dans l'outil de l'ENS Lyon. De mon côté, je dois comprendre la structure d'annotation disponible à utiliser, déterminer comment l'exporter et la traiter.

5.1 La structure d'annotation

TXM ne permet pas une organisation hiérarchique de la structure d'annotation. Or, nous avons besoin de définir une arborescence d'annotation. Voici par, exemple, ce que nous voulons et ce que TXM nous offre :



Quant à l'écran de saisie des annotations, en voici une copie :

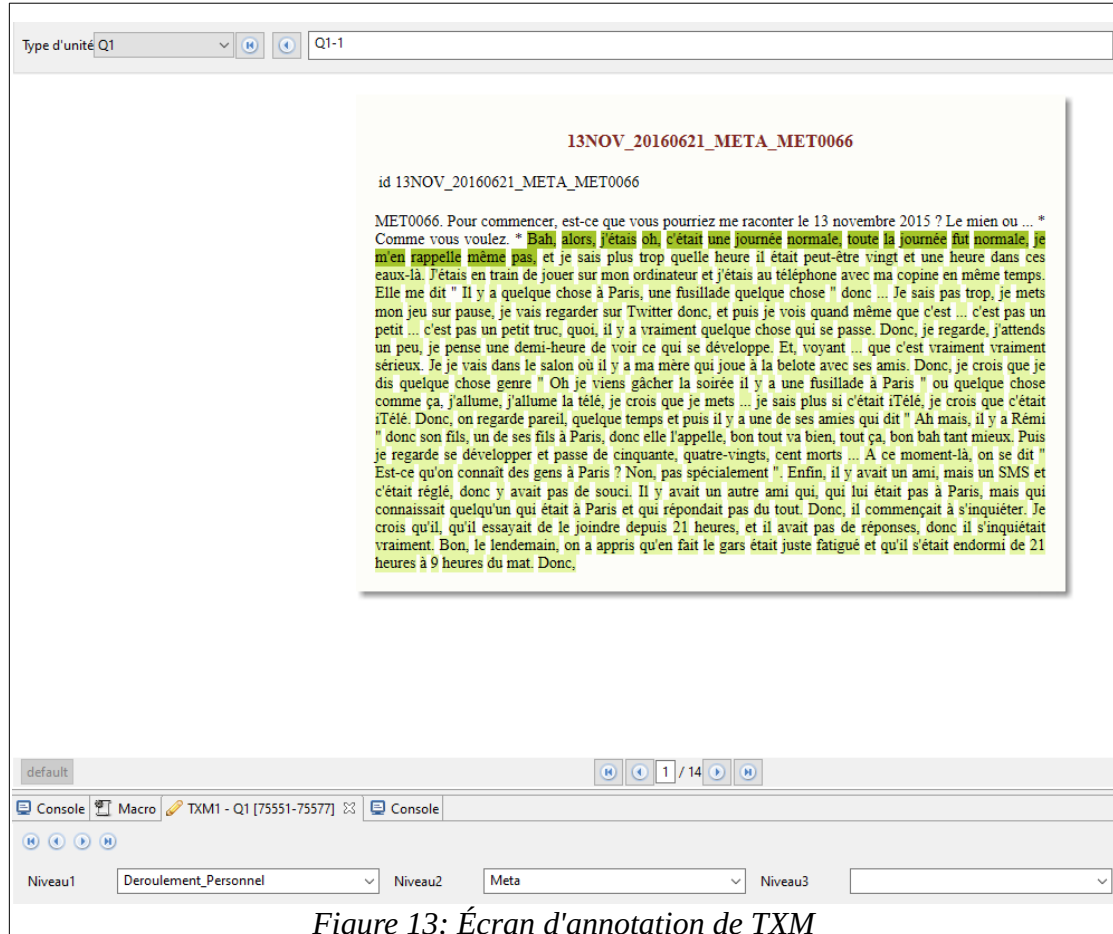


Figure 13: Écran d'annotation de TXM

Sur la partie haute de cet écran, la zone « Type d'unité » permet de choisir la structure d'annotation (Q1 est le nom que nous avons donnée à la structure dédiée à la Question I), au milieu le texte coloré est la partie annotée, en gras la partie sélectionnée pour l'annotation en cours, enfin, en bas, les valeurs données à chaque champ d'annotation.

En l'absence de hiérarchisation de ces champs je propose une discipline d'annotation : <Niveau 1> contiendra la valeur que nous voulons donner au premier niveau, <Niveau 2> celle du second niveau et ainsi de suite. Cette règle de remplissage est un peu contraignante pour l'utilisateur mais il faut bien palier la défaillance de TXM. Par contre elle est très contraignante pour moi puisque la couche d'acquisition de notre outil va devoir reconstruire la hiérarchie conçue par l'annotateur.

Notons que dans cet exemple nous n'avons que trois niveaux mais que je dois définir une solution compatible avec une profondeur hiérarchique infinie. En réalité, puisque nous travaillons sur l'humain, et que chaque segment textuels est une digression, nous n'excéderons pas une profondeur de 4. Au-delà, c'est difficilement concevable tant pour le locuteur que pour l'annotateur.

5.2 Représentation des données

Les deux contraintes ci-dessus, reconstruire la hiérarchie et prévoir une arborescence infinie, m'imposent un modèle et une méthode :

- Les données exportées de TXM seront stockées dans un arbre hiérarchique de largeur et de profondeur infinies,
- Pour construire une telle organisation des données, mon algorithme devra être récursif.

La récursivité est une technique de programmation peu utilisée parce que complexe à maintenir et qui peut s'avérer dangereuse si elle n'est pas maîtrisée. Une fonction est dite récursive si elle s'appelle elle-même. Le risque est de créer une boucle infinie et de remplir la mémoire de l'ordinateur et d'occuper sa CPU à 100 %. Dans le milieu

professionnel, elle est souvent proscrite pour cette raison. Pour mon projet je décide de coder cette récursivité puisque, en l'absence de méta-données sur l'arborescence de la part de TXM, mon programme va découvrir la profondeur et la largeur des données pendant leur import.

Je sais que cette étape dans ma démarche Agile est cruciale : je construis à ce moment le noyau central de la solution. Je dois m'assurer que cette base de mon édifice est des plus solides. Je réalise donc un jeu de données de tests complexe dans TXM après avoir défini un ensemble de cas d'école :

1. Un segment de texte portant une annotation de niveau 1 suivi d'un second texte portant le même niveau d'annotation
2. Une annotation de niveau 1 suivi d'une annotation de niveau 1 et 2 suivi d'une annotation de niveau 1
3. Une annotation de niveau 1 et 2 suivi d'une annotation de niveau 1
4. etc.

Tous les cas aux limites, c'est à dire, ceux qui, je le pense, représentent une complexité d'annotation peu probable dans la plupart des discours, doivent me permettre de clore le sujet de l'acquisition des données et de me concentrer sur leur représentation.

Enfin, le format des exports : comme toutes les applications bien faites TXM propose le format XML pour ses exports. C'est un format approprié aux arbres hiérarchiques et de nombreuses bibliothèques Python sont disponibles pour lire ce type de fichier.

5.3 Réalisation

Si nous avons déjà, lors du cours d'apprentissage automatique, pratiqué le décodage de données XML avec Python, aucun cours ne nous a enseigné la réalisation d'interface utilisateurs. Je vais donc devoir faire des recherches sur les standards du langage pour générer des écrans de visualisation et des graphiques.

Une autre démarche aurait pu être d'apprendre ce qu'il est possible de faire avec les bibliothèques Python concernant l'interface utilisateurs puis de construire les maquettes sur la

base de cette connaissance mais c'est une démarche que je refuse. En matière de design et d'ergonomie (on parle d'UX pour User Experience et d'UI pour User Interface) la technique ne doit pas diriger la réflexion. Le contraire est toujours préférable : la définition d'une expérience utilisateur la plus aboutie possible doit piloter les développements.

Je passe donc quelques jours à me former à l'écriture d'interfaces avec Python. Il faut, non seulement que je sois capable d'écrire une interface classique à base de boutons, zones de saisie et listes de choix, mais que je puisse également générer des graphiques comme des camemberts ou des histogrammes. Heureusement la communauté des développeurs Python sur le web est prolifique, je trouve de nombreuses ressources et exemples en ligne pour acquérir rapidement les premières bases de tkinter, la librairie de référence en la matière.

Il me reste à définir le format général de mon écran principal en gardant en mémoire que c'est aussi à partir de ce noyau que je ferai évoluer l'outil. Voici le schéma d'interface que je conçois :

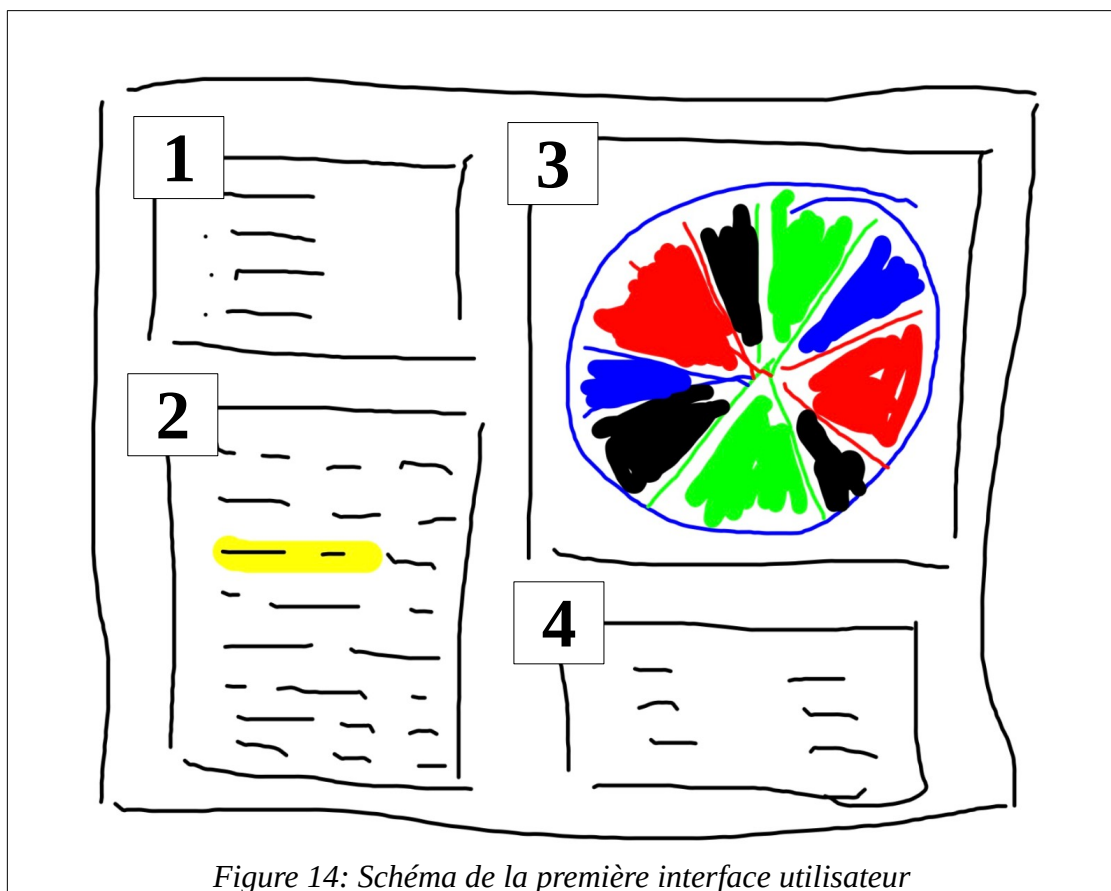


Figure 14: Schéma de la première interface utilisateur

1

Cette zone contiendra la liste des textes annotés importés.

2

Ici sera affiché le texte représenté par le graphique.

3

La zone du graphique.

4

Je prévois d'afficher ici quelques statistiques sur le texte.

Cet écran n'a qu'une seule fonction : présenter les graphiques des données acquises après les imports de TXM. L'interface de cette première version contient déjà un principe structurant : l'interactivité. Dans la zone 3 je prévois que l'utilisateur pourra cliquer sur chaque part du camembert afin d'identifier la partie du texte qu'elle représente (surlignée en jaune dans la zone 2) et de mettre à jour les statistiques de la zone 4. C'est une des leçons apprises en utilisant TXM et en parcourant le web sur des solutions d'annotation et de représentation graphique de textes : l'utilisateur doit avoir la main sur les données qu'ils consultent et l'informatique doit lui apporter une valeur ajoutée maximum.

La fonction d'import n'est, elle, disponible que via un programme séparé, exécuté en ligne de commande. Ma priorité est de faire valider les fonctions principales de générations automatiques des graphiques et de rassurer ma responsable de stage sur les chances d'atteindre nos objectifs dans le délai des 4 mois.

Avec la méthode Agile, les besoins doivent être priorisés afin de les planifier dans des livraisons successives. J'utilise trois niveaux de priorités :

- Indispensable : le produit ne peut être fini sans.
- Appréciable : c'est une des motivations du projet, une attente forte des utilisateurs.
- Agréable : serait idéal si les délais et la charge induite le permettent.

C'est ainsi que je construis la liste du reste à faire dans laquelle nous allons piocher pour réaliser chaque nouvelle version.

Une seconde conséquence de l'agilité est la montée en puissance : plus le projet avance et plus l'équipe est compétente. Pour les techniciens leur capacité augmente grâce à une meilleure connaissance du sujet, une maîtrise des outils utilisés et pour les utilisateurs, leurs besoins se précisent et s'alimentent des propositions des développeurs. C'est bien ce que l'on constate : j'acquière un savoir faire sur les bibliothèques Python et ma perception des attentes de mon encadrante se précise, tandis qu'elle découvre de nouvelles possibilités qui l'inspire (l'interactivité et les statistiques pour commencer).

Il me faut environ 20 jours pour être présenter cette première version :

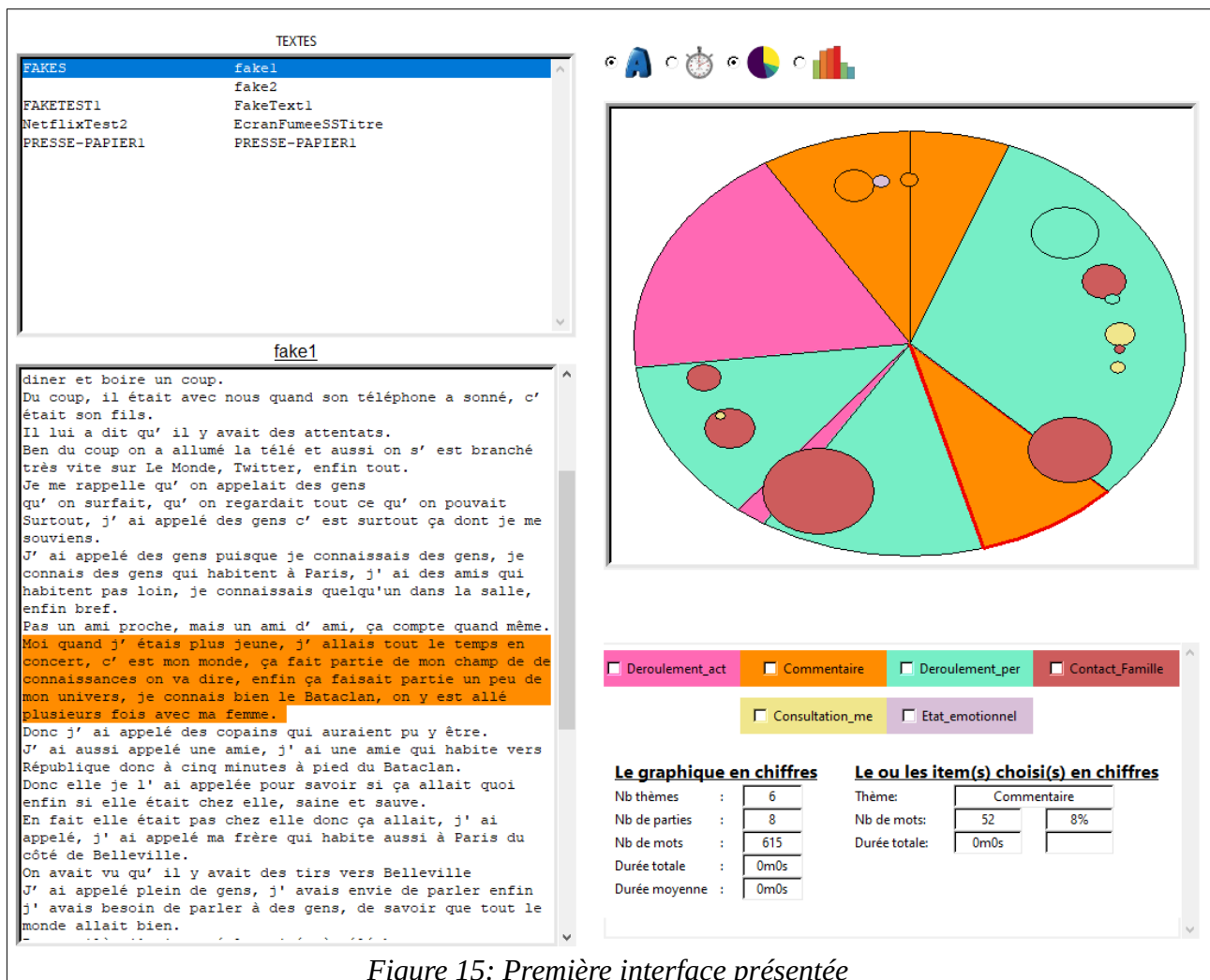
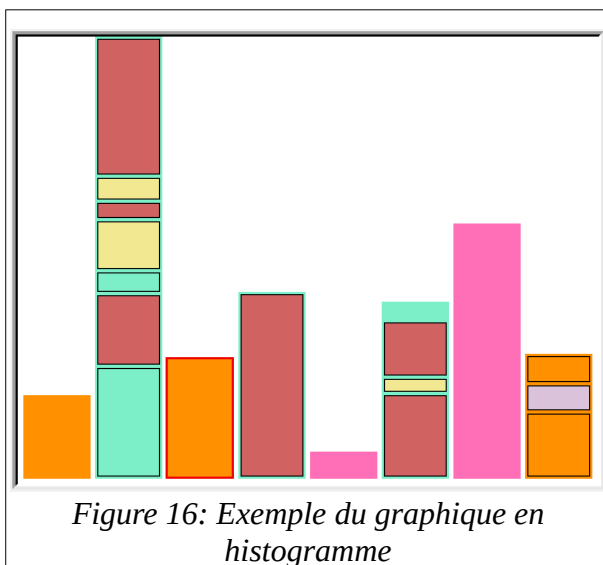


Figure 15: Première interface présentée

Toutes les zones du schéma précédent sont là ainsi que l'interactivité.

Voici le format en histogramme :



5.4 Précisions sur les informations présentées

Les graphiques

Nous avons expliqué plus haut que les données étaient organisées hiérarchiquement. Concrètement, cela signifie qu'un segment de texte consacré à décrire le déroulement de la journée (donc à répondre précisément à la Question I), peut contenir un ou des segments, portant par exemple sur la famille, les médias qui à leur tour peuvent contenir des segments, etc. Tout l'enjeu des graphiques est de donner cette vision de l'imbrication de segments dans les segments.

Dans la représentation en forme de camembert, les parts représentent un premier niveau de segment, les cercles internes représentent un second niveau et si nous avons un troisième niveau, d'autres cercles, plus proches du centre seraient dessinés.

Dans le graphique en histogramme, de la même façon, chaque barre peut contenir des boîtes représentant les segments de second niveau qui, à leur tour, peuvent contenir d'autres boîtes, à la manière des poupées russes.

La dimension de chaque forme dépend de sa représentation dans le texte selon deux critères : le nombre de mot ou le temps de parole, selon ce que l'utilisateur choisit ici :



Couleurs

A chaque valeur d'annotation est attribuée une couleur. L'outil contient une palette de couleurs. Ces couleurs sont utilisées pour les formes, le surlignage du texte et les cases à cocher présentes sous le graphique (voir interactivité ci-dessous).

Interactivité

Chaque forme est cliquable et permet d'accéder à plusieurs informations :

- Dans le texte à gauche, la partie concernée par la forme sur laquelle l'utilisateur a cliqué est surlignée avec sa couleur.
- Dans la partie statistiques : le thème de la forme (ou segment), des informations sur son nombre de mots, le rapport de ce nombre sur le texte complet, la durée de parole et le rapport entre cette durée et la durée totale.

Concernant le temps de parole, l'information est contenue dans les données de TXM. Cependant, je n'ai pas eu accès aux textes protégés du corpus de l'Etude 1000, les textes « fakes » donnés par ma responsable de stage en sont dépourvus.

Les boîtes à cocher permettent de modifier la vision des données en proposant, non plus de choisir un segment unique, mais de faire apparaître la répétition d'une valeur d'annotation dans plusieurs parties du texte. Voici ce que permet, par exemple, le choix de la valeur « Déroulement personnel » :

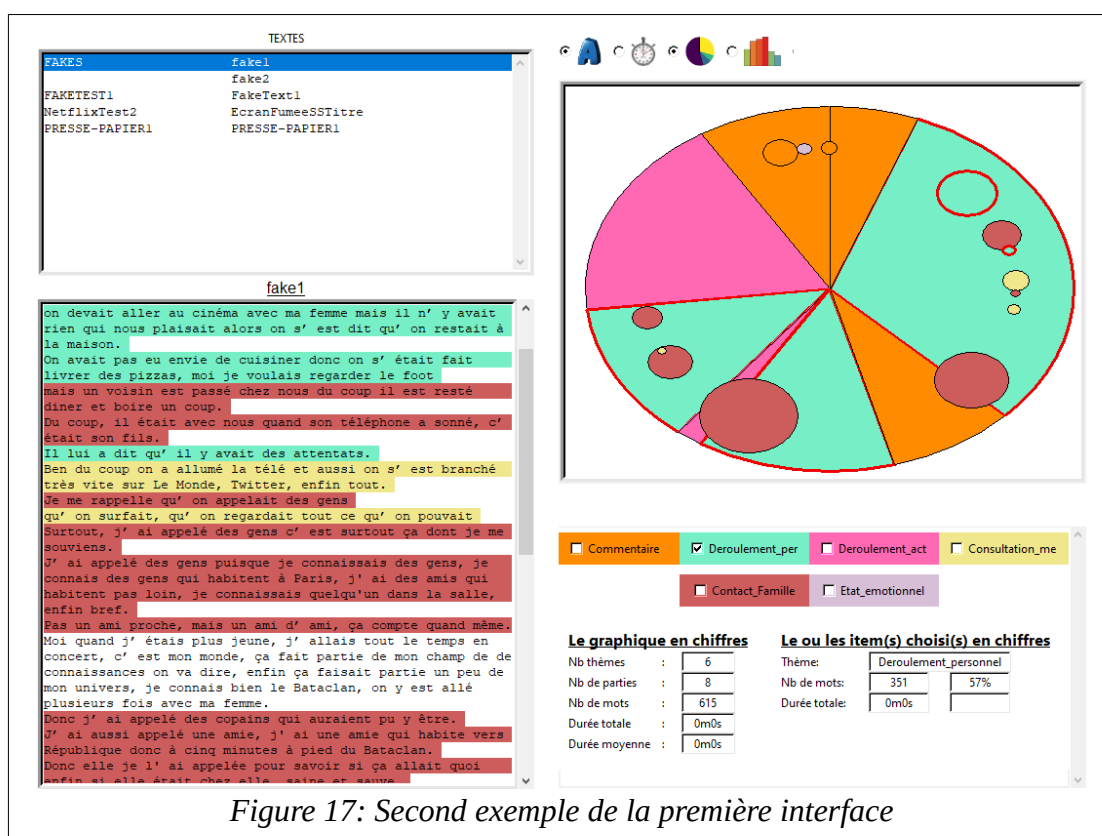


Figure 17: Second exemple de la première interface

Sur ce schéma, toutes les parties annotées et sous-parties contenues dans le segment « Déroulement personnel » sont sélectionnées :

- Dans le graphique toutes les formes concernées sont entourées en rouge,
- Dans le texte, tous les passages sont surlignés,
- Les statistiques sont mises à jour en conséquence.

Il est possible de valider plusieurs boîtes à cocher, dans ce cas seul le champ thème est alimenté différemment : il contient '*' pour indiquer que plusieurs valeurs sont choisies. Ces comportements de l'interface sont identiques pour les 2 types de graphiques.

Statistiques complètes

La colonne de gauche des statistiques contient, pour le texte entier :

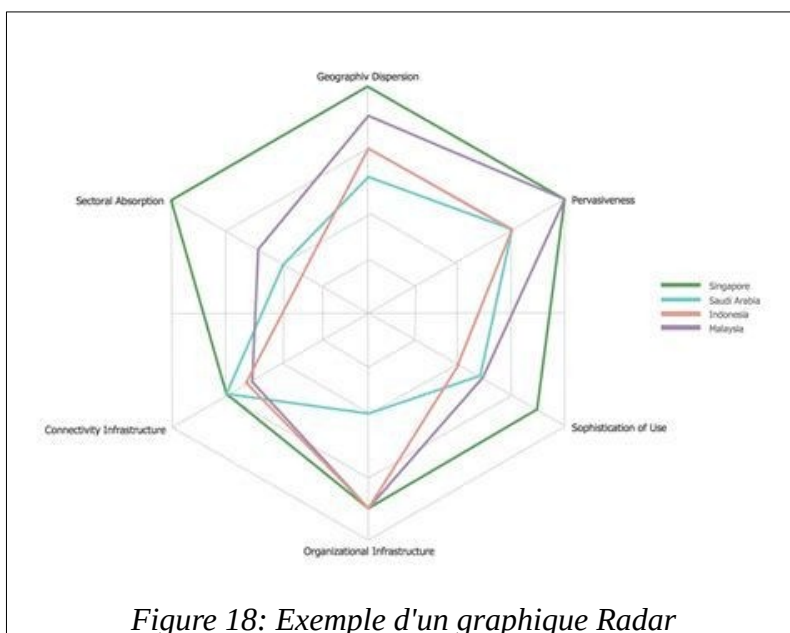
- Le nombre de segments ou thèmes annotés,
- Le nombre de parties, soit le nombre de segment de premier niveau,
- Le nombre de mots,
- Le temps de parole total,
- Le temps de parole moyen par segment.

J'avais prévu en première instance, de comptabiliser le nombre de phrases par thème mais cela n'avait aucun sens puisque chaque segment peut contenir plusieurs phrases ou seulement une partie (dans ce cas une phrase peut appartenir à deux segments). Le travail de l'annotateur permet justement de montrer ce fonctionnement supra de la pensée du locuteur.

6 Évolutions fonctionnelles

A l'issue de la présentation de cette première version, Marie Chagnoux est à la fois rassurée et inspirée. Rassurée sur la possibilité que nous aurons de finaliser quelque chose d'utile à valeur ajoutée et inspirée car certaines de mes idées lui permettent d'imaginer des évolutions. En voici deux : dans un souci pédagogique, l'affichage des graphique pourrait être animé. Cela permettrait de dérouler le cheminement du discours et, en quelque sorte, de suivre la chronologie de ses digressions. Seconde évolution : sur la partie affichant le texte, il serait intéressant de pouvoir visualiser tous les segments annotés. Ainsi, nous comblerions un manque de TXM qui ne sait afficher qu'avec une seule couleur l'ensemble de la partie annotée (cf Figure 9). En termes de priorités, aux dires de mon encadrante, la fonction d'animation est à considérer comme « Agréable », celle concernant la coloration du texte serait « Appréciable ».

De mon côté, je sais qu'il me reste beaucoup de choses à implémenter dont une fonctionnalité essentielle : la génération d'une vue identique à la figure 1. Par ailleurs, la vision des segments intra-texte m'inspire une forme graphique : le radar.

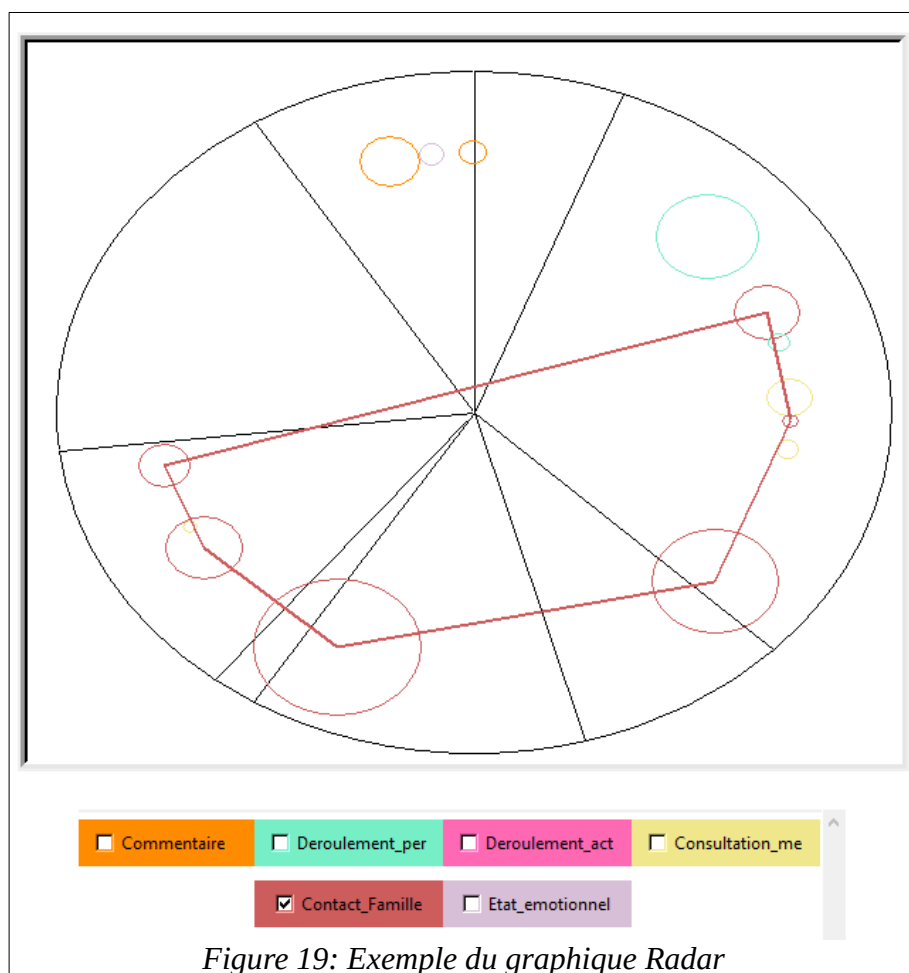


Cette mise en forme pourrait faire apparaître les liens sous-jacents à la réponse principale : tous les segments dont la valeur est identique pourraient être reliés. Et nous comprendrions l'étymologie commune de texte et textile.

6.1 Nouveaux graphiques, nouvelles fonctionnalités

Le radar

A partir du graphique en cercle partitionné, j'ajoute la forme radar. Les couleurs des parts ne sont maintenues que pour les contours des formes et les boîtes à cocher permettront à l'utilisateur de choisir quels liens il veut voir. Ci-dessous un exemple avec la valeur « Contact_Famille » sélectionnée :



Cette figure révèle les rappels successifs du même sujet en parallèle de la réponse à la Question I.

On pourrait me reprocher d'avoir réalisé ce graphique avant le graphique cible attendu par Madame Chagnoux. Je l'ai priorisé en raison de son faible coût de réalisation.

La coloration du texte

C'est une des fonctionnalités importantes pour ma responsable de stage qui semble relativement aisée à réaliser et qui va pourtant me poser quelques problèmes. La gestion des couleurs d'affichage d'un texte dans une zone d'édition n'est pas une fonctionnalité simple pour tkinter et, par ailleurs, puisque nos annotations sont disposées sur plusieurs niveaux, des choix s'imposent : faut-il colorier d'une seule couleur un segment qui contient plusieurs sous-parties ou bien faut-il conserver les couleurs des sous-parties ? J'opte pour la seconde option fournissant visuellement plus d'informations.

J'en parle à mon utilisatrice en l'avertissant que la multitude de couleurs risque de rendre le texte difficile à lire. C'est assez peu important puisque ce qui compte c'est la vision macro d'un ensemble de segments constituant les propos.

Voici l'aspect obtenu par cette implémentation :

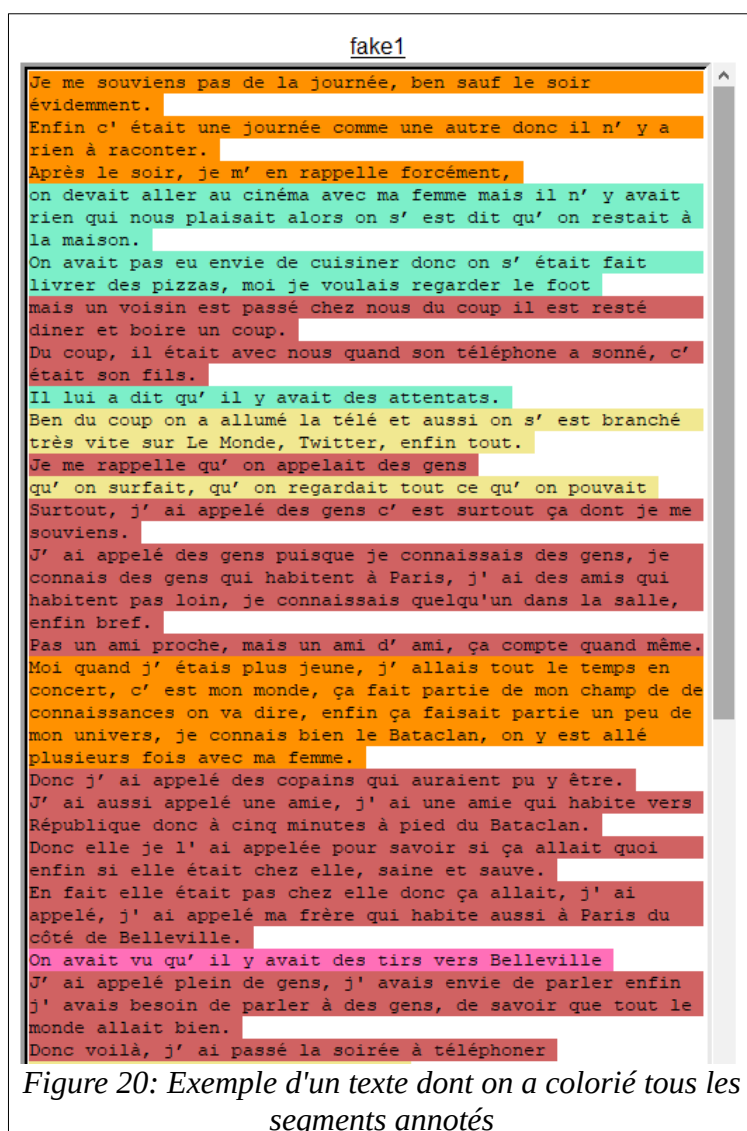


Figure 20: Exemple d'un texte dont on a colorié tous les segments annotés

L'animation

Pour la fonction d'animation il m'a suffit d'ajouter un délai dans le traçage du graphique et un bouton de lancement pour sa réalisation. Correction : après quelques tests, je me rends compte d'un effet de bord. Lorsque l'animation est activée je dois désactiver tous les autres boutons sinon, dans le cas où l'utilisateur cliquerait sur l'histogramme alors que le camembert se dessine, la figure obtenue n'aurait plus aucun sens. Il faut donc que je désactive tous les boutons et que je crée un bouton d'arrêt de l'animation pour permettre à l'utilisateur d'interrompre le ralentissement de l'affichage. Voici les deux boutons :

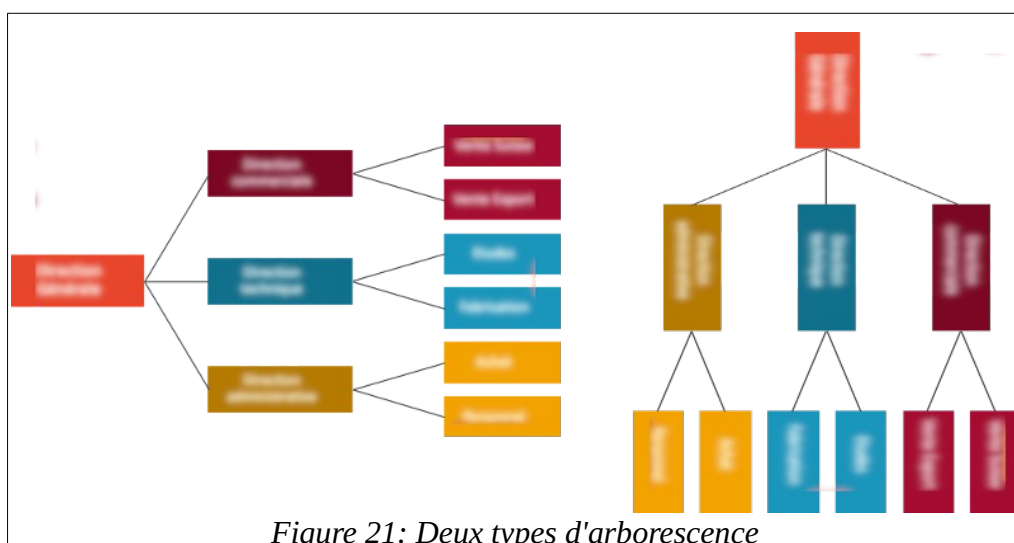


L'arbre

Il est enfin temps de réaliser la figure tant espérée. Le graphique de la figure 1 contient deux difficultés de réalisation : il n'est pas circonscrit et des liens doivent être dessinés entre chaque nœud.

Il n'est pas circonscrit car je dois toujours avoir en tête le caractère infini de l'arborescence. Que ce soit en largeur ou en profondeur, c'est à dire en nombre de segments et en hiérarchie entre les segments.

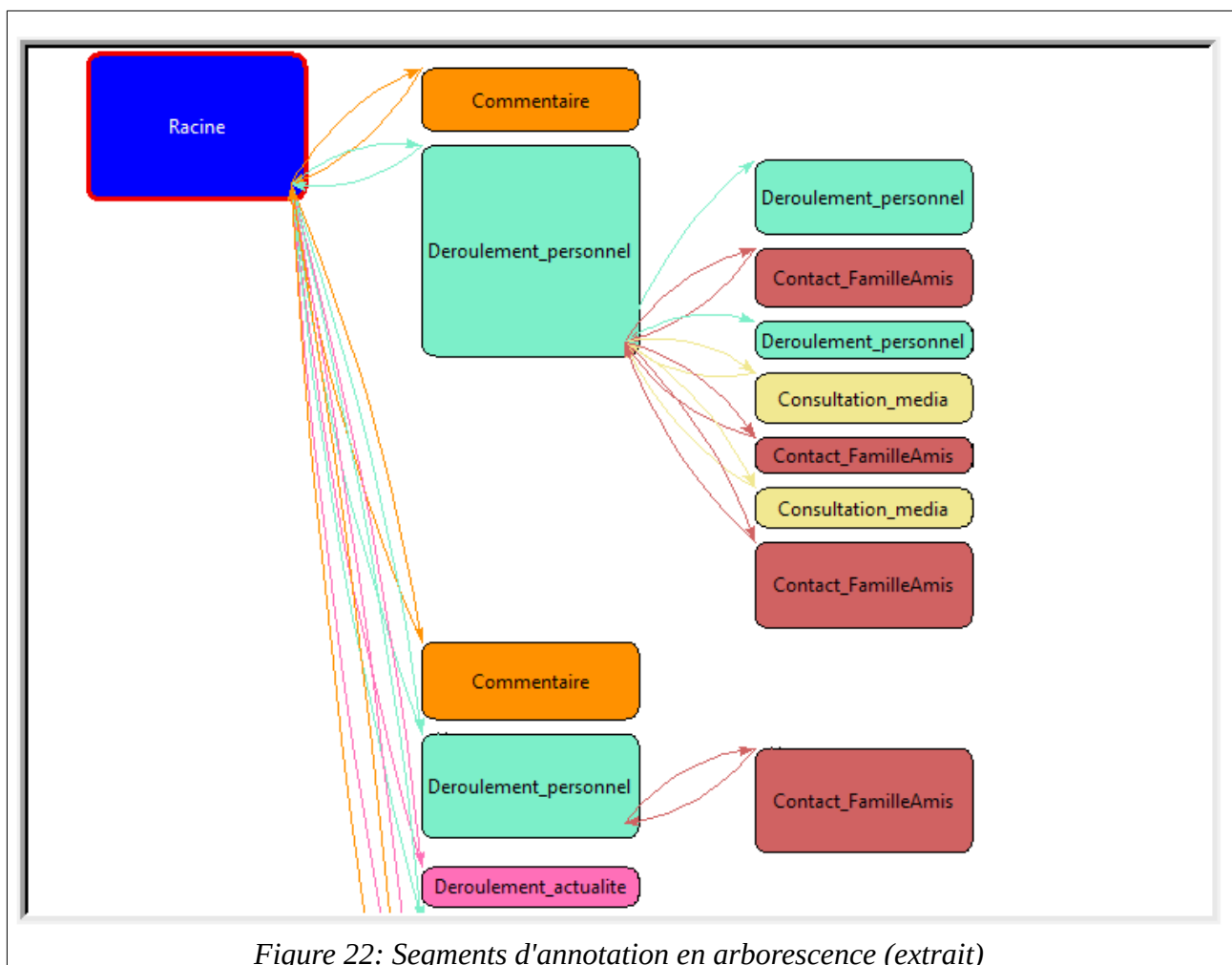
Dans la science de la représentation des arbres deux courants s'affrontent : le format horizontal et le format vertical :



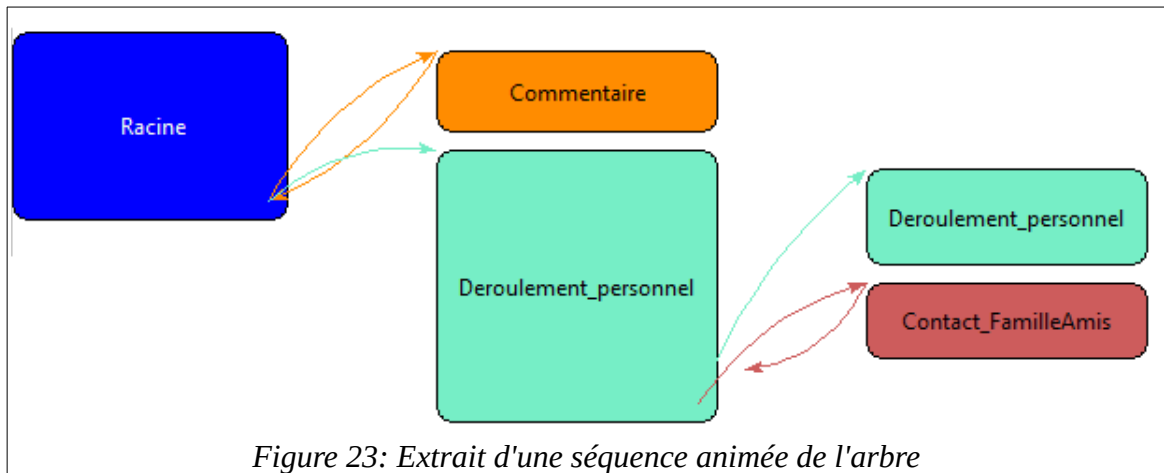
Je choisis le format vertical qui me semble correspondre à la forme d'un texte dans notre langue : de gauche à droite et de haut en bas.

Quant aux liens, ils ont, dans le schéma de la publication de Marie Chagnoux, deux propriétés : ils sont doubles pour représenter les deux sens de la pensée (digression et retour au sujet précédent) et courbes. Enfin, les nœuds sont de simples cercles numérotés. Cette numérotation n'est pas un objectif puisqu'elle fait référence à l'annotation. Dans un souci de confort pour l'utilisateur, je prévois d'inclure la valeur du segment à chaque nœud. De même, j'ajoute la possibilité de se déplacer dans le graphique à l'aide de la souris par un simple glisser (comme avec une feuille de papier), ce qui n'était pas nécessaire dans les mises en forme précédentes.

Voici le graphe en arborescence :



Sur ce type de schéma, l'animation est également disponible, ce qui prend tout son sens :



Chaque segment apparaît dans l'ordre du discours et chaque flèche se déploie pour illustrer les digressions et retours successifs. On remarquera que dans cet arbre certaines boîtes n'ont qu'une flèche. La grande boîte « Déroulement_personnel » n'a pas de retour vers la racine car ses sous-parties ne sont pas entièrement parcourues ; la petite boîte « Deroulement_personnel », elle, n'est pas considérée comme une digression, elle n'a donc pas non plus ce lien arrière.

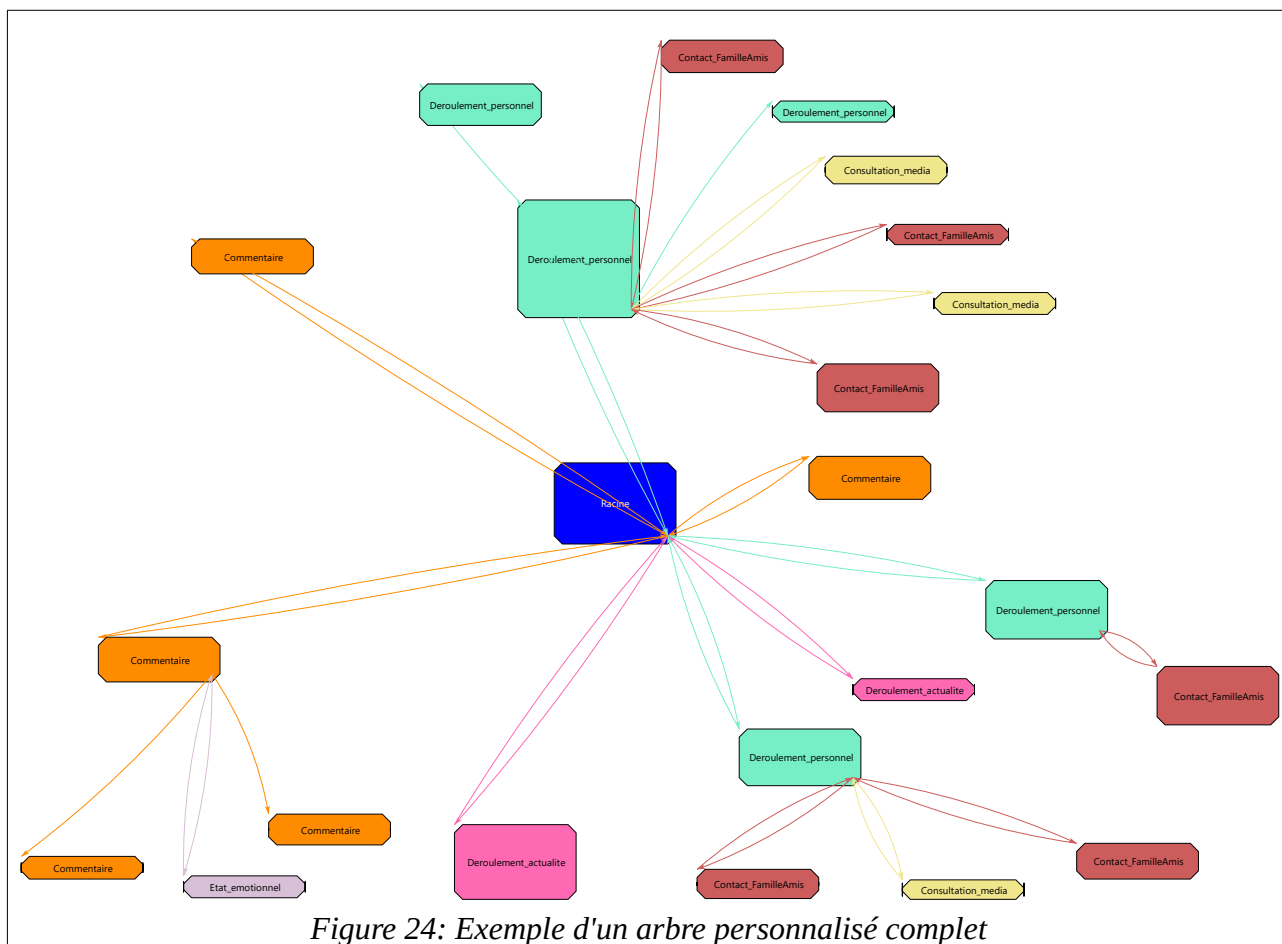
Ce formalisme est à quelque chose près celui de la première figure présentée dans ce mémoire. Il lui manque une chose que je vais ajouter dans l'arbre personnalisé.

L'arbre personnalisé

Pour être tout à fait compatible avec le besoin de Marie Chagnoux il faudrait que cet arbre puisse être mis en forme manuellement. La figure 1 n'est pas LA mise en forme à respecter, c'est une mise en forme adaptée à un exemple. Notre outil doit offrir la plus grande autonomie à l'utilisateur. Pour cela il faut lui donner la main sur la disposition des éléments de l'arbre. Mais il faut aussi l'aider dans cette mise en forme, qu'elle soit aussi confortable que possible en l'assistant raisonnablement.

J'ajoute donc la possibilité de déplacer chaque boîte de l'arbre comme l'utilisateur le souhaite tout en automatisant la mise à jour des liens aller et retour. Bien sûr, il faut aussi conserver la fonctionnalité d'animation.

Voici une mise en forme réalisée avec l'arbre personnalisé :



Cette vue est exactement l'objectif que nous nous étions fixé, le cœur de notre projet est terminé. L'outil affiche, dans une première étape, l'arbre standard, l'utilisateur a ensuite le loisir de déplacer chaque élément comme il le veut.

Cependant, les nouvelles fonctionnalités de mise en forme en appellent d'autres. Étant donné la taille des arbres, il n'est pas possible d'organiser les nœuds sans être capable de prendre du recul sur le schéma. Il faut donc que j'ajoute une fonction de zoom avant/arrière. Il est également indispensable qu'après un travail de mise en forme, l'on puisse sauvegarder son arbre personnalisé. Si, au contraire, l'on souhaite revenir à la forme par défaut il faut pouvoir l'obtenir d'un clic.

La fonction zoom avant/arrière est utilisable avec l'aide de la souris associée à la touche <ctrl>. J'ajoute aussi les boutons + et – qui serviront en cas d'utilisation d'un touchpad, notamment avec les MacBook d'Apple. Enfin, lorsque l'on quitte le fichier, une proposition de sauvegarde de l'arbre est faite à l'utilisateur.

Il reste encore un besoin que j'avais mis en stock dans la liste des fonctions indispensables pour mon encadrante : la sauvegarde de l'image générée. Puisque cet outil doit servir à illustrer ses recherches, il faut qu'elle puisse enregistrer les mises en formes obtenues suite à ses annotations. J'ajoute un petit appareil photo sous le graphique pour lui permettre de stocker les clichés qu'elle voudra réutiliser. C'est cette fonctionnalité qui m'a permis de générer la figure précédente.

Voici la partie droite de l'écran final :

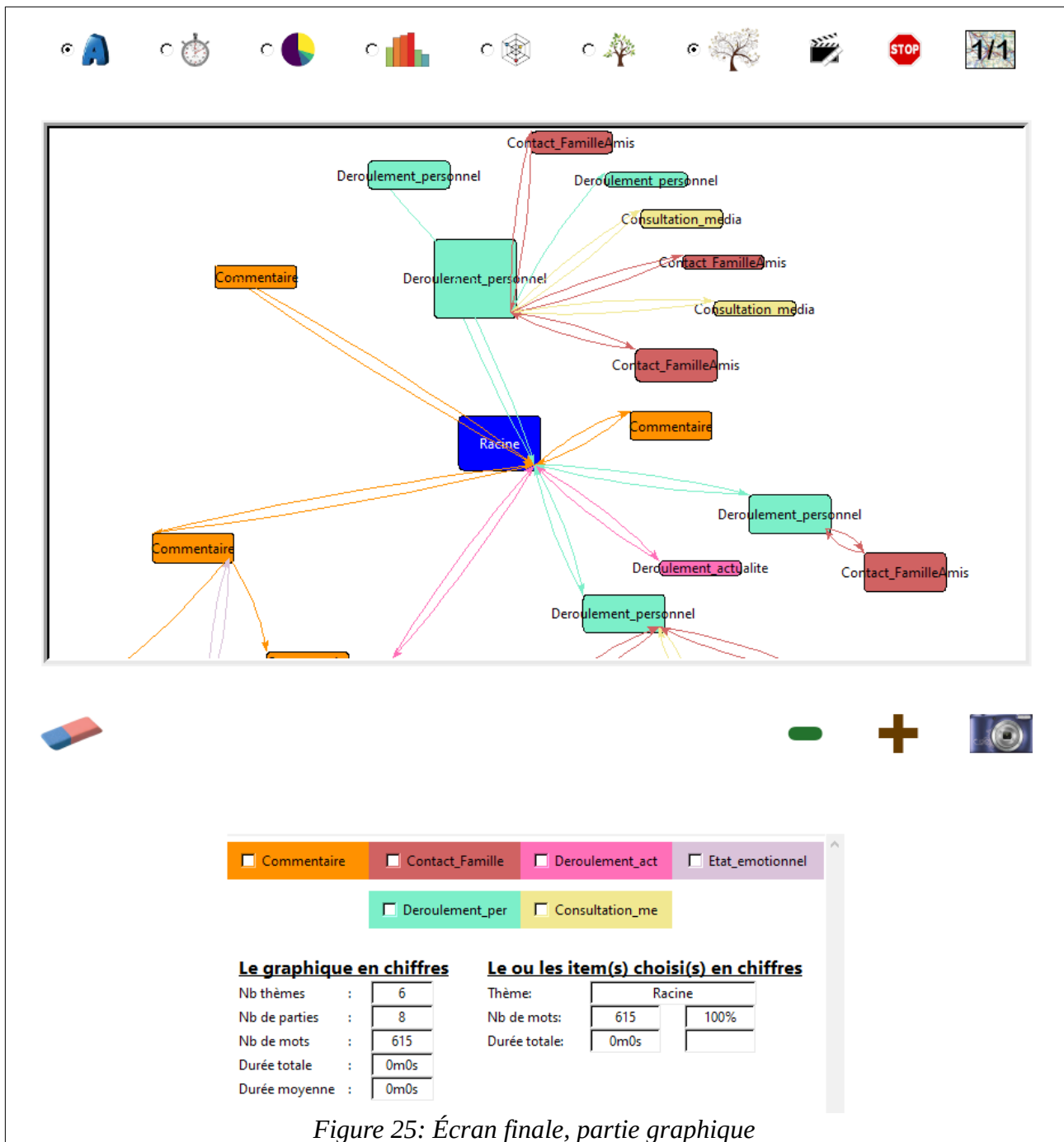



Figure 25: Écran finale, partie graphique

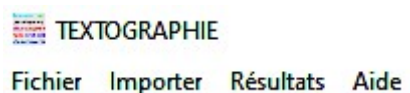
La gomme sous le graphique permet de revenir à la mise en forme standard sur l'arbre personnalisé et sur tous les autres graphiques elle permet de désélectionner les formes et les segments choisis via les cases à cocher.

L'icône  permet la réinitialisation à un zoom nul.

6.2 Intégration de l'import

Puisque l'essentiel est fait, il faut penser à l'indispensable. La fonctionnalité d'import d'un fichier XML de TXM n'étant disponible qu'en ligne de commande, il est temps de l'intégrer à l'outil. Je passe donc d'une interface simpliste à un logiciel complet contenant un menu et plusieurs items.

Puisque nous changeons d'échelle, il faut baptiser notre solution. Je choisis de l'appeler Textographie, ce qui permet un bon raccourci pour décrire son contenu. Il lui faut aussi une icône :



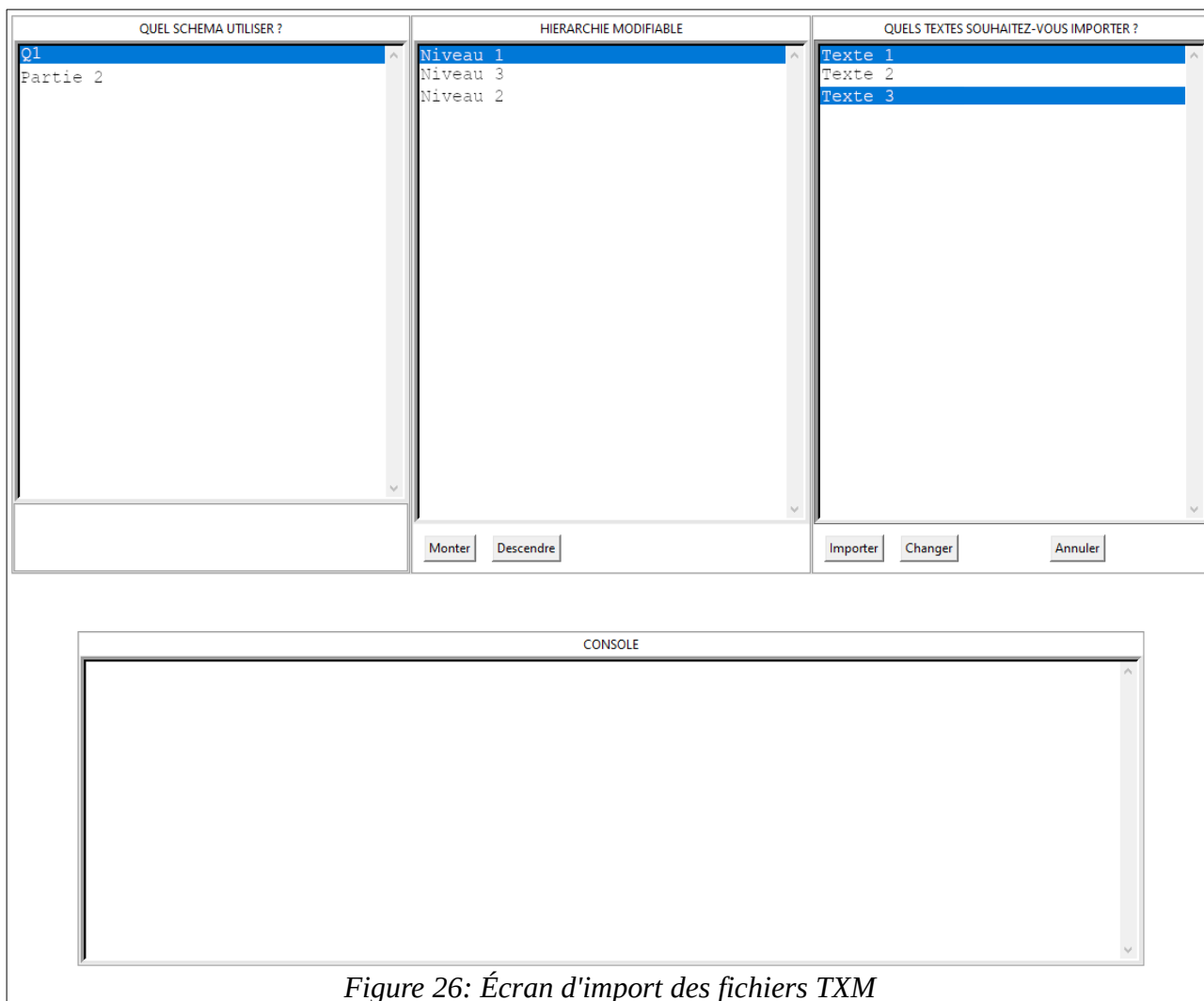
Les graphiques seront accessibles par l'item <Résultats>, les imports avec <Importer> et je réserve de la place pour de futures fonctions dans <Fichier>.

L'écran d'importation n'est pas trivial. L'absence de schéma hiérarchique dans la structure d'annotation de TXM oblige à nouveau à l'utilisateur à valider ou modifier l'arborescence partiellement déduite du fichier XML à importer. Par ailleurs, plusieurs schémas peuvent être définis dans TXM et chaque export peut contenir plusieurs textes. L'utilisateur devra donc :

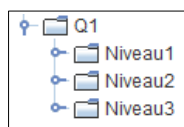
- Choisir le schéma à utiliser,
- Valider ou modifier la hiérarchie du schéma choisi,
- Choisir les textes à importer.

Quand il aura validé ces options, il faudra lui permettre d'obtenir le compte rendu d'import de chaque fichier.

Après avoir choisi le fichier TXM à utiliser, voici l'écran de validation des paramètres d'import :



On retrouve ci-dessus les éléments cités. La hiérarchie est modifiable grâce aux boutons <Monter> et <Descendre> pour donner à Textographie le format de la structure initiale, pour rappel :



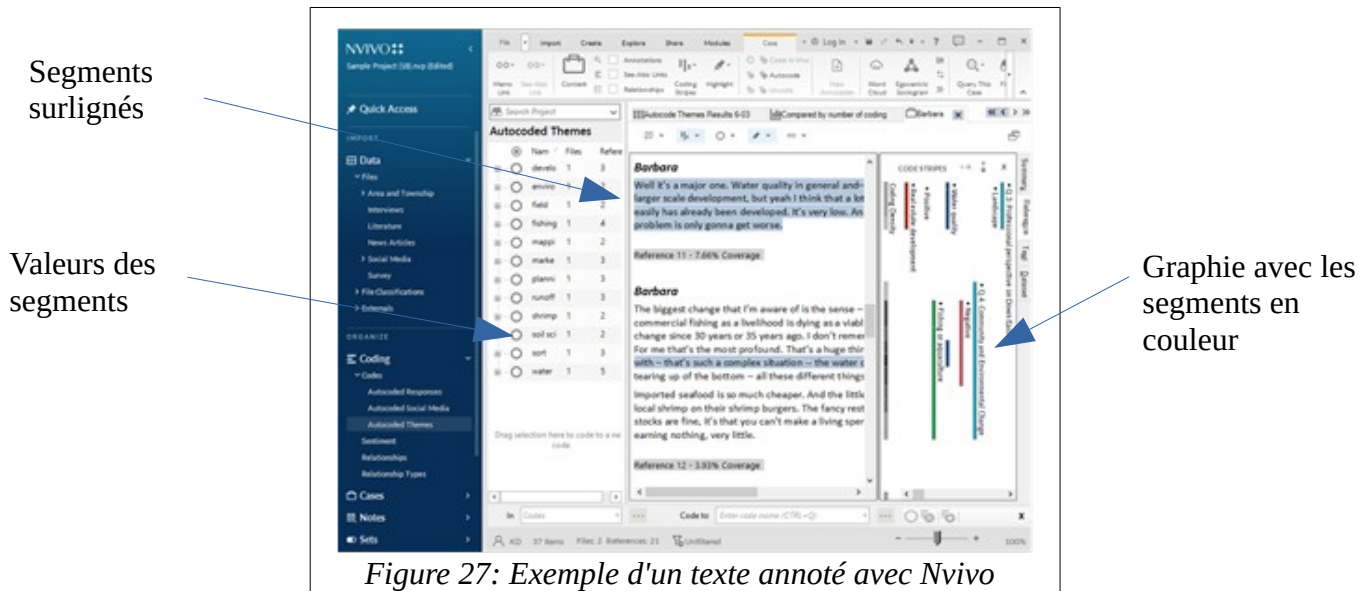
Dans l'exemple ci-dessus on constate qu'elle doit être corrigée.

La zone <CONSOLE> permet à l'utilisateur de suivre les comptes rendus d'import.

6.3 Derniers besoins

Nvivo

Puisque Textographie est presque terminée et qu'il reste quelques semaines de stage, Madame Chagnoux me demande si je pourrais intégrer l'import de textes annotés de Nvivo. Nvivo, l'outil professionnel que j'ai rapidement présenté au début, est un logiciel très complet qui contient déjà quelques fonctionnalités d'affichage des segments annotés.



Il est plus riche que TXM, ce qui me facilite la travail puisque, cette fois, une hiérarchie d'annotation peut être définie et qu'elle est intégrée aux exports XML. Par contre, je n'ai pas de connexion avec l'équipe technique du produit, il va donc falloir que j'induisse la méthode d'importation à partir des exemples que je peux générer. Enfin, comme je n'ai le droit qu'à une version de tests utilisable 14 jours, il me faut travailler efficacement.

Je reviens sur mon code d'importation et de construction de l'arbre des données initial et ajoute le traitement des textes Nvivo. Dans ma démarche de pilotage par la vision utilisateur (appelée *User centric*), je fais en sorte que cela soit le plus transparent possible dans l'interface. Il me faut une semaine environ pour ajouter ce format de fichier.



Cette fois la hiérarchie n'est pas modifiable et le schéma est unique.

Gestion des couleurs

Une dernière fonction tient à cœur à Madame Chagnoux : la gestion des couleurs. Dans ces publications elle applique une charte graphique dans laquelle les couleurs sont définies. Elle souhaite donc, pour harmoniser le contenu de ses documents pouvoir choisir les couleurs qui seront utilisées lors de la génération automatique des graphiques.

Je développe donc un bel écran proposant une palette très large de couleurs afin de définir des nuanciers. Dans ma démarche *user centric* je prends soin de proposer une interface confortable pour des utilisateurs non informaticiens. Voici cette interface (que j'intègre dans le menu <Fichier> :

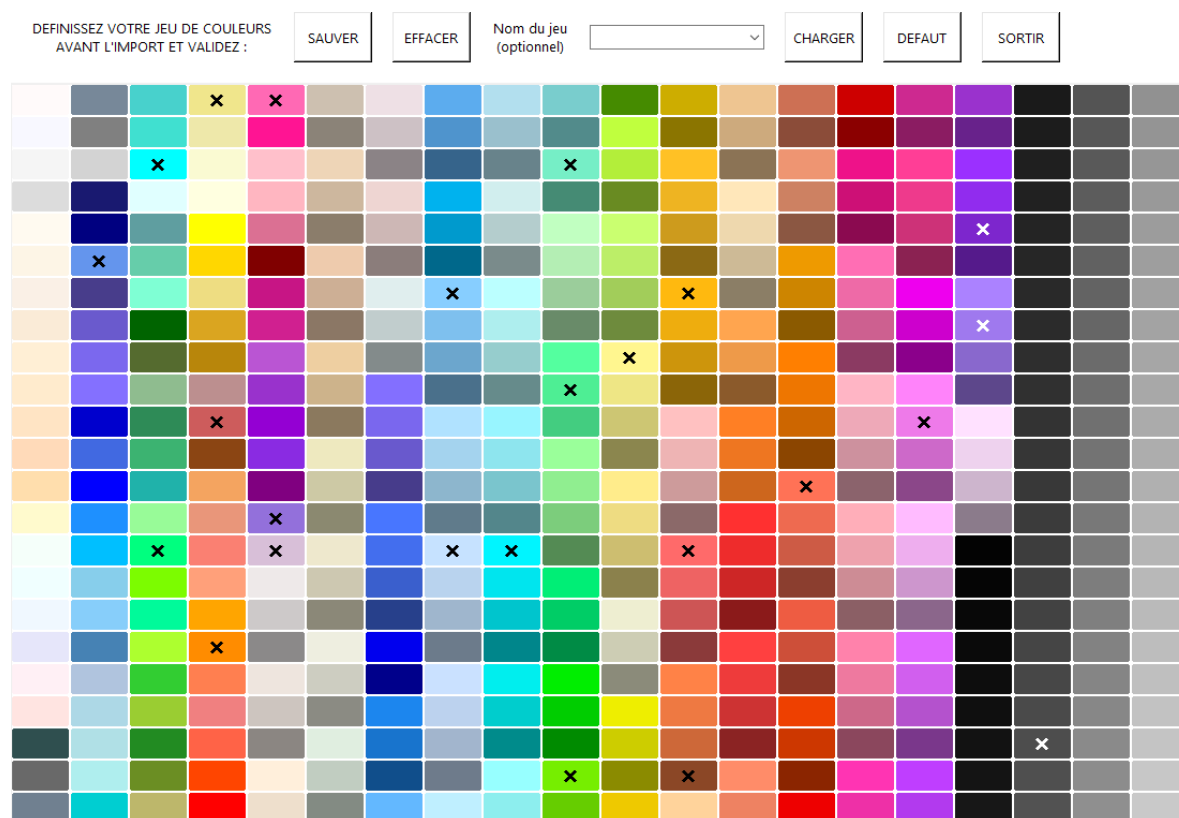


Figure 29: Premier écran de gestion des nuanciers

Chaque couleur peut être ajoutée au nuancier en cliquant dessus, le bouton <EFFACER> permet de tout désélectionner, les boutons <SAUVER> et <CHARGER> permettent d'enregistrer et lire un nuancier défini.

Le travail sur ce type d'écran est des plus agréable car il produit un résultat esthétique et permet d'offrir une bonne fluidité des actions à l'utilisateur. Mais ce n'est pas ce qui convient à Marie Chagnoux. J'ai commis l'erreur de ne pas lui demander une expression de besoins en bonne et due forme. Ce dont elle a besoin c'est de choisir très précisément les codes couleurs de ses nuanciers. Il lui faut une zone de saisie pour donner les codes couleurs hexadécimaux des valeurs de sa charte. J'ajoute donc ce second écran :

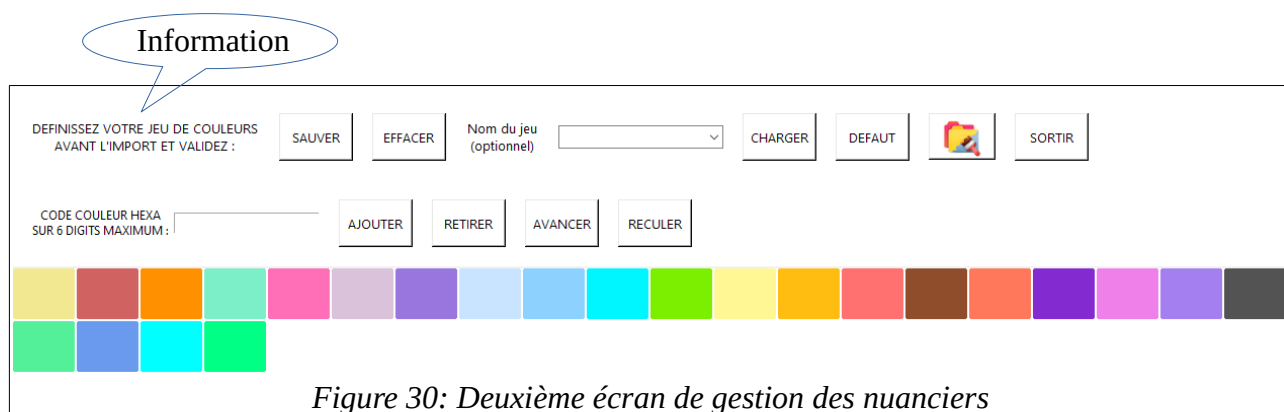



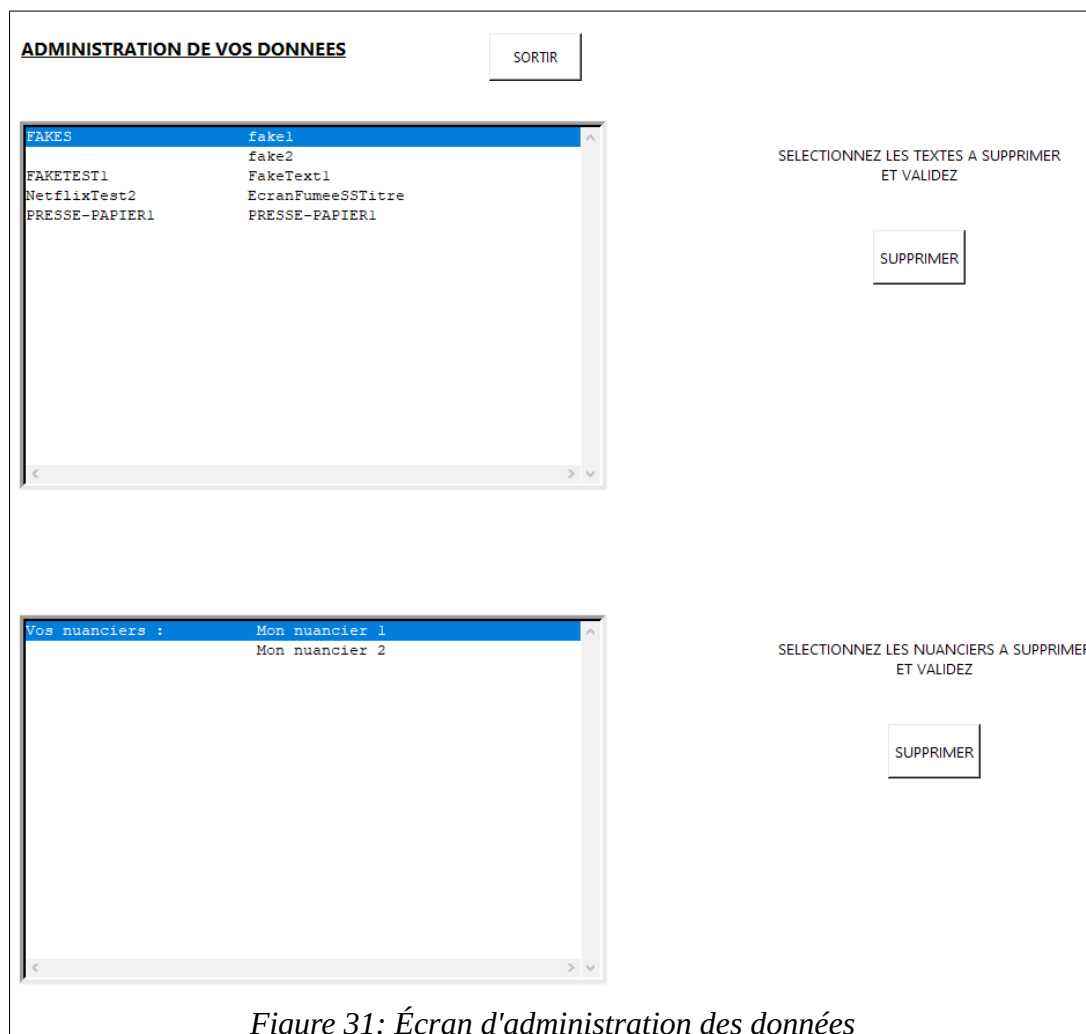
Figure 30: Deuxième écran de gestion des nuanciers

Les boutons <AJOUTER>, <RETIRER> permettent d'insérer ou de supprimer une couleur, <AVANCER> et <RECUER> permettent de les classer. Le bouton  permet de basculer sur l'écran des couleurs précédent. Cet écran est conforme à sa demande.

Il faut remarquer une contrainte liée à la gestion des nuanciers : les couleurs sont intégrées à l'import des données. J'avais, depuis le début, en tête qu'il nous faudrait pouvoir gérer les couleurs des graphiques donc, lors de la construction des arbres de données au moment de l'acquisition des textes de TXM (puis plus tard de Nvivo), les couleurs étaient déjà mémorisées (jeu de couleurs par défaut). A présent, l'utilisateur est averti de ce comportement dont il n'avait pas conscience (bulle «Information»). La raison est simple : lorsque l'on a importé un texte et que l'on a généré et capturé ses représentations graphiques et ne veut pas que ce travail soit modifié par un autre texte qui utiliserait le même nuancier modifié pour respecter une autre charte.

Administration des données

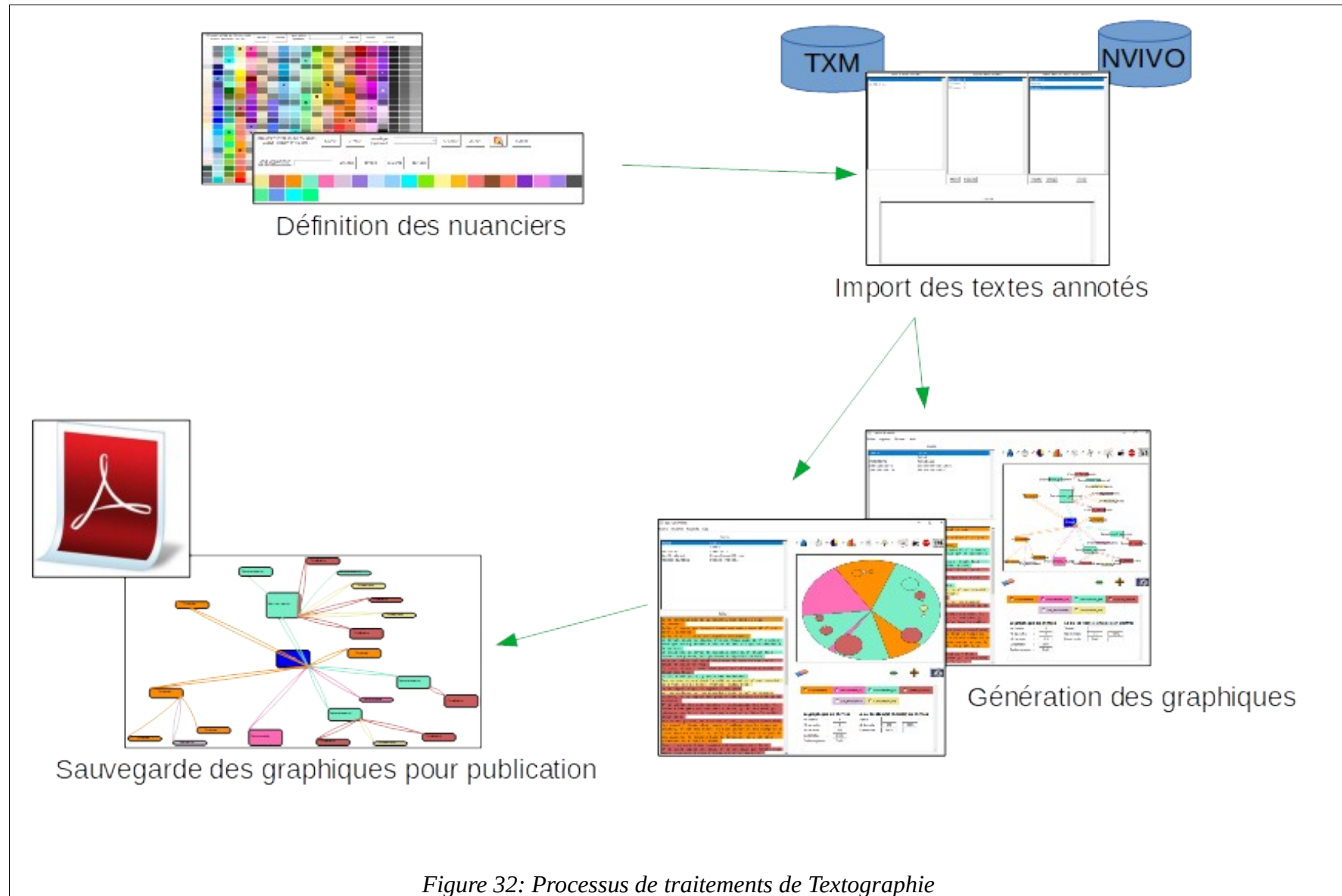
Enfin, pour que Textographie soit une application autonome, il faut donner la possibilité aux utilisateurs de gérer leurs données. Deux types de données sont produites par l'application : les résultats d'imports (dans lesquels figurent aussi les arbres personnalisés) et les nuanciers. Il faut donc créer un dernier écran de purge des fichiers contenant ces informations.



Cette interface simple permet de sélectionner des textes et/ou des nuanciers puis de demander leur suppression.

7 Processus de traitements

Voici, en synthèse, le processus complet de traitement de Textographie :



8 Déploiement

8.1 Livraison pour Python

Comme expliqué dans le chapitre « Organisation et méthode », j'ai choisi Python pour réaliser ce logiciel afin de mettre toutes les chances de notre côté de finir dans le délais des quatre mois du stage. Ce langage offre également la garantie d'une portabilité sur tous les systèmes d'exploitation d'ordinateur personnel.

En contrepartie, pour le faire fonctionner il faut installer un interpréteur Python (téléchargeable gratuitement) auquel il faut adjoindre les bibliothèques utilisées par Textographie. Je rédige donc une procédure d'installation^x de notre outil et la fait valider par Marie Chagnoux qui l'applique sur son ordinateur.

Comme je voulais que cette installation soit légère et simple j'ai peu utilisé de bibliothèques Python. Il n'y en a que cinq : pour le décodage XML (2), la gestion du clavier, le son et la sauvegarde des images graphiques (les clichés). Tkinter qui gère toutes les interfaces est un standard du langage qui ne nécessite aucune installation complémentaire (je l'ai choisi en grande partie pour ça).

La procédure fonctionne bien et ma responsable de stage peut utiliser l'outil.

8.2 Plus de serpent au Paradis

Mais ce mode d'installation ne me plaît pas car il nécessite des manipulations auxquelles les utilisateurs non avertis ne sont pas habitués. Je cherche donc une méthode pour générer une application qui ne nécessiterait pas de procédure d'installation manuelle et qui maintiendrait la compatibilité avec plusieurs systèmes d'exploitation.

Python étant très riche en bibliothèques, voici les solutions que je choisis :

- cx_Freeze : permet de générer un package sous Windows qui proposera à l'utilisateur d'installer automatiquement un exécutable complet dans le répertoire de son choix.

L'outil est extrêmement simple à utiliser : je lui donne le nom du fichier source qui contient l'entrée principale et il génère l'exécutable.

- py2app : c'est l'équivalent de l'outil précédent dans le monde Apple. Il fonctionne de la même façon pour moi et permet de livrer une App que le système reconnaît .

Python a disparu, vive Python !

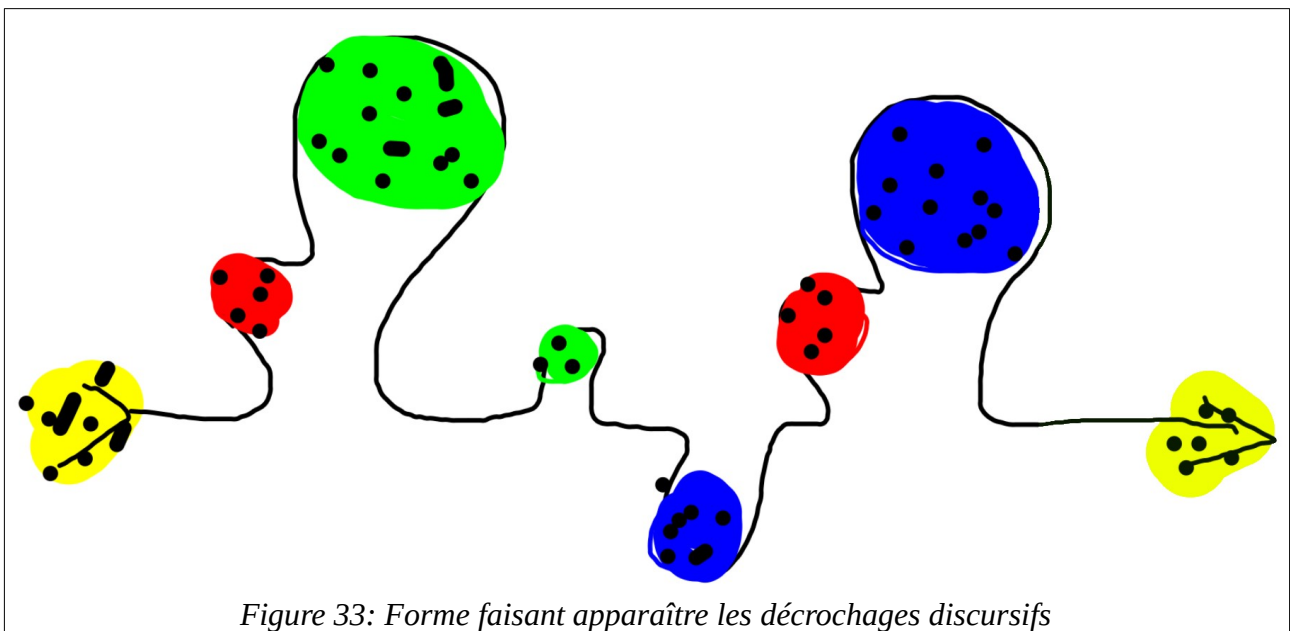
Les deux versions sont téléchargeables ici : <https://jmboucher.fr/accueil/logiciels/>

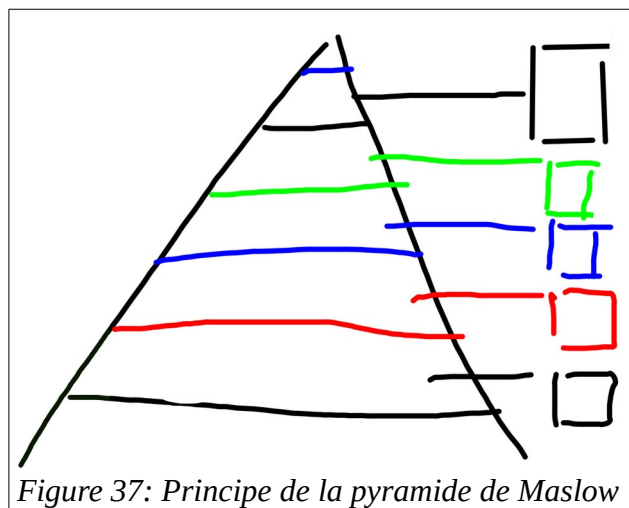
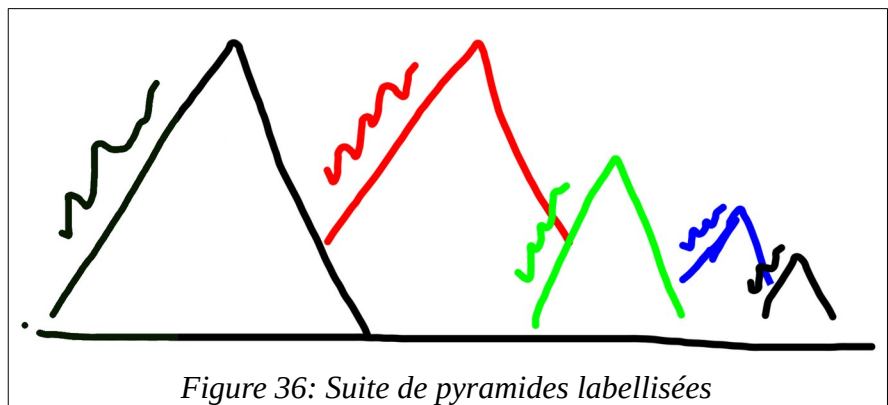
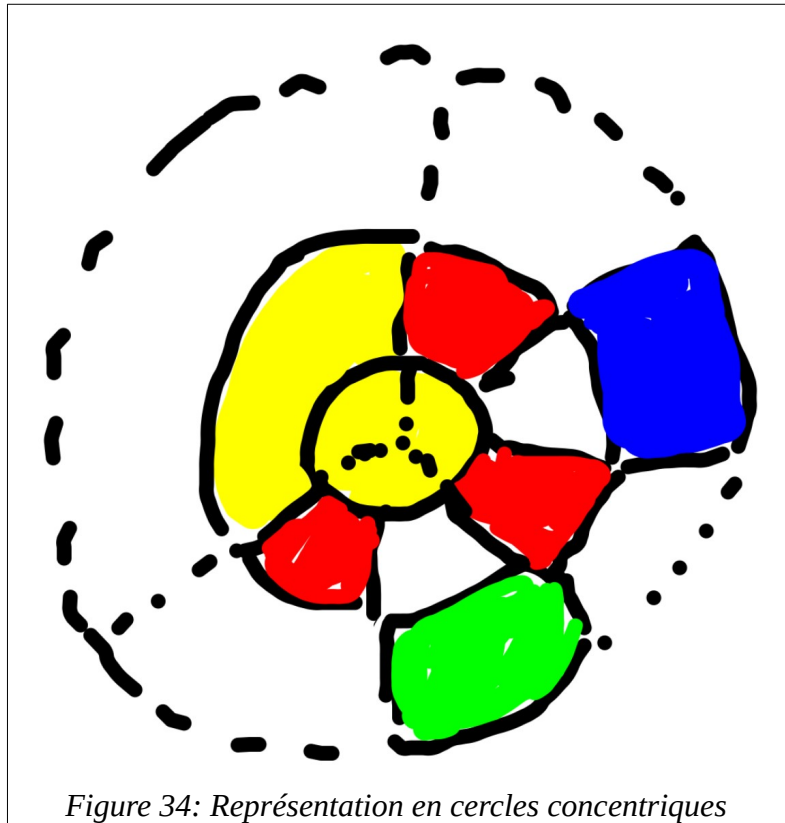
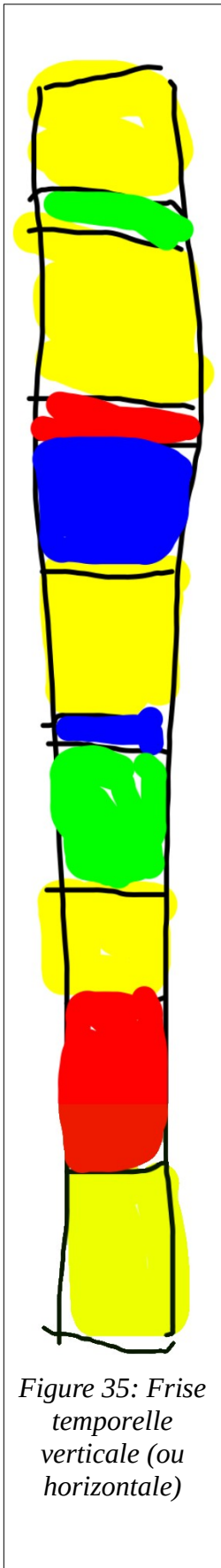
9 Partage et transmission

Cet outil a été présenté dans une version intermédiaire au séminaire Etude 1000 de Metz du 16 avril 2021 au cours duquel certains participants ont manifesté leur intérêt pour l'utiliser dans le cadre de leurs recherches.

Il doit encore être enrichi pour faciliter la gestion des couleurs après import et de nouvelles formes graphiques pourront probablement être ajoutées. On pourrait, par exemple, enrichir les liens des arbres avec les mots, expressions ou ponctuations utilisés pour passer d'un sujet à un autre.

D'un point de vue général, la représentation graphique d'annotations semble être un domaine assez pauvre en solutions, beaucoup de travail reste à réaliser. Voici quelques schémas que j'avais imaginés qui inspireront peut-être pour une version 2 :





J'ai évoqué avec ma responsable de stage l'idée d'une extension de Textographie avec un module d'annotation de textes mais de nombreux logiciels proposant déjà ce type de fonctionnalités enrichies d'abondantes ramifications, il ne semble pas utile d'ouvrir cette voie.

10 Remerciements

Je remercie Marie Chagnoux de m'avoir accueilli si facilement et de m'avoir consacré un peu de son temps de recherche et de professorat, mais je lui suis surtout reconnaissant de son ouverture d'esprit qui m'a permis de développer mes idées et a énormément contribué à faire de ces quatre mois une parenthèse créative et productive.

Enfin, merci à l'équipe TXM pour sa disponibilité et son support qui m'ont fait gagner du temps de formation et ses conseils pour réaliser mes premières lignes de codes.

11 Index des figures

Index des figures

Figure 1: Graphe discursif (Informer sans s'engager : variations de prise en charge énonciative dans les sujets d'actualité), [Chagnoux 2008].....	5
Figure 2: Boucle de gestion de projet itérative (Agile).....	7
Figure 3: ATISHS Analyseur de Textes Innovant pour les Sciences de l'Homme et de la Société.....	8
Figure 4: Eric Delcroix, IBM Watson Personality.....	8
Figure 5: Sémato, Représentation graphique de la sémantique d'un ensemble de textes.....	8
Figure 6: analyse du langage .com, La sémiologie pour faire émerger du sens.....	8
Figure 7: Intensité et type d'émotions dominantes dans le discours des pro- et anti-fusion (Loubet, 2019, revue l'Espace Politique).....	10
Figure 8: Les 4 groupes de témoins de l'Etude 1000.....	12
Figure 9: Planning de l'Etude 1000.....	13
Figure 10: La pluridisciplinarité de l'Etude 1000.....	13
Figure 11: Esquisse 1, Camembert.....	15
Figure 12: Esquisse 2, Histogramme.....	15
Figure 13: Écran d'annotation de TXM.....	16
Figure 14: Schéma de la première interface utilisateur.....	19
Figure 15: Première interface présentée.....	21
Figure 16: Exemple du graphique en histogramme.....	22
Figure 17: Second exemple de la première interface.....	23
Figure 18: Exemple d'un graphique Radar.....	25
Figure 19: Exemple du graphique Radar.....	26
Figure 20: Exemple d'un texte dont on a colorié tous les segments annotés.....	27
Figure 21: Deux types d'arborescence.....	28
Figure 22: Segments d'annotation en arborescence (extrait).....	29
Figure 23: Extrait d'une séquence animée de l'arbre.....	30
Figure 24: Exemple d'un arbre personnalisé complet.....	31
Figure 25: Écran finale, partie graphique.....	32
Figure 26: Écran d'import des fichiers TXM.....	34
Figure 27: Exemple d'un texte annoté avec Nvivo.....	35
Figure 28: Extrait de l'import de fichiers Nvivo.....	35
Figure 29: Premier écran de gestion des nuanciers.....	36

<i>Figure 30: Deuxième écran de gestion des nuanciers.....</i>	<i>37</i>
<i>Figure 31: Écran d'administration des données.....</i>	<i>38</i>
<i>Figure 32: Processus de traitements de Textographie.....</i>	<i>39</i>
<i>Figure 33: Forme faisant apparaître les décrochages discursifs.....</i>	<i>41</i>
<i>Figure 34: Représentation en cercles concentriques.....</i>	<i>42</i>
<i>Figure 35: Frise temporelle verticale (ou horizontale).....</i>	<i>42</i>
<i>Figure 36: Suite de pyramides labellisées.....</i>	<i>42</i>
<i>Figure 37: Principe de la pyramide de Maslow.....</i>	<i>42</i>

12 NOTES ET COMPLÉMENTS

- i Marie Chagnoux, 2008, IRIT-MELODI, *Informers sans s'engager : variations de prise en charge énonciative dans les sujets d'actualité*. <https://hal.archives-ouvertes.fr/hal-00336855>
- ii Marie Chagnoux, 2008, IRIT-MELODI, *Vers un outil de visualisation de la dynamique textuelle : l'exemple des phénomènes citationnels et modaux*. <https://hal.archives-ouvertes.fr/hal-00336859v2/>
- iii Eric Delcroix, 2020, Ed Production cyberlab : : "l'API IBM Watson Personality utilise l'analyse linguistique pour déduire des caractéristiques sur la personnalité, les besoins intrinsèques, et les valeurs d'un individu, depuis les communications que l'utilisateur a accepté de rendre disponible à travers des médias comme des courriels, des messages textes, des médias sociaux, des posts sur les blogs/forums, etc.Source: IBM aidera votre psy avec une intelligence artificielle". <https://www.pinterest.fr/pin/241294492512669651/>
- iv ATISHS : *Analyseur de Textes Innovant pour les Sciences de l'Homme et de la Société de l'Université de Franche Comté*. <http://www.nooj-association.org/atishs.html> Le projet ATISHS a été financé en partie par la région Bourgogne-Franche-Comté. L'équipe ATISHS à l'Université de Franche-Comté : Magali Bigey, Isabelle Hure, Virginie Léthier, Anne Collet-Parizot, Max Silberstein, Izabella Thomas.

Outil des Humanités Numériques (Digital Humanities) utilisé pour analyser des textes littéraires, des corpus journalistiques, des entretiens psychologiques, des sondages, etc. qui utilise des ressources linguistiques pour effectuer des analyses de contenu : analyse de discours, linguistique de corpus, analyse littéraire et narrative et analyse terminologique.

- v *La sémiologie pour faire émerger du sens*, 2008, analysedulangage.com <http://www.analysedulangage.com/index.php/semiologie/>

La sémiologie pour faire émerger du sens. Discipline universitaire dotée d'une méthodologie rigoureuse et scientifique, la sémiologie permet d'analyser tous les éléments de la communication avec un regard lucide. A la fois garante du bon usage des mots (versus la langue de bois), des signes graphiques (versus les contre-sens), des structures narratives (versus le « bullshit ») et de la bonne distance comportementale (versus l'ethnocentrisme), la sémiologie est la discipline reine de la pertinence. Elle structure quatre identités complémentaires : verbale, visuelle, symbolique et comportementale.

- vi Sémato, *Représentation graphique de la sémantique d'un ensemble de textes*. Les graphes construits présentent les quatre principaux niveaux de la description linguistique offerte par Sémato : les lemmes, les champs sémantiques, les synapsies et les thèmes. Pour plus d'information sur ces quatre niveaux de description, on lira : [La technologie linguistique de Sémato](#). A propos du graphique des thèmes présenté : Les thèmes sont obtenus de façon automatique par la fonction GTH (trouvée sur la page des Thèmes de Sémato Texte), ou par vos efforts de construction assistée (avec l'AST ou Assistant Scripteur de Thèmes), ou encore de manière manuelle par des arimages directs aux phrases ou textes du corpus.
- vii Loubet, 2019, revue l'Espace Politique, *Intensité et type d'émotions dominantes dans le discours des pro- et anti-fusion*.

https://www.researchgate.net/figure/Graphique-2-Intensite-et-type-demotions-dominantes-dans-le-discours-des-pro-et_fig3_343045257

viii TXM : <http://textometrie.ens-lyon.fr/>

ix NVivo : <https://www.qsrinternational.com/nvivo-qualitative-data-analysis-software/home/>

x Procédure d'installation sous MacOS pour exécution du code Python (après avoir installer l'interpréteur) :

1. Créer un répertoire <Textographie> sur le bureau et y déposer les fichiers sources

2. Dans IDLE saisir la ligne :

```
import sys; sys.executable
```

=> Cela donne le chemin d'accès à Python, ça devrait être :

```
/Library/Frameworks/Python.framework/Versions/3.9/bin/python3.9
```

=> On l'appelle CHEMIN_PYTHON ci-dessous

4. Dans un terminal installer les packages python utilisés :

```
CHEMIN_PYTHON -m pip install keyboard
```

```
CHEMIN_PYTHON -m pip install canvasvg
```

```
CHEMIN_PYTHON -m pip install playsound
```

```
CHEMIN_PYTHON -m pip install bs4
```

```
CHEMIN_PYTHON -m pip install lxml
```

5. Saisir :

```
cd Desktop
```

```
cd Textographie
```

```
chmod u+x Textographie
```

Fermer le terminal

6. Configurer le mode terminal pour qu'il se ferme automatiquement :

Voir <https://www.hebergementwebs.com/informatique/comment-fermer-automatiquement-le-terminal-macos-a-la-fin-d-un-processus>

7. Modifier le fichier "Textographie" dans le dossier <Textographie>

Remplacer "nom_du_user" par le nom de l'utilisateur

8. Lancer l'application "Textographie" (sans extension) du dossier <Textographie>